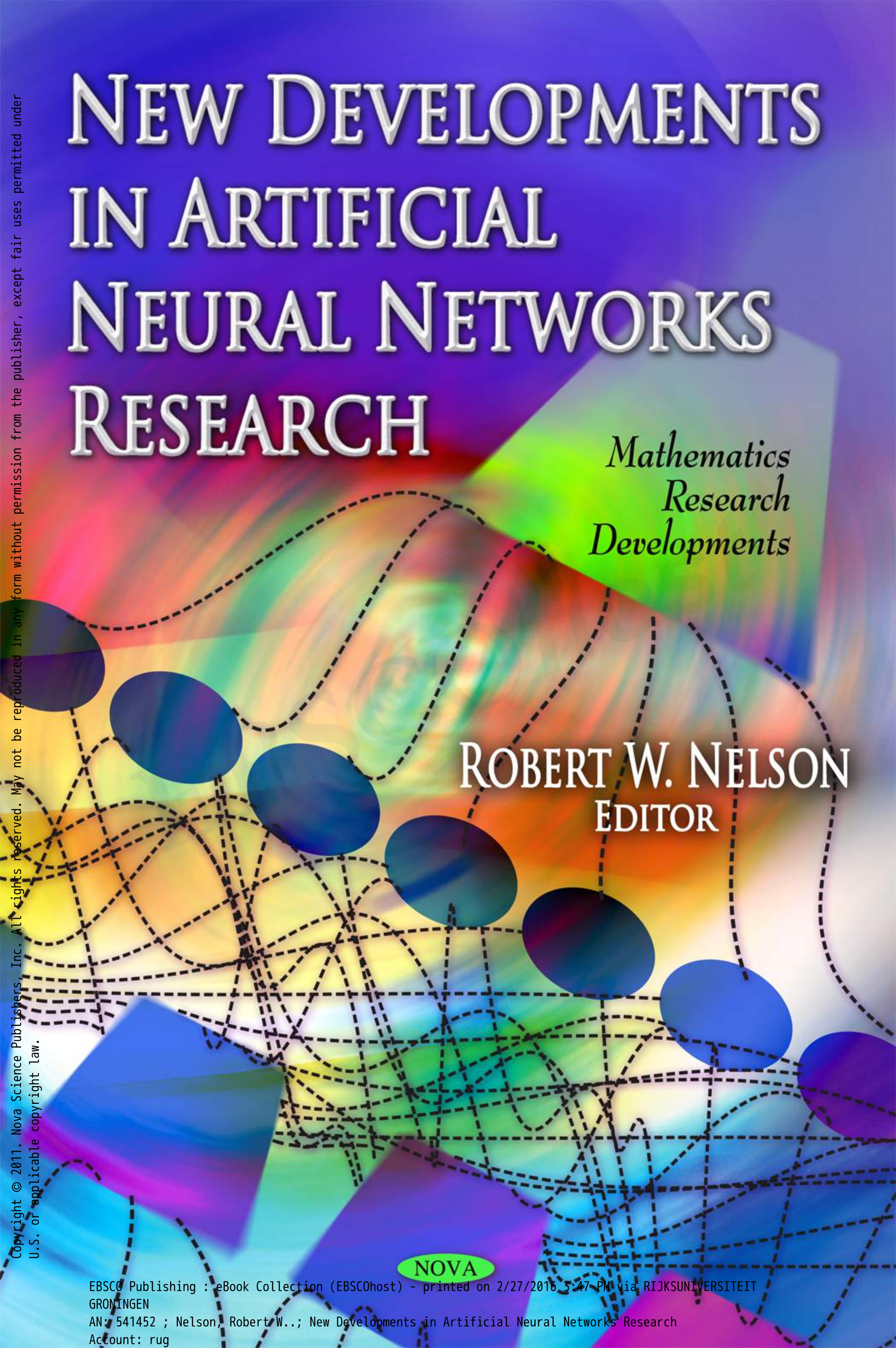


New Developments in Artificial Neural Networks Research

Mathematics Research Developments

Robert W. Nelson
Editor

NOVA

MATHEMATICS RESEARCH DEVELOPMENTS

NEW DEVELOPMENTS IN ARTIFICIAL NEURAL NETWORKS RESEARCH

ROBERT W. NELSON

EDITOR

Nova Science Publishers, Inc.
New York

For permission to use material from this book please contact us:
Telephone 631-231-7269; Fax 631-231-8175
Web Site: http://www.novapublishers.com

NOTICE TO THE READER

The Publisher has taken reasonable care in the preparation of this book, but makes no expressed or implied warranty of any kind and assumes no responsibility for any errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of information contained in this book. The Publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or in part, from the readers' use of, or reliance upon, this material. Any parts of this book based on government reports are so indicated and copyright is claimed for those parts to the extent applicable to compilations of such works.

Independent verification should be sought for any data, advice or recommendations contained in this book. In addition, no responsibility is assumed by the publisher for any injury and/or damage to persons or property arising from any methods, products, instructions, ideas or otherwise contained in this publication.

This publication is designed to provide accurate and authoritative information with regard to the subject matter covered herein. It is sold with the clear understanding that the Publisher is not engaged in rendering legal or any other professional services. If legal or any other expert assistance is required, the services of a competent person should be sought. FROM A DECLARATION OF PARTICIPANTS JOINTLY ADOPTED BY A COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND A COMMITTEE OF PUBLISHERS.

Additional color graphics may be available in the e-book version of this book.

LIBRARY OF CONGRESS CATALOGING-IN-PUBLICATION DATA

New developments in artificial neural networks research / editor, Robert W. Nelson.
p. cm.
Includes index.
ISBN 978-1-62081-964-7 (E-Book)
1. Neural networks (Computer science) 2. Neural networks (Computer science)--Research. I. Nelson, Robert W.
QA76.87.N4957 2011
006.3'2--dc23
2011019862

Published by Nova Science Publishers, Inc. † New York

CONTENTS

Preface **vii**

Chapter 1 A Neural Network Based Visual Servo System: Learning Visual Kinematics of a Scene **1**
Ebrahim A. Mattar Al-Gallaf

Chapter 2 Prediction of Mechanical Properties of Polypropylene/Wgrt Composites Via Uniform Design and Artificial Neural Networks **23**
Shu Ling Zhang' Kaushik Pal and Jin Kuk Kim

Chapter 3 Modeling of Computer-Assisted Learning Using Artificial Neural Networks **41**
Fahad A. Al-Zahrani, Hassan. M. Mustafa, Ayoub Al-Hamadi and U. A. Khashaba

Chapter 4 Prediction of Hole Quality in Drilling GFRE Using Artificial Neural Networks **59**
U. A. Khashaba, I. A. El-Sonbat, A. I. Selmy and A. A. Megahed

Chapter 5 Performance Analysis of Liquid Desic Cant Dehumidification System Using Artificial Neural Networks **77**
P. Gandhidasan and M. A. Mohandes

Chapter 6 Application of the Artificial Neural Networks on Visually Guided Robot **91**
Emregul Ersan, Vedat Topuz and Ayca Gokhan Ak

Chapter 7 ANN-Based Approaches to Study the Nanoscale CMOS Devices **109**
F. Djeffal and T. Bendib

Chapter 8 Artificial Neural Networks in Chromatography and Spectroscopy **123**
Iva Rezić and Tomislav Rolich

Chapter 9 A New Candidate List Strategy for Architecture Selection in Artificial Neural Networks **139**
Cagdas Hakan Aladag

Chapter 10 Artificial Neural Network Based Estimation of Global and Diffuse Fraction of Solar Radiation Using Meteorological Parameters **151**
Shafiqur Rehman and Mohamed Mohandes

Chapter 11 Effects of Micro Pore Characteristics on Strength of Cement Mortar Using Artificial Neural Network **169**
Ali Ugur Ozturk and Okan Onal

Chapter 12 Tracing the Drainage Divide: the Future Challenge for Predictive Medicine **181**
Enzo Grossi

Chapter 13 Artificial Neural Network (ANN) Prediction of a Full-Scale Municipal Wastewater Treatment Plant Performance **193**
Erkan Sahinkaya and Sibel Pulcu Yıldız

Chapter 14 Positioning Control of an under-Actuated Robot Manipulator Using Artificial Neural Network Inversion Technique **207**
Ahmad Azlan Mat Isa, H. M. A.A. Al-Assadi and Ali T. Hasan

Chapter 15 Nonlinear Forecasting with a Hybrid Approach Combining Sarima, Arch and Ann **221**
Erol Egrioglu, Cagdas Hakan Aladag and Cem Kadilar

Chapter 16 Towards a Formalisation of Evolving Connectionist Systems **229**
Michael J. Watts

Chapter 17 Learning with Heterogeneous Neural Networks **257**
Llu ́ıs A. Belanche-Mu ̃noz

Chapter 18 Tuning Differential Evolution for Artificial Neural Networks **277**
Magnus Erik Hvass Pedersen, Andrew John Chipperfield

Chapter 19 Artificial Neural Networks: A Chemist's Perspective in Modeling and Design of Human Immunodeficiency Virus-1 (HIV-1) Non-Nucleoside Reverse Transcriptase Inhibitors (NNRTIs) **295**
Nitin S. Sapre, Swagata Gupta,Nilanjana Pancholi and Neelima Sapre

Chapter 20 Pricing in Banking and Finance by Utilizing Artificial Neural Networks **317**
Vincenzo Pacelli

Index **331**

PREFACE

This book gathers the most current research from across the globe in the study of artificial neural networks. Topics discussed include a neural network based visual servo system; modeling of computer-assisted learning using artificial neural networks; prediction of hole quality in drilling GFRE using artificial neural networks; ANN-based approaches to study the nanoscale CMOS devices; artificial neural networks in chromatography and spectroscopy; heterogeneous neural networks and tuning differential evolution for artificial neural networks.

Chapter 1 – Robotics arm visual servo system does suffer from a number of issues, as related to speed and complexity. One of which is the complicated kinematics relations, in addition the needed computational time to execute a task. This manuscript highlights a mechanism through which to approximate the inter-related visual kinematics relations that are part of visual servo closed loop system through an artificial neural network system. A main issue that hinders visual servo system is related to the complicated and time variant feature Jacobian matrix. The methodology followed here is based on the concept of integration of ANN with an Image Based Visual Servoing (IBVS) system. Artificial neural networks have been employed here to learn and approximate the relations that relate an object moving to a robotics arm movement through a visual servo. For validating the presented concept, one of the closed loop visual servo system already developed, have been used here to verify the presented concept. A learning Artificial Neural Network has proven to be an effective approach in learning the nonlinear relationships that govern kinematics relations used in robotics arm visual servo. In this respect, this chapter will explore how a trained artificial neural net will be used to achieve a very precise robotics arm movement by extracting the visual information from a moving object.

Chapter 2 – Recycling represents a valid alternative to the disposal of post-consumer materials if it is possible to obtain new materials with good properties. In this chapter, Polypropylene (PP)/waste ground rubber tire powder (WGRT) or waste polypropylene (WPP)/waste ground rubber tire (WGRT) powder composites were studied with respect to the effect of bitumen and maleic anhydride-grafted styrene-ethylene-butylene-styrene (SEBS-g-MA) content by using the design of experiments (DOE) approach, whereby the effect of the four polymers content on the final mechanical properties were predicted. Uniform design method was especially adopted for its advantages. Optimization was done using hybrid artificial neural network-genetic algorithm (ANN-GA) technique. The results indicated that the composites showed fairly good ductibility provided that it had a relatively higher concentration of bitumen and SEBS-g-MA under the studied condition. A quantitative relationship was presented between the material concentration and the mechanical properties as a set of contour plots, which were confirmed experimentally by testing the optimum ratio.

Chapter 3 – The present work belongs to a rather challenging interdisciplinary issue associated with neuroscience modeling, educational psychology, and cognitive sciences for searching two optimum teaching methodologies at children's classrooms. Firstly, that one associated with trying effectively to find an optimal methodology for "how reading should be taught?" at children's classrooms. However, the second topic is closely related to optimum teaching method for solving long division problems following subsequent mathematical steps such as: divide, multiply, subtract, bring down, and repeat (if necessary). Searching for optimality leaning/teaching educational topics will depend upon comparative assessments and analysis for different experimental educational methodologies that applied at children's classrooms. Herein, presented assessment processes comprise application of visual and/or auditory tutorial materials in addition to the classical classrooms teaching methodologies. The comparative assessment processes are performed by using realistic Artificial Neural Network (ANN) simulation programs, and/or mathematically formulated modeling. In addition, a computer-assisted learning (CAL) module is designed carefully aiming to develop a specified multimedia tutorial material.

Conclusively, it has been shown that optimal teaching methodology performance attained if and only if visual and auditory tutorial materials have been both presented simultaneously to reinforce the retention of learned material topics. Multi-sensory associative memories in addition to Pavlovian classical conditioning theories are applicable, via designed CAL package, at children's classrooms. It is worthy to note that comparative results obtained are interesting, after field application of suggested CAL package, for association tutorials with teacher's voice. Finally, presented study results in high recommendation for application of novel teaching trends aiming to improve learning quality in children's two reading and mathematical topics.

Chapter 4 – The weight and fuel savings offered by composite materials make them attractive not only to the military, but also to the civilian aircraft, space, and automobile industries. In these industries, drilled holes are extensively implemented for structure assembly. The presence of hole defects due to drilling reduces the stiffness and strength of a laminate and hence its load carrying capacity. The main objective of the present work is to develop artificial neural networks (ANNs), with back-propagation training routine, for predicting the machinability parameters in drilling glass fiber reinforced epoxy (GFRE) composites with different machining conditions (feed, speed, and drill pre-wear). Machinability parameters were characterized by thrust force, torque, peel-up and push-out delaminations, and surface roughness of drilled holes.

The inputs to the neural networks used for predicting delamination size and surface roughness are: spindle speed, feed, drill pre-wear, thrust force, and torque. The values of the thrust force and torque that are fed as inputs to the above networks are predicted using ANNs developed for predicting each of them. Several attempts were performed to achieve the best neural network by changing both of network structure (i.e. the number of hidden layers and the number of units within each hidden layer) and the initial values of the connection weights and thresholds. The best obtained network structure for predicting thrust force, torque, peel-up delamination, push-out delamination, surface roughness were 3-5-1, 3-3-1, 5-11-1, 5-7-3-1, and 5-7-3-1, respectively. The developed ANNs predict the machinability parameters (thrust force, torque, peel-up and push-out delaminations and surface roughness) with acceptable errors for the most confirmation tests.

Chapter 5 – The heart of the liquid desiccant cooling system is the dehumidification process which is influenced by many parameters. Different types of dehumidification equipment have been developed and a variety of analytical models have been employed to analyze the dehumidification process. The dehumidification process involves simultaneous heat and mass transfer and reliable transfer coefficients are required in order to analyze the system. This has been proved to be difficult and many assumptions are made to simplify the analysis.

Artificial Neural Network (ANN) is widely used as an innovative way to tackle complex and ill-defined problems. The present research proposes the use of ANN based model in order to simulate the relationship between inlet parameters and the performance of the dehumidifier. For the analysis, randomly packed dehumidifier is chosen since the packed tower facilitates high mass transfer by providing a large surface area in a relatively small volume. Lithium chloride is selected as the liquid desiccant due to its stability with high performance.

A multilayer ANN is used to investigate the performance of dehumidifier. For training ANN models, data is obtained from analytical equations. The training process implies adjustment of connection weights and biases so that the differences between ANN output and the desired output are minimized. Eight parameters are used as inputs to the ANN, namely: air and desiccant flow rates, air and desiccant inlet temperatures, air inlet humidity, the desiccant inlet concentration, dimensionless temperature ratio, and the inlet temperature of the cooling water. The output of the ANN is the water condensation rate. The predicted water condensation rate by the ANN is validated with experimental data and the value of R^2 is found to be 0.9251. Results and the performance of the developed system are presented in this chapter.

Chapter 6 – In this chapter, the carrying of an object at a workspace, which was perceived by vision, to another location was realized by a robot arm with five axes. Basic image process techniques were used for object recognition and position determination. If the desired object was inside the workspace, the inverse kinematics solution was realized, and then after coordinates of the object's location was sent to the robot arm. The inverse kinematics solution of the robot arm was performed with artificial neural networks (ANN) model (Multi Layer Perceptron-MLP and Radial Basis Function (RBF) Neural Network) based on the forward kinematics solution. For an inverse kinematics solution of the robot, the training data set was created in the ANN method by using the robot's forward kinematics values first and then, ANN modeling was realized. After the robot's inverse kinematics solution was realized, the determined joint angle values were directed to the EDUBOT robot arm and moving the object to the desired location was realized successfully. Experimental results presented in this chapter indicate that RBF is more efficient solution than MLP for inverse kinematic solution of visually guided robot.

Chapter 7 – Over the past three decades, the primary driver of the exponential improvements in integrated circuit performance has been the scaling of transistor dimensions. The inherent benefits of MOSFET scaling are the speed improvement and energy reduction associated with a binary-logic transition. As the MOSFET is scaled below the 100 nm technology node the advantages of MOSFET scaling are diminished by the short channel effects. Ultra-thin film body multigate structures become to be envisaged as a possible alternative to the conventional devices, due to its enormous potentiality to push back the integration limits to which conventional bulk transistor are subjected [1-4]. The main advantage of this architecture is to offer a reinforced electrostatic coupling between the conduction channel and the gate electrode. In other terms, a multigate structure can efficiently sandwich (and thus very well control, electrostatically speaking) the semiconductor element playing the role of the transistor channel, which can be a Silicon thin layer or nanowire. Moreover, it is known for its higher drive current, improved subthreshold slope, improved short channel effect control and potential circuit design flexibility [1–4]. As shown in Figure 1, with two gates controlling the channel, short-channel effects can be greatly suppressed. Due to the fact that simulation of nanoscale CMOS circuits has been the primary factor driving improvements in integrated circuit performance and cost, which contributes to the rapid growth of the semiconductor industry, there is a need to develop a new theory and modeling techniques that capture the physics of quantum transport accurately to guide the design for nanoscale CMOS circuits.

Chapter 8 – The purpose of this chapter is to demonstrate the application of artificial neural networks in modern chemical investigation, mainly in chromatography and spectroscopy.

The first part of the chapter presents a fast and simple procedure for estimation of steel materials corrosion in artificial sweat solution with usage of artificial neural networks (ANN). For the purpose of the mathematical modeling, optical emission spectroscopy experimental data were used to train and test the most appropriate model of artificial neural networks. Evaluation was performed by comparing the experimental data and values estimated by ANN and by calculating the correlation coefficient and mean absolute error. It was concluded that ANN can be easily applied for estimation of corrosion processes.

The second part of this chapter presents a combination of artificial neural networks and genetic algorithms (GA) in optimization of thin layer chromatographic separation of seven components from their mixture. As a goal of optimization, a resolution factor was calculated for different mixture model solutions. Afterwards the prediction of the same parameter was performed by ANN and GA, and very good correlation between predicted and calculated data was observed. Therefore it can be concluded that the developed combination of ANN and GA may be successfully used in many different chromatography investigations, like high performance liquid chromatography (HPLC), ion chromatography (IC), gas chromatography (GC) and other similar techniques.

Based on their current results the authors expect that artificial neural networks and their combination with other methods (such as genetic algorithms) will be implemented in many different chemical and analytical applications in a near future.

Chapter 9 – In the past decade, there have been many implementations in which artificial neural networks (ANN) successfully applied to many areas of science and engineering. One of these areas is time series forecasting. ANN method has been preferred to conventional time series forecasting models because of its easy usage and providing accurate results. In spite of the fact that ANN produces accurate forecasts in many time series implementations, there are still some problems with using this method. When ANN method is utilized to forecast time series, selection of the components of the method is a vital issue for obtaining good forecast values. These components such as architecture structure, learning algorithm and activation function have important effect on the performance of ANN. An important decision is the selection of architecture structure that consists of determining the numbers of neurons in the layers of a network. Therefore, to making a good choice for architecture selection, various approaches have been proposed in the literature. Aladag (2009a; 2009b) proposed a method based on tabu search algorithm to determine the best ANN architecture. He showed that accurate forecasts are obtained when the proposed method is employed for architecture selection. In his proposed algorithm, he utilized a candidate list strategy in which six architectures are examined. In this study, the tabu search algorithm proposed by Aladag (2009b) is tried to be improved by defining a new candidate list strategy in which only four architectures are examined. The beer consumption in Austria and the electricity consumption in Turkey time series are forecasted by ANN and the applicability of the proposed strategy is shown.

Chapter 10 – Measured meteorological parameters such as air temperature and relative humidity values recorded between 1998 and 2002 for Abha city in Saudi Arabia were used for the estimation of global solar radiation (GSR) and fraction of diffuse solar radiation (DSR) in future time domain using artificial neural network method. The estimations of GSR and DSR were made using three combinations of data sets namely, (i) day of the year and daily maximum air temperature as inputs and global solar radiation as output, (ii) day of the year and daily mean air temperature as inputs and global solar radiation as output and (iii) time day of the year, daily mean air temperature and relative humidity as inputs and global solar radiation as output. The measured data between 1998 and 2001 was used for training the neural networks while the remaining 240 days' data from 2002 as testing data. The testing data was not used in training the neural networks.

Obtained results show that neural networks are well capable of estimating global and diffuse solar radiation from temperature and relative humidity. Hence the methodology can be used for estimating GSR and DSR for locations where only temperature and humidity data are available.

Chapter 11 – Cementitious materials comprise a great part in construction process of structures such as buildings, bridges, roads and dams. The most expected properties of structural members prepared with cement mortar or concrete, are strength and durability. These structural members are supposed to have strength values determined in the structural analysis and to be durable against aggressive media in their service life. These characteristics are the most effective criteria in civil and material engineering. Therefore, these two parameters depend mainly on the pore structure and its characteristics of structural members. Nowadays, scientists and engineers are using new computer technologies, simulations and experimental techniques try to perform to characterize the inner structure of structural materials in order to define microstructural formations and the effects of microstructural phases such as pores on macro properties.

New image capturing tools and their improved magnification capacity induced researchers to have an expanded view on investigation of microstructures. In addition, the results of these studies are simply not enough to realize the simulation of effects of inner structure. Some numerical and statistical methods performed by computers are needed at this stage. Artificial neural network (ANN) is one of these methods. In last decades, artificial neural network applications have become more considerable issue in engineering applications. In the scope of this chapter, pore area ratio values represent total pore area amount in a polished section of cement mortars were determined. Also, some pore characteristics representing the probability of channels between pores are investigated. The pore amounts and these pore characteristics are related to compressive strength values of cement mortars in order to establish a microstructure – macro property relationship. Thus, nondestructive methodologies and artificial neural network have been used in the prediction of a macro property, which only be determined by destructive testing techniques.

Chapter 12 – The concept of the drainage divide in a flat country can be taken as a metaphor of health vs disease in a medical setting. In the medical landscape the authors could consider spring location as the equivalent of genetic background at birth. Rivers become people's life trajectories. The seas the rivers ultimately flow into are the outcomes, for example healthy aging vs. premature death due to chronic disease. Depending on the genetic background (starting point), the influence of life events or particular life styles (hills, peaks, ridges) on the fate of the person (river) would be negligible or determinant in inducing a particular trajectory to a specific outcome.

If the principal aim of predictive medicine is to predict the future direction of a person's life in terms of health on the basis of available data, then in this metaphor the problem is to predict what direction the river will take, given its spring location in a particular country and the environmental data available. Two rivers whose "springs" are very close to each other can easily flow in opposite directions.

At present, despite the incredible development of genetic testing technologies, defining the role of genetic predisposition of a subject in determining future outcome still is an elusive target.

In other words, at present, with the exception of extreme situations, the authors are not able to translate efficiently the precise location of individual springs – in the prediction of the river path . The essay discuss how the use of newer mathematical approaches like those inherent to artificial neural networks environment the next future could really offer the chance to trace the health – disease drainage divide. This is the future challenge for predictive medicine.

Chapter 13 – Performance of a treatment plant highly depends on plant's operation and it is generally difficult to predict the performance due to several uncertainties. Also, many of activated sludge processes produce poor effluent quality due to escape of sludge from secondary clarifier as a result of excess filamentous growth. In this context, this study aims at modeling a real scale activated sludge process using a popular artificial neural network (ANN) to make its management easier. Effluent COD, effluent BOD_5 and SVI were used as model output parameters. The developed ANN model was very successful as an excellent to reasonable match was obtained between the measured and the predicted parameters of effluent COD ($R = 0.90$), effluent BOD_5 ($R = 0.83$) and SVI ($R=0.84$). Hence, the ANN based model can be used to predict a full scale wastewater treatment plant performance and to control the operational conditions for improved process performance.

Chapter 14 – This chapter is devoted to solve the positioning control problem of under-actuated robot manipulator. Artificial Neural Networks Inversion technique was used where a network representing the forward dynamics of the system was trained to learn the position of the passive joint over the working space of a 2R under-actuated robot. The obtained weights from the learning process were fixed and the network was inverted to represent the inverse dynamics of the system, and then used in the estimation phase to estimate the position of the passive joint for a new set of data the network was not previously trained for in order to show the success of the control strategy.

Data used in this research are recorded experimentally from sensors fixed on the robot joints in order to overcome whichever uncertainties presence in the real world such as ill-defined linkage parameters, links flexibility and backlashes in gear trains.

The technique was implemented in two phases, the first phase was the forward learning phase that used to obtain the training weights which are used in the second phase which is the inverse estimation phase that is used to estimate the passive joint's position for any set of data the network was not trained for. The results were verified experimentally to show the ability of the proposed technique to solve the problem efficiently.

Chapter 15 – Time series forecasting is a vital issue for many institutions. In the literature, many researchers from various disciplines have tried to improve forecasting models to reach more accurate forecasts. It is known that real life time series has a nonlinear structure in general. Therefore, conventional linear methods are insufficient for real life time series. Some methods such as autoregressive conditional heteroskedastiacity (ARCH) and artificial neural networks (ANN) have been employed to forecast nonlinear time series. ANN has been successfully used for forecasting nonlinear time series in many implementations since ANN can model both the linear and nonlinear parts of the time series. In this study, a novel hybrid forecasting model combining seasonal autoregressive integrated moving average (SARIMA), ARCH and ANN methods is proposed to reach high accuracy level for nonlinear time series. It is presented how the proposed hybrid method works and in the implementation, the proposed method is applied to the weekly rates of TL/USD series between the period January 3, 2005 and January 28, 2008. This time series is also forecasted by using other approaches available in the literature for comparison. Finally, it is seen that the proposed hybrid approach has better forecasts than those calculated from other methods.

Chapter 16 – Evolving Connectionist Systems (ECoS) are a class of constructive artificial neural networks that grow their structure as they learn. They have been widely applied to many problems. Among their advantages are fast, efficient training and a resistance to catastrophic forgetting. An attempt has previously been made to formally describe their operation and behaviour. This formalisation has some objections associated with them. This chapter describes the basic algorithms behind ECoS networks, critiques the previous formalisation and presents a new theory of ECoS networks that overcomes the objections associated with the previous work. Finally, the formalisation is tested on two well-known benchmark data sets.

Chapter 17 – This chapter studies a class of neuron models that computes a user-defined similarity function between inputs and weights. The neuron transfer function is formed by composition of an adapted logistic function with the quasi-linear mean of the partial input-weight similarities. The neuron model is capable of dealing directly with mix- tures of continuous as well as discrete quantities, among other data types and there is provision for missing values. An artificial neural network using these neuron models is trained using a breeder genetic algorithm until convergence. A number of experiments are carried out in several real-world problems in very different application domains described by mixtures of variales of distinct types and eventually showing missing values. This heterogeneous network is compared to a standard radial basis function network and to a multi-layer perceptron networks and shown to learn from with su- perior generalization ability at a comparable computational cost. A further important advantage of the resulting neural solutions is the great interpretability of the learned weights, which is done in terms of weighted similarities to prototypes.

Chapter 18 – The efficacy of an optimization method often depends on the choosing of a num- ber of behavioural parameters. Research within this area has been focused on devising schemes for adapting the behavioural parameters during optimization, so as to alleviate the need for a practitioner to select the parameters manually. But these schemes usually introduce new behavioural parameters that must be tuned. This study takes a different approach in which finding behavioural parameters that yield good performance is con- sidered an optimization problemin its own right and can therefore be attempted solved by an overlaid optimization method. In this work, variants of the general purpose op- timization method known as Differential Evolution have their behavioural parameters tuned so as to work well in the optimization of an Artificial Neural Network. The re- sults show that DE variants using so-called adaptive parameters do not have a general performance advantage as previously believed.

Chapter 19 – The modern drug discovery has entered a realm wherein artificial intelligence plays a pivotal role in handling the huge overflow of the data and extracting vital information related to structure-activity/property relationships and rational drug design. In this communication, the authors present an overview of applications of Artificial Neural Networks (ANN) specifically in the drug discovery arena of Acquired Immunodeficiency Syndrome (AIDS) with special reference to modeling and design of Non-nucleoside Reverse Transcriptase Inhibitors (NNRTIs) of Human Immunodeficiency Virus Type-1 (HIV-1). Although both linear and non-linear techniques are appropriate in distinguishing the biologically active from the inactive compounds, ANN has consistently shown better predictive ability in case of training and test data sets both, thus can serve as an excellent tool for statistical modeling. Also, a comparison of ANN technique with linear and other non-linear techniques is portrayed in brief. Recent trends in application of ANN technique in drug development for AIDS are also discussed in the communication. The flexibility of the neural architecture serves in better understanding of the trained dataset. Since the last two decades, ANN has emerged as a tool to enhance a descriptive model, though the higher computational cost encountered in deriving a neural network compared to linear methods might be of concern, but certainly not a limiting factor.

Chapter 20 – The objective of this paper is to analyze the theme of the application of the intelligent learning systems (such as artificial neural networks, expert systems, fuzzy models and genetic algorithms) in the pricing in banking and finance.

All firms must settle a price for the services or products which they offer. The price is an important element of the marketing mix, being also an important source of income for the firm. As the competition in the financial systems has intensified, nowadays the settlement of correct prices has become an essential element for the marketing strategy of a bank.

The price must not be considered as a purely financial problem, calculated by estimating only the costs to which a margin for profit will be added. The settlement of the price must consider also the stakeholders point of view.

A peculiarity of the pricing in banking is also a partial lack of transparency which make difficult to understand the variables to analyze in order to forecast the phenomenon.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 1

A NEURAL NETWORK BASED VISUAL SERVO SYSTEM: LEARNING VISUAL KINEMATICS OF A SCENE

Ebrahim A. Mattar Al-Gallaf[*]
Associate Professor of Intelligent Control, Department of Electrical
and Electronics Engineering, College of Engineering,
University of Bahrain, Kingdom of Bahrain

ABSTRACT

Robotics arm visual servo system does suffer from a number of issues, as related to speed and complexity. One of which is the complicated kinematics relations, in addition the needed computational time to execute a task. This manuscript highlights a mechanism through which to approximate the inter-related visual kinematics relations that are part of visual servo closed loop system through an artificial neural network system. A main issue that hinders visual servo system is related to the complicated and time variant feature Jacobian matrix. The methodology followed here is based on the concept of integration of ANN with an Image Based Visual Servoing (IBVS) system. Artificial neural networks have been employed here to learn and approximate the relations that relate an object moving to a robotics arm movement through a visual servo. For validating the presented concept, one of the closed loop visual servo system already developed, have been used here to verify the presented concept. A learning Artificial Neural Network has proven to be an effective approach in learning the nonlinear relationships that govern kinematics relations used in robotics arm visual servo. In this respect, this chapter will explore how a trained artificial neural net will be used to achieve a very precise robotics arm movement by extracting the visual information from a moving object.

[*] Corresponding author: ebrgallaf@eng.uob.bh

I. Introduction

Visual servo is a technique that is used to control a motion of a robotics system using the visual information to achieve a task. Visionary data is acquired from a camera that is mounted directly on a robot manipulator or on a mobile robot, in which case, motion of the robot induces camera motion. Differently, the camera can be fixed, so that can observe the robot motion. In this sense, visual servo control relies on techniques from image processing, computer vision control theory, kinematics, dynamic and real time computing. Robotics visual servoing has been recently introduced by robotics, AI and control communities. This is due to the significant number of advantages over blind robotic systems. Researchers have been demonstrated that visual servoing is an effective and a robust framework to control robotics systems while relying on visual information as feedback. An image-based scheme task is said to be completely performed if a desired image is acquired by a robotic system. Numerous advances in robotics have been inspired by the biological systems.

Visual servoing aims to control a robotics system through artificial vision in a way as to manipulate an environment, comparable to humans actions. Intelligence-based visual control has also been introduced by research community as a way to supply robotics system even with more cognitive capabilities, [2]. A number of research on the field of intelligent visual robotics arm control have been introduced. Visual servoing has been classified as using visual data within a control loop, enabling visual-motor (hand-eye) coordination. There have been different structures of visual servo systems. However, the main two classes are; Position-based visual servo systems (PBVS), and the Image-based visual servo systems (IBVS). In this chapter, we shall concentrate on the second class, which is the Image-based visual servo systems.

An Image Based Visual Servoing using Takagi-Sugeno fuzzy neural network controller has been proposed by Miao et. al. [3]. In this chapter, a Takagi-Sugeno Fuzzy Neural Network Controller (TSFNNC) based Image Based Visual Servoing (IBVS) method is proposed. Firstly, the eigenspace based image compression method is explored, which is chosen as the global feature transformation method. After that, the inner structure, performance and training method of T-S neural network controller are discussed respectively. Besides, the whole architecture of the TS-FNNC is investigated.

Panwar and Sukavanam in [4] have introduced Neural Network Based Controller for Visual Servoing of Robotic Hand Eye System. For Panwar and Sukavanam, in their paper a neural network based controller for robot positioning and tracking using direct monocular visual feedback is proposed. Visual information is provided using a camera mounted on the end-effector of a robotics manipulator. A PI kinematics controller is proposed to achieve motion control objective in an image plane. A Feedforward Neural Network (FFNN) is used to compensate for the robot dynamics. The FFNN computes the required torques to drive the robot manipulator to achieve desired tracking. The stability of combined PI kinematics and FFNN computed torque is proved by Lyapunov theory.

Luis and Carlos [5], have prsented a research throught which a new control scheme for visual servoing that takes into account the delay introduced by image acquisition and image processing. The capabilities (steady-state errors, stability margins, step time response, etc.) of the proposed control scheme and of previous ones are analytically analyzed and compared. Several simulations and experimental results were provided to validate the analytical results and to illustrate the benefits of the proposed control scheme. In particular, it was shown that this new control scheme clearly improves the performance of the previous ones.

Alessandro et. al. [6], in their paper, they proposed an image-based visual servoing framework. Error signals are directly computed from image feature parameters, thus obtaining control schemes which do not need neither a 3-D model of the scene, nor a perfect knowledge of the camera calibration matrix. Value of the depth Z for each considered feature must be known. Thus they proposed a method to estimate on-line the value of Z for point features while the camera is moving through the scene, by using tools from nonlinear observer theory. By interpreting Z as a continuous unknown state with known dynamics, they build an estimator which asymptotically recovers the actual depth value for the selected feature.

In [7] an adaptive visual servo regulation control for camera-in-hand configuration with a fixed camera extension was presented by Chen, et. al. In this chapter, image-based regulation control of a robot manipulator with an uncalibrated vision system is discussed. To compensate for the unknown camera calibration parameters, a novel prediction error formulation is presented. To achieve the control objectives, a Lyapunov-based adaptive control strategy is employed. The control development for the camera-in-hand problem is presented in detail and a fixed-camera problem is included as an extension. Epipolar Geometry Toolbox [8], was also created to grant MATLAB users with a broaden outline for a creation and visualization of multi-camera scenarios. In addition, to be used for the manipulation of visual information, and the visual geometry. Functions provided, for both class of vision sensing, the pinhole and panoramic, include camera assignment and visualization, computation, and estimation of epipolar geometry entities. Visual servoing has been classified as using visual data within a control loop, enabling visual-motor (hand-eye) coordination. Image Based Visual Servoing Using Takagi-Sugeno Fuzzy Neural Network Controller has been proposed by Miao et. al. [9]. In their study, a T-S fuzzy neural controller based IBVS method was proposed. Eigenspace based image compression method is firstly explored which is chosen as the global feature transformation method. Inner structure, performance and training method of T-S neural network controller are discussed respectively. Besides that, the whole architecture of TS-FNNC is investigated. For robotics arm visual servo, this issue has been formulated as a function of object feature Jacobian. Feature Jacobian is a complicated matrix to compute for real-time applications. For more feature points in space, the issue of computing inverse of such matrix is even more hard to achieve.

1.2. Main Contribution

This manuscript is presenting a research framework which was oriented to develop a robotics visual servo system that relies on approximating complicated nonlinear visual servo kinematics. It concentrates on approximating differential visual changes (object features) relations relating objects movement in a world space to object motion in a camera space (usually time-consuming relations as expressed by time-varying Jacobian matrix), hence to a robotics arm joint space. The research is also presenting how a trained Neural Network can be utilized to learn the needed approximation. The research whole concept is based on utilizing and mergence three fundamentals. The first is a robotics arm-object visual kinematics, the second is the Epipolar Geometry relating different object scenes during motion, and a learning artificial neural system. To validate the concept, the visual control loop algorithm developed by Rives has been thus adopted to include a learning neural system. Results have indicated that, the proposed visual servoing methodology was able to produce a considerable accurate results.

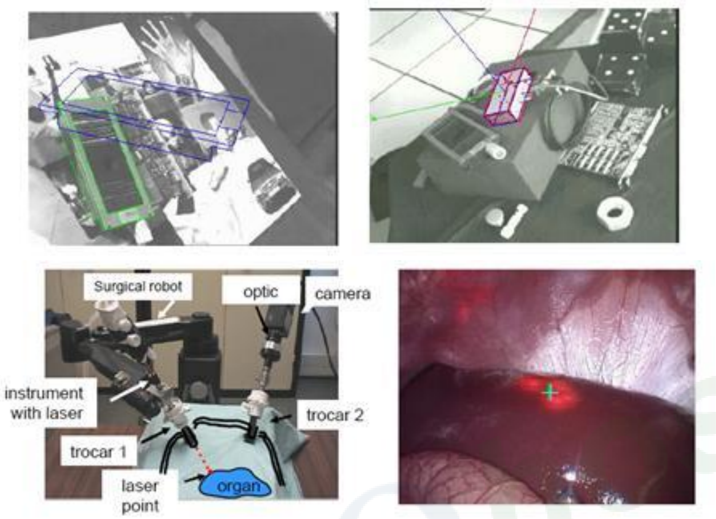


Figure 1. Visual servoing has a number of applications in medical and other industrial fields fields, [1].

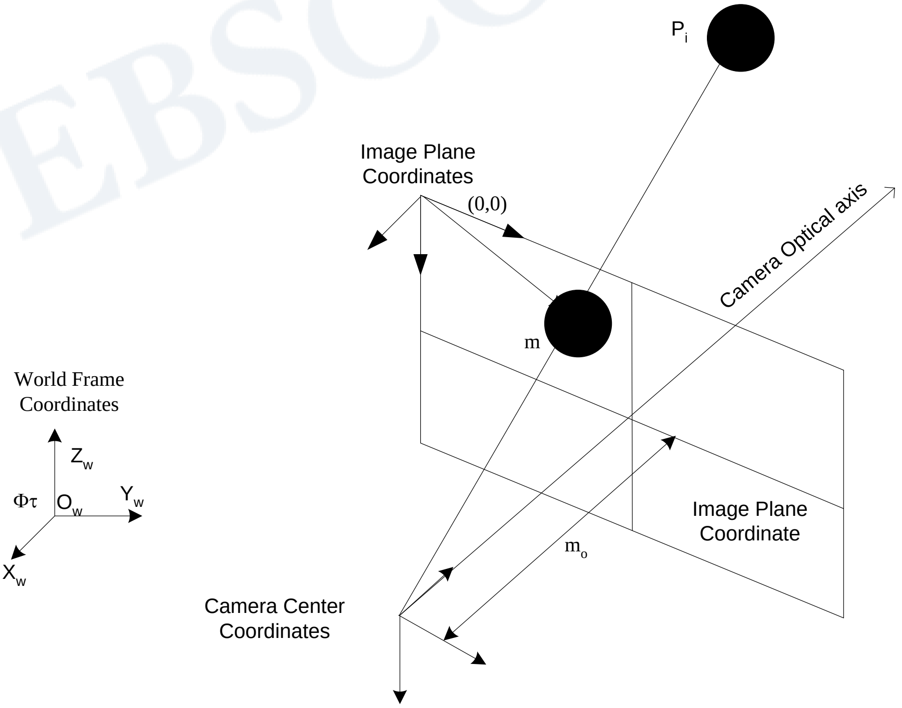


Figure 2. Pinhole Camera Geometry.

1.3. Chapter Organization

This manuscript has been sub-divided into six main sections. In section (1) we introduce concept of robotics visual servo background as related to visual servo. In section (2), we present a background for single scene via signal camera system. Double scene, as known as Epipolar Geometry is presented in section (3). Neural based IBVS is presented in Section (4), whereas simulated Rives algorithm is presented in section (5). Section (5) presents a case study and a simulation result of the modified Rives algorithm. Finally, Section (6) presents the chapter conclusions.

II. Single Camera Model : Object to Camera Plane Kinmeatics

To assemble a closed loop visual servo system, a loop is to be closed around the robotics system. In this study, this is a Puma-560 robotics arm, with a Pinhole camera system. For analyzing closed loop visual kinematics, we shall employ a Pinhole Camera Model for capturing object features. For whole representation, details of a Pinhole camera model in terms of image plane (ξ^a , ψ^a) location are expressed in terms (ξ , ψ , and ζ), as in Equ. (1):

In reference to Figure (2), we can express (ξ^a , ψ^a) locations as expressed in terms (ξ , ψ , and ζ) :

$$\begin{cases} \xi^a = \phi^a \dfrac{\xi}{\zeta} \\ \psi^a = \phi^a \dfrac{\Psi}{\zeta} \end{cases} \tag{1}$$

Both (ξ^a , ψ^a) location over an image plane is thus calculated by Equ. (1). For thin lenses (as the Pinhole camera model), camera geometry are thus represented by, [10] :

$$\frac{1}{\zeta^a} - \frac{1}{\zeta} = \frac{1}{\phi} \tag{2}$$

In reference to [10], using Craig Notation, ${}^{B}P$ denotes coordinate of point P in frame B. For translation case :

$${}^{B}P = {}^{A}P + {}^{B}O_A \tag{3}$$

${}^{B}O_A$ is a coordinate of the origin O_A of frame (A) in a new coordinate system (B). Rotations are given :

$$ {}^{B}_{A}R = \left({}^{B}i_{A}\,{}^{B}j_{A}\,{}^{B}k_{A}\right) = \begin{pmatrix} {}^{A}i_{B}^{T} \\ {}^{A}j_{B}^{T} \\ {}^{A}k_{B}^{T} \end{pmatrix} \tag{4} $$

In Equ (4), ${}^{B}i_{A}$ is a frame axis coordinate of (A) in another coordinate (B). In this respect, for rigid transformation we have :

$$ {}^{B}P = {}^{B}_{A}R\,{}^{A}P $$

$$ {}^{B}P = {}^{B}_{A}R\,{}^{A}P + {}^{B}O_{A} \tag{5} $$

For more than one consecutive rigid transformations, (for example to frame C), i.e. form frames $A \to B \to C$, coordinate of point P in frame C can then be expressed by :

$$ \begin{aligned} {}^{B}P &= {}^{B}_{A}R({}^{B}_{A}R\,{}^{A}P + {}^{B}O_{A}) + {}^{C}O_{B} \\ {}^{B}P &= {}^{C}_{B}R\,{}^{B}_{A}R\,{}^{A}P + ({}^{C}_{B}R\,{}^{A}O_{A} + {}^{C}O_{A}) \end{aligned} \tag{6} $$

For homogeneous coordinates, it looks very concise to express ${}^{B}P$ as :

$$ \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}R & {}^{B}O_{A} \\ O^{T} & 1 \end{bmatrix} \begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} \tag{7} $$

$$ \begin{bmatrix} {}^{C}P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{C}_{B}R & {}^{C}O_{B} \\ O^{T} & 1 \end{bmatrix} \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} \tag{8} $$

$$ \begin{bmatrix} {}^{C}P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{C}_{B}R & {}^{C}O_{B} \\ O^{T} & 1 \end{bmatrix} \begin{bmatrix} {}^{B}_{A}R & {}^{B}O_{A} \\ O^{T} & 1 \end{bmatrix} \begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} \tag{9} $$

We could express elements of a transformation Γ by writing :

$$ \Gamma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{bmatrix} = \begin{bmatrix} A & \Omega \\ O^{T} & 1 \end{bmatrix} \tag{10} $$

as becoming offline transformation. If ($A = R$), i.e., a rotation matrix (Γ), once $R^{T}R = I$, then :

$$ \Gamma = \begin{bmatrix} R & \Omega \\ O^{T} & 1 \end{bmatrix} \tag{11} $$

Euclidean transformation preserves parallel lines and angles, on the contrary, affine preserves parallel lines but not angles, Introducing a normalized image plane located at focal length $\phi = 1$. For this normalized image plane, pinhole (C) is mapped to the origin of an image plane using :

$$\hat{P} = (\hat{u} \quad \hat{v})^T \tag{12}$$

$\hat{C}$ and P are mapped to :

$$\hat{P} = \begin{bmatrix} \hat{u} \\ \hat{u} \\ 1 \end{bmatrix} = \frac{1}{\zeta^C} \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} P^C \\ 1 \end{bmatrix} \tag{13}$$

$$\hat{P} = \frac{1}{\zeta^C} \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \xi^c \\ \psi^c \\ \zeta^c \\ 1 \end{bmatrix} \tag{14}$$

we also have :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{\zeta^C} \begin{bmatrix} k\phi & 0 & u_0 \\ 0 & k\phi & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi^c \\ \psi^c \\ \zeta^c \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{\zeta^C} \begin{bmatrix} k\phi & 0 & u_0 \\ 0 & k\phi & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \xi^c \\ \psi^c \\ \zeta^c \\ 1 \end{bmatrix} \tag{15}$$

Now once $\kappa = k\phi$ and $\beta = L\phi$, then we identify these parameters κ, β, u_o, v_o as intrinsic camera parameters. In fact, they do present the inner camera imaging parameters. In matrix notation, this can be expressed as :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{\zeta^C} \begin{bmatrix} \kappa & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \xi^c \\ \psi^c \\ \zeta^c \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{\zeta^C} \begin{bmatrix} \kappa & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \Omega \\ O^T & 0 \end{bmatrix} \begin{bmatrix} \xi^v \\ \psi^v \\ \zeta^v \\ 1 \end{bmatrix} \tag{16}$$

Both (R) and (Ω) extrinsic camera parameters, do represent coordinate transformation between camera coordinate system and world coordinate system . Hence, any (u, v) point in camera image plan is evaluated via the following relation :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{\zeta^C} M_1 M_2 p^w \quad = \frac{1}{\zeta^C} M p^w \tag{17}$$

Here (M) in Equ (17) is referred to as a Camera Projection Matrix. We are given (a) a calibration rig, i.e., a reference object, to provide the world coordinate system , and (2) an image of the reference object. The problem is to solve (a) the projection matrix, and (b) the intrinsic and extrinsic parameters.

COMPUTING A PROJECTION MATRIX : In a mathematical way, we are given $(\xi_i^w \quad \psi_i^w \quad \zeta_i^w)$ and $(u_i \quad v_i)^T$ for *i* = (1 …… n), we want to solve for M_1 and M_2 :

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \frac{1}{\zeta^C} M_1 M_2 \begin{bmatrix} \xi_i^w \\ \psi_i^w \\ \zeta_i^w \\ 1 \end{bmatrix} = \frac{1}{\zeta^C} M \begin{bmatrix} \xi_i^w \\ \psi_i^w \\ \zeta_i^w \\ 1 \end{bmatrix} \tag{18}$$

$$\zeta_i^c = \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \end{bmatrix} \begin{bmatrix} \xi_i^w \\ \psi_i^w \\ \zeta_i^w \\ 1 \end{bmatrix} \tag{19}$$

$$\zeta_i^c u_i = \sigma_{21} \xi_i^w + \sigma_{22} \psi_i^w + \sigma_{23} \zeta_i^w + \sigma_{24} \tag{20}$$

$$\zeta_i^c u_i = \sigma_{31} \xi_i^w + \sigma_{32} \psi_i^w + \sigma_{33} \zeta_i^w + \sigma_{34}$$

Obviously, we can let $\sigma_{34} = 1$. This means, the projection matrix is scaled by σ_{34} . We have, $KM = U$, and $K \in \Re^{2n \times 11}$ is a matrix, a 11-D vector, and $U \in \Re^{2n-D}$ vector. A least square solution of equation $KM = U$ is thus given by :

$$M = K^+ U \text{ or } M = K^T K^{-1} K^T U \tag{21}$$

K^+ is the pseudo inverse of matrix K, and m and m_{34} consist of the projection matrix M. We now turn to double scene.

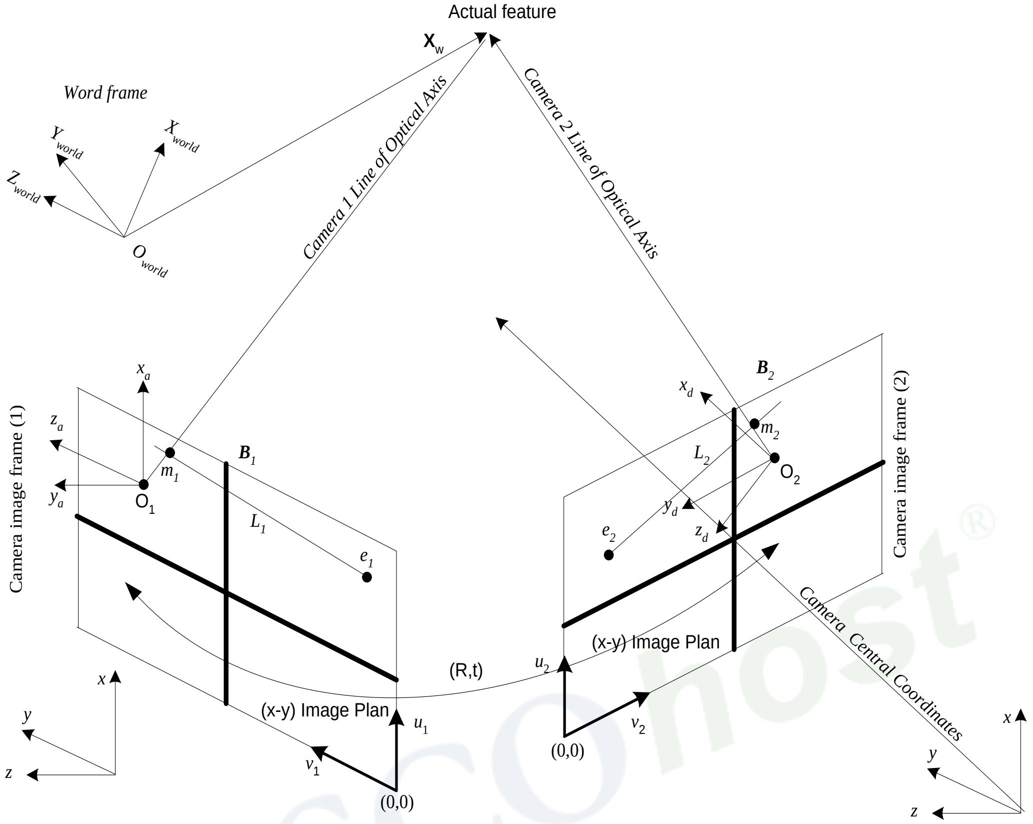


Figure 3. Camera image frame and Epipolar Geometry.

III. Double Camera Scene Analysis Epipolar Geometry

In this section, we shall consider an image resulting from two camera views. For two perspective views of the same scene taken from two separate viewpoints O_1 and O_2, this is illustrated in Figure (3). Also we shall assume that (m_1 and m_2) are representing two separate points in two the views. In other words, perspective projection through O_1 and O_2, of the same point X_w, in both image planes Λ_1 and Λ_2. In addition, by letting (c_1) and (c_2) be the optical centers of two scene, the projection E_1 (E_2) of one camera center O_1(O_2) onto the image plane of the other camera frame Λ_2(Λ_1) is the epipole geometry. We can expressed such an epipole geometry in homogeneous coordinates in terms $\tilde{E}_1$ and $\tilde{E}_2$:

$$\tilde{E}_1 = \begin{pmatrix} E_{1x} & E_{1y} & 1 \end{pmatrix}^T \text{ and } \tilde{E}_2 = \begin{pmatrix} E_{2x} & E_{2y} & 1 \end{pmatrix}^T \tag{22}$$

One of the main parameters of an epipolar geometry is the fundamental Matrix H (which is $\Re \in 3\times3$).

In reference to the H matrix, it conveys most of the information about the relative position and orientation (t, R) between the two different views. Moreover, the fundamental matrix H algebraically relates corresponding points in the two images through the Epipolar Constraint. For instant, let the case of two views of the same 3-D point X_{w}, both characterized by their relative position and orientation (t, R) and the internal, hence H is evaluated in terms of K_1 and K_2 (extrinsic camera parameters), [8] :

$$\mathrm{H} = \mathrm{K}_2^{-\mathrm{T}} [\mathrm{t}]_{\mathrm{x}} \mathrm{R} \mathrm{K}_1^{-1} \tag{23}$$

In such a case, a 3-D point (X_{w}) is projected onto two image planes, to points (m_2) and (m_1), as to constitute a conjugate pair. Given a point (m_1) in left image plane, its conjugate point in the right image is constrained to lie on the epipolar line of (m_1). The line is considered as the projection through C_2 of optical ray of m_1. All epipolar lines in one image plane pass through an epipole point. This is the projection of conjugate optical centre : $\tilde{\mathrm{E}}_1 = \tilde{\mathrm{P}}_2 \begin{pmatrix} \mathrm{c}_1 \\ 1 \end{pmatrix}$ Parametric equation of epipolar line of $\tilde{\mathrm{m}}_1$ gives $\tilde{\mathrm{m}}_2^{\mathrm{T}} = \tilde{\mathrm{E}}_2 + \lambda \mathrm{P}_2 \mathrm{P}_1^{-1} \tilde{\mathrm{m}}_1$. In image coordinates this can be expressed as :

$$\mathrm{u} = [\mathrm{m}_2]_1 = \frac{[\tilde{\mathrm{e}}_2]_1 + \lambda[\tilde{\mathrm{v}}]_1}{[\tilde{\mathrm{e}}_2]_3 + \lambda[\tilde{\mathrm{v}}]_3} \tag{24}$$

$$\mathrm{v} = [\mathrm{m}_2]_2 = \frac{[\tilde{\mathrm{e}}_2]_2 + \lambda[\tilde{\mathrm{v}}]_2}{[\tilde{\mathrm{e}}_2]_3 + \lambda[\tilde{\mathrm{v}}]_3} \tag{25}$$

here $\tilde{\mathrm{v}} = \mathrm{P}_2 \mathrm{P}_2^{-1} \tilde{\mathrm{m}}_1$ is a projection operator extracting the (i^{th}) component from a vector. When (c_1) is in the focal plane of right camera, the right epipole is an infinity, and the epipolar lines form a bundle of parallel lines in the right image. Direction of each epipolar line is evaluated by derivative of parametric equations listed above with respect to (λ) :

$$\frac{\mathrm{du}}{\mathrm{d}\lambda} = \frac{[\tilde{\mathrm{v}}]_1 [\tilde{\mathrm{e}}_2]_3 - [\tilde{\mathrm{v}}]_3 [\tilde{\mathrm{e}}_2]_1}{([\tilde{\mathrm{e}}_2]_3 + \lambda[\tilde{\mathrm{v}}]_3)^2} \tag{26}$$

$$\frac{\mathrm{dv}}{\mathrm{d}\lambda} = \frac{[\tilde{\mathrm{v}}]_2 [\tilde{\mathrm{e}}_2]_3 - [\tilde{\mathrm{v}}]_3 [\tilde{\mathrm{e}}_2]_2}{([\tilde{\mathrm{e}}_2]_3 + \lambda[\tilde{\mathrm{v}}]_3)^2} \tag{27}$$

The epipole is rejected to infinity once $[\tilde{\mathrm{E}}_2]_3 = 0$. In such a case, direction of the epipolar lines in right image doesn't depend on any more. All epipolar lines becomes parallel to vector $([\tilde{\mathrm{E}}_2]_1 \quad [\tilde{\mathrm{E}}_2]_2)^{\mathrm{T}}$. A very special occurrence is once both epipoles are at infinity. This happens once a line containing (c_1) and (c_2), the baseline, is contained in both focal planes, or the retinal planes are parallel and horizontal in each image as in Figure (3). The right pictures plot the epipolar lines corresponding to the point marked in the left pictures. This procedure is called rectification [8]. If cameras share the same focal plane the common retinal plane is constrained to be parallel to the baseline and epipolar lines are parallel.

IV. Neural Net Based Image-Based Visual Servo Control (ANN-IBVS)

Over the last section we have focused more in single and double camera scenes, i.e. representing the robot arm visual sensory. In this section, we shall focus on Image-Based Visual Servo (IBVS) which uses locations of object features on image planes (epipolar) for direct visual feedback. For instant, re-consider Figure (1), where it is desired to move a robotics arm in such away that camera's view changes from (initial) to (final) view, and feature vector from (ϕ_0) to (ϕ_d). Here (ϕ_0) may comprise coordinates of vertices, or areas of the object to be tracked. Implicit in (ϕ_d) is the robot is normal to, and centered over features of an object, at a desired distance. Elements of the task are thus specified in image space. For a robotics system with an end-effector mounted camera, viewpoint and features are functions of relative pose of the camera to the target, (${}^c\xi_t$). Such function is usually nonlinear and cross-coupled. A motion of end-effectors DOF results in complex motion of many features. For instant, a camera rotation can cause features to translate horizontally and vertically on the same image plane, as related via the following relationship :

$$\phi = f({}^c\xi_t) \qquad (28)$$

Equ (22) is to be linearized. This is to be done around an operating point :

$$\delta\phi = {}^f J_c({}^c x_t)\delta\, {}^c x_t \qquad (29)$$

$${}^f J_c({}^c x_t) = \frac{\partial\phi}{\partial^c x_t} \qquad (30)$$

In Equ (24), ${}^f J_c({}^c x_t)$ is the Jacobian matrix, relating rate of change in robot arm pose to rate of change in feature space. Variously, this Jacobian is referred to as the feature Jacobian, image Jacobian, feature sensitivity matrix, or interaction matrix [10]. Assume that the Jacobian is square and non-singular, then :

$${}^c\dot{x}_t = {}^f J_c({}^c x_t)^{-1}\dot{f} \qquad (31)$$

from which a control law can be expressed by :

$${}^c\dot{x}_t = [K][{}^f J_c({}^c x_t)^{-1}](f_d - f(t)) \qquad (32)$$

will tend to move the robotics arm towards desired feature vector. In Equ (26), K^f is a diagonal gain matrix, and (t) indicates a time varying quantity. Object posture rates ${}^c\dot{x}_t$ is converted to robot end-effector rates. A Jacobian , ${}^fJ_c({}^cx_t)$ as derived from relative pose between the end-effecter and camera, $({}^cx_t)$ is used for that purpose. In this respect, a technique to determine a transformation between a robot's end-effector and the camera frame is given by Tsai and Lenz [11] . In turn, an end-effector rates may be converted to manipulator joint rates using the manipulator's Jacobian [12], as follows :

$$\dot{\theta}_t = {}^{t6}J_\theta^{-1}(\theta)^{t6}\dot{x}_c \tag{33}$$

$\dot{\theta}_t$ represents the robot joint space rate. A complete closed loop equation can then be given by :

$$\dot{\theta}_t = K^{t6}J_\theta^{-1}(\theta)^{t6}J_\theta^f J_c^{-1\,c}x_t(f_d - f(t)) \tag{34}$$

For achieving this task, an analytical expression of the error function is given by :

$$\varphi = Z^+\varphi_1 + \gamma(I_6 - Z^+Z)\frac{\partial\varphi_2}{\partial X} \tag{35}$$

Here, $\gamma \in \Re^+$ and Z^+ is pseudo inverse of the matrix Z, $Z \in \Re^{m\times n} = \Re(Z^T) = \Re(J_1^T)$ and J is the Jacobian matrix of task function as $J = \frac{\partial\varphi}{\partial X}$. Due to modeling errors, such a closed-loop system is relatively robust in a possible presence of image distortions and kinematics parameter variations of the Puma 560 kinematics. A number of researchers also have demonstrated good results in using this image-based approach for visual servoing. It is always reported that, the significant problem is computing or estimating the feature Jacobian, where a variety of approaches have been used [12]. The proposed IBVS structure of Weiss [13,14], controls robot joint angles directly using measured image features. Non-linearities include manipulator kinematics and dynamics as well as the perspective imaging model. Adaptive control was also proposed, since ${}^fJ_\theta^{-1}({}^c\theta)$, is pose dependent, [15]. In this study, changing relationship between robot pose, and image feature change is learned during the motion via a learning neural system. The learning neural system accept s a weighted set of inputs (stimulus) and responds.

Artificial Neural Networks : Biological Inspiration

Animals are able to react adaptively to changes in their external and internal environment, and they use their nervous system to perform these behaviours. An appropriate model/simulation of the nervous system should be able to produce similar responses and behaviours in artificial systems. The nervous system is build by relatively simple units, the neurons, so copying their behaviour and functionality should be the solution [16]. The human brain is a part of the central nervous system, it contains of the order of 10^{+10} neurons. Each can activate in approximately 5ms and connects to the order of 10^{+4} other neurons giving 10^{+14} connections as shown in Figure (4.2) , [17].

The neuron as shown in Figure (4) made of, [16]:

- Synapses: Gap between adjacent neurons across which chemical signals are transmitted: (known as the input) .
- Dendrites: Receive synaptic contacts from other neurons.
- Cell body / soma: Metabolic centre of the neuron: processing.
- Axon: Long narrow process that extends from body: (known as the output).

Information transmission happens at the synapses as shown in Figure (5). The spikes travelling along the axon of the pre-synaptic neuron trigger the release of neurotransmitter substances at the synapse. The neurotransmitters cause excitation or inhibition in the dendrite of the post-synaptic neuron. The integration of the excitatory and inhibitory signals may produce spikes in the post-synaptic neuron. The contribution of the signals depends on the strength of the synaptic connection [16]. An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANN system, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANN system as well [18].

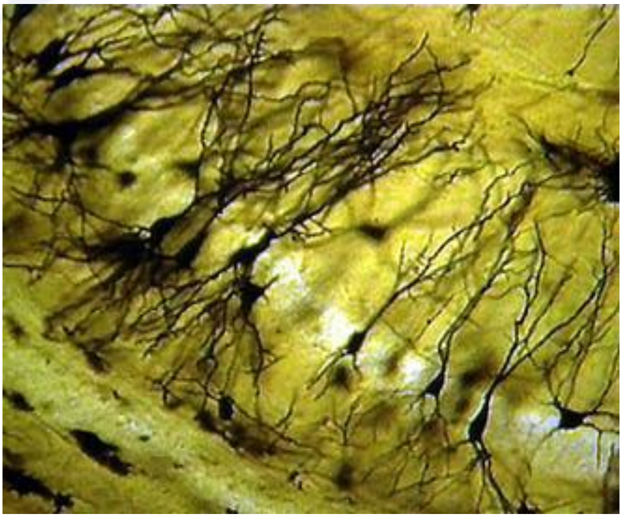


Figure 4. Human Brain nerve Systems (Nuerons), [16].

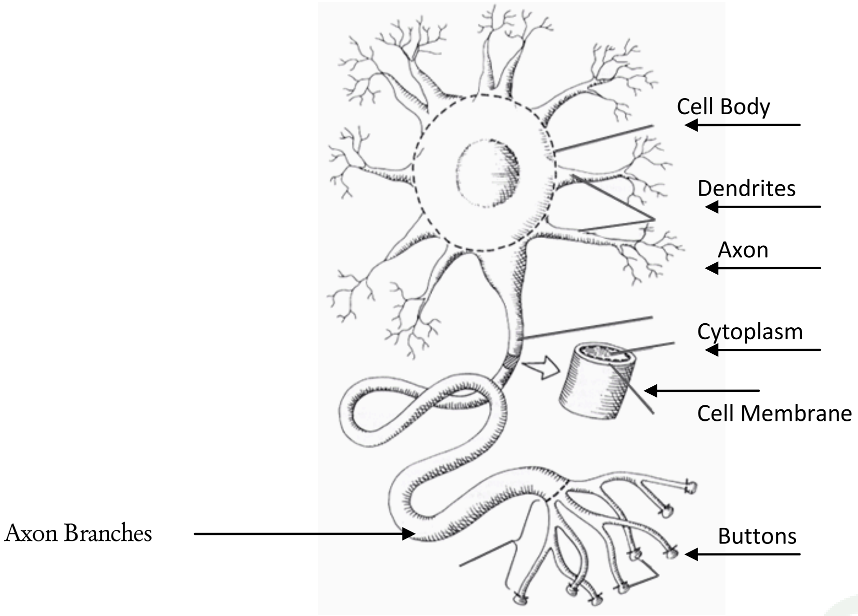


Figure 5. Multipolar Neuron, [16].

The four-layer feed-forward neural network with (n) input units, (m) output units and N units in the hidden layer, shown in the Figure (8), and as will be further discussed later. Figure (8). exposes a one possible neural network architecture that have been used. In reference to the Figure (8), every node is designed in such away to mimic its biological counterpart, the neuron. Interconnection of different neurons forms an entire grid of the used ANN that have the ability to learn and approximate the nonlinear visual kinematics relations. The used learning neural system composes of four layers. The input, output layers, and two hidden layers.

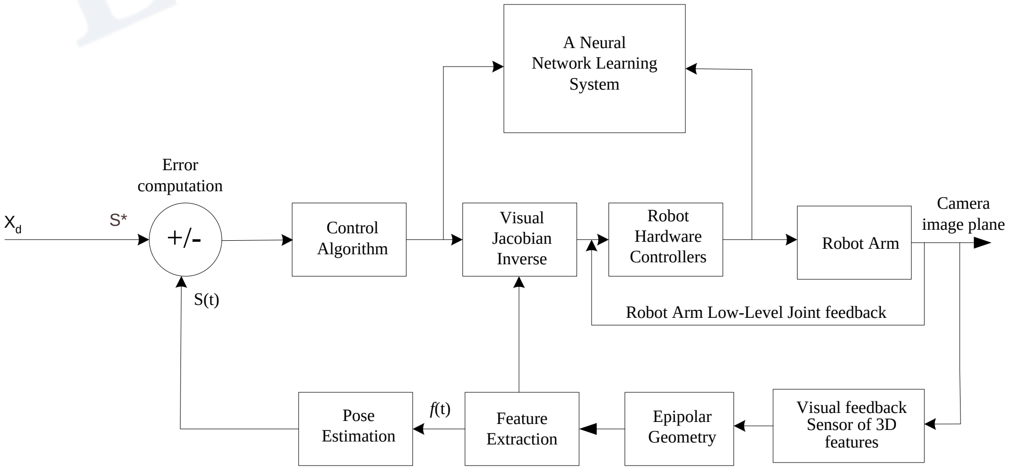


Figure 6. Neural system based Visual servo.

If we denote ${}^{w}\nu_{c}$ and ${}^{w}\omega_{c}$ as the camera's linear and angular velocities with respect to the robot frame respectively, motion of the image feature point as a function of the camera velocity is obtained by :

$$\dot{\gamma} = -\frac{\alpha\lambda}{{}^{c}p_{c}} \begin{bmatrix} 0 & 0 & \frac{{}^{c}p_{x}}{{}^{c}p_{z}} & \frac{{}^{c}p_{x}\,{}^{c}p_{x}}{{}^{c}p_{z}} & -\frac{{}^{c}p_{x}\,{}^{c}p_{x}}{{}^{c}p_{z}} & {}^{c}p_{x} \\ 1 & -1 & \frac{{}^{c}p_{y}}{{}^{c}p_{z}} & \frac{{}^{c}p_{x}\,{}^{c}p_{x}}{{}^{c}p_{z}} & -\frac{{}^{c}p_{x}\,{}^{c}p_{x}}{{}^{c}p_{z}} & -{}^{c}p_{x} \end{bmatrix} \begin{bmatrix} {}^{c}R_{w} & 0 \\ 0 & {}^{c}R_{w} \end{bmatrix} \begin{bmatrix} {}^{w}\nu_{c} \\ {}^{w}\omega_{c} \end{bmatrix} \tag{30}$$

Instead of using coordinates ${}^{x}P_{c}$ and ${}^{y}P_{c}$ of the object feature described in camera coordinate frame, which are a priori unknown, it is usual to replace them by coordinates (u) and (v) of the projection of such a feature point onto the image frame.

V. Simulation Visual Servoing with Pin-Hole Camera for 6-DOF Robotics Arm (A Case Study)

Visual servoing with a pin-hole camera for a 6-DOF robotics arm is simulate here. The system under study and simulation is shown Figure (6). During simulations the task has been performed using 6-DOF Puma manipulator with 6 revolute joints and a camera that can provide position information of the robot tip and the target in the robot workplace. The robot direct kinematics is given by the set of equations of Puma 560 robotics system, as documented in [15]. Kinematics and dynamics equations are already well known in the literature. For the purpose of comparison, the used example is based on visual servoing system developed by Rives [1]. The robotics system are has been servoing to follow an object that is moving in a 3-D working space. Object has been characterized by (8-features) marks, this has resulted in 24, $\Re \in^{8\times3}$ size, feature Jacobian matrix. This is visually sown in Figure (7). The object 8-features will be mapped to the movement of the object in the camera image plane through defined geometries. Changes in features points, and the differentional changes in robot arm, constitute the data that will be used for training the ANN. The used ANN architecture is hence shown in Figure (8).

Visual DATA Generation

The foremost ambition of this visual servoing is to drive a 6-DOF robot arm, as simulated with Robot Toolbox [14], and equipped with a pin-hole camera, as simulated with Epipolar Geometry Toolbox, EGT [8], from a starting configuration toward a desired one using only image data provided during the robot motion. For the purpose of setting up the proposed method, Rives algorithm has been run a number of time before hand. In each case, the arm was servoing with different object posture and a desired location in the working space. The EGT function to estimate the fundamental matrix H, given U_1 and U_2, for both scenes in which U_1 and U_2 are defined in terms of eight (ξ, ψ, ζ) feature points :

$$U_1 = \begin{bmatrix} \xi_1^1 & \xi_1^2 & \cdots & \xi_1^8 \\ \psi_1^1 & \psi_1^2 & \cdots & \xi_1^8 \\ \zeta_1^1 & \zeta_1^2 & \cdots & \zeta_1^8 \end{bmatrix} \text{ and } U_2 = \begin{bmatrix} \xi_2^1 & \xi_2^2 & \cdots & \xi_2^8 \\ \psi_2^1 & \psi_2^2 & \cdots & \xi_2^8 \\ \zeta_2^1 & \zeta_2^2 & \cdots & \zeta_2^8 \end{bmatrix}$$

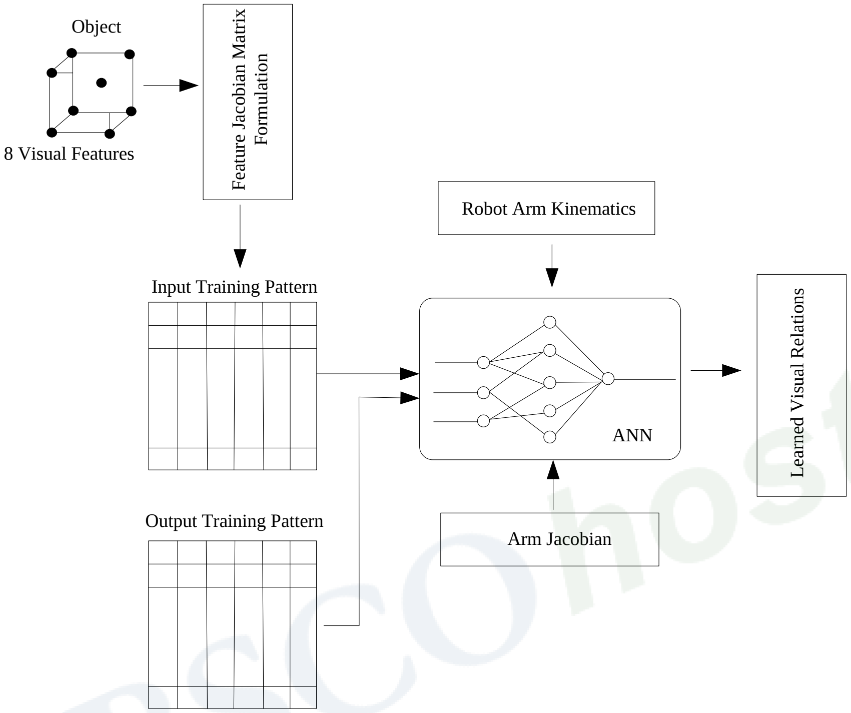


Figure 7. Features based data gathering.

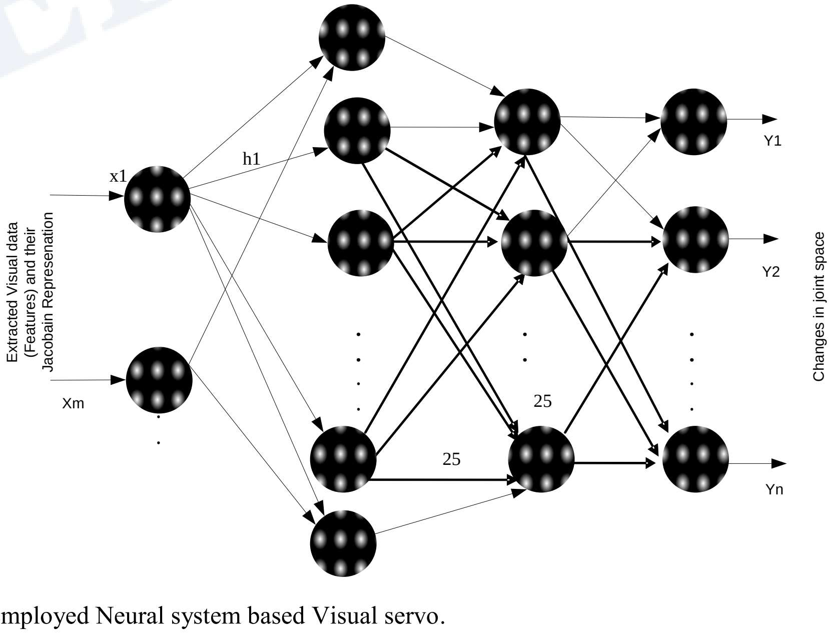

Extracted Visual data (Features) and their Jacobain Represenation
x1
h1
Xm
25
25
Y1
Y2
Yn
Changes in joint space

Figure 8. Employed Neural system based Visual servo.

Large training patterns have been gathered and classified, therefore. Gathered patterns at various loop locations gave an inspiration to a feasible size of learning neural system. Four layers artificial neural system has been found a feasible architecture for that purpose. The net maps 24 (3×8 feature points) inputs characterizing object cartesian feature position and arm joint positions into the (six) differential changes in arm joints positions. The network is presented with some arm motion in various directions. Once the neural system has learned with presented patterns and required mapping, it is ready to be employed in the visual servo controller. Trained neural net was able to map nonlinear relations relating object movement to differentional changes in arm joint space. Object path of motion was defined and simulated via Rives Algorithm, as given in [1], after such large number of running and patterns, it was apparent that the learning neural system was able to capture such nonlinear relations.

An Execution Phase

Execution starts primary while employing learned neural system within the robotics dynamic controller (which is mainly dependent on visual feature Jacobian). In reference to Figure (6), visual servoing dictates the visual features extraction block. That was achieved by the use of the Epipolar Toolbox. For assessing proposed visual servo algorithm, simulation of full arm dynamics has been achieved using kinematics and dynamic models for the Puma 560 arm. Robot Toolbox has been used for that purpose. In this respect, Figure (9) shows an aerial view of actual object posture and the desired one. This is prior to visual servoing to take place. The figure also indicates some scene features. Figure (10) shows the Robot arm-camera servoing and approaching towards a desired object posture. ANN was fed with defined pattern during arm movement. Epipolars have been used to evaluate visual features and the update during arm movement.

Object Visual Features Migration

Figure (11) shows error between the Rives Algorithm and the proposed ANN based visual servo for the first (60) iterations. Results suggest high accuracy of identical results, indicating that a learned neural system was able to servo the arm to desired posture. Difference in error was recorded within the range of (4×10^{-6}) for specific joint angles. Figure (12) shows migration of the eight visual features as seen over the camera image plan. Just the once the Puma robot arm was moving, concentration of features are located towards an end within camera image plane. In Figure (13), it is shown the object six dimensional movement. They indicate that they are approaching the zero reference. As an validaition of the enural network ability to servo the robotics arm toward the object, finally in Figure (14) we show that the trained ANN visual servo controller does approach zero level of movement, as for different training patterns for different arm postures.

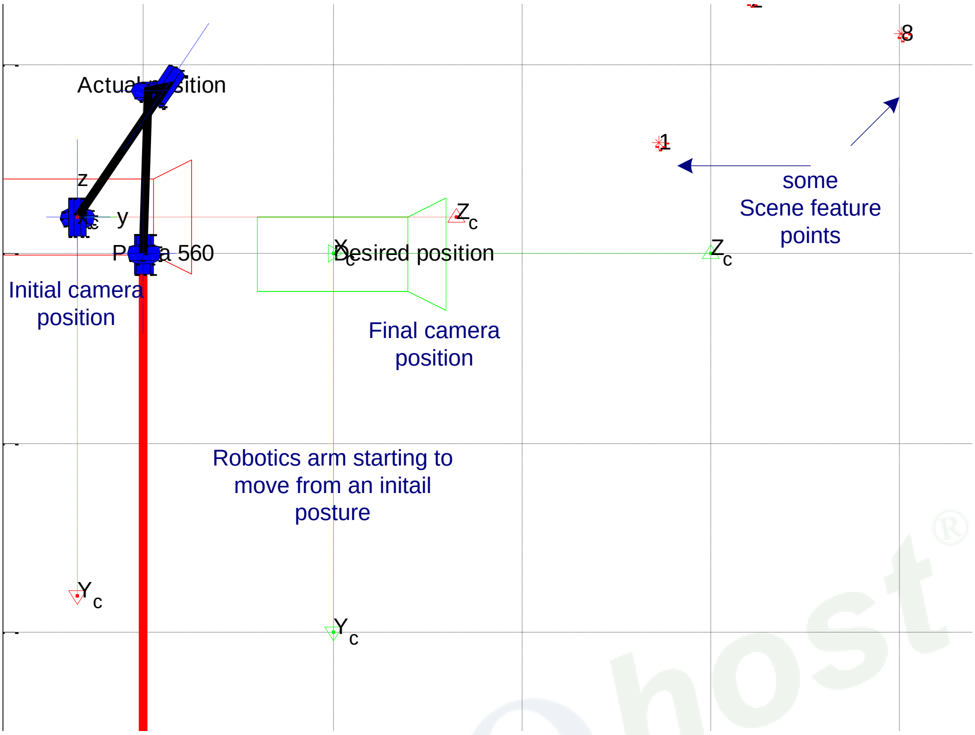


Figure 9. Top view. Actual object position and desired position before the servoing.

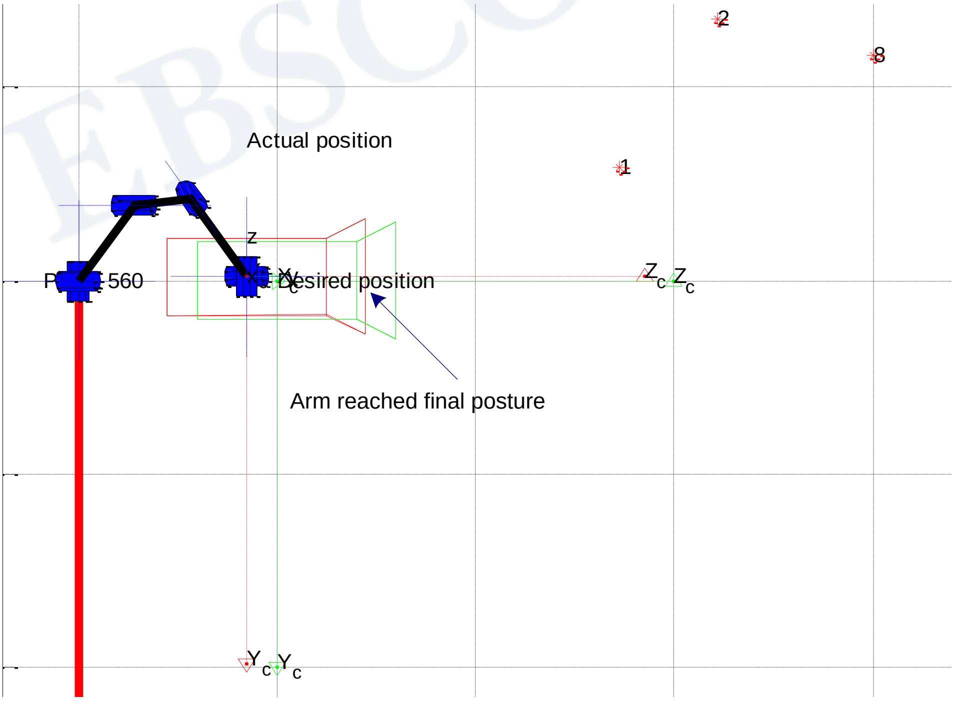


Figure 10. Robot am-camera servoing towards a desired object posture.

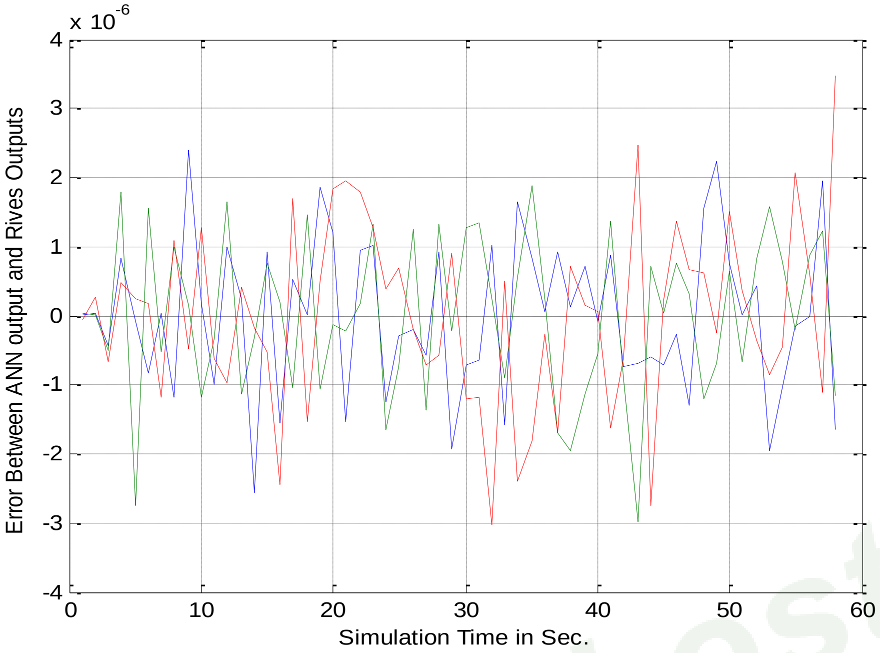


Figure 11. Error. Rives results and proposed ANN based visual servo.

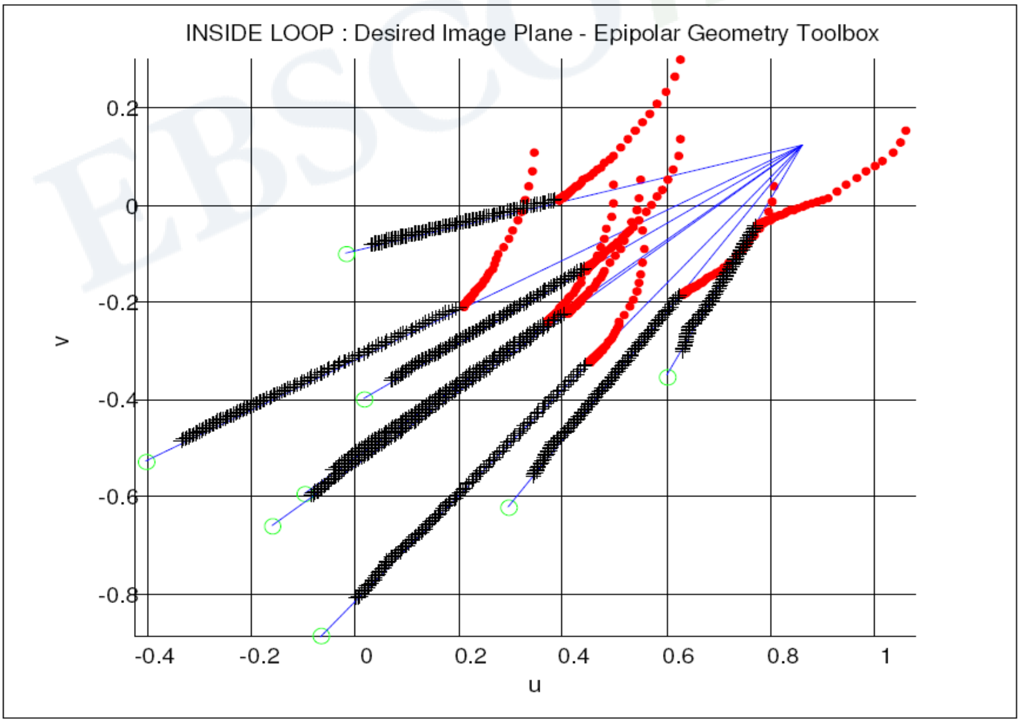


Figure 12. Migration of eight visual features as seen over the camera image plan.

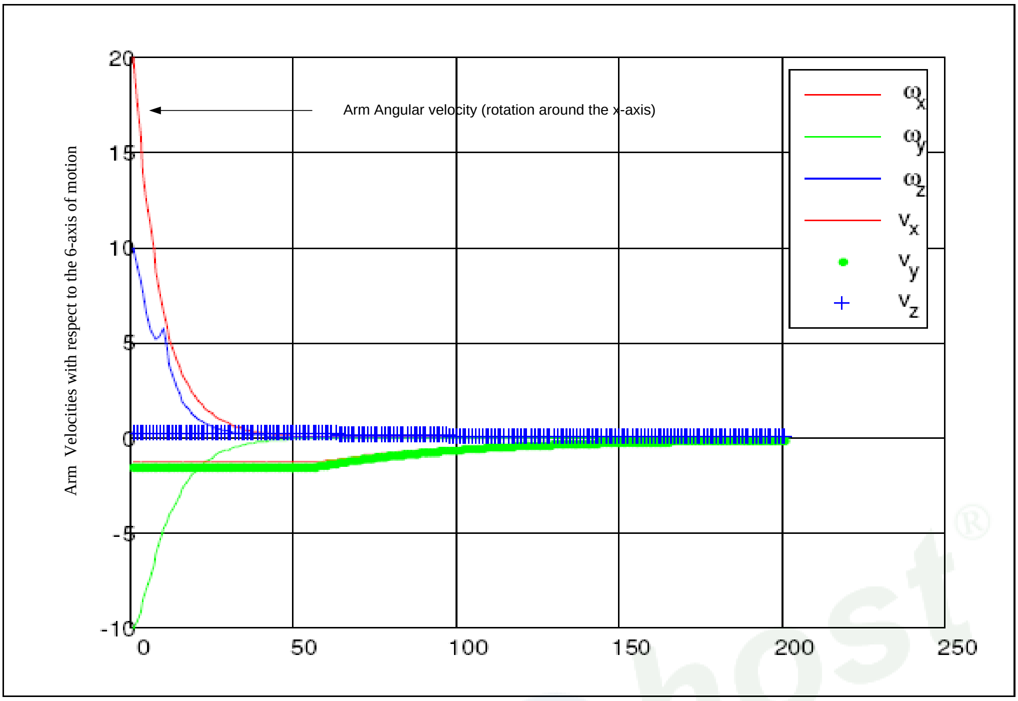


Figure 13. Object six dimensional movement. Approaching the zero reference.

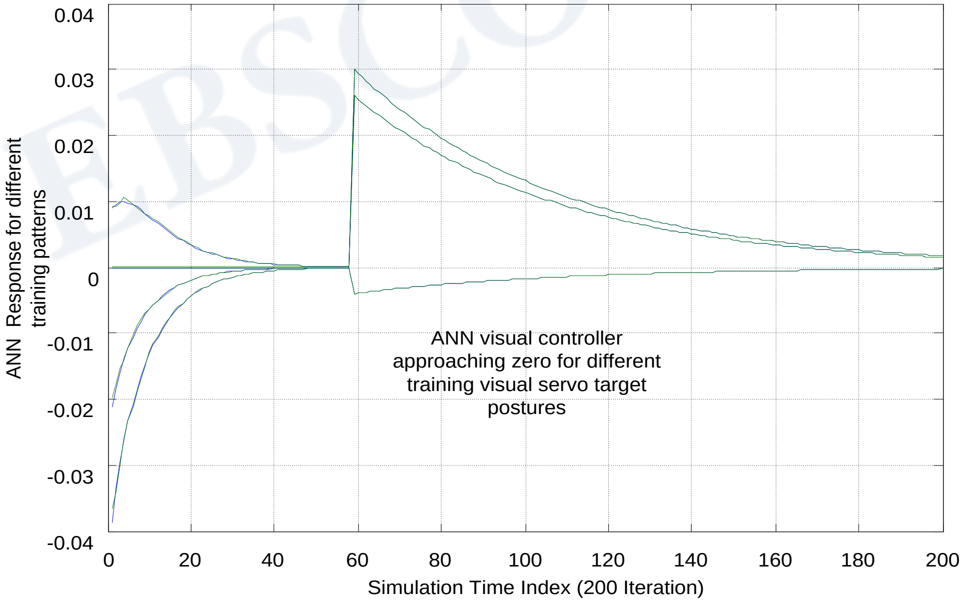


Figure 14. ANN visual servo controller approaching zero level for different training visual servo target postures.

CONCLUSIONS

Servoing a robotics arm towards an object movement using visual information is an issue that has been discussed by a number of research for the last twenty years. This manuscript has discussed a mechanism to learn the kinematics and feature-based Jacobian realtions that are used for robotics arm visual servo system. In this respect, the concept introduced has been based on the employment of an artificial neural network system trained to learn a mapping relating kinematics and changes in visual loop kinematics. Changes in a v loop visual Jocobain depends heavily on a robotics arm 3-D posture, in addition, it depends on features associated with an object under visual servo (to be tracked). Results have shown that, trained neural network can be used to learn the complicated visual relations relating an object movement to an arm joint space movement. The proposed methodology has also resulted in a great deal of accuracy. The proposed methodology was applied to the well-know Image Based Visual Servoing already discussed by Rives in [10].

REFERENCES

[1] Martinet, P. (2004). *“Applications in Visual Servoing”,* IEEE-RSJ IROS’04 Conference, September 28th , Japan.

[2] Perez Cisneros, M. (2004). *"Intelligent Model Structures in Visual servoing, "* Ph.D., University of Manchester, Institute for Science and Technology.

[3] Miao, H., Zengqi, S. & Fujii, M. (2007). "Image Based Visual Servoing Using Takagi-Sugeno Fuzzy Neural Network Controller, " *IEEE 22nd International Symposium on Intelligent Control*, ISIC 2007, Singapore, *vol.1*, Issue 3, 53-58.

[4] Panwar, V. & Sukavanam, N. (2007). "Neural Network Based Controller for Visual Servoing of Robotic Hand Eye System," *Engineering Letters, 14:1*, EL_14_1_26, Advance Online Publication.

[5] Luis Gracia & Carlos Perez-Vidal, (2009). "A New Control Scheme for Visual Servoing," *International Journal of Control, Automation, and Systems, vol. 7*, issue (5), pp. 764-776 DOI 10.1007/s12555-009-0509-9.

[6] Alessandro De Luca Giuseppe Oriolo Paolo Robuffo Giordano, (2007). " On-Line Estimation of Feature Depth for Image-Based Visual Servoing Schemes, " 2007 IEEE International *Conference on Robotics and Automation Roma*, Italy, *1014*, April.

[7] Chen, J., Dawson, M., Dixon, E. & Aman, B. (2008). " Adaptive Visual Servo Regulation Control for Camera-in-hand Configuration With a Fixed-Camera Extension, " Proceedings of the 46th IEEE Conference on Decision and Control, CDC, 2008, pp. 2339-2344, New Orleans, LA, United States.

[8] Eleonora, A., Gian, M. & Domenico, P. (2004). " Epipolar Geometry Toolbox, For Use with MATLAB, " *User Guide, vo1. 1*, December.

[9] Miao, H., Zengqi, S. & Masakazu, F. (2007). "Image Based Visual Servoing Using Takagi-Sugeno Fuzzy Neural Network Controller,", *22nd IEEE International Symposium on Intelligent Control Part of IEEE Multi-conference on Systems and Control Singapore,* 1-3.

[10] Gian, M., Eleonora, A. & Domenico, P. (2004). *"The Epipolar Geometry Toolbox (EGT) for MATLAB, "* Technical Report, 07-21-3-DII University of Siena, Siena, Italy,.

[11] Lenz, K. & Tsai, Y. (1988)." Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-hand Relationship, " Proceedings Computer Vision and Pattern Recognition, 1988 CVPR apos., *Computer Society Conference on vol*. no. 5, Issue 9, 67-75.

[12] Croke, P. (1994). " High-Performance Visual Closed-Loop Robot Control ", Thesis submitted in total fulfillment of the Requirements for the *Degree of Doctor of Philosophy*, July.

[13] Weiss, L., Sanderson, A. & Neuman, A. C. (1987). "Dynamic visual servo control of robots: An adaptive image-based approach", *IEEE Journal on Robotics and Automation*, *3(5)*, 404–417.

[14] Corke, P. (2002). "*Robotics Toolbox for MatLab*," April.

[15] Craig, J. (2004). "Introduction to Robotics: Mechanics and Control, " Textbook, International Edition," *Prentice Hall*.

[16] Pellionisz, A. (1989). "About the Geometry Intrinsic to Neural Nets," International *Joint Conference on Neural Nets,* Washington, D.C. *vol. 1*. 711-715.

[17] Shields, M. & Casey, M. C. (2008). " A Theoretical Framework for Multiple Neural Network Systems, " *Neurocomputing*, *vol. 71(7-9)*, 1462-1476.

[18] Aleksander, I. & Morton, H. (1995). *"An Introduction to Neural Computing*," 2nd edition.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 2

PREDICTION OF MECHANICAL PROPERTIES OF POLYPROPYLENE/WGRT COMPOSITES VIA UNIFORM DESIGN AND ARTIFICIAL NEURAL NETWORKS

Shu Ling Zhang[1], Kaushik Pal[2*] and Jin Kuk Kim[2]
[1]Alan G. MacDiarmid Lab, College of Chemistry, Jilin University, Changchun, 130012, People's Republic of China
[2]Polymer Science and Engineering, School of Nano and Advanced aterials Engineering, Gyeongsang National University, Jinju 660-701, Gyeongnam, Republic of Korea

ABSTRACT

Recycling represents a valid alternative to the disposal of post-consumer materials if it is possible to obtain new materials with good properties. In this chapter, Polypropylene (PP)/waste ground rubber tire powder (WGRT) or waste polypropylene (WPP)/waste ground rubber tire (WGRT) powder composites were studied with respect to the effect of bitumen and maleic anhydride-grafted styrene-ethylene-butylene-styrene (SEBS-g-MA) content by using the design of experiments (DOE) approach, whereby the effect of the four polymers content on the final mechanical properties were predicted. Uniform design method was especially adopted for its advantages. Optimization was done using hybrid artificial neural network-genetic algorithm (ANN-GA) technique. The results indicated that the composites showed fairly good ductibility provided that it had a relatively higher concentration of bitumen and SEBS-g-MA under the studied condition. A quantitative relationship was presented between the material concentration and the mechanical properties as a set of contour plots, which were confirmed experimentally by testing the optimum ratio.

Keywords: waste ground rubber tire powder; composites; mechanical properties; uniform design; artificial neural network.

[*] Tel: +82-55-751-5299, Fax: +82-55-753-6311
Email: pl_kshk@yahoo.co.in

1. INTRODUCTION

Our laboratory has been focusing considerable attention on development of technologies to effectively recycle waste ground rubber tire [1-3]. In the first stage of the 10 year project (2000~2002), we developed technologies to produce ultra fine powder from waste ground rubber tire. In the second stage (2003~2005), we developed technologies to manufacture thermoplastic elastomers from waste ground rubber tire powder. In the third stage (2006~2009), we are developing technologies for mass production of TPEs based on waste ground rubber tire powder. Recycling of waste ground rubber tire requires special techniques because waste ground rubber tire is a thermoset material, which cann't be reprocessed like thermoplastics. Powder utilization is one of the attractive techniques for effective utilization of waste ground rubber tire. A promising way of 'recycling' waste ground rubber tire powder (WGRT) is to incorporate it into thermoplastics to obtain thermoplastic elastomers (TPEs) and a perfect choice for thermoplastic is polypropylene (PP) or waste polypropylene (WPP) due to their low cost and the resulting protection of environment. One of major criteria for a thermoplastic elastomer is elongation at break is more than 100%. In order to achieve the target, bitumen and compatibilizer were added to the polypropylene/waste ground rubber tire powder (PP/WGRT) or waste polypropylene/waste ground rubber tire powder (WPP/WGRT) blend systems. According to the former papers, we know that bitumen acts as a plasticizer for polyolefin [4]. In addition, bitumen also acts as a devulcanized agent for WGRT [5,6]. The interface adhesion between PP or WPP and WGRT is usually very weak due to the crosslinked structure of WGRT. In order to solve the problem, some attempts were made to produce thermoplastic rubbers [7,8]. It was early recognized that WGRT should be devulcanized or at least partially devulcanized to facilitate the molecular entanglement between PP and WGRT. In order to further improve the interface adhesion between PP or WPP and WGRT, the compatibilizer and bitumen are added together.

The sharp market competition makes it important to shorten the development cycle of products and reduce the costs. The application of statistical experimental design and analysis technology in the rubber industry provides composites with a convenient, accurate and quantitative means. Statistical design of experiments (DOE) has been long used to provide efficient approaches to optimize process parameters and rubber formulary in rubber processing [9]. Most statistical experimental designs usually adopted in rubber composites are two level factorial design, screening design and response surface design, in which the response surface design especially has been used widely because of its use of less experimental trials and its capability of fitting quadratic regression equations [10]. However, with the increase of experimental factors, the number of coefficients of the quadratic equation increases exponentially and hence, does the number of experimental trials. More recently, a new statistical design of experiments, the uniform design method is found to overcome this problem. Keeping this in mind, a three factor with seven trials uniform experimental design U_7 (7^3) was chosen and experiments were carried out.

Table 1. Uniform design table ($U_7(7^3)$) and the corresponding results for PP/WGRT/Bitumen/SEBS-g-MA composites

Exp.no.	Factors				Results	
	PP (wt.%)	Bitumen (wt.%)	WGRT (wt.%)	SEBS-g-MA (phr)	Tensile strength (MPa)	Elongation at break (%)
1	65	13.5	21.5	6.7	12.3	380.0
2	35	18	47	10	7.8	339.0
3	60	0	40	13.3	11.5	139.3
4	40	4.5	55.5	3.3	10.4	165.9
5	50	9	41	20	10.3	342.0
6	55	22.5	22.5	0	10.7	300.0
7	45	27	28	16.7	8.9	505.7

Uniform design (UD) is a useful and simple method, which was first proposed by Fang and co-workers [11]. Generally speaking, UD is a form of "space filling" design for computer experiments [12]. In order to establish a UD, one needs to find suitable design points that are scattered uniformly on the experimental domain. However, if we restrict the domain to certain lattice points, then UD is also an efficient fractional factorial design. For a given measure of uniformity *M*, a uniform design has the smallest M-value over all fractional factorial designs with *n* runs and m q-level factors. Examples of successful applications of the UD method for improving technologies in various fields have been consistently reported [13-17].

Table 2. Uniform design table ($U_7(7^3)$) and the corresponding results for WPP/WGRT/Bitumen/SEBS-g-MA composites

Exp.no.	Factors				Results	
	WPP (wt.%)	Bitumen (wt.%)	WGRT (wt.%)	SEBS-g-MA (phr)	Tensile strength (MPa)	Elongation at break (%)
1	65	13.5	21.5	6.7	12.1	188.7
2	35	18	47	10	8.9	256.4
3	60	0	40	13.3	12.6	157.2
4	40	4.5	55.5	3.3	11.6	131.2
5	50	9	41	20	11.2	207.6
6	55	22.5	22.5	0	10.6	239.4
7	45	27	28	16.7	8.7	393.1

Table 3. Comparison of experimental properties and those predicted by RCAD for PP/WGRT/Bitumen/SEBS-g-MA composites

Properties	Predicted by RCAD	Experimental value
Tensile strength (MPa)	9.4	9.3
Elongation at break (%)	500.2	534.8

An artificial neural network (ANN) approach is a powerful mathematical tool in recognizing and modeling of material properties. They have been used in diverse applications in control, robotics, pattern recognition, forecasting, power systems, manufacturing, optimization, signal processing, etc. The idea to solve engineering problems using neural networks was developed in the 1940s in the United States. It has been introduced into the fields of materials recently [18,19]. They are actually a computing system containing simple processing elements known as units or nodes connected by links. Each neuron generates an output signal and this is in the case that the weighted sum of inputs is greater than an actuation value. Output of each neuron is related to the inputs through transfer or actuation functions. These subsystems are organized in a series of layers. In each neuron, the scalar input *p* is transmitted through a connection that multiplies its strength by the scalar weight *w*, to form the product *wp*, again a scalar. This product may be added to a scalar bias (much like a weight, except that it has a constant input of 1) to form the final argument of the transfer function. A neural network is not constructed for the solution of a specific mathematical problem and no mathematical or engineering principles are applied to determine the system output. They just apply the gained knowledge from previously solved examples to build a system of neurons that learn how to solve a new examined problem by adapting the intensity of the links between nodes [20].

On the other hand, genetic algorithm (GA) is a direct search algorithm that is based on the natural evolution concept coming from Darwin's theory of evolution. It is actually a probability-based optimization algorithm, started with an initial population of design variables. In GA, natural selection increases the surviving capability of the populations over the foregoing generations. The characteristics of each design are used to generate a fitness value indicating its level of performance with respect to the other designs in the population. Designs that perform the best (i.e., have the highest fitness value) are given the greatest probability of breeding with other good designs so that their characteristics can be passed to future generations. ANN and GA are the most promising natural computation techniques. In recent years, ANN has become a very powerful and practical method to model very complex nonlinear systems [21-25] and can be found in various research fields for parameter optimization [22-24,26].

In the present chapter, polypropylene (PP)/waste ground rubber tire powder (WGRT) or waste polypropylene/waste ground rubber tire powder (WPP/WGRT) composites were prepared. The effect of bitumen and maleic anhydride-grafted styrene-ethylene-butylene-styrene (SEBS-g-MA) of various concentrations on the mechanical properties of the above composites was studied. Instead of studying the effect of each factor one at a time and finally optimizing, design of experiments (DOE) methodology was chosen and uniform design method was especially adopted involving a minimum of only seven experiments. Optimization was done using a hybrid artificial neural network (ANN)-genetic algorithm (GA) approach. The primary focus of this article is to predict the mechanical properties of PP/WGRT or WPP/WGRT composites companied with bitumen and SEBS-g-MA using ANN-GA approach with minimum number of experiments. Results from the predicted properties were also experimentally confirmed.

Table 4. Comparison of experimental properties and those predicted by RCAD for WPP/WGRT/Bitumen/SEBS-g-MA composites

Properties	Predicted by RCAD	Experimental value
Tensile strength (MPa)	9.2	8.9
Elongation at break (%)	506.8	461.6

2. Experimental

2.1. Materials

Polypropylene (PP R520Y, MFI=1.8 g/10min) was manufactured by SK Corporation in Korea. Waste polypropylene (WPP) was supplied by Korean TR Co. Ltd. Maleic anhydride-grafted styrene-ethylene-butylene-styrene (SEBS-g-MA, Kraton FG-1901X) were obtained from Shell Chemical Co. Ltd. Bitumen (X-4) was obtained from Waterproofbank Company in Korea. Waste ground rubber tire powder (WGRT) of about 50 meshes was produced by Hongbok Industries in Korea. The composition of WGRT powder was polymer content of 48.5% with a natural rubber (NR) and a styrene-co-butadiene rubber (SBR). The other composition of WGRT powder was organic additive, carbon block and ash content of 13.4%, 27.7% and 10.4% respectively.

2.2. Sample Preparation

Waste ground rubber tire powder (WGRT) was devulcanized in a 30 mm single-screw extruder. First WGRT was passed through the extruder at the screw speed of 40 rpm and temperature profile of 110/120/140 °C. Afterwards, the extruded WGRT was extruded again together with the bitumen at the screw speed of 20 rpm and temperature profile of 90/100/110 °C. Finally, PP/WGRT/Bitumen/SEBS-g-MA or WPP/WGRT/Bitumen/SEBS-g-MA composites were produced in a L40/D19 twin-screw extruder at the screw speed of 180 rpm and temperature profile of 170/180/200/210/210/215/215/210 °C. The final samples were molded at temperature profile of 220/230/240/250 °C by injection for the measurements of tensile properties.

2.3. Characterization

The tensile properties of the dumbbell samples were measured according to ASTM D412, using a Lloyd LR10K tensile testing machine, with crosshead speed of 500 mm/min, and the average value of mechanical properties was calculated using at least 5 samples.

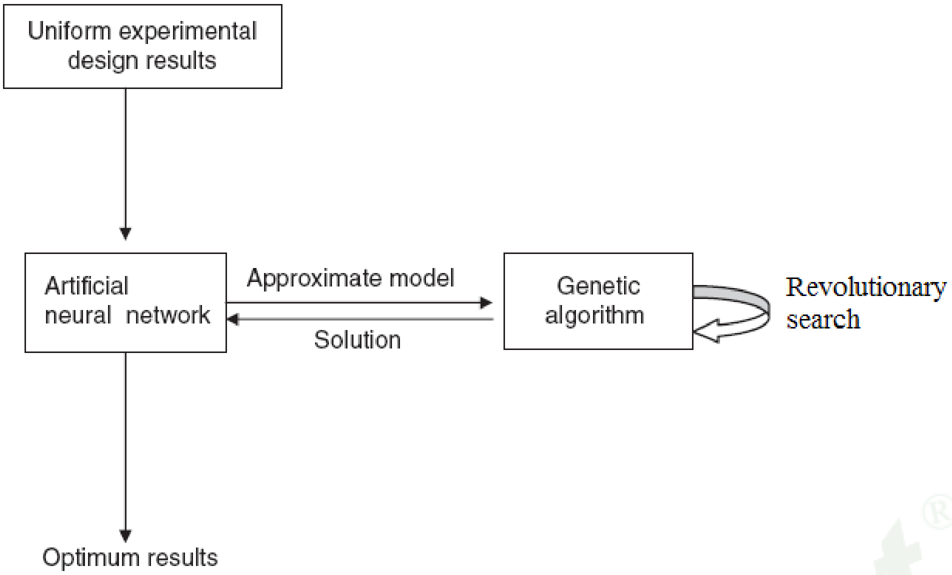


Figure 1. Optimization methodology in RCAD.

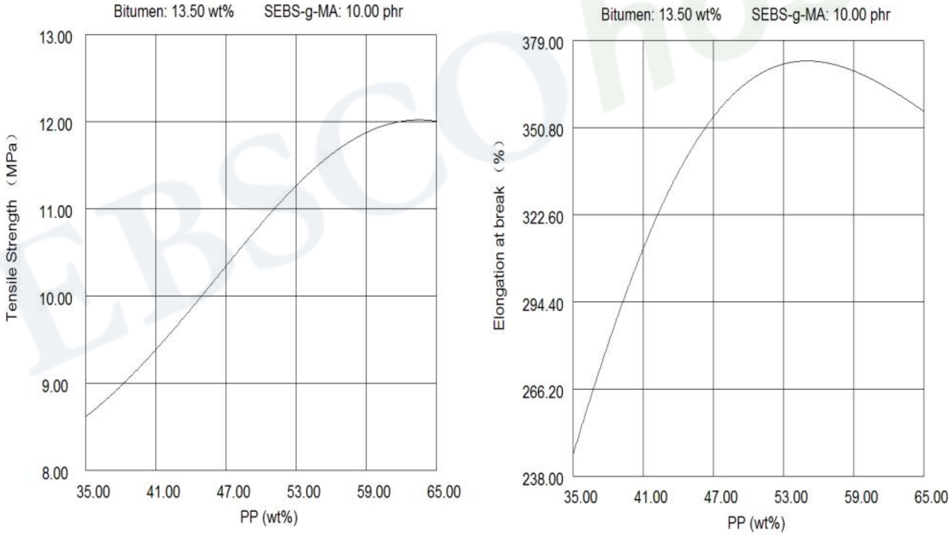


Figure 2. Predicted properties of RCAD with respect to PP concentration at a constant bitumen (13.50 wt%) and SEBS-g-MA (10.00 phr) concentration for PP/WGRT/Bitumen/SEBS-g-MA composites.

2.4. Stages in Experimentation

The various steps involved in execution of design of experiment [27], which was employed in this study: (1) recognition and statement of the problem (2) choice of factors and levels (3) selection of the response variables (4) design of experiments (i.e., layout for carrying out the trial) (4) conduct of the experiment (5) analysis of data (6) selection of optimum combination of parameters and (7) trial implementation to confirm the result. The first step is the recognition and statement of the problem, which is the objective of the study to determine the effect of bitumen and SEBS-g-MA and develop a thermoplastic elastomer based on PP and WGRT possessing good ductibility by involving a minimum number of experiments. Four important factors identified to have maximum impact on the properties of the blends are PP (x1), bitumen (x2), WGRT (x3) and SEBS-g-MA (x4) contents. The ranges of the factors to be evaluated are: 35 wt% ≤ x1 ≤ 65 wt%, 0 wt% ≤ x2 ≤ 27 wt%, 0 phr ≤ x4 ≤ 20 phr and x1 + x2 + x3 = 100%. Although there are four factors involved, the existence of x1 + x2 + x3 = 100% decides to take the U_7 (7^3) involving three factors (x1, x2, x4) and seven experimental runs (Table 1). The effect of each factor one at a time and finally optimizing, design of experiments (DOE) methodology was chosen and uniform design method was especially adopted involving a minimum of only seven experiments. The response variables selected for study are tensile strength and elongation at break, as these are the chief and critical factors to any industrial application.

2.5. Optimization by Hybrid Artificial Neural Network – Genetic Algorithm

The uniform design data was used as the input for optimization procedure. Due to the good approximation ability of neural network and the robust evolutionary searching performance of GA, this hybrid ANN-GA strategy was particularly selected for optimization in this study. It also has the primary advantage of being used for optimization of processes without explicitly knowing the forms of objective functions. The application of this strategy is recently finding increased applications in many different scientific and engineering disciplines owing to its accuracy in prediction/optimization and flexibility. The searching mechanism of the hybrid strategy can be briefly described as follows. Firstly, as approximate models of the practical problem, ANNs are constructed with certain training algorithms based on a collection of training samples. Then, the GA is employed to explore good solutions among solution space. Once the GA generates a new solution, the ANN will be used to determine its fitness value for the GA to continue its searching process. Until the stopping criterion of the GA is satisfied, the strategy will output the best solution resulted by the GA and its performance determine by detail evaluation based on actual problem. In other word, the objective values of solutions can be predicted efficiently by the ANN with certain extent of precision degree due to its good approximation performance, so as to improve the searching efficiency of GA by without paying much computational time for evaluation. On the other hand, the GA can guarantee promising solutions due to its effective and robust searching performance for this current optimization problem. A proprietary software package developed in Qingdao University of Science and Technology, China called Rubber Computer Aided Design (RCAD) [28] was used for this purpose. The schematic diagram for optimization of RCAD is shown in Figure 1.

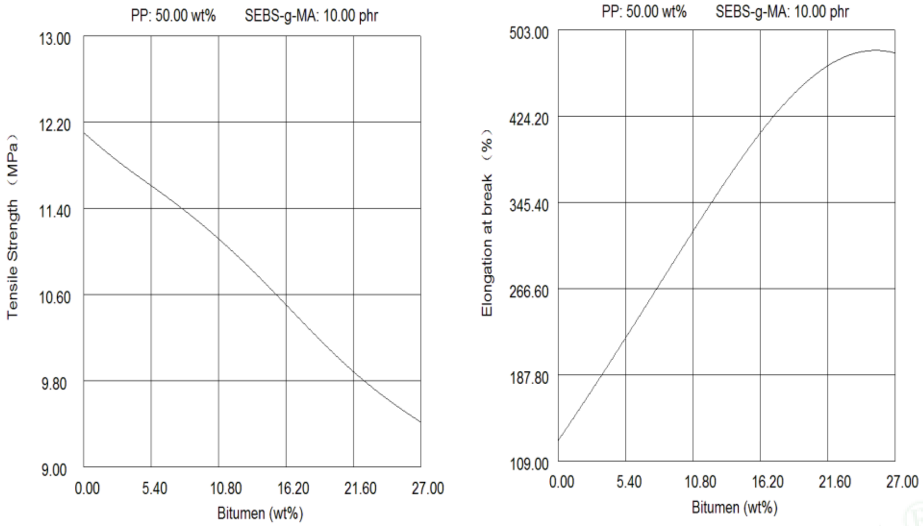


Figure 3. Predicted properties of RCAD with respect to bitumen concentration at a constant PP (50.00 wt%) and SEBS-g-MA (10.00 phr) concentration for PP/WGRT/Bitumen/SEBS-g-MA composites.

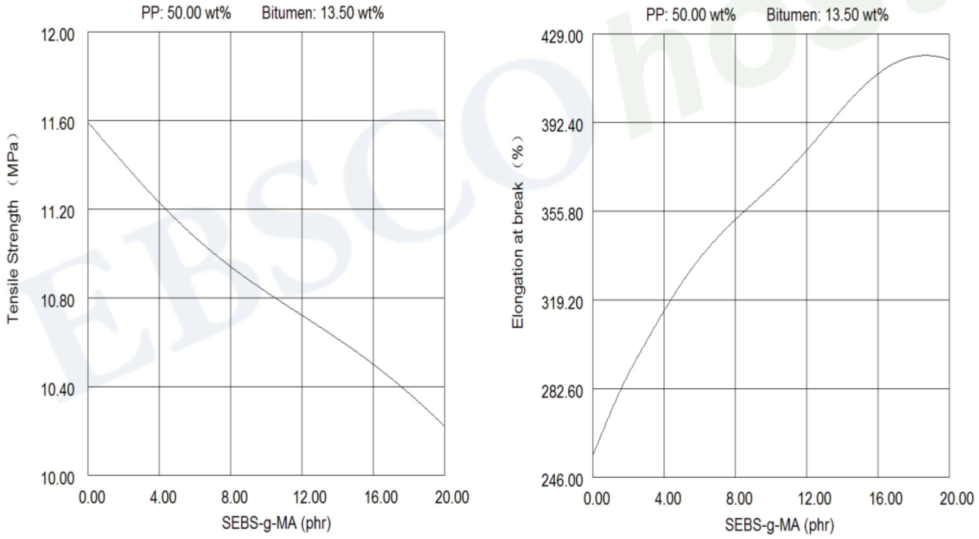


Figure 4. Predicted properties of RCAD with respect to SEBS-g-MA concentration at a constant PP (50.00 wt%) and bitumen (13.50 wt%) concentration for PP/WGRT/Bitumen/SEBS-g-MA composites.

3. Results and Discussion

As observed in our preliminary studies, mixtures of PP or WPP and WGRT do not lead to the formation of composites having appreciable mechanical properties. Accordingly, it was presumed that incorporation of bitumen and SEBS-g-MA in the PP/WGRT or WPP/WGRT composites could produce the desired properties. Bitumen was chosen because of its plasticized effect for PP or WPP [4] and its devulcanized effect for WGRT [5,6]. Namely the sulfur crosslink in WGRT are broken under mechanical stress (the first extrusion) [5] and the components of bitumen react with the sulfur in WGRT (the second extrusion) [6]. SEBS-g-MA was also selected for its use as a good compatibilizer. Namely SEBS-g-MA has a certain extent compatibility with PP or WPP but also MA group in SEBS-g-MA can react with phenolic OH group in WGRT. In our former studies [29], it has been proved that the presence of bitumen could improve the elongation at break of polypropylene (PP)/waste ground rubber tire powder (WGRT) composites due to the devulcanized and plasticized effects of bitumen. Moreover, the presence of compatibilizer (SEBS-g-MA) could also enhance the elongation at break of polypropylene (PP)/waste ground rubber tire powder (WGRT) composites. This should origin from the compatibilized effect of SEBS-g-MA. On the basis of the addition of compatibilizer (SEBS-g-MA), the improvement of the composites in the elongation at break increased with the content of bitumen. This should result from the increase of the devulcanized and plasticized effects of bitumen. On the basis of WGRT treated with bitumen, the properties of the composites showed different change with the addition of different compatibilizer. However, the composition of the plastic/elastomer ratio in every trial of the above experiments was changeless. Thus, it was difficult to character the mechanical properties in complete range of the composites. It is noteworthy that the uniform design can solve this problem.

Most importantly, the composition of the plastic/elastomer ratio in these composites is not kept constant in order to study the mechanical properties in complete range of the composites. The uniform design and the experimental results for PP/WGRT/Bitumen/SEBS-g-MA and WPP/WGRT/Bitumen/SEBS-g-MA composites are listed in Table 1 and 2, respectively. It is interesting to note from Table 1 and 2 that tensile strength and percentage elongation at break (EB) are almost inversely related i.e. with increase in the tensile strength a simultaneous decrease in the elongation at break. This is confirmed by the prediction of the RCAD for PP/WGRT/Bitumen/SEBS-g-MA composites, which is shown in Figure 2-4. At a constant bitumen (13.50 wt%) and SEBS-g-MA (10.00 phr) concentration, the variation of the different properties with respect to the PP content is presented in Figure 2. The tensile strength is found to increase with increase in the PP content, while the elongation at break is found to increase with increase in the PP content until 55.00 wt% and then starts decreasing fast. This is due to the presence of bitumen (13.50 wt%) and SEBS-g-MA (10.00 phr) for the available WGRT to get dispersed and thereby having a gradual increase to get a maximum elongation at break of 370 %. But, when PP further increases, namely WGRT further decreases, it is obvious that bitumen (13.50 wt%) and SEBS-g-MA (10.00 phr) could not disperse well in the available WGRT and thereby having a fast decrease from the maximum elongation at break of 370 %. At a constant PP (50.00 wt%) and SEBS-g-MA (10.00 phr) concentration, the variation of the different properties with respect to the bitumen content is presented in Figure 3. The tensile strength is seen to decrease with increase in the bitumen content, whereas the elongation at break is seen to increase due to the increase in the devulcanizated and plasticized effect of bitumen. At a constant PP (50.00 wt%) and bitumen (13.50 wt%) concentration, the variation of the different properties with respect to the SEBS-g-MA content is presented in Figure 4. The change of mechanical properties behaves similarly i.e. the tensile strength is seen to decrease with increase in the SEBS-g-MA content, whereas the elongation at break is seen to increase due to the increase in the compatibilized effect of SEBS-g-MA. The aforementioned phenomena are confirmed by the prediction of the

RCAD for WPP/WGRT/Bitumen/SEBS-g-MA composites, which is shown in Figure . 5-7. At a constant bitumen (13.50 wt.%) and SEBS-g-MA (10.00 phr) concentration, the variation of the different properties with respect to the WPP content is presented in Figure . 5. The tensile strength is found to almost increase with increase in the WPP content, while the elongation at break is found to increase with increase in the WPP content until 52.00 wt.% and then starts decreasing fast. This is due to the presence of bitumen (13.50 wt.%) and SEBS-g-MA (10.00 phr) for the available WGRT to get dispersed and thereby having a gradual increase to get a maximum elongation at break of 268 %. But, when WPP further increases, namely WGRT further decreases, it is obvious that bitumen (13.50 wt.%) and SEBS-g-MA (10.00 phr) could not disperse well in the available WGRT and thereby having a fast decrease from the maximum elongation at break of 268 %. At a constant WPP (50.00 wt.%) and SEBS-g-MA (10.00 phr) concentration, the variation of the different properties with respect to the bitumen content is presented in Figure . 6. The tensile strength is seen to decrease with increase in the bitumen content, whereas the elongation at break is found to increase with increase in the bitumen content until 24.00 wt.% and then starts decreasing fast. This may origin from the increase in the devulcanizated and plasticized effect of bitumen and thereby having a gradual increase to get a maximum elongation at break of 400 %. But, when bitumen further increases, it is possible that bitumen as plasticizer changes into bitumen as diluent agent and thereby having a fast decrease from the maximum elongation at break of 400 %. At a constant WPP (50.00 wt.%) and bitumen (13.50 wt.%) concentration, the variation of the different properties with respect to the SEBS-g-MA content is presented in Figure . 7. The tensile strength is seen to decrease with increase in the SEBS-g-MA content, whereas the elongation at break is seen to increase due to the increase in the compatibilized effect of SEBS-g-MA.

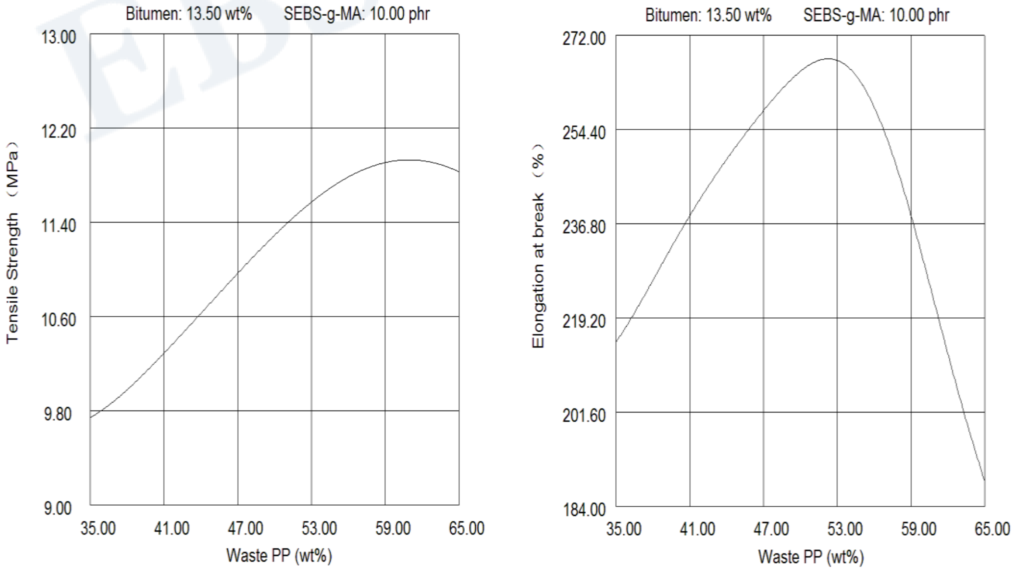


Figure 5. Predicted properties of RCAD with respect to PP concentration at a constant bitumen (13.50 wt%) and SEBS-g-MA (10.00 phr) concentration for WPP/WGRT/Bitumen/SEBS-g-MA composites.

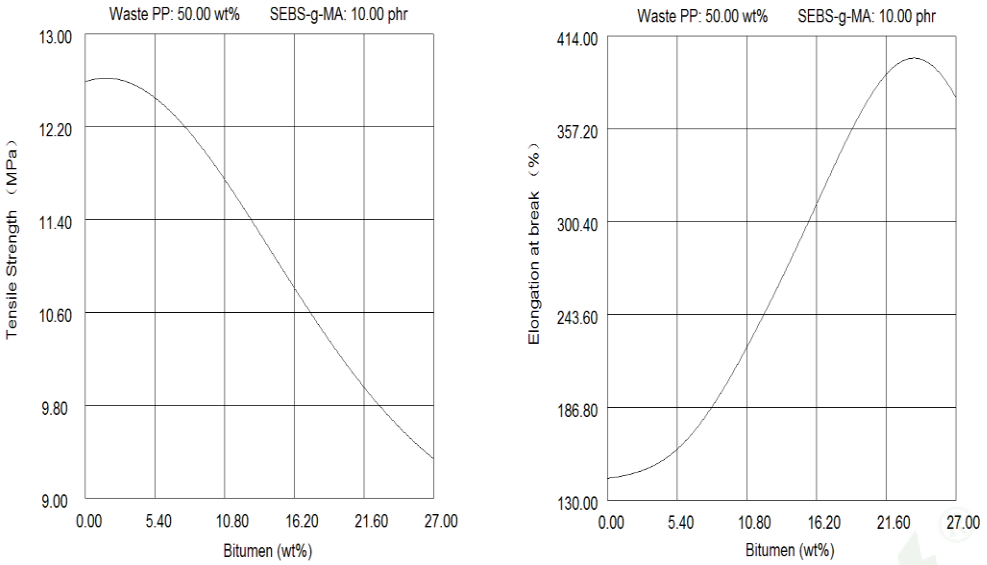


Figure 6. Predicted properties of RCAD with respect to bitumen concentration at a constant PP (50.00 wt%) and SEBS-g-MA (10.00 phr) concentration for WPP/WGRT/Bitumen/SEBS-g-MA composites.

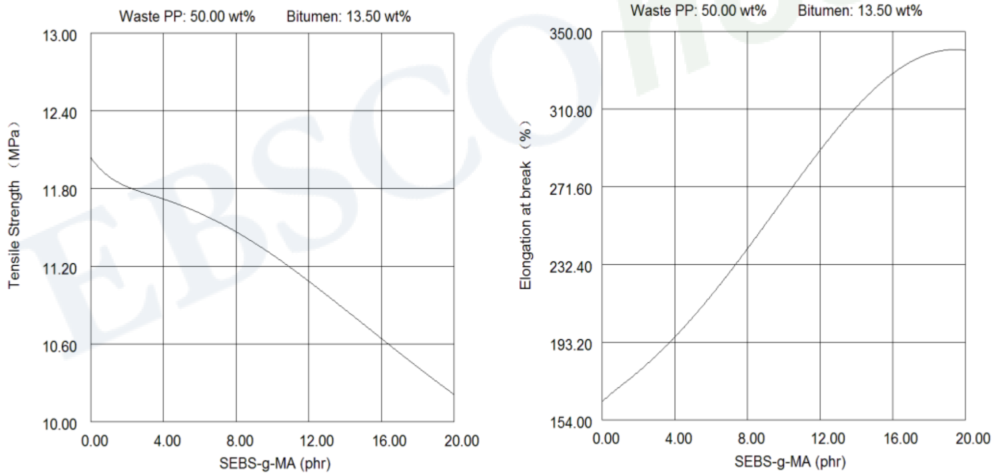


Figure 7. Predicted properties of RCAD with respect to SEBS-g-MA concentration at a constant PP (50.00 wt%) and bitumen (13.50 wt%) concentration for WPP/WGRT/Bitumen/SEBS-g-MA composites.

The RCAD program also gave two-dimensional contour plots after the simulation. Two special cases are considered. Figure 8 shows the contour plots of the effect of PP and bitumen at constant SEBS-g-MA loadings (10 phr) on the tensile strength and elongation at break respectively. We can find that tensile strength is maximum in the region 49 wt% ≤ PP ≤ 65 wt% & 0 wt% ≤ Bitumen ≤ 10.8 wt%, while elongation at break is maximun in the region 42 wt% ≤ PP ≤ 65 wt% & 20.5 wt% ≤ Bitumen ≤ 27 wt%. Figure 9 shows the contour plots of the effect of PP and SEBS-g-MA at constant bitumen loading (13.50 wt%) on the tensile strength and elongation at break respectively. We can find that tensile strength is maximum in the region 52 wt% ≤ PP ≤ 65 wt% & 0 phr ≤ SEBS-g-MA ≤ 8 phr, while elongation at break is maximun in the region 38 wt% ≤ PP ≤ 61 wt% & 13 phr ≤ SEBS-g-MA ≤ 20 phr. Figure 10 shows the contour plots of the effect of bitumen and SEBS-g-MA at constant PP loading (50.00 wt%) on the tensile strength and elongation at break respectively. We can find that tensile strength is maximum in the region 0 phr ≤ SEBS-g-MA ≤ 4 phr & 0 wt% ≤ Bitumen ≤ 5.4 wt% while elongation at break is maximun in the region 12 phr ≤ SEBS-g-MA ≤ 20 phr & 20 wt% ≤ Bitumen ≤ 27 wt%. Figure . 11 shows the contour plots of the effect of WPP and bitumen at constant SEBS-g-MA loadings (10 phr) on the tensile strength and elongation at break respectively. We can find that tensile strength is maximum in the region 53 wt.% ≤ WPP ≤ 65 wt.% & 0 wt.% ≤ Bitumen ≤ 8 wt.%, while elongation at break is maximun in the region 51 wt.% ≤ WPP ≤ 58 wt.% & 20 wt.% ≤ Bitumen ≤ 27 wt.%. Figure . 12 shows the contour plots of the effect of WPP and SEBS-g-MA at constant bitumen loading (13.50 wt.%) on the tensile strength and elongation at break respectively. We can find that tensile strength is maximum in the region 52 wt.% ≤ WPP ≤ 65 wt.% & 0 phr ≤ SEBS-g-MA ≤ 7 phr, while elongation at break is maximun in the region 44 wt.% ≤ WPP ≤ 56 wt.% & 15 phr ≤ SEBS-g-MA ≤ 20 phr. Figure . 13 shows the contour plots of the effect of bitumen and SEBS-g-MA at constant WPP loading (50.00 wt.%) on the tensile strength and elongation at break respectively. We can find that tensile strength is maximum in the region 0 phr ≤ SEBS-g-MA ≤ 7 phr & 0 wt.% ≤ Bitumen ≤ 8.1 wt.% while elongation at break is maximun in the region 13 phr ≤ SEBS-g-MA ≤ 20 phr & 18 wt.% ≤ Bitumen ≤ 27 wt.%.

In order to check the validity of the prediction, an experiment is needed to be done. According to Figure 7 and 13, two trials with PP=50 wt%, Bitumen=25 wt%, WGRT=25 wt%, SEBS-g-MA=20.00 phr and WPP=50 wt%, Bitumen=25 wt%, WGRT=25 wt%, SEBS-g-MA=20.00 phr were selected for the vertification.

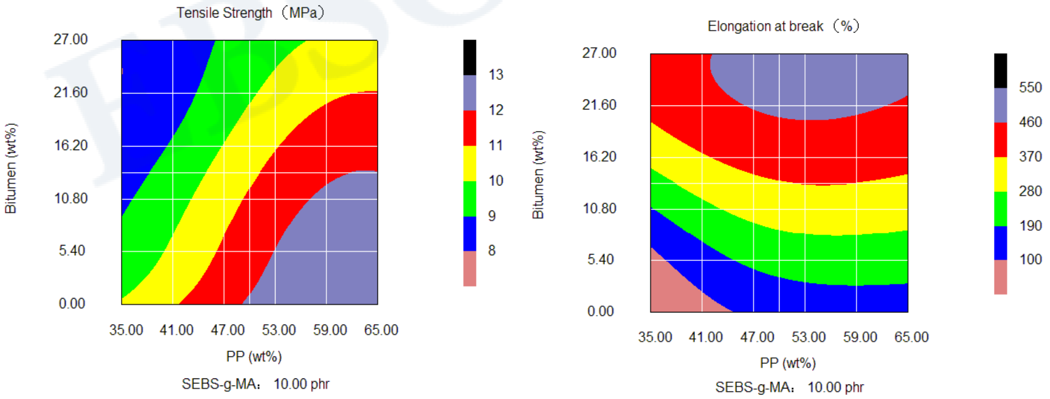


Figure 8. Contour plots of the effect of PP and bitumen on tensile strength and elongation at break for PP/WGRT/Bitumen/SEBS-g-MA composites.

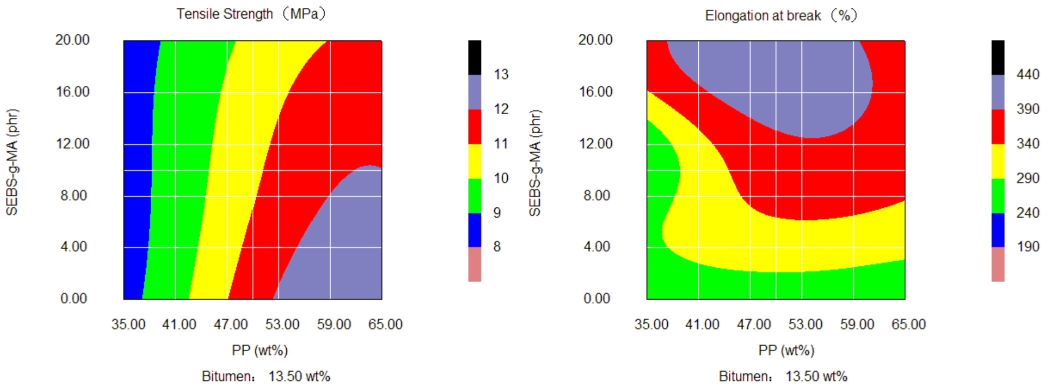


Figure 9. Contour plots of the effect of PP and SEBS-g-MA on tensile strength and elongation at break for PP/WGRT/Bitumen/SEBS-g-MA composites.

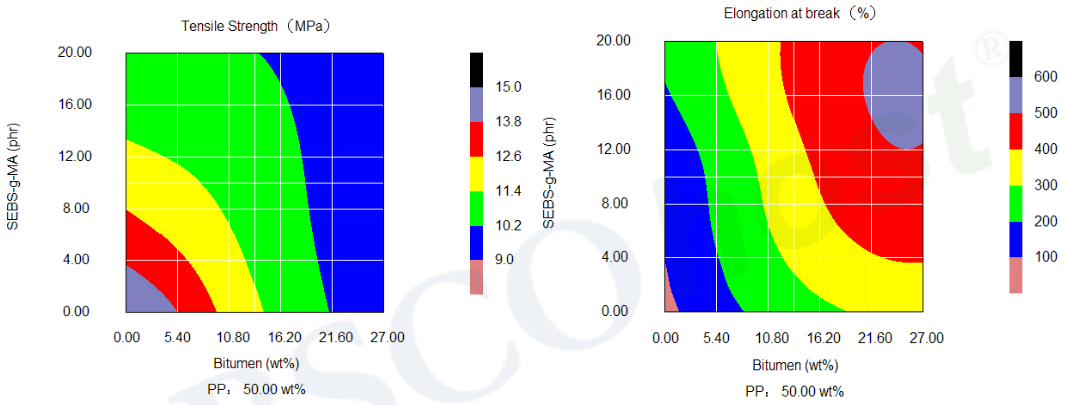


Figure 10. Contour plots of the effect of bitumen and SEBS-g-MA on tensile strength and elongation at break for PP/WGRT/Bitumen/SEBS-g-MA composites.

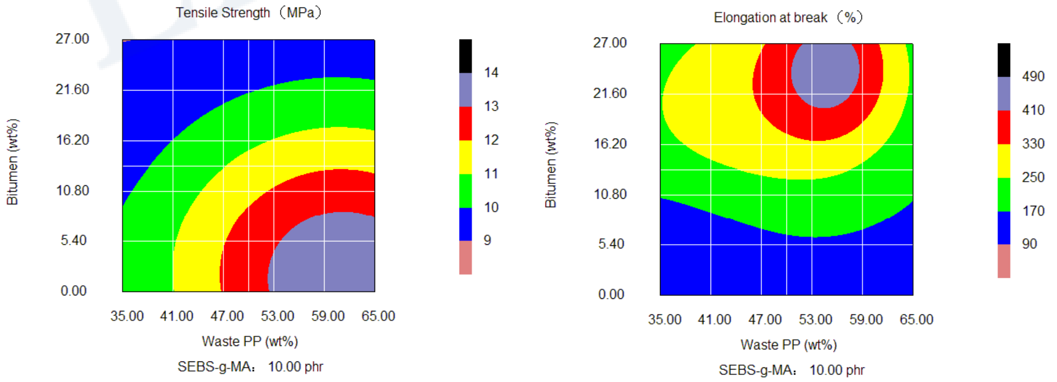


Figure 11. Contour plots of the effect of PP and bitumen on tensile strength and elongation at break for WPP/WGRT/Bitumen/SEBS-g-MA composites.

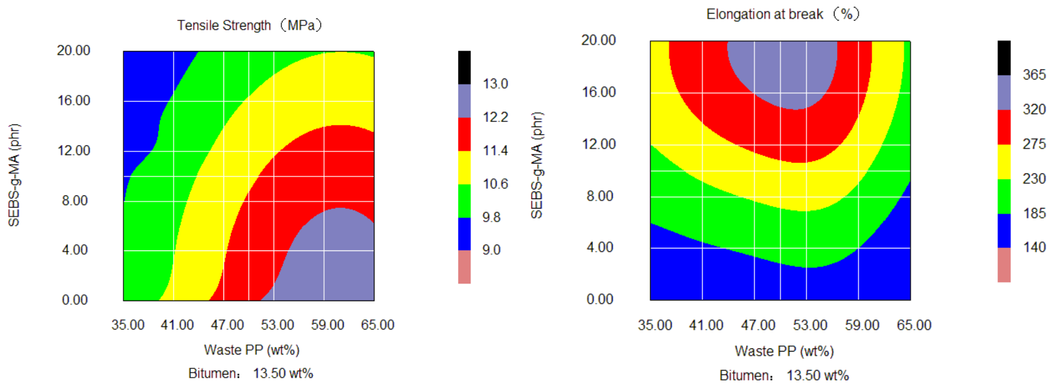


Figure 12. Contour plots of the effect of PP and SEBS-g-MA on tensile strength and elongation at break for WPP/WGRT/Bitumen/SEBS-g-MA composites.

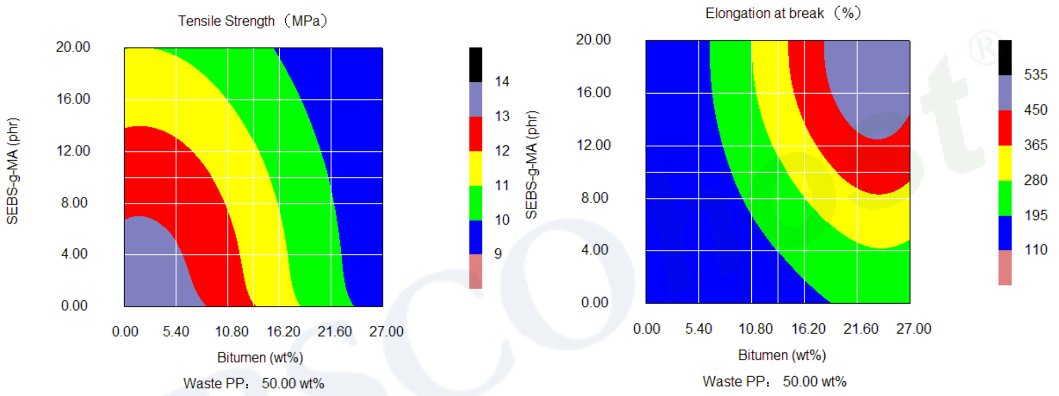


Figure 13. Contour plots of the effect of bitumen and SEBS-g-MA on tensile strength and elongation at break for WPP/WGRT/Bitumen/SEBS-g-MA composites.

The materials according to this recipe were extruded at the same processing conditions as done for the seven uniform design-based experiments. Table 3 and 4 show the comparison of properties between the predicted and the experimental values for the above trials, respectively. It was observed that the predicted and the experimentally determined properties are fairly close to each other and thereby confirming that uniform experimental design-based ANN-GA optimization procedure can be used for the prediction of properties with a fair degree of accuracy.

CONCLUSION

PP/WGRT or WPP/WGRT composites with bitumen and SEBS-g-MA of various concentrations were investigated using DOE methodology to optimize a recipe for commercial applications having high mechanical properties. A proprietary software package called Rubber Computer Aided Design (RCAD) was used for this purpose. RCAD utilizes the uniform design technique for the design of starting experiments which was selected for reducing the number of preliminary experiments when compared to traditional simultaneous methods, followed by a hybrid ANN-GA technique for optimization and prediction of the composite properties. Three material factors namely PP or WPP, bitumen and SEBS-g-MA concentration leading to seven trials gave an optimized recipe with respect to tensile strength and elongation at break for property prediction. The predicted optimum formulation was found to agree with the experimental recipe thereby validating the accuracy of the uniform design method-based ANN-GA optimization procedure. No doubt that the use of RCAD is convenient for the commercial application of PP/WGRT or WPP/WGRT composites with an appropriate property.

REFERENCES

[1] Kim, JK; Burford, RP. Study on powder utilization of waste tires as a filler in rubber compounding. *Rubber Chemistry and Technology*, 1999, 71, 1028-1041.

[2] Kim, JK; Park, JW. The biological and chemical desulfurization of crumb rubber for the rubber compounding. *Journal of Applied Polymer Science*, 1999, 72, 1543-1549.

[3] Shanmugharaj, AM; Kim, JK; Ryu, SH. Modification of rubber powder with peroxide and properties of polypropylene/rubber composites. *Journal of Applied Polymer Science*, 2007, 104, 2237-2243.

[4] Fawcett, AH; McNally, T. Blends of bitumen with various polyolefins., *Polymer*, 2000, 41, 5315-5326.

[5] Lievana, E; Karger-Kocsis, J. Use of ground tyre rubber (GTR) in thermoplastic polyolefin elastomer compositions. Progress in Rubber, *Plastics and Recycling Technology*, 2004, 20(1), 1-10.

[6] Wen, GA; Zhang, Y; Zhang, YX; Sun, K; Chen, ZY. Vulcanization characteristics of asphalt/SBS blends in the presence of sulfur. *Journal of Applied Polymer Science*, 2001, 82, 989-996.

[7] Lee, SH; Maridass, B; Kim, JK. Dynamic reaction inside co-rotating twin screw extruder. II. waste ground rubber tire powder/ polypropylene blends. *Journal of Applied Polymer Science*, 2007, 106, 3209-3219.

[8] Shanmugharaj, AM; Kim, JK; Ryu, SH. UV surface modification of waste tire powder: Characterization and its influence on the properties of polypropylene/waste powder composites. *Polymer Testing*, 2005, 24, 739-745.

[9] Del, Vecchio, Basics of Response Surface Methods, In: Understanding Design of Experiments: A Primer for Technologists, 83, Chapter 15, Hanser/Gardner Publications. Inc., Cincinati, USA, 1997.

[10] Maridass, B; Gupta, BR. Performance Optimization of a Counter Rotating Twin Screw Extruder for Recycling Natural Rubber Vulcanizates using Response Surface Methodology, *Polymer Testing*, 2004, 23(4), 377-385.

[11] Fang, KT; Lin, DKJ; Winker, P; Zhang, Y. Uniform Design: *Theory and Applications,* Technometrics, 2000, 42(3), 237-248.

[12] Cao, Y; Zheng, YY; Fang, B. Optimization of Polymerase Chain Reaction-amplified Conditions using the Uniform Design Method, *J. Chem. Technol. Biotechnol.*, 2004, 79, 910-913.

[13] Lee, AWM; Chan, WF; Yuen, FSY; Tse, PK; Liang, YZ; Fang, KT. An Example of a Sequential Uniform Design: Application in *Capillary Electrophoresis. Chemometr*. Intell. Lab. Syst. 1997, 39, 11-18.

[14] Guan, FY; Wu, HF; Luo, Y. A Novel Strategy for Systematic Optimization of Micellar Electrokinetic Chromatography Separations. *Anal Chim Acta*., 1997, 342, 133-144.

[15] Leung, SYL; Chan, WH; Leung, CH; Luk, CH. Screening the Fabrication Conditions of Ultrafilration Membranes by using Uniform Design and Regression Analysis Methods. *Chemometr. Intell. Lab. Sys*., 1998, 40, 203-213.

[16] Fang, Q; Yeung, HW; Leung, HW; Huie, CW. Micelle Mediated Extraction and Preconcentration of Ginsenosides from Chinese Herbal Medicine. *J. Chromatogr*., 2000, 904, 47-55.

[17] Xu, CP; Yung, JW. Optimization of Submerged-Culture Condittions for Mycelial Growth and Exo-biopolymer Production by Auricularia Polytricha (wood ears fungus) using the Methods of Uniform Design and Regression *Analysis. Biotechnol. Appl. Biochem*, 2003, 38, 193-199.

[18] Ebube, NK; Ababio, GO; Adeyey, CM. Preformulation Studies and Characterization of the Physicochemical Properties of Amorphous Polymers using Artificial Neural Networks. *Int. J. Pharm*., 2000, 196, 27-31.

[19] White, H. Artificial Neural Networks: *Approximation and Learning Theory, Blackwell*, New York, 1992.

[20] Goldberg, DE. Genetic Algorithms in Search, Optimization, and Machine Llearning, Addison-Wesley, London, 1989.

[21] Sadeghi, BHM. ABP-Neural Network Predictor Model for Plastic Injection Molding Process. *J. Mater. Process Technol*., 2000, 103(3), 411-416.

[22] Chow, TT; Zhang, GO; Lin, Z; Song, CL. Global Optimization of Absorption Chiller System by Genetic Algorithm and Neural Network. *Energy Build*, 2002, 34(1), 103-109.

[23] Cook, DF; Ragsdale, CT; Major, RL. Combining a Neural Network with a Genetic Algorithm for Process Parameter Optimization. *Eng. Appl. Artif. Intell*., 2000, 13(4), 391-396.

[24] Woll, SLB; Cooper, DJ. Pattern-based Closed-loop Quality Control for the Injection Molding Process. *Polym. Eng. Sci*., 1997, 37(5), 801-812.

[25] Chen, CR; Ramaswany, HS. Modeling and Optimization of Variable Retort Temperature (VRT) Thermal Processing using Coupled Neural Networks and Genetic Algorithms. *J. Food Eng*., 2002, 53(3), 209-220.

[26] Pzcelik, B; Erzurumlu, T. Determination of Affecting Dimensional Parameters on Warpage of thin Shell Plastic Parts using Integrated Response Surface Method and Genetic Algorithm. Int. *Commun. Heat Mass Transfer*., 2005, 32, 1085-1094.

[27] Montgomery, DC. Design and analysis of experiments. John Wiley and Sons Inc., New York, USA, 1991.

[28] Xin, ZX; Kim, JK; Liu, L; Liu, M; Gao, L; Shi, X; Chen, X. Application of the Uniform Experimental Design Method for Rubber Formulation, In: Proceedings of the International Rubber Conference IRC-2004, *Beijing, People's Republic of China*, Vol 3, 478-482.

[29] Zhang, SL; Xin, ZX; Zhang, ZX; Kim, JK. Characterization of the properties of thermoplastic elastomers containing waste rubber tire powder. *Waste management*, 2009, 29, 1480-1485.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 3

MODELING OF COMPUTER-ASSISTED LEARNING USING ARTIFICIAL NEURAL NETWORKS

Fahad A. Al-Zahrani[1*]***, Hassan. M. Mustafa***[2**]***,***
Ayoub Al-Hamadi[3***] ***and U. A. Khashaba***[4****]

[1]Computer Engineering Department, Faculty of Computer and Information System, Umm Al-Qura University, Makkah Saudi Arabia
[2]Computer Engineering Department, Faculty of Engineering, Albaha University
[3]Institute for Electronics, Signal Processing and Communications (IESK), Otto-von-Guericke-University Magdeburg
[4] Department of Production Engineering & Mechanical Systems Design, Faculty of Engineering King Abdul Aziz University, KSA

ABSTRACT

The present work belongs to a rather challenging interdisciplinary issue associated with neuroscience modeling, educational psychology, and cognitive sciences for searching two optimum teaching methodologies at children's classrooms. Firstly, that one associated with trying effectively to find an optimal methodology for "how reading should be taught?" at children's classrooms. However, the second topic is closely related to optimum teaching method for solving long division problems following subsequent mathematical steps such as: divide, multiply, subtract, bring down, and repeat (if necessary). Searching for optimality leaning/teaching educational topics will depend upon comparative assessments and analysis for different experimental educational methodologies that applied at children's classrooms. Herein, presented assessment

[*] Corresponding author: E-mail: fahad.alzahrani@gmail.com
[**] E-mail: mustafa_hasan47@yahoo.com; On leave from the Educational Technology Department, Faculty of Specified Education-Banha University Egypt
[***] E-mail: Ayoub.Al-Hamadi@ovgu.de
[****] E-mail: khashabu@zu.edu.eg & ukhashaba@kau.edu.sa; On leave from the Department of Mechanical Design and Production Engineering,Zagazig University, Zagazig, Egypt.

processes comprise application of visual and/or auditory tutorial materials in addition to the classical classrooms teaching methodologies. The comparative assessment processes are performed by using realistic Artificial Neural Network (ANN) simulation programs, and/or mathematically formulated modeling. In addition, a computer-assisted learning (CAL) module is designed carefully aiming to develop a specified multimedia tutorial material.

Conclusively, it has been shown that optimal teaching methodology performance attained if and only if visual and auditory tutorial materials have been both presented simultaneously to reinforce the retention of learned material topics. Multi-sensory associative memories in addition to Pavlovian classical conditioning theories are applicable, via designed CAL package, at children's classrooms. It is worthy to note that comparative results obtained are interesting, after field application of suggested CAL package, for association tutorials with teacher's voice. Finally, presented study results in high recommendation for application of novel teaching trends aiming to improve learning quality in children's two reading and mathematical topics.

Keywords: Artificial Neural Network, Hebbian learning, Pavlov's classical conditioning, Associative memories, Computer assisted learning, Multimedia applications

1. INTRODUCTION

For a long time, psycho-linguistics researchers and educationalists were searching effectively for an optimal tutoring method for "how reading should be taught to children?" [1]. Optimality of teaching elementary of English language reading (letters & words) has been addressed by Rayner et al. [2]. This searching direction motivated by a great debate given by researchers at fields of psychology, linguistic and, cognitive sciences were continuously in cooperation [2,3]. Steps of mathematical processes such as: divide, multiply, subtract, bring down, and repeat are necessary to optimize learning/teaching methodology for solving long division problems [4].

Herein, the adopted optimality searching approach is inspired by realistic modeling and simulation of computer assisted learning (CAL) as well as classical teaching performance by using relevant Artificial Neural Network (ANN) learning paradigms. Commonly, presented analysis for leaning/teaching topics has been fulfilled by realistic modeling originated from simulation unsupervised learning paradigm on the basis of Hebb's self-organized learning rule [5]. The model was constructed based on Pavlovian classical conditioning for associative memory phenomenon (between visual and auditory stimuli signal) [6].

CAL packages for leaning/teaching topics, which simulating visual and/or auditory tutorial materials, are experimentally tested in educational field on children of about 11 years of age. Dominant optimality of teaching reading phonically over other methodologies has been proven by realistic ANN modeling along with educational field testing results, Rayner et al. [2]. Additionally, obtained field results were supported well by cognitive multimedia theory that suggests simultaneously presenting visual and auditory material, on the basis of memory association, to reinforce the retention of learned materials [7].

Interestingly, referring to more recently announcement by National Institutes of Health (NIH) in UAS children in elementary school were qualified to learn "basic building blocks" of cognition. Children after about 11 years of age take these building blocks and use them [8].

Consequently, it is convenient to suggest pupils' samples as children of about11 years of age for searching optimality of methodologies considering both leaning/teaching topics [2,9,10].

In the present work, a generalized learning model is presented along with a descriptive conceptual view (block diagram) on adopted ANN learning model. Detailed description of the learning/teaching model using visual and auditory tutorial materials that associated with reading topic was investigated. More detailed analysis of that model with mathematical formulation for associative memory function is introduced. Simulation results after running of suggested ANN learning model in addition to an evaluation of environmental noise effect on learning performance are presented.

2. GENERALIZED ANN LEARNING MODEL

2.1. Interactive Educational Model

In educational practice, learning/teaching modelling motivated mainly by two essential cognitive leaner's brain functions [11-13]. Firstly, pattern classification/recognition function based on visual/audible interactive signals stimulated by CAL packages. Secondly, associative memory function is used, which is originally based on classical conditioning motivated by Hebbian learning rule [5]. Both functions are commonly required to perform efficiently interactive educational processes for suggested learning/teaching topics in accordance with environmental learning and behaviourism [11-16].

Inputs to the neural network learning model, Figure1, are provided by environmental stimuli (unsupervised learning). The correction signal for the case of learning with a teacher is given by responses outputs of the model that will be evaluated by either the environmental conditions or by the teacher. Finally, the tutor plays a role in improving the input data (stimulating learning pattern), by reducing noise and redundancy of model pattern input. According to tutor's experience, the model can be provided with clear data by maximizing its signal to noise ratio. Details of mathematical formulation describing memory association between auditory and visual signals are shown in fourth section.

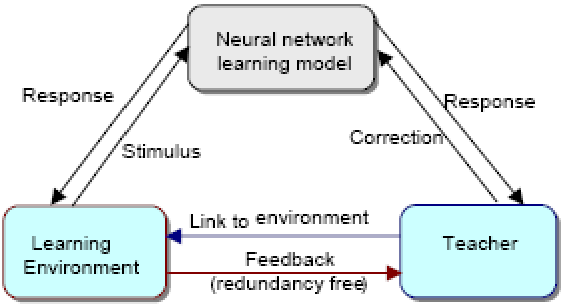


Figure 1. Illustrates a general view for interactive educational process, adapted from [10].

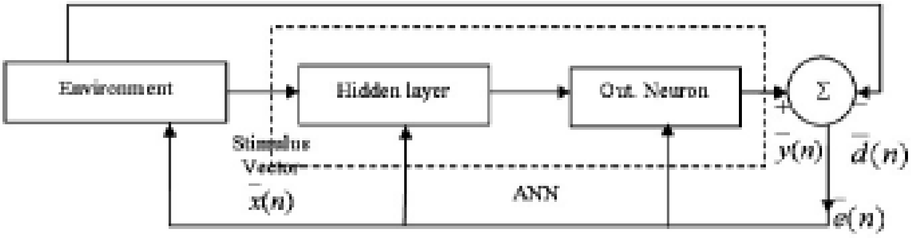


Figure 2. Generalized ANN block diagram, adapted from [7].

2.2. Basic ANN Model

Figure 2 shows the basic block diagram for suggested ANN learning/teaching model that simulates two diverse learning paradigms.

The paradigms are presented by interactive learning/teaching process, as well as other self-organized learning. The first paradigm is concerned with classical (supervised by tutor) learning observed at our classrooms (face to face tutoring). Accordingly, this paradigm proceeds interactively via bidirectional communication process between teacher and his learner(s) [5,11]. The second paradigm performs self-organized (unsupervised) tutoring process [14].

Referring to Figure 2, the error vector at any time instant (n) observed during learning processes is given by:

$$\bar{e}(n) = \bar{y}(n) - \bar{d}(n) \tag{1}$$

Where $\bar{e}(n)$ is the error correcting signal controlling adaptively the learning process, $\bar{x}(n)$ is the input stimulus, $\bar{y}(n)$ is the output response vectors, and $\bar{d}(n)$ is the desired numeric value(s).

The following equations are easily deduced:

$$V_{\mathrm{k}}(n) = X_j(n) W_{kj}^T(n) \tag{2}$$

$$Y_{\mathrm{k}}(n) = \varphi(V_{\mathrm{k}}(n)) = (1 - \mathrm{e}^{-\lambda V_{\mathrm{k}}(n)}) / (1 + \mathrm{e}^{-\lambda V_{\mathrm{k}}(n)}) \tag{3}$$

$$e_{\mathrm{k}}(n) = \left| d_{\mathrm{k}}(n) - y_{\mathrm{k}}(n) \right| \tag{4}$$

$$W_{\mathrm{kj}}(n+1) = W_{\mathrm{kj}}(n) + \Delta W_{\mathrm{kj}}(n) \tag{5}$$

Where *X* is input vector, *W* is the weight vector, φ is an activation (odd sigmoid) function characterized by λ as gain factor and *Y* as its output. e_k is the error value, and d_k is the desired output. Noting that $\Delta W_{kj}(n)$ is the dynamical change of weight vector value connecting the $k^{\underline{th}}$ and $i^{\underline{th}}$ neurons . Eqs. (2-5) are commonly applied for both the supervised (interactive learning with a tutor), and the unsupervised (learning though students' self-study) paradigms. The dynamical changes of weight vector value for supervised phase are given as following:

$$\Delta W_{kj}(n) = \eta e_k(n) X_j(n) \tag{6}$$

where, η is the learning rate value during learning process. However, for unsupervised paradigm, the dynamical change of weight vector value is given by:

$$\Delta W_{kj}(n) = \eta Y_k(n) X_j(n) \tag{7}$$

Noting that $e_k(n)$ in (6) is substituted by $y_k(n)$ at any arbitrary time instant (*n*) during learning process.

3. Modeling of Teaching Reading

3.1. Neurobiological Model Concepts

The concept of reading model, from neurobiological point of view, is presented with some biological hypotheses. These hypotheses are derived according to observed cognitive/behavioral tasks during the experimental learning process. Generally, the output response signal varies in its strength as inspired by original Pavlov's psycho-experimental work (classical conditioning) [6]. Therein, the learning response strength signal is measured quantitatively as the number of salivation drops.

In accordance with neurobiology, the adopted model is designed basically to fulfill better systematic investigations of the previously measured performance of classical conditioning experiments. So, the strength of response signal is dependent upon the transfer properties of the output motor neuron stimulating salivation gland. The structure of the model following the original Hebbian learning rule in its simplified form was illustrated in Figure 3.

Each of lettered circles A, B, and C presents a neuron cell body, however the line connecting cell bodies represents an axon that terminates synaptic junctions. Both of two signals released out from sound and sight sensory neurons A and C are represented respectively, by y_1 and y_2. The activation functions of neurons A and C are respectively $\phi_A(\lambda)$ and $\phi_C(\lambda)$. Both are suggested to be fractional of signum function as follows:

$$\phi_A(\lambda) = \begin{cases} \delta & \lambda \geq 0 \\ -\delta & \lambda < 0 \end{cases} \tag{8}$$

$$\phi_C(\lambda) = \begin{cases} (1-\delta) & \lambda \geq 0 \\ -(1-\delta) & \lambda < 0 \end{cases} \tag{9}$$

$$\text{Such that} \quad 0 \leq \delta < 1$$

Where δ is an implicit factor that representing the interrelation between motivational and reinforcement learning. However, in practice δ factor may be time dependent during progress of training phases. For simplest, it is considered as constant for average signal decay through auditory and visual nervous pathways (δ and 1-δ) respectively. More details about learning convergence (latency) time due to effect of dynamically changing threshold value *θ(t)*, and choosing of δ factor are presented elsewhere [12].

3.2. Reading Model

Learning how to read is an essential step in the educational ladder, especially for children during their primary school years. Thus, failure to achieve reading ability leads to nearly permanent learning disability during the following more advanced educational stages. For a long time, psycho-linguistics researchers as well as educationalists were trying effectively to find an optimal method for "how reading should be taught?" [1,2]. During the last decade phonics method is replaced, at several USA schools, by other guided reading methods that are performed using literature-based activities [3]. Nevertheless, comparative evolutional analysis for both methodological approaches proved the superiority of phonics method [10,17]. It is worth to mention that learning by phonics is performed directly by concurrent association between the pronounced sound (phoneme) and its corresponding letter/word.

In nature reading process two basic brain functions are considered to perform that process efficiently. The first function is to classify seen/heard patterns i.e. pattern classification. From neuronal network point of view this function is originated in the perception and essentially requires supervisor's (teacher's) instructions for learning completion. Secondly, the associative memory function is originally based on classical conditioning motivated by Hebbian learning rule [5]. It is belongs to the principle of learning without a teacher (unsupervised). The response outputs of the model will be evaluated either by the environmental conditions (unsupervised learning) or by the teacher. Finally, teacher experience plays an important role in improving the input data (stimulating the learning) by maximizing signal to noise ratio and redundancy of the model input.

The generalized simplified form of the model structure that following originality of Hebbian learning rule (with single neuronal output) is shown in Figure 4. It simulates realistically the process of teaching/learning children "How to read?". It complies with associative memorization concept motivated by classical conditioning (unsupervised) learning. I_1 and I_2 represent the inputs of sound (heard) stimulus and visual (sight) stimulus respectively. A and C represent two sensory neurons (receptors), while B and D are presenting nervous subsystems developing output responses. The generalized simple structure drives an output response (pronunciation) that is as O_1. However, O_2 is the output response that obtained when input sound is considered as conditioned stimulus. Hence visual recognition as conditioned response of the heard letter/word is obtained as output O_2.

The output response signal varies as shown in the original Pavlov experimental work [6], where it is measured quantitatively via the exactness of pronouncing letter/word. In accordance with biology, the strength of response signal is dependent upon the transfer properties of the output motor neuron stimulating salivation gland. In the present work, the transfer properties of output motor neurons simulate pronunciation as unconditioned response (UCR) for heard phoneme (sound signal). However, this pronounced output is considered as conditioned response (CR) when input stimulus is given by only sight (seen letter/word). In order to justify the superiority and optimality of the phonic methodology over other teaching to read approach an elaborated mathematical formulation will be introduced later.

The suggested ANN learning model obeys original self-organized Hebbian learning rule [5]. Hence, simulation of the reading process is inspired by that rule in analogous manner to previous Pavlovian conditioning model [6]. The input stimuli to the model are considered as either conditioned or unconditioned. Visual and audible signals are considered interchangeably for training the model to get desired responses at the output. Moreover, the model obeys more elaborate mathematical analysis for Pavlovian learning process [12]. In addition, the model is modified to follow the general Hebbian algorithm and correlation matrix memory.

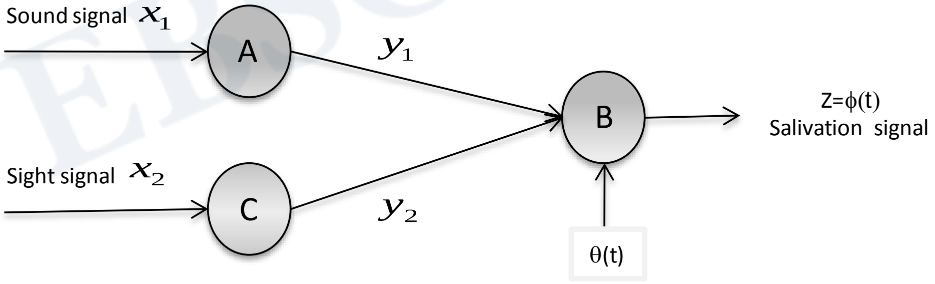


Figure 3. Structure of Neurobiological model.

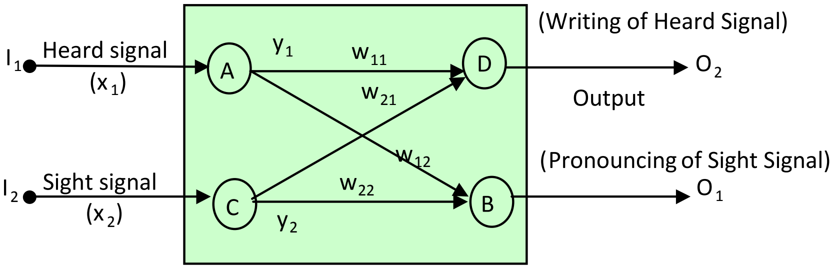


Figure 4. Generalized reading model of phonics methodology, (adapted from [17]).

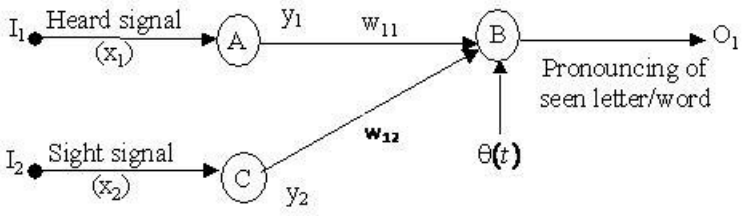


Figure 5. Structure of the first model for pronouncing seen letter/word (adapted from [17]).

Figure 5 presents the classical conditioning learning process inspired by Pavlov's psycho experimental work [6]. Where each of lettered circles A, B, and C presents a neuron cell body. The line connecting cell bodies represents an axon that terminates synaptic junctions. Both of two signals released out from sound and sight sensory neurons A and C are represented by y_1 and y_2.

Mathematical description with some simplifications of any neuronal cell arguments was illustrated as follows: the differential equation describing electrical neural activity was given by;

$$\frac{do_i}{dt} = \sum_{j=1}^{n} f\left(y_{ij}\right) - j\left(o_i\right) \tag{10}$$

where, y_{ij} represents the activity at the input (j) of neuron (i) , $f(y_{ij})$ indicates the effect of input on membrane potential, $j(o_i)$ is nonlinear loss term combining leakage signals, saturation effects occurring at membrane in addition to the dead time till observing output activity signal. The steady state solution of Eq.10, proved to be presented as transfer functions. Assuming, the linearity of synaptic control effect, the output response signal is given as:

$$O_i = \phi\left(\sum_{j=1}^{n} w_{ij} y_{ij} - \theta_i\right) \tag{11}$$

where, ϕ has two saturation limits, the function ϕ may be linear above a threshold and zero below or linear within a range but flat above θ_i is the threshold (offset) parameter, and w_{ij} synaptic weight coupling between two neuron *(i)* and *(j)*. Specifically, the function (ϕi) is recommended to be chosen as a ramp or sigmoid signal function [7]. However, the ramp function was used to represent the output response of the model, because of it’s mathematically similarity to sigmoid function. Referring to the weight dynamics described by the famous Hebb’s learning law, the adaptation process for synaptic interconnections is given by the following modified equation:

$$\frac{d\omega_{ij}}{dt} = \eta z_i y_{ij} - a\left(o_i\right)\omega_{ij} \tag{12}$$

Where, the first right term corresponds to the unmodified learning (Hebb's law) and η is a positive constant. The second term represents active forgetting; $a(o_i)$ is a scalar function of the output response (o_i). Referring to the model structure given in Figure4, the adaptation equation of the single neuronal model (between *i, j* nodes) is as follows:

$$\dot{w}_{ij} = -aw_{ij} + \eta y_i x_{ij} \tag{13}$$

Where w_{ij} represents synaptic connectivity between node *i* and node *j*, y_i is the output obtained from node *i*. The values of η, y_i and x_{ij} are assumed all to be non-negative quantities, η is the proportionality constant. The values of the constants η and *a* are less than one. The solution of the above equation is graphically illustrated elsewhere [18].

Assuming the ratio of the values of η and *a*; to be $(\eta / a = 1)$. Thus a linear neuron model for the output is fulfilled as suggested for generalized Hebbian algorithm. The study of the interrelations between the neural network models and experimental results of classical condition are presented earlier**,** [12,17].

4. Mathematical Formulation of Associative Memory Model

Referring to Figures 4 & 5, suggested reading model obeys the concept of Pavlovian learning .That is by considering the two inputs I_1, I_2 represent sound (heard) stimulus and visual (sight) stimulus respectively. However, the outputs O_1, O_2 represent pronouncing (reading), and image recognition (writing) processes respectively. In order to justify the superiority and optimality of phonic approach over other teaching to read methods an elaborated mathematical formulation is introduced to show how to perform reading tasks, in addition to writing recognized letters/words by audible signal. In more details, for the generalized learning/teaching topic model, Figure 4, consider X and Y to map input and output vectors presented as (I_1, I_2) and (O_1, O_2) respectively. Consequently, the input (conditioned/unconditioned) stimulus vector X with m-dimensionality (composed of m components) was decomposed into two smaller sub-vectors. The first small sub-vector with r-dimensionality (components) simulates the auditory/heard stimulus signal. However the other sub-vector has the rest (m-r)-dimensions simulates the visual/seen stimulus signal. Similarity the output vector Y considered to be decomposed into two smaller sub-vectors representing (conditioned/unconditioned) responses. Consider q pairs of patterns relating to X and Y vectors that formulate learning convergence of reading activities. The kth pair of patterns is represented by the key vector X_k and the memorized Y_k.

Let the key vector X_k with *m* input space dimensionality, which represents implicitly both of heard phoneme (sound signal) stimulus and its corresponding seen letter/word (visual signal) stimulus. Hence, this implicit vector should be decomposed into two smaller sub-vectors each with dimensionality less than *m*. Assume that sound signal is simulated as vector with *r* dimensionality. Obviously the correlated visual signal is simulated as a vector with dimension (*m*-*r*). This means in practical application that vanishing any of two smaller vectors implies non-existence of either input stimuli. In other words input conditioned or unconditioned stimulus is detected by measuring input space dimensionality of vector X_k. Similarly, the memorized vector Y_k represents two unconditioned/ conditioned response to the input stimulus vector X_k. However the dimensionality of that memorized response vector differs from that for input vector. Consequently, considering that Y_k vector have l dimensionality, hence it decompose into two smaller vectors as unconditioned and/or conditioned responses. Let Y_k vector with *l*-dimensionality implicitly includes both output response signals, i.e. when pronouncing signal vector have s-dimensionality the other recognizing process of seen letter/word is simulated as a vector with (*l*-s) dimensionality.

The ANN model obeys the mathematical analysis for Pavlovian learning process [6,12]. The model is modified to follow the general Hebbian algorithm and correlation matrix memory. So, the simulated reading process by suggested model performs analogously to the original Pavlovian conditioning experimental work [6,12]. The input stimuli to the model are considered as either conditioned or unconditioned stimuli. Visual and audible signals are considered interchangeably for training the model to get desired responses at the output of the model. Details about the mathematical formulation are given as following:

Consider $X_k^{'}$ and $X_k^{''}$ are the two vectors simulating heard and seen input stimuli respectively. Similarly $Y_k^{'}$ and $Y_k^{''}$ are the two vectors simulating pronouncing and visual recognizing output responses respectively. The two expected unconditioned responses are described in matrix form by the equation:

$$Y_k^{'} = W(k) \cdot X_k^{'} \qquad , k = 1,2,3,\ldots,q \tag{14}$$

Where *W*(*k*) is a weight matrix determined solely by the input-output pair $(X_k^{'}, Y_k^{'})$

$$y_{ki} = \sum_{j=1}^{r} w_{ij}(k) \cdot x_{kj} \qquad , i = 1,2,\ldots,r \tag{15}$$

Where $w_{ij}(k),\ j = 1,2,\ldots,r$ are the synaptic weights of neuron *i* corresponding to the *k*th pair of associated patterns of the input-output pair $(X_k^{'}, Y_k^{'})$. The equivalent form of y_{ki} is expressed by;

$$y_{ki} = \left[w_{i1}(k), w_{i2}(k), \ldots, w_{ir}(k)\right] \begin{bmatrix} x_{k1} \\ x_{k2} \\ \cdots\cdots \\ x_{kr} \end{bmatrix} \quad ; i = 1,2,\ldots,s \tag{16}$$

The visual input stimulus $X_k^{''}$ results in recognizing output response (seen letter/word) $Y_k^{''}$

$$y_{ki} = \left[w_{ir+1}(k), w_{ir+2}(k), \ldots, w_{im-r}(k)\right]\begin{bmatrix} x_{kr+1} \\ x_{kr+2} \\ \ldots\ldots \\ x_{km-r} \end{bmatrix} \tag{17}$$

$$i = s+1,2,3,\ldots,l$$

Similarly, for conditioned response, the input hearing stimulus $X_k^{'}$ results in recognizing of visual signal $Y_k^{''}$. However input seen letter/word stimulus $X_k^{''}$ results in pronouncing that letter/word as conditioned response vector $Y_k^{'}$, which expresses the reading activity, Eq.18.

$$y_{ki}^{'} = \left[w_{ir+1}(k), w_{ir+2}(k), \ldots, w_{im-r}(k)\right]\begin{bmatrix} x_{kr+1}^{''} \\ x_{kr+2}^{''} \\ \ldots\ldots \\ x_{km-r}^{''} \end{bmatrix} \tag{18}$$

$$i = 1,2,3,\ldots,s$$

In a similar manner the other conditioned response for recognizing heard phoneme is described by;

$$y_{ki}^{''} = \left[w_1(k), w_2(k), \ldots, w_r(k)\right]\begin{bmatrix} x_{kr+1}^{'} \\ x_{kr+2}^{'} \\ \ldots\ldots \\ x_{km-r}^{'} \end{bmatrix} \quad ; i = 1,2,\ldots,s \tag{19}$$

As a result of the above equation the memory matrix that represents all q-pairs of pattern associations is given by $m*l$ memory correlation matrix as follows

$$M = \sum_{k=1}^{q} W(k)$$

where *W(k)* weight matrix is defined by

$$W(k) = \begin{bmatrix} w_{11}(k) & w_{12}(k) & \ldots & w_{1m}(k) \\ w_{21}(k) & w_{22}(k) & \ldots & w_{11}(k) \\ \ldots & \ldots & \ldots & \ldots \\ w_{l1}(k) & w_{l2}(k) & \ldots & w_{lm}(k) \end{bmatrix} \tag{20}$$

This weight matrix relating input stimulus vector with m-dimensionality X_k connected by synapses via output response vector Y_k with *l*-dimensionality. The complete relation for input/ output relation is given by the following equation

$$\begin{bmatrix} y_{k1} \\ y_{k2} \\ .... \\ y_{kl} \end{bmatrix} = \begin{bmatrix} w_{11}(k) & w_{12}(k) & ... & w_{1m}(k) \\ w_{21}(k) & w_{22}(k) & ... & w_{11}(k) \\ ... & ... & ... & ... \\ w_{l1}(k) & w_{l2}(k) & ... & w_{lm}(k) \end{bmatrix} \cdot \begin{bmatrix} x_{k1} \\ x_{k2} \\ .... \\ x_{km} \end{bmatrix} \tag{21}$$

It is worthy to note that, the above equation represents memory correlation matrix after learning convergence. So this matrix is given in other way as:

$$M = Y \cdot X^T \tag{22}$$

The above equation illustrates that all values of memory matrix M elements present synaptic weights relating key pattern X with memorized stored patterns Y. In other words, the relation between input patterns to the proposed model and that model's output patterns is tightly closed by the steady state values of the memory matrix M after reaching of learning convergence. Noting, that learning process obeys well the above presented ANN model performance (referring to Figure 2); which valid for both suggested learning/teaching topics.

5. Simulation Results

5.1. Children's Achievement Analysis

At this section realistic simulation results are illustrated in Figure 6 as well as in Table 1. Those are obtained after computer running of an ANN model adapted from realistic learning simulation with considering various learning rate values. It is worthy to note that learning rate value associated to CAL with teacher's voice proved to be higher than CAL without voice. Simulation curves at Figure 6 illustrate statistical comparison for two learning processes with two different learning rates. The lower learning rate (η = 0.1) may be relevant for simulating classical learning process. However, higher learning rate (η = 0.5) could be analogously considered to indicate the case of CAL process applied without teacher's voice.

The experimental results in Tables 2 and 3 were obtained after performing three different learning experiments. The results in Table 2 are classified in accordance with different students' learning styles following three teaching methodologies. Firstly, the classical learning style is carried out by students-teacher interactive in the classroom. Secondly, learning is taken place using a suggested software learning package without teacher's voice association. The last experiment is carried out using CAL package that is associated with teacher's voice. This table gives children's achievements (obtained marks) considering that maximum mark is 100. The statistical analysis of the three experimental marking results is given in details at Table 3.

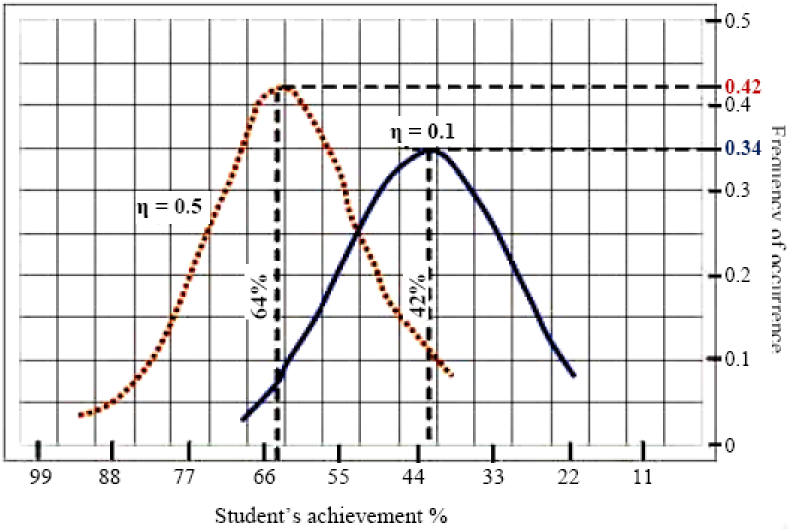


Figure 6. Simulation results presented by statistical distribution for children's achievements versus the frequency of occurrence.

Table 1. Simulation results for different learning rate values η

Learning rate value	Children's average achievement score (M)	Variance σ	Standard deviation $\sqrt{\sigma}$	Coefficient of variation $\rho = \sqrt{\sigma}$ /M	Improvement of teaching quality
η = 0.1	42	428.5	20.7	0.61	--
η = 0.5	64	918.1	30.3	0.47	66%

Table 2. Illustrates children's marks after performing three educational experiments

Classical Learning	35	43	29	50	37	17	10	60	20	48	15	55	40	8	20
CAL (without voice)	39	29	52	60	50	68	62	30	55	42	40	59	48	70	2
CAL (with voice)	65	70	50	75	45	50	62	90	85	50	80	90	58	55	60

Table 3. Illustrates statistical analysis of above obtained children's marks

Teaching Methodology	Children's average achievement score (M)	Variance σ	Standard deviation $\sqrt{\sigma}$	Coefficient of variation $\rho = \sqrt{\sigma}$ /M	Improvement of teaching quality
Classical	32.46	265.32	16.28	0.5	--
CAL (without tutor's voice)	46.80	297.49	17.24	0.36	44.1%
CAL (with tutor's voice)	64.33	283.42	16.83	0.26	98.2%

Table 4. Effect of noisy environment on learning of reading convergence

Noise Power	0.2	0.1	0.05
(s/n) Ratio	5	10	20
t (cycles)	85	62	47

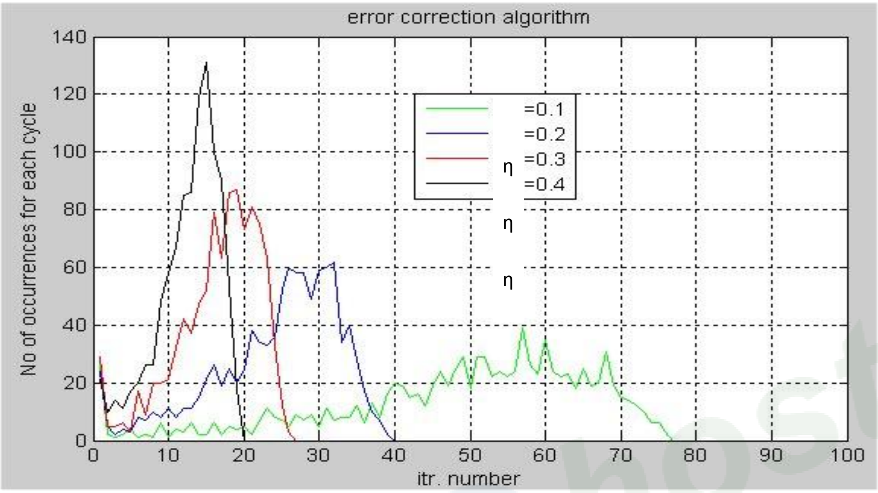


Figure 7. Statistical distribution of learning convergence time for different learning rate values.

Lindstrom [19] found that participants could only remember 20% of the total materials when they were presented with visual material only, 40% when they were presented with both visual and auditory material, and about 75% when the visual and auditory material were presented simultaneously. The results in Table 3 agree with the results of Lindstrom [19].

5.2. Effect of Noisy Environment on Learning Performance

Naturally, ideal (noiseless) learning environment is not available in practice. Usually, it is environmental learning data is vulnerable to contaminations by either external or internal noisy conditions. So, learning/teaching processes in our classrooms, for either suggested topics, have to be accomplished under influenced effect of some environmental noise [20]. Therefore considerable attention has been paid in this subsection to report the effect of either noisy CAL environment or noisy teacher on learning convergence time.

The simulation results were obtained via association between the two input stimuli (visual and auditory) following classical conditioning learning [6,12]. Obtained results for optical character recognition under different noise levels are given in a tabulated form as in Table 4. Practically, the best way to teach children how to read is carried out under the effect of less noisy data.

The noise effect is measured by signal to noise ratio value (s/n) versus the number of training cycle**s (t).** Conclusively, an interesting remark observed considering relation between number of training cycles values (convergence learning time), and noisy environmental data in case of application of adopted ANN learning model.

The convergence time cycles (t), of learning process is inversely proportional to signal to noise ratio values, (s/n) and learning rate values. On the other hand, it is directly proportional to noise power value(s) [2]. Additionally, the evaluated relation between learning rate values and noisy data (teacher) appeared considering unsupervised learning. The convergence time of learning process is reached after 47, 62, 85 training cycles when noise power is 0.05, 0.1 and 0.2, respectively, as shown at Table 4.

Running the ANN simulation program with three numerical learning rate values (0.2, 0.1 and 0.05) resulted in the training cycles (3, 5, and 10) respectively. Figure7 shows the statistical distribution for the relation between number of training cycles and learning rates. This distribution is seems to be similar to Gaussian (normal) as it has a bell shape, which proved the realistic of the developed model.

CONCLUSION

This article presents some interesting results associated with reaching optimal methodologies for two suggested tutorial topics. More precisely, presented research work adopted the above introduced ANN model that is based on cognitive associative learning with visual and auditory materials. Interestingly, The introduced mathematical justification model (given at section 4) as well as obtained results after running our suggested model are supported by following recent founding:

1. The statistical distribution for the relation between number of training cycles and learning rates seems to be similar to Gaussian (normal) as it has a bell shape, which proved the realistic of the developed model.
2. The obtained field testing results were supported well by cognitive multimedia theory (visual and auditory material should be simultaneously presented, on the basis of memory association, to reinforce the retention of learned materials).
3. The results of ANN model agree with the experiential results of Lindstrom.
4. As future expected extension of presented paper, it is highly recommended to consider more elaborate investigational analysis and evaluations for other behavioral and cognitive learning phenomena observed at educational field (such as learning creativity, improvement of learning performance, learning styles,……etc.) using ANNs modeling. As consequence of all presented in above, it is worthy to recommend implementation of modified ANNs models, to be realistically applicable in solving more educational phenomena issues associated with diverse cognitive styles observed at educational field activities.
5. The prospective modification of suggested CAL package is measurement of time response (learning) parameter may be promising for more elaborate of learning performance measurement in practice application (educational field classrooms). This time parameter is recommended for educational field practice [17,21]. For very recently published research work, this parameter suggested for of e-learning systems performance measurement [22].

REFERENCES

[1] Rayner, K., Foorman, B. R., Perfetti, C. A. & Pesetsky, Seiedenberg, M. S. (2003). How Should Reading be Taught?. *Majallat Aloloom (Scientific American*), *19(6/7)*, 4-11.

[2] Rayner, K., Foorman, B. R., Perfetti, C. A., Pesetsky, D. & Seidenberg, M. S. (2001). How Psychological Science Informs the Teaching of Reading. Psychological Science *In Public interest*., *2(2)*, 31-74.

[3] Jeanne, S. Chall, (1996). Learning to read, the great debate. New York: McGraw Hill

[4] Eric, H. & Johnson, A. Learning and Reasoning about Interruption. Proceeding of the 5th *International Conference on Multimodal Interfaces*, Nov.

[5] Hebb, D. O. (1949). The Organization of Behavior. Wiley, New York.

[6] Pavlov, I. P. (1927). Conditional Reflex, an Investigation of the Psychological Activity of the Cerebral Cortex. New York , Oxford University press.

[7] Yu, C. H., Digangi, S., Jannasch-Pennell, A., Stay, V., Lo, W. J. & Kilic, Z. (2003). On Analysis and Evaluation of Multi-Sensory Cognitive Learning as Facilitated in a Multimedia Tutorial for Item Response Theory. Systemics, *Cybernetics and Informatics*, *5(4)*, 13-22.

[8] Swaminathan, N. (2007). Cognitive Ability Mostly Developed Before Adolescence, NIH Study Says, *NIH announces preliminary findings from an effort to create a database that charts healthy brain growth and behavior*. Scientific American letter, May 18, 2007.

[9] Al-Zahrani, F. A., Mustafa, H. M. & Al-Hamadi, A. (2010). On Analysis And Evaluation Of Multi-Sensory Cognitive Learning Of A Mathematical Topic Using Artificial Neural Networks", *Journal Of Telecommunications*, *1(1)*, 99-104.

[10] Mustafa, H. M. & Al-Hamadi, A. (2009). On Teaching Quality Improvement of A Mathematical Topic Using Artificial Neural Networks Modeling (With A Case Study). published at 10th (Anniversary) International Conference Models in Developing Mathematics Education" to be held in Dresden, Saxony, Germany on September 11-17.

[11] Haykin, S. (1999). Neural Networks, Englewood Cliffs, NJ. Prentice-Hall, 50-60.

[12] Hassan H. & Watany M. (2000). "On Mathematical Analysis of Pavlovian Conditioning Learning Process using Artificial Neural Network Model. 10th Mediterranean Electro technical *Conference, Cyprus*.

[13] Mustafa, H. M. (2007). On Quantifying Learning Creativity Using Artificial Neural Networks (Nero-physiological Cognitive Approach) Published at National Conference on *Applied Cognitive Psychology*. India,29 –30 November.

[14] Fukaya, M., kitagwa M. & Okabe, Y. (1987). *Two level Neural Networks: Learning by Interaction with Environment.* 1st ICNN, San Diego.

[15] Mustafa, H. M. (2009). *On Evaluation of Virtual Improvement of Learning Creativity by Application of Computer Assisted Learning Using Artificial Neural Networks Modeling.* 6th Annual International Conference on Remote Engineering and Virtual Instrumentation REV2009, University of Bridgeport, CT, USA, 18-22 June.

[16] Mustafa, H. M. (2005). *On Simulation of Adaptive Learner Control Considering Students' Cognitive Styles using Artificial Neural Networks (ANNs).* Published at CIMCA05, Austria 28-30 Nov.

[17] Mustafa, H. M, Al-Hamadi, A. & Al-Saleem, S. (2007). Towards Evaluation of Phonics Method for Teaching of Reading Using Artificial Neural Networks (A Cognitive Modeling Approach). Published at IEEE Symposium on Signal Processing and Information *Technology Seventh Symposium*, Cairo.

[18] Freeman, J. A. (1994). Simulating Neural Networks with Mathematics, Addison-Wesley publishing company.

[19] Lindstrom, R. L. (1994). The Business Week Guide to Multimedia Presentations: Create Dynamic Presentations that Inspire, New York: McGraw-Hill.

[20] Ghoaimy M. A., Al–Bassiouni, A. M. & Mustafa, H. M. (1994). A Learning of Neural Networks using Noisy Data, Second International Conference On Artificial Intelligence *Application, Cairo, Egypt*, 389-399.

[21] Mustafa, H. M. (2004). Evaluation of Learning/Training Convergence Time Using Neural Network (ANNs). Published at international conference of *Electrical Engineering*, 542-549, 24-26 Nov.

[22] Mustafa, H. M. (2009). "On Simulation of E-learning Convergence Time using Artificial Neural Networks. Published at the 7th International Conference on Education and Information Systems, Technologies, and Applications (EISTA), Orlando, USA, on July 10-13.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 4

PREDICTION OF HOLE QUALITY IN DRILLING GFRE USING ARTIFICIAL NEURAL NETWORKS

U. A. Khashaba[1*], I. A. El-Sonbat[2*], A. I. Selmy[2*] and A. A. Megahed[2*]

[1]Department of Production Engineering & Mechanical Systems Design, Faculty of Engineering King Abdul Aziz University, Jeddah 21589, KSA
[2]Mechanical Design and Production Engineering Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt

ABSTRACT

The weight and fuel savings offered by composite materials make them attractive not only to the military, but also to the civilian aircraft, space, and automobile industries. In these industries, drilled holes are extensively implemented for structure assembly. The presence of hole defects due to drilling reduces the stiffness and strength of a laminate and hence its load carrying capacity. The main objective of the present work is to develop artificial neural networks (ANNs), with back-propagation training routine, for predicting the machinability parameters in drilling glass fiber reinforced epoxy (GFRE) composites with different machining conditions (feed, speed, and drill pre-wear). Machinability parameters were characterized by thrust force, torque, peel-up and push-out delaminations, and surface roughness of drilled holes.

The inputs to the neural networks used for predicting delamination size and surface roughness are: spindle speed, feed, drill pre-wear, thrust force, and torque. The values of the thrust force and torque that are fed as inputs to the above networks are predicted using ANNs developed for predicting each of them. Several attempts were performed to achieve the best neural network by changing both of network structure (i.e. the number of hidden layers and the number of units within each hidden layer) and the initial values of

[*] Corresponding author: khashabu@zu.edu.eg & ukhashaba@kau.edu.sa; On Leave from the Department of Mechanical Design and Production Engineering, Zagazig University, Zagazig, Egypt

the connection weights and thresholds. The best obtained network structure for predicting thrust force, torque, peel-up delamination, push-out delamination, surface roughness were 3-5-1, 3-3-1, 5-11-1, 5-7-3-1, and 5-7-3-1, respectively. The developed ANNs predict the machinability parameters (thrust force, torque, peel-up and push-out delaminations and surface roughness) with acceptable errors for the most confirmation tests.

Keywords: Drilling, Composites, Thrust force, Torque, Delamination size, Surface roughness, Artificial Neural Networks (ANNs).

1. INTRODUCTION

Machining composite materials is a rather complex task owing to its heterogeneity, heat sensitivity, and to the fact that reinforcements are extremely abrasive. Conventional machining methods should be adapted in such a way that they diminish thermal and mechanical damage. Drilling is a frequently practiced machining process in industry owing to the need for component assembly in mechanical pieces and structures. The drilling of laminate composite materials is significantly affected by the tendency of these materials to delaminate and the fibers to bond from the matrix under the action of machining forces (thrust force and torque). The presence of delamination reduces the stiffness and strength of a laminate and hence its load carrying capacity. Delamination can often be the limiting factor in the use of composite materials for structural applications, particularly when subjected to compressive, shear and fatigue type of loads and when exposed to moisture and other aggressive environments over a long period of time.

Artificial neural networks (ANNs) have recently been introduced into the field of polymer composites. Inspired by the biological nervous system, ANNs can be used to solve a wide variety of complex scientific and engineering problems. Like their biological counterparts, ANNs can learn from examples, and therefore can be trained to find solutions of the complex non-linear, multi-dimensional functional relationships without any prior assumptions about their nature; further, the network is built directly from experimental data by its self-organizing capabilities [1].

Stone and Kishnamurthy [2] developed a thrust force controller to minimize the delamination associated with drilling in graphite-epoxy laminate. A neural network control scheme was implemented which required a neural network identifier to model the drilling dynamics and a neural network controller to learn the relationship between feed rate and the desired thrust force. The robustness of the controller was demonstrated by varying some of the drilling parameters, spindle speed and drill diameter. Enemuoh et al. [3] used an intelligent sensor fusion technique based on artificial neural network to predict on-line delamination during drilling of an advanced fiber composite beam (AS4/PEEK). The fusion model included two drilling parameters (feed rate and cutting speed), two drilling conditions (tool material and tool geometry) and two sensors (thrust force and acoustic emission).

Chakraborty [4] aimed at developing an artificial neural network model for detection of extent of delamination, its shape and location in a graphite/epoxy composite laminate using natural frequencies as inputs and corresponding size, shape and location of delamination as outputs of the network. Hundreds of finite element models have been run to generate natural frequencies up to ten modes for various combinations of size, shape and location of an embedded delamination in a laminate and these data have been used to train a back propagation neural network for future prediction of delamination in the laminate. Sardiñas et al. [5] proposed a multi-objective optimization of the drilling process of a laminate composite material. Two mutually conflicting objectives are optimized: material removal rate, which represents the productivity; and delamination factor, which characterizes the superficial quality. A micro-genetic algorithm was implemented to carry out the optimization process. An a posteriori approach was used to obtain a set of optimal solutions. Finally, the obtained outcomes were arranged in graphical form (Pareto's front) and analyzed to make the proper decision for different process preferences.

Srinivasa Rao et al. [6] used the multi-variable linear regression analysis to make the correlation between the delamination factor and the drilling parameters; feed rate, spindle speed and drill diameter, when drilling glass/epoxy woven mat cross-ply laminates. Karnik et al. [7] predicted the delamination factor at the entrance side of drilled CFRP plates, using the multilayer feed forward ANN model trained by error-back propagation training algorithm, with spindle speed, feed rate and point angle as the inputs to the developed ANN. Drilling experiments are conducted as per full factorial design using cemented carbide (grade K20) twist drills that serve as input–output patterns for ANN training.

Surface roughness is a commonly encountered problem in machined surfaces. It is defined as the finer irregularities of surface texture, which results from the inherent action of the production process. Consequently, surface roughness has a great influence on product quality, and the part functional properties such as lubricant retentivity, void volume, load bearing area, and frictional properties [8]. Surface roughness cannot be controlled as accurately as geometrical form and dimensional quality as it fluctuates according to many factors such as machine tool structural parameters, cutting tool geometry, workpiece and cutting tool materials, environment, etc. In other words, surface quality is affected by the machining process, e.g. by changes in the conditions of either the workpiece, tool or machine tool. Surface roughness changes over a wide range in response to these parameters [9].

Behnam and Suman [10] proposed an indirect method to monitor the cutting force (thrust force) during the drilling operation and map it to the corresponding hole quality using artificial neural networks. They performed drilling experiments to evaluate the effect of the thrust force, speed and feed-rate on hole quality in drilling of aluminum alloys, 2024-T3, 7075-T3. Variables under consideration for evaluation of the hole quality are the average burr height and surface roughness parameter R_a. Mathews and Shunmugam [11] carried out a reaming process in EN4 steel work samples for the purpose of condition monitoring. To enhance the capability of sensor system in monitoring all the relevant aspect of the cutting process, a multisensor strategy on acoustic emission, force and vibration signals were considered. An artificial neural network, using Back-propagation algorithm, was trained using thrust, torque, acoustic emission and vibration parameters as input vectors and surface roughness parameter R_a, roundness error and residual stress as output vectors.

Enemuoh et al. [3] used an intelligent sensor fusion technique based on artificial neural network to predict on-line surface roughness during drilling of an advanced fiber composite beam (AS4/PEEK). The fusion model included two drilling parameters (feed rate and cutting speed), two drilling conditions (tool material and tool geometry) and two sensors (thrust force and acoustic emission). The final network predicted surface roughness with error ranging from 0% and 5%. Tsao and Hocheng [12] proposed an experimental approach, when drilling woven WFC200 fabric carbon fiber/epoxy matrix, to the prediction and evaluation of thrust force and surface roughness produced by candle stick drill using regression analysis of experiments and ANN. The authors found the feed rate and the drill diameter are recognized as the most significant factors affecting the thrust force, while the feed rate and spindle speed are seen to make the largest contribution to the surface roughness. In the confirmation tests, ANN is demonstrated more effective than multi-variable regression analysis for the prediction and evaluation of drilling-induced thrust force and surface roughness in drilling of the used composite material.

This paper summarizes an approach for prediction of hole quality (represented by delamination at both drill entry and exit sides of the hole as well as surface roughness, R_a parameter, of the produced hole wall) resulting from drilling GFRE specimens, using the feed-forward artificial neural networks (ANNs) technique trained with the back-propagation routine. The inputs to the neural networks were; spindle speed, feed, drill pre-wear, thrust force, and torque.

2. Artificial Neural Networks

Artificial neural network algorithms are regarded as multivariate nonlinear analytical tools capable of recognizing patterns from noisy complex data and estimating their nonlinear relationships. Their major advantages include superior learning, noise suppression, and parallel data processing capabilities [13]. Further, the network is built directly from experimental data by its self-organizing capabilities [1].

Artificial neural networks have highly interconnected structure similar to brain cells of human neural networks and consist of large number of simple processing elements called neurons, which are arranged in different layers in the network. Each network consists of an input layer, an output layer and one or more hidden layers. One of the well-known advantages of ANN is that the ANN has the ability to learn from the sample set, which is called training set, in a supervised or unsupervised learning process. Once the architecture of network is defined, then through learning process, weights are calculated so as to present the desire output [14].

In the present work, in order to, develop the required neural networks, from all of the holes that were machined in the experimental work, holes were selected randomly as; 100 holes for training patterns, 25 holes for cross validation patterns (which were used as a training stoppage criterion [15]), and 12 holes were used as test patterns. All of the developed ANNs are of multi-layer perceptron type and trained using the back-propagation routine [16,17]. A NeuroSolutions software (version 5) [18] was used in the training, validating and testing process of the developed neural networks. For training, a value of 0.7 was selected for the momentum term, a starting value of 1.0 was assigned for the learning rate, and maximum epochs of 40,000 were chosen with batch weight update method. Each network was obtained by training five times (runs) starting with five different initial weight values, the run and epoch numbers which gave the minimum mean square error (MSE) of the validation sets were chosen as the required net. The criterion of selecting the best net from the tried networks is the minimum mean square of the differences between the measured and the predicted values obtained from the net, applied for the test data sets.

3. Experiment Work

3.1. Specimen Preparation

Quasi-isotropic polymeric composite laminates were fabricated from epoxy resin reinforced by woven E-glass fiber using hand lay- up process presented earlier [19]. Care must be taken when cutting and laying the woven glass fiber layers. The cutting must be through the warp and weft threads to ensure right angles of all layers. The laminates consist of 25 layers with 8.3 mm thickness. The fiber volume fraction (V_f) was determined experimentally using the ignition technique according to ASTM D3171-99. The average value of V_f was 35%.

3.2. Drilling Processes

A StankoImport (Moskva - SSSP) radial drill machine, 21 spindle rotational speeds (range from 20 to 2000 RPM) and 12 longitudinal feeds (range from 0.056 to 2.5 mm/rev.) was used for drilling the specimens. All specimens were drilled using cemented carbide drill with 8 mm diameter. The specimens were drilled under dry cutting conditions with five spindle speeds (V = 6.41, 12.71, 20.25, 32.03, and 50.63 m/min), five feeds (f = 0.056, 0.112, 0.22, 0.315, 0.45 mm/rev) and five drill pre-wear values (fresh drill plus four artificially introduced pre-wear values; W = 7, 19, 26, 34 gm x10^{-4}). Specimens were clamped, on the dynamometer, between two plates each has a center hole of 26 mm.

3.3. Thrust Force and Torque Measurements

In the present work a two component dynamometer, based on strain-gage sensor, has been designed and manufactured to measure the thrust force and torque during the drilling processes. Details about drilling set-up, and the drill dynamometer are illustrated elsewhere, Khashaba et al. [20,21].

3.4. Delamination Measurements

The surface delamination was measured using the “AutoCad method”, which suitable for quasi-transparent composite materials. Details for the technique set-up and the measuring processes are published earlier, Khashaba [20]. The size of the delamination is defined as the difference between the maximum damage radius (R_{max}) and the drilled hole radius (R = 4 mm).

3.5. Surface Roughness Measurements

The surface roughness of the drilled hole wall (R_a) was measured using “Rank Taylor Hobson Surtronic 3+” surface roughness measuring instrument. The cut-off and traversing length values were taken 0.8 mm and 4 mm, respectively [22]. The surface roughness value of each specimen is the arithmetic average value of three surface roughness measurements made across the lay, at three different positions along the circumference of the hole wall, separated by around 60^o - 90^o [22,23].

At the end of the experimental work there were 125 holes. Additional 12 holes were drilled with cutting conditions different from that of the experimental work for the purpose of testing the developed neural networks for generalization. The experimental results of thrust force, torque, peel-up delamination, push-out delamination and surface roughness are presented elsewhere, Khashaba et al. [24].

4. Results and Discussion

The effect of the machining parameters (feed, speed and drill pre-wear) on the machining parameters (thrust force, torque, delamination and surface roughness) are discussed elsewhere, Khashaba et al. [22,24].

4.1. Anns for Thrust Force and Torque Prediction

Among the inputs to the ANNs for delamination size and surface roughness prediction, there are thrust force and torque. Thus, before using the developed networks, thrust force and torque generated during the drilling process should be predicted in order to eliminate the need for measuring each of thrust force and torque at every prediction process. The value of the thrust force and torque that is fed to the ANNs is the average value of the maximum five peaks in drilling process of the hole.

To obtain the best neural network structure for each of thrust force and torque, eleven neural networks were developed by changing both of the number of hidden layers and the number of hidden units within each hidden layer.

(1) ANNs for thrust force prediction

Figure 1 shows the schematic diagram of the neural net for predicting the thrust force (F_t). Three inputs were fed to the network; cutting speed, feed, and drill pre-wear. Table 1 represents the tried neural networks that were used for the prediction of F_t. From the Table it can be seen that the best neural network for predicting F_t is the net which has the structure of 3 input units, 5 hidden units in the hidden layer and one output node (F_t), i.e. 3-5-1 structure, where its test data set error is the lowest value (1086.91 N^2). Figure 2 shows the relationship between the measured and the predicted values of F_t for this best net, including the results of training, validation and test data sets.

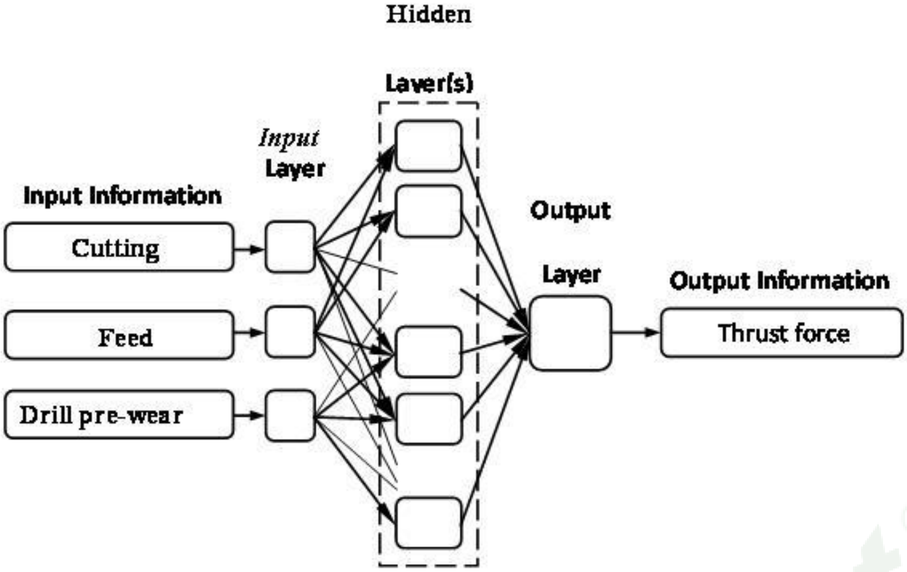


Figure 1. Schematic diagram of the neural network for predicting thrust force.

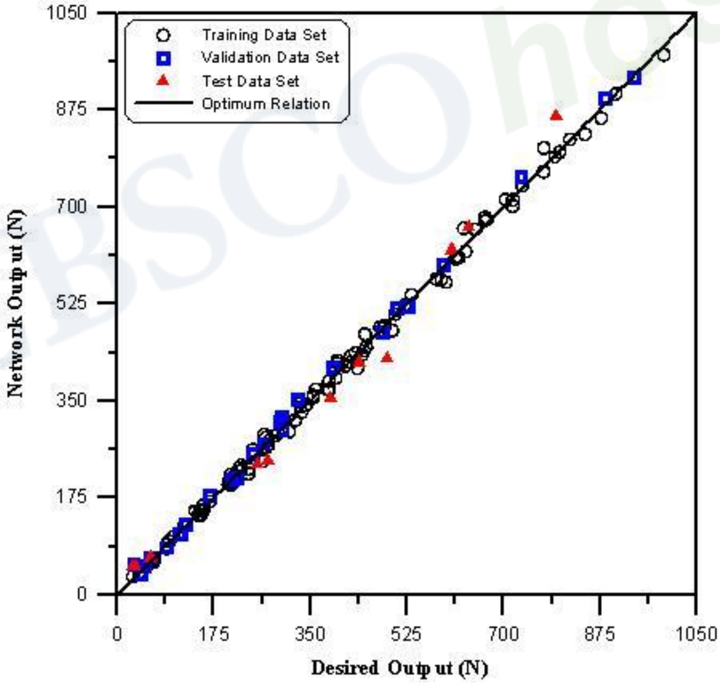


Figure 2. Relationship between actual (desired) and predicted (output) values of the best obtained network for predicting thrust force.

(2) ANNs for torque prediction

Figure 3 shows the schematic diagram of the neural net for predicting the torque (*T*). Three inputs were fed to the network; cutting speed, feed, and drill pre-wear. Table 2 represents the tried neural networks that were used for the prediction of *T*. From the Table it can be seen that the best neural network for predicting *T* is the net which has the structure of 3 input units, 3 hidden units in one hidden layer, and one output node (*T*), i.e. 3-3-1 structure, where its test data set error is the lowest value (0.00978 (N.m)^2). Figure 4 shows the relationship between the measured (target) and the predicted (output) values of *T* for this best net, including the results of training, validation and test data sets.

Table 1. Trials for the ANNs for predicting thrust force

No.	Network Structure	Training Set MSE (N^2)	Validation Set MSE (N^2)	Test Set MSE (N^2)
1	3-3-1	563.0180	826.8925	2756.5914
2	3-5-1*	126.4593	144.6190	1086.9103
3	3-7-1	91.7994	128.0033	1955.1853
4	3-9-1	72.8345	93.6432	1858.2038
5	3-3-3-1	525.1276	716.8111	3157.5226
6	3-3-5-1	471.7167	620.1074	1813.8737
7	3-5-3-1	94.0000	137.3710	5324.9907
8	3-5-5-1	80.3308	119.2037	3282.0877
9	3-5-8-1	110.4572	109.7060	2961.7219
10	3-8-5-1	84.7874	108.3428	9412.8380
11	3-8-8-1	62.2545	117.9673	2875.9578

* The best network structure

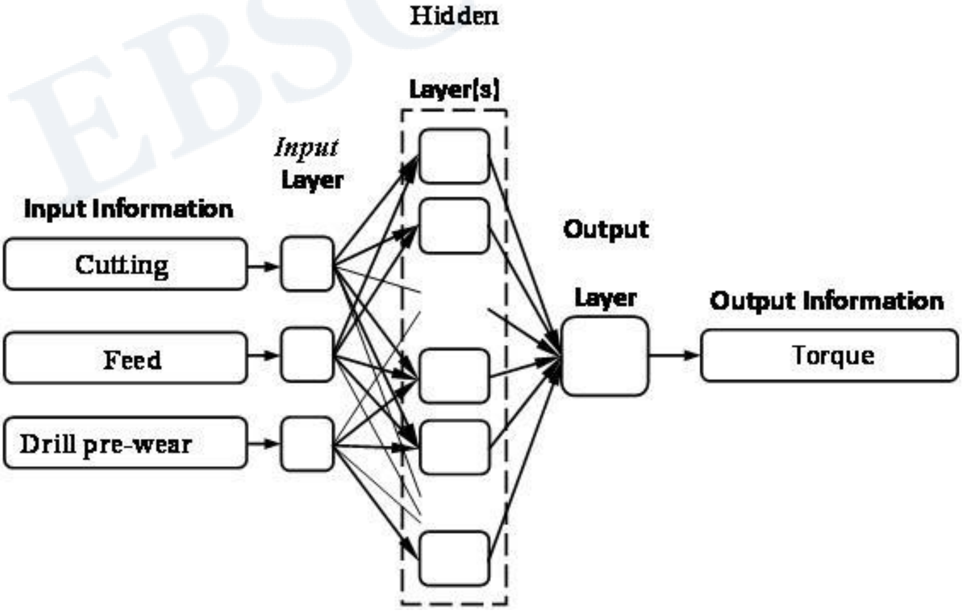


Figure 3. Schematic diagram of the neural network for predicting torque.

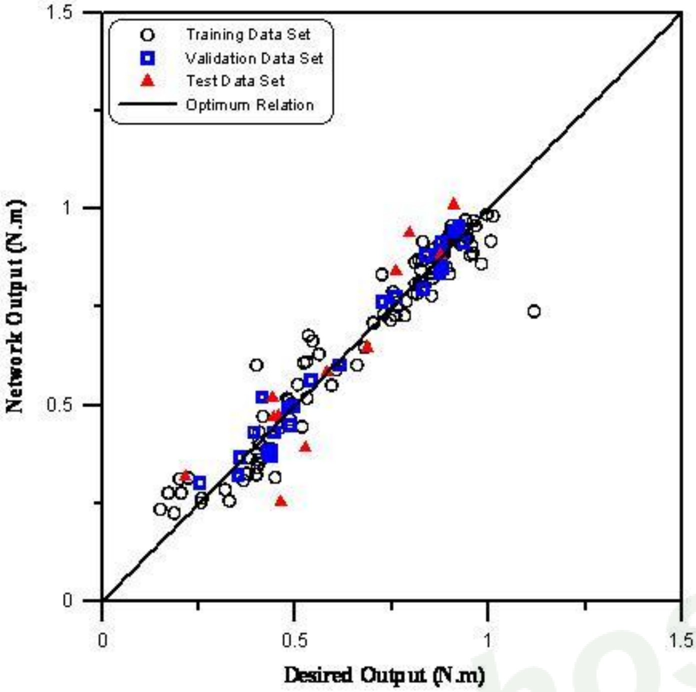


Figure 4. Relationship between actual (desired) and predicted (output) values of the best obtained network for predicting torque.

Table 2. Trials for the ANNs for predicting torque

No.	Network Structure	Training Set MSE $(N.m)^2$	Validation Set MSE $(N.m)^2$	Test Set MSE $(N.m)^2$
1	3-3-1*	0.00450	0.00159	0.00978
2	3-5-1	0.00385	0.00118	0.01099
3	3-7-1	0.00374	0.00119	0.01242
4	3-9-1	0.00327	0.00129	0.01430
5	3-3-3-1	0.00387	0.00151	0.01234
6	3-3-5-1	0.00372	0.00129	0.01387
7	3-5-3-1	0.00363	0.00112	0.01104
8	3-5-5-1	0.00357	0.00107	0.01191
9	3-5-8-1	0.00363	0.00135	0.01129
10	3-8-5-1	0.00345	0.00120	0.01208
11	3-8-8-1	0.00311	0.00108	0.01317

* The best network structure.

4.2. ANNs for Delamination Size Prediction

Figure 5 shows the schematic diagram of the neural net for predicting the delamination size at drill entrance and drill exit. For each output, five inputs were fed to the network (spindle speed, feed, drill pre-wear, thrust force, and torque).

To obtain the best neural network structure, eleven neural networks were developed by changing both of the number of hidden layers and the number of hidden units within each hidden layer. Tables 3 and 4 represent the tried networks for predicting delamination at drill entry and exit sides respectively. From these Tables, the best network structures, among the present trials, for predicting delamination size at hole entry and exit sides are 5-11-1 (with a MSE of 0.0499 mm^2) and 5-7-3-1 (with a MSE of 0.1203 mm^2) respectively.

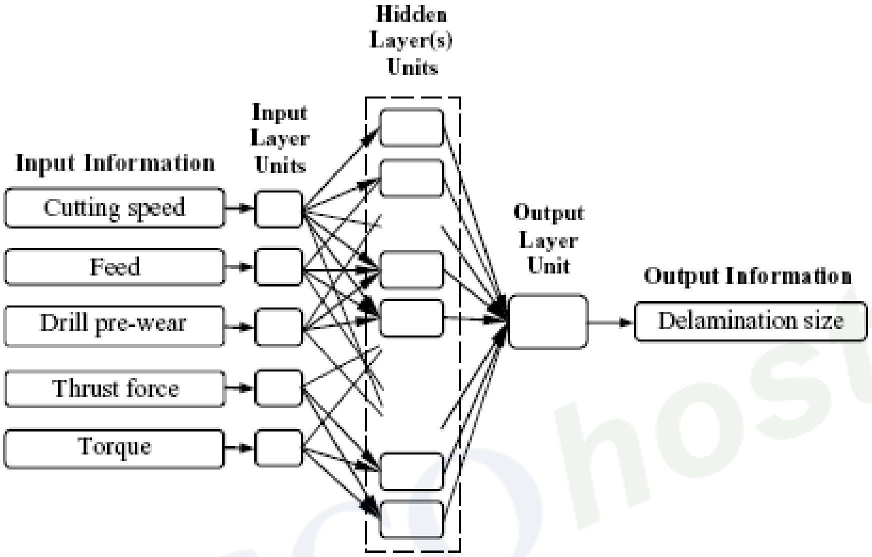


Figure 5. Schematic diagram of the neural network for predicting delamination size at drill entrance and exit sides.

Table 3. Trials for the ANNs for predicting delamination at drill entrance

No.	Network Structure	Training Set MSE (mm^2)	Validation Set MSE (mm^2)	Test Set MSE (mm^2)
1	5-3-1	0.0107	0.0114	0.0781
2	5-5-1	0.0049	0.0077	0.0716
3	5-8-1	0.0029	0.0105	0.0570
4	5-11-1*	0.0026	0.0061	0.0499
5	5-3-3-1	0.0138	0.0141	0.0785
6	5-3-7-1	0.0131	0.0138	0.0626
7	5-7-3-1	0.0019	0.0140	0.0755
8	5-7-7-1	0.0037	0.0101	0.0634
9	5-7-11-1	0.0021	0.0063	0.0646
10	5-11-7-1	0.0017	0.0074	0.0540
11	5-11-11-1	0.0016	0.0093	0.0802

* The best network structure.

Table 4. Trials for the ANNs for predicting delamination at drill exit

No.	Network Structure	Training Set MSE (mm^2)	Validation Set MSE (mm^2)	Test Set MSE (mm^2)
1	5-3-1	0.0368	0.0529	0.2646
2	5-5-1	0.0172	0.0478	0.1717
3	5-8-1	0.0161	0.0511	0.1898
4	5-11-1	0.0105	0.0386	0.2303
5	5-3-3-1	0.0403	0.0492	0.2834
6	5-3-7-1	0.0271	0.0394	0.2489
7	5-7-3-1*	0.0114	0.0612	0.1203
8	5-7-7-1	0.0352	0.0577	0.2313
9	5-7-11-1	0.0178	0.0505	0.2411
10	5-11-7-1	0.0308	0.0562	0.2441
11	5-11-11-1	0.0288	0.0550	0.2721

* The best network structure.

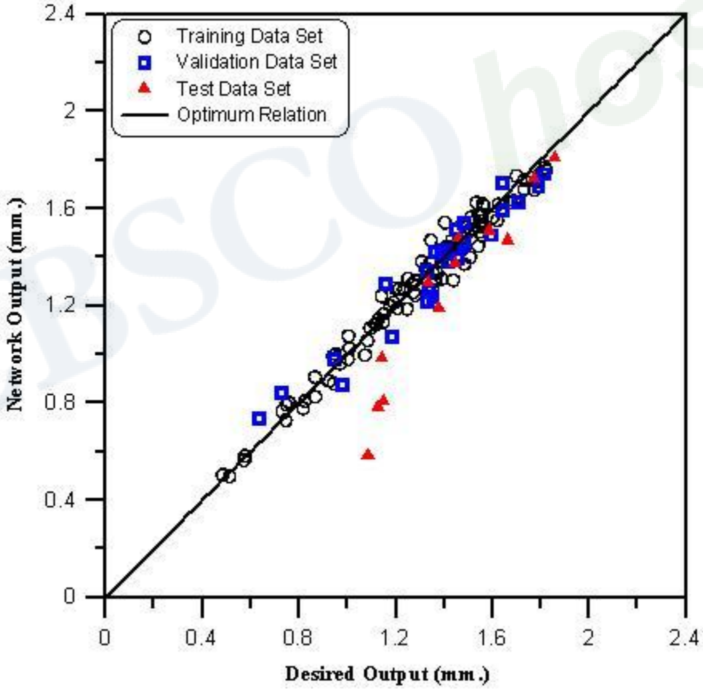


Figure 6. Relationship between actual (desired) and predicted (output) values of the best obtained network for predicting delamination size at drill entrance.

Figures 6 and 7 illustrate the relationships between the actual (desired) and predicted (network output) values of the best obtained neural network structures for predicting delamination at hole entry and exit respectively.

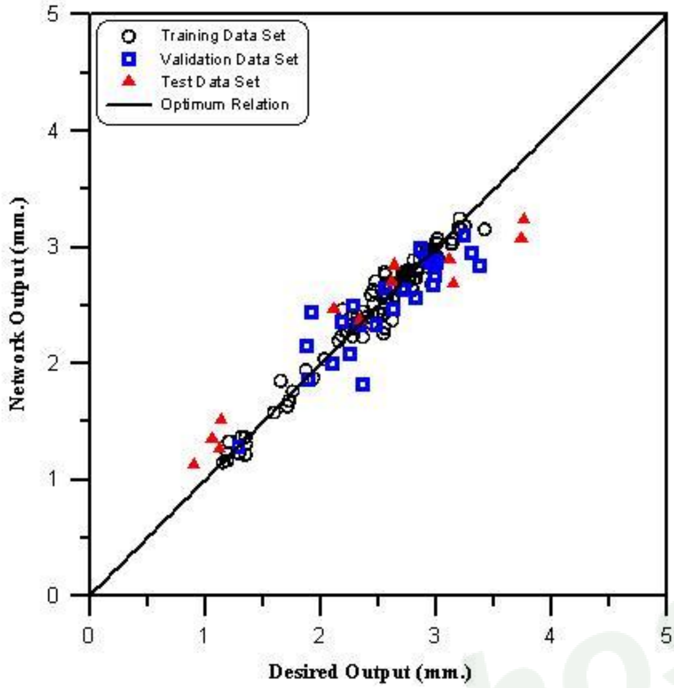


Figure 7. Relationship between actual (desired) and predicted (output) values of the best obtained network for predicting delamination at drill exit.

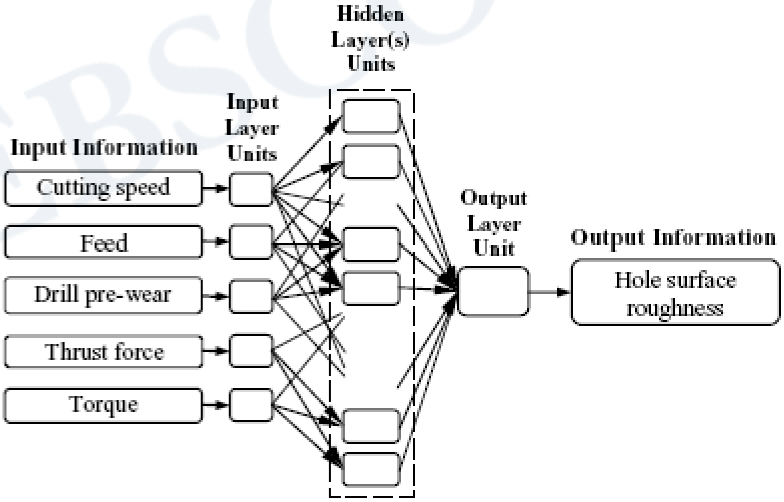


Figure 8. Schematic diagram of the neural network for predicting surface roughness.

4.3. Anns for Surface Roughness Prediction

Figure 8 shows the schematic diagram of the neural net for predicting the surface roughness, assessed by the roughness average height parameter (R_a). Five inputs were fed to the network (spindle speed, feed, drill pre-wear, thrust force, and torque).

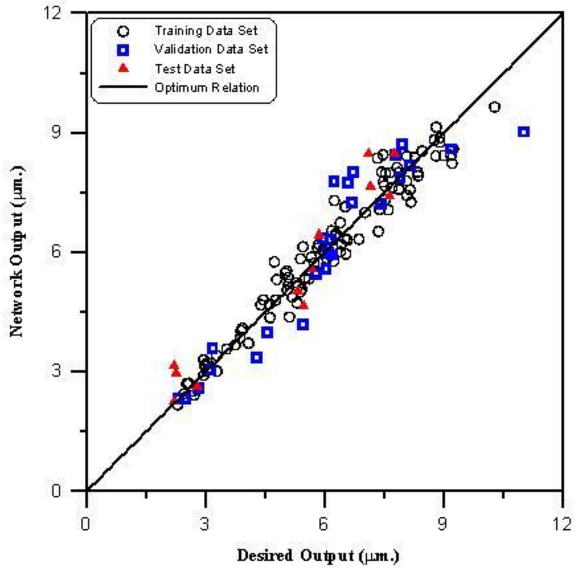


Figure 9. Relationship between actual (desired) and predicted (output) values of the best obtained network for predicting R_a parameter.

Table 5. Trials for the ANNs for predicting R_a parameter

No.	Network Structure	Training Set MSE (μm^2)	Validation Set MSE (μm^2)	Test Set MSE (μm^2)
1	5-3-1	0.6027	1.1581	0.6173
2	5-5-1	0.2692	0.7582	0.8672
3	5-8-1	0.1741	0.6162	1.3545
4	5-11-1	0.1219	0.6416	3.6262
5	5-3-3-1	0.4025	0.4805	0.6490
6	5-3-7-1	0.3779	0.5408	0.5695
7	5-7-3-1*	0.1766	0.5908	0.4381
8	5-7-7-1	0.1606	0.4659	0.5102
9	5-7-11-1	0.1021	0.6813	0.7555
10	5-11-7-1	0.0530	0.4885	0.7461
11	5-11-11-1	0.0790	0.4010	1.3869

* The best network structure

Table 6. Correlation coefficients for training, validation and test data sets

Correlation coefficients			
Best Networks	Training Data Set	Validation Data Set	Test Data Set
Thrust force (3-5-1)	0.999	0.999	0.992
Torque (3-3-1)	0.963	0.985	0.919
Drill Entry (5-11-1)	0.987	0.966	0.956
Drill Exit (5-7-3-1)	0.980	0.883	0.967
R_a (5-7-3-1)	0.976	0.936	0.962

To obtain the best neural network structure, twenty-seven neural networks were developed by changing both of the number of hidden layers and the number of hidden units within each hidden layer. Table 5 represents the tried networks for predicting roughness parameter R_a. From the Table, the best network structure, among the performed trials, for predicting R_a is 5-7-3-1 (with a MSE of 0.4381 μm^2).

Figure 9 illustrates the relationship between the actual (desired) and predicted (network output) values of the best obtained neural network structure for predicting R_a.

In order to investigate the performance of the best obtained networks, correlation coefficients between the targets and the corresponding network outputs (actual-predicted relationship) were estimated. Correlation coefficient is a measure of how well the variation in the outputs is explained by the targets, if the number is equal to one, it means that there is a perfect correlation between the targets and the ANNs outputs.

Table 6 shows the values of the correlation coefficients for each of training, validation and test data sets for the developed ANNs. From this Table, it is clear that the values of the correlation coefficients approach to one which give strong indication that the obtained ANNs can be used effectively to model and predict thrust force, torque and hole quality represented by delamination size (at drill entrance and exit sides) and surface roughness of hole wall (assessed by the roughness parameter R_a) resulting from drilling GFRE specimens.

4.4. Confirmation Test

The cutting conditions used in the confirmation tests are shown in Table 7. Tables 8, 9 and 10 indicate the comparison between the experimental values of the machinability parameters (thrust force, torque, peel-up and push-out delamination and surface roughness) and the predicted values using ANNs developed in this study. The results in these tables show that ANNs predict the machinability parameters with acceptable errors for the most confirmation tests.

Table 7. Cutting conditions in confirmation tests

Test No.	Cutting conditions		
	Speed m/min	Feed Mm/rev.	Wear gx10^{-4}
1	10.05	0.08	0
2	15.834	0.16	0
3	40.212	0.08	0
4	25.133	0.16	0
5	15.834	0.16	7
6	40.212	0.9	7
7	7.917	0.315	19
8	20.253	0.16	19
9	10.053	0.08	26
10	25.133	0.16	26
11	40.212	0.315	34
12	25.133	0.22	34

Table 8. Experimental confirmation of thrust force and torque and comparison with ANNs

Test No.	Thrust force			Torque		
	F_t (N) Exp.	F_t (N) ANNs	Error %	T (N.m) Exp.	T (N.m) ANNs	Error %
1	34.11	52.20	53.01	0.216	0.319	47.76
2	59.88	66.73	11.45	0.444	0.471	5.99
3	28.16	51.32	82.23	0.462	0.255	44.77
4	61.56	68.44	11.17	0.525	0.393	25.23
5	254.73	237.59	6.73	0.442	0.518	17.34
6	637.78	665.34	4.32	0.909	1.011	11.23
7	489.36	426.92	12.76	0.877	0.882	0.59
8	386.67	355.44	8.08	0.581	0.585	0.64
9	273.83	242.61	11.40	0.455	0.474	4.20
10	438.24	419.95	4.17	0.685	0.647	5.54
11	795.09	862.93	8.53	0.794	0.939	18.23
12	606.44	622.18	2.60	0.760	0.840	10.62

Table 9. Experimental confirmation of delamination and comparison with ANNs

Test No.	Delamination					
	Peel-up (mm) Exp.	Peel-up (mm) ANNs	Error %	Push-out (mm) Exp.	Push-out (mm) ANNs	Error %
1	1.125	0.782	30.42	1.121	1.265	12.93
2	1.143	0.985	13.77	1.137	1.516	33.36
3	1.085	0.583	46.29	0.903	1.131	25.27
4	1.148	0.808	29.66	1.059	1.348	27.26
5	1.461	1.480	1.30	2.338	2.372	1.49
6	1.858	1.811	2.50	3.149	2.685	14.72
7	1.772	1.726	2.54	2.634	2.847	8.11
8	1.336	1.295	3.00	3.116	2.899	6.94
9	1.377	1.189	13.59	2.111	2.467	16.91
10	1.446	1.377	4.72	2.615	2.704	3.42
11	1.664	1.471	11.60	3.756	3.244	13.63
12	1.585	1.510	4.75	3.735	3.074	17.70

Table 10. Experimental confirmation of surface roughness and comparison with ANNs

Test No.	Surface roughness		
	R_a (μm) Exp.	R_a (μm) ANNs	Error %
1	2.21	2.259	2.20
2	2.79	2.637	5.50
3	2.26	2.957	30.83
4	2.21	3.168	43.34
5	5.33	5.038	5.49
6	5.68	5.599	1.42
7	5.48	4.660	14.96
8	5.85	6.440	10.08
9	7.76	8.479	9.27
10	7.11	8.495	19.48
11	7.64	7.433	2.70
12	7.16	7.648	6.81

CONCLUSIONS

In this work the artificial neural networks technique, with back-propagation training routine was developed for prediction the machinability parameters in drilling GFRE composites. Several attempts were made to obtain the best structure of these networks. The results obtained lead to the following conclusions;

- The best obtained network structure for predicting thrust force, torque, peel-up delamination, push-out delamination, surface roughness were 3-5-1, 3-3-1, 5-11-1, 5-7-3-1, and 5-7-3-1, respectively.
- As the correlation coefficients for each of training, validation and test data sets for the best developed networks approach to one, therefore the ANNs predict the machinability parameters (thrust force, torque, peel-up and push-out delaminations and surface roughness) with acceptable errors for the most confirmation tests.

REFERENCES

[1] Zhang, Z. & Friedrich, K. (2003). Artificial neural networks applied to polymer composites: A review. *Composites Science and Technology*, *63*, 2029–2044.

[2] Stone, R. & Krishnamurthy, K. (1996). A neural network thrust force controller to minimize delamination during drilling of Graphite-Epoxy laminates. International *Journal of Machine Tools & Manufacture*, *36(9)*, 985-1003.

[3] Enemuoh, U. E., El-Gizawy, A. S. & Okafor, C. A. (1999). Neural network based sensor fusion for on-line prediction of delamination and surface roughness in drilling AS4/PEEK composites. *T NAMRI of SME*, *27*, 263–268.

[4] Chakraborty, D. (2005). Artificial neural network based delamination prediction in laminated composites. *Materials and Design*, *26*, 1–7.

[5] Sardiñas, R. Q. & Pedro Reis, Davim, J. P. (2006). Multi-objective optimization of cutting parameters for drilling laminate composite materials by using genetic algorithms. *Composites Science and Technology*, *66*, 3083–3088.

[6] Srinivasa Rao, B., Rudramoorthy, R., Srinivas, S. & Nageswara Rao, B. (2008). Effect of drilling induced damage on notched tensile and pin bearing strengths of woven GFR-Epoxy composites. *Materials Science and Engineering A*, *472*, 347–352.

[7] Karnik, S. R., Gaitonde, V. N., Campos Rubio, J., Esteves Correia, A., Abrão A. M. & Paulo Davim, J. (2008). Delamination analysis in high speed drilling of carbon fiber reinforced plastics (CFRP) using artificial neural network model. *Materials and Design*, *29*, 1768–1776.

[8] El-sonbaty, I. A., Khashaba, U. A., Selmy, A. I. & Ali, A. I. (2008). Prediction of Surface Roughness Profiles for Milled Surfaces Using an Artificial Neural Network and Fractal Geometry Approach. *Journal of Materials Processing Technology*, *200*, 271-278.

[9] Karayel, D. (2009). Prediction and control of surface roughness in CNC lathe using artificial neural network. *Journal of Materials Processing Technology*, *209*, 3125-3137.

[10] Behnam, B. & Suman, S. (1998). Mapping the cutting force to the hole quality using neural networks. *Intelligent Engineering Systems Through Artificial Neural Networks*, *8*, 773-778.

[11] Mathews, P. G. & Shunmugam, M. S. (1999). Neural-network approach for predicting hole quality in reaming. *International Journal of Machine Tools & Manufacture*, *39(5)*, 723-730.

[12] Tsao, S. S. & Hocheng, H. (2008). Evaluation of thrust force and surface roughness in drilling composite material using Taguchi analysis and neural network. *Journal of Materials Processing Technology*, *203*, 342–348.

[13] Abu-Mahfouz, I. (2003). Drilling wear detection and classification using vibration signals and artificial neural network", *International Journal of Machine Tools and Manufacture*, *43*, 707–720.

[14] Hayajneh, M. T., Hassan, A. M. & Mayyas, A. T. (2009). Artificial neural network modeling of the drilling process of self-lubricated aluminum/alumina/graphite hybrid composites synthesized by powder metallurgy technique. *Journal of Alloys and Compounds*, *478*, 559–565

[15] Swingler, K. (1996). *Applying neural networks: A practical guide*. London: Academic Press Inc.

[16] Haykin, S. (1999). *Neural networks: A comprehensive foundation: Second Edition.* USA: Prentice-Hall Inc.

[17] Hertz, J., Krogh, A. & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. USA: Addison-Wesley Publishing Company Inc.

[18] *Neurosolutions software*, version 5, http://www.nd.com/, 2006.

[19] Khashaba, U. A. (2003). Fracture behavior of woven composites containing various cracks geometry. *J Comp Mater*, 37-1, 5–21.

[20] Khashaba, U. A. (2004). Delamination in drilling GFR-thermoset composites, *Journal of Composite Structures*, *63(3–4)*, 313-327.

[21] Khashaba, U. A., Seif, M. A. & Elhamid, M. A. (2007). Drilling analysis of chopped composites, Composites Part A: *Applied Science and Manufacturing*, *38*, 61–70.

[22] Khashaba, U. A., El-Sonbaty, I. A., Selmy, A. I. & Megahed, A. A. (2010). Machinability Analysis in Drilling Woven GFR/Epoxy Composites:Part I- Effect of Machining Parameters. *Composites*: Part A, 41, 391-400.

[23] Durão, L. M. P., Gonçalves, D. J. S., Tavares, J. M. R. S., de Albuquerque, V. H. C., Vieira, A. A. & Marques, A. T. (2010). Drilling tool geometry evaluation for reinforced composite laminates, *Composite Structures*, 92-7, 1545-1550.

[24] Khashaba, U. A., El-Sonbaty, I. A., Selmy, A. I. & Megahed, A. A. (2010). Effect of Drill Wear on Cutting Forces and Hole Quality in Drilling GFRE Composites. Submitted for publication in Composites Part A: *Applied Science and Manufacturing*.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 5

PERFORMANCE ANALYSIS OF LIQUID DESIC CANT DEHUMIDIFICATION SYSTEM USING ARTIFICIAL NEURAL NETWORKS

P. Gandhidasan and M. A. Mohandes
King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

ABSTRACT

The heart of the liquid desiccant cooling system is the dehumidification process which is influenced by many parameters. Different types of dehumidification equipment have been developed and a variety of analytical models have been employed to analyze the dehumidification process. The dehumidification process involves simultaneous heat and mass transfer and reliable transfer coefficients are required in order to analyze the system. This has been proved to be difficult and many assumptions are made to simplify the analysis.

Artificial Neural Network (ANN) is widely used as an innovative way to tackle complex and ill-defined problems. The present research proposes the use of ANN based model in order to simulate the relationship between inlet parameters and the performance of the dehumidifier. For the analysis, randomly packed dehumidifier is chosen since the packed tower facilitates high mass transfer by providing a large surface area in a relatively small volume. Lithium chloride is selected as the liquid desiccant due to its stability with high performance.

A multilayer ANN is used to investigate the performance of dehumidifier. For training ANN models, data is obtained from analytical equations. The training process implies adjustment of connection weights and biases so that the differences between ANN output and the desired output are minimized. Eight parameters are used as inputs to the ANN, namely: air and desiccant flow rates, air and desiccant inlet temperatures, air inlet humidity, the desiccant inlet concentration, dimensionless temperature ratio, and the inlet temperature of the cooling water. The output of the ANN is the water condensation rate. The predicted water condensation rate by the ANN is validated with experimental data and the value of R^2 is found to be 0.9251. Results and the performance of the developed system are presented in this chapter.

1. INTRODUCTION

Air dehumidification is not only an important industrial process of water vapor removal from moist air but also in comfort cooling of buildings and this can reduce the electric power consumption and/or eliminate the chlorofluorocarbons (CFC's). Dehumidified air is required for many industrial applications with particular reference to chemical, metallurgical, combustion and air conditioning industries. It is used for drying air in air handling systems in commercial and domestic buildings. In a desiccant air conditioning system, moisture is removed from the air by bringing it in contact with the desiccant material and followed with sensible cooling of air by a conventional vapor compression or vapor absorption cooling system or evaporative cooling system [1].

In storage applications, the space has to be maintained at a specific humidity and temperature, so that the products stored do not absorb moisture from the surroundings. Hygroscopic materials such as sugar, flour, plastics and other types of synthetic materials and chemicals are often stored in humidity controlled areas, thus keeping them ready for use and handling. It is also used in controlled atmosphere storage for hygroscopic products such as candies, dried fruits or vegetables, baked goods, controlling the ripening of fruit or drying of meat and fish and drying of crops. In the electronics industry, moisture can cause a multitude of problems ranging from pitting of electrical contacts due to excessive arcing under high humidity conditions to the bursting of casings of electrical heating elements. Food stuffs such as potato chips, instant coffees, soda crackers and other hygroscopic materials require only a very small amount of water vapor to cause them to congeal or become rancid and stale after a very short period of time.

In pharmaceutical industry, dry air is required for storing raw materials and chemicals to prevent contamination, in addition to processing and packaging of the finished products. Other special applications are for wind tunnels and altitude chambers where fogging occurs and interferes with observations and photography. In textile industry, the humidity affects the quality and quantity of all types of natural and synthetic fibres. The efficiency of a drying operation depends largely on humidity control. Drying is slow and inefficient under high humidity conditions. When air is used for combustion processes, the water content must be low for efficient operation so that it minimizes fuel consumption.

2. DEHUMIDIFICATION METHODS

Dehumidification of air can be accomplished by different methods [2]. It can be achieved by either cooling (refrigeration method) or increasing the pressure of air by compression process (mechanical method) or by adsorption/absorption of moisture by solid or liquid material, known as desiccants. In the refrigeration method, the moisture content of the air is reduced by lowering its temperature below the dew point temperature. Dehumidification can also be accomplished by using surface condensers or by using cold water sprays such that the moisture in the air is condensed. Air compression constitutes another method of reducing the moisture content of air. When air is compressed, the partial pressure of the water vapor in the water-air mixture is raised to the point where moisture can be condensed from the air at a higher temperature. This system is suitable for small volumes of air to be dehumidified. But the cost of the compressor, high power requirements and the cooling water requirement make this system impractical.

Desiccant materials have a high affinity for water vapor. A desiccant dehumidifier is a device that employs a desiccant material to produce a dehumidification effect. The process involves exposing the desiccant material to high relative humidity air stream, allowing it to attract and retain some of the water vapor. This process is followed by exposing the same desiccants to a lower relative humidity air stream which has the affect of drawing the retained moisture from the desiccant. The first air stream is the air that is being dehumidified while the second air stream is used only to regenerate the desiccant material. Two major categories of desiccants are adsorbents and absorbents. Adsorbent materials hold water molecules in pores at their surface, no chemical change results. Absorbents go through chemical change as they attract and retain water vapor. Absorbents generally can attract and hold greater quantities of water per kilogram of desiccant material.

In adsorption method, solid desiccants like silica gel, activated alumina, activated bauxite, microsieves, etc. which have great affinity for water are used for dehumidifying the air. This system is suitable in the field where low dew points are desired. As the air to be processed is passed through the desiccant bed, the moisture in the air is condensed out in the pores of the desiccant. During the process the latent heat of vaporization of the moisture condensed is converted to sensible heat thereby raising the temperature of the air. In conventional units, gas, steam or electrical energy are used to regenerate the adsorbent beds. Attempts have been made to use solar energy as the heat input in this system. Many space cooling systems using solid desiccants have been proposed [3, 4]. In these systems dehumidification of room air is achieved by using a desiccant bed and it is regenerated by solar heated air. The regeneration temperature for this system is in the range of 80 to 95°C.

3. Liquid Desiccant Dehumidification

Air may be dehumidified when sprayed with a suitable liquid. The aqueous solution of absorbent would seem to have the best prospects for the dehumidification process. If a solution of water with a substance such as calcium chloride, lithium chloride or bromide, glycols, sodium chloride, etc. is brought into contact with air either singly or in combination, the air attempts to assume the vapor pressure of the solution [5]. At a given temperature the solution of any of these substances has a lower vapor pressure than water itself. A spray of a hygroscopic solution will therefore be especially effective for dehumidification. The regeneration temperature required for liquid desiccant system is in the range of 50–65°C. Where only moderately low humidities are required and where large quantities of air are to be dehumidified, the liquid absorbents are ideally suitable.

Although many methods are available for dehumidification operation, liquid desiccant system is considered in the present study due to the following several design and performance advantages:

- Flexibility in operation and in its location.
- Readily circulated by means of a small pump.

- Absorbing inorganic and organic contaminants in the air and hence, improved indoor air quality.
- Ability of using low grade thermal energy due to the requirement of low regeneration temperature.
- Simultaneous cooling during dehumidification.
- Ability to use low-cost cooling tower water for removing sensible heat from the dehumidification process.
- Ability to store the energy in the form of chemical energy.
- Part of the weak desiccant solution can be over-concentrated and mixed with the remaining solution.
- Single regenerator for multiple conditioners.

Some of the demerits of liquid desiccant systems include problems associated with corrosion (except for glycols), carryover of solution into the air stream and crystallization of desiccant material. However, to reduce corrosiveness some inhibitors can be added to the solution. Among the various available liquid desiccant materials, lithium chloride is chosen for the current study since it has good desiccant characteristics and does not vaporize in air at ambient conditions. Lithium chloride can absorb up to 1,200 times their dry weight in water [6]. The authors developed an Artificial Neural Network (ANN) model to predict the vapor pressures of liquid desiccants and validated the model results for three different liquid desiccants namely: lithium chloride, calcium chloride, and lithium bromide with experimental data [7].

4. Dehumidification Equipment

In a liquid desiccant dehumidifier the air is dehumidified when exposed to cool and strong desiccant solutions. The dehumidification process can be accomplished in equipment such as a finned-tube surface in a column, spray tower, or packed tower.

4.1. Finned-Tube Surface

This type of dehumidifier was first proposed by Turner [8] in which the desiccant solution is sprayed over fins and air is blown through fin passages. Cooling water is circulated through the tubes. However, such equipment requires unreasonably high air velocity in certain cases. It is also difficult to control the liquid film on the fin. Therefore, Peng and Howell [9] proposed an alternative finned-tube surface arrangement. The interfacial surface area is assumed to be equal to the fin surface as long as the liquid flow occupies less than 20% of the volume. However, in the experimental studies there were many problems because of the flooding which is coupled with choking. The reason for this is the structure of the fins and the liquid desiccant (triethylene glycol). The liquid is sprayed over the tubes of the fin tube surfaces in a column where the air is blown upward through the column. The cooling water that circulates inside the fin tube keeps the operation isothermal. However, in some cases, extremely large air flow rates are required, and it is difficult to control the liquid film over the fin tubes.

4.2. Spray Tower

In this arrangement large surface area for heat and mass transfer is obtained by breaking the liquid into small droplets with the help of nozzles [10]. This system has many advantages such as low cost, low pressure drop on air side, compact size, and simple operation. However, this system has some drawbacks that include large pressure drop on liquid side and chances of liquid carryover.

4.3 Packed Tower

Packed tower configuration has received more attention because of a large rate of heat and mass transfer per unit volume. Other advantages include compactness, high efficiency and large contact time. In a counter-flow packed tower dehumidifier, a strong and cool desiccant solution is distributed from the top and allowed to trickle down through the tower in a thin film covering the packing material surfaces as shown in Figure 1. The humid air enters the dehumidifier at the bottom. The moisture content of the air in the unit is controlled by adjusting the temperature and the concentration of the desiccant. Since the desiccant solution vapor pressure is less than the water vapor pressure in the air, the moisture transfer takes place from the air to the desiccant solution. The dehumidified air exits at the top. A warm and diluted (weak) solution leaves at the bottom of the dehumidifier. Two types of packing are generally used viz. random packing and regular (structured) packing. Although regular packing such as cellulose rigid media pads, wood grids, and expanded metal lash packing, etc. offers low pressure drop for the air stream at the expense of costly installation, random packing such as Berl saddles, Raschig rings and Intalox saddles, etc. provide good contact between air and the desiccant solution. Hence, a random packing is chosen for the present study. The performance of the dehumidifier is measured by the amount of water condensed from the humid air.

The dehumidification process involves simultaneous heat and mass transfer and reliable transfer coefficients are required in order to analyze the dehumidifier. This has been proved to be difficult and many assumptions are made to simplify the analysis. For this purpose different theoretical models have been developed to study the performance of the dehumidifier and this is out of the scope of the present work. Due to the high number of variables involved in the process, the analysis becomes increasingly complex. The objective of the current study is to examine the possibilities of using ANN to predict the performance of the random packed dehumidifier. ANN is widely used as an innovative way to overcome the limitations of physical modeling of complex and ill-defined problems.

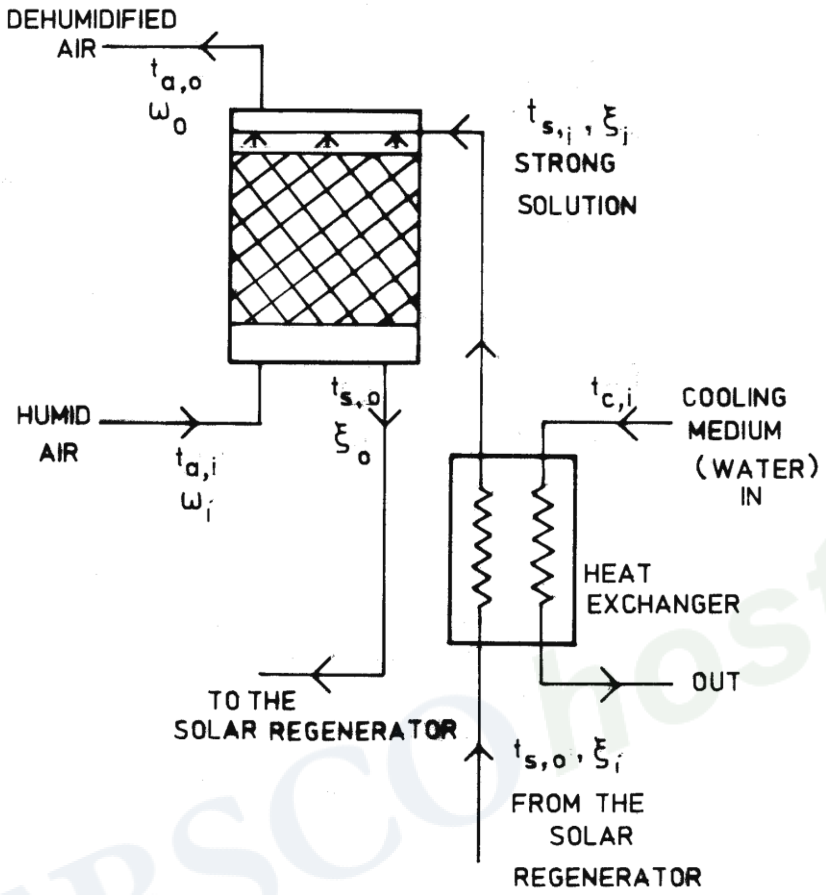


Figure 1. Schematic of air dehumidification process with liquid desiccant.

5. Thermodynamic Analysis of the Dehumidifier

The assumptions and the concept of the dehumidifier performance analysis are adapted from [5]. Referring to Figure 1, air at humidity ratio, $\omega_{a,i}$, and temperature, $T_{a,i}$, enters the bottom of the dehumidifier and leaves at the top with humidity ratio, $\omega_{a,o}$, and temperature, $T_{a,o}$. The desiccant solution enters at the top of the dehumidifier with temperature, $T_{s,i}$, and concentration, $\xi_{s,i}$, and leaves at the bottom with temperature, $T_{s,o}$, and concentration, $\xi_{s,o}$. The mass flow rate of air and the desiccant solution per unit cross-sectional area of the dehumidifier is defined as G_a and G_s, respectively.

The dimensionless temperature difference ratio can be defined as,

$$\beta = \frac{T_{a,i} - T_{a,o}}{T_{a,i} - T_{s,i}} \tag{1}$$

In order to obtain the desiccant solution temperature at the inlet of the dehumidifier in terms of the solution temperature at the outlet, the effectiveness of the heat exchanger is defined as,

$$\varepsilon = \frac{T_{s,o} - T_{s,i}}{T_{s,o} - T_{c,i}} \tag{2}$$

The effectiveness of the heat exchanger is assumed as constant at 0.6.

Water condensation rate is given by,

$$M = \frac{1}{\lambda}\left[\frac{G_s C_s \varepsilon}{(1-\varepsilon)}\left(T_{s,i} - T_{c,i}\right) - G_a C_a \beta\left(T_{a,i} - T_{s,i}\right)\right] \tag{3}$$

The humidity ratio of air at the outlet of the dehumidifier is given by,

$$\omega_{a,o} = 0.622\left(\frac{p_{a,o}}{P - p_{a,o}}\right) \tag{4}$$

where

$$p_{a,o} = p_{a,i} - \left(\frac{M\lambda}{G_a Y}\right) \tag{5}$$

$$Y = \frac{0.622\,\lambda}{P} \tag{6}$$

$$p_{a,i} = \frac{\frac{P\omega_{a,i}}{0.622}}{1 + \frac{\omega_{a,i}}{0.622}} \tag{7}$$

The desiccant temperature at the outlet of the dehumidifier is given by,

$$T_{s,o} = \frac{T_{s,i} - \varepsilon T_{c,i}}{(1-\varepsilon)} \tag{8}$$

The concentration of the desiccant at the outlet of the dehumidifier can be found as

$$\xi_{s,o} = \frac{\xi_{s,i}}{1 + \frac{M}{G_s}} \tag{9}$$

The above analysis shows the dependence of the dehumidification process on operational parameters such as air and desiccant flow rates, the desiccant concentration and its temperature as well as the air humidity ratio and its temperature at the inlet of the dehumidifier. The latent heat of condensation and the specific heat of the fluids are assumed to be constants. However, the cooling water inlet temperature to the heat exchanger is considered as one of the variables in the present study. Further, the performance of soft computing methodology, trained ANN based on multilayer algorithm is used for the performance analysis of the dehumidifier.

6. Artificial Neural Network Model

The current interest in artificial neural networks (ANNs) is largely due to their ability to mimic natural intelligence in its learning from experience [11]. They learn from examples by constructing an input-output mapping without explicit derivation of the model equation. ANNs have been used in a broad range of applications including: pattern classification [12, 13], function approximation, optimization, prediction and automatic control [14] and many others.

An artificial neural network consists of many interconnected identical simple processing units called neurons. Each connection to a neuron has an adjustable weight factor associated with it. Every neuron in the network sums its weighted inputs to produce an internal activity level a_i,

$$a_i = \sum_{j=1}^{n} w_{ij} x_{ij} - w_{io} \tag{10}$$

where w_{ij} is the weight of the connection from input j to neuron i, x_{ij} is input signal number j to neuron i, and w_{io} is the threshold associated with unit i. The threshold is treated as a normal weight with the input clamped at -1. The internal activity is passed through a nonlinear function φ to produce the output of the neuron y_i,

$$y_i = \varphi(a_i) \tag{11}$$

Several forms of differentiable activation functions have been used with the most popular being the logistic function of the form:

$$\varphi(a_i) = \frac{1}{1 + \exp(-a_i)} \tag{12}$$

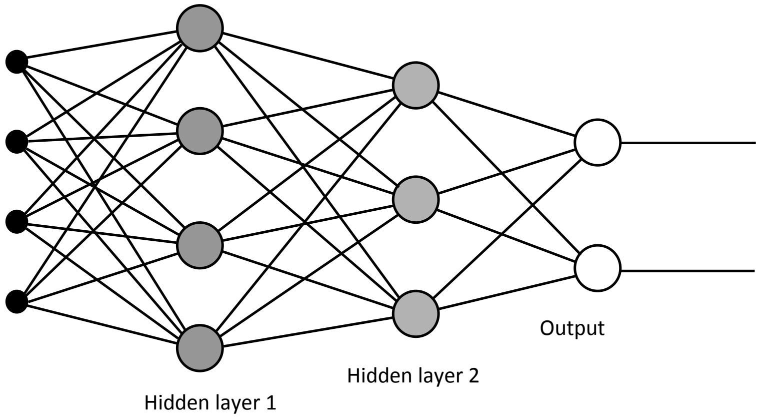


Figure 2. Structure of artificial neural network.

The weights of the connections are adjusted during the training process to achieve the desired input/output relation of the network.

Artificial neural networks come in many different paradigms, some require topologies with total interconnection among neurons and others require arrangement in layers. A multilayer feedforward network has its neurons organized into layers with no feedback or lateral connections. Layers of neurons other than the output layer are called hidden layers. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. Figure 2 shows a three-layer feedforward network.

The backpropagation algorithm [15] is a supervised iterative training method for multilayer feedforward nets with sigmoidal nonlinear threshold units. It uses training data consisting of *S* input-output pairs of vectors that characterizes the problem. Using a generalized least-mean-square algorithm the backpropagation algorithm minimizes the mean square difference between the real network output and the desired output [16]. The error function that the backpropagation algorithm minimizes is the average of the square difference between the output of each neuron in the output layer and the desired output. The error function can be expressed as:

$$E = \frac{1}{S} \sum_{s} \sum_{k} \left(d_{sk} - o_{sk} \right)^2 \tag{13}$$

where *s* is the index of the *S* training pair of vectors, *k* is the index of elements in the output vector, d_{sk} is the *k*th element of the *s*th desired pattern vector, and o_{sk} is the *k*th element of the output vector when pattern *s* is presented as input to the network.

Minimizing the cost function represented in Eq. (13) results in an updating rule to adjust the weights of the connections between neurons. The weight adjustment of the connection between neuron *i* in layer *m* and neuron *j* in layer *m* + 1 can be expressed as:

$$\Delta w_{ji} = \eta \delta_j o_i \tag{14}$$

where i is the index of units in layer m, η is the learning rate, o_i is the output of unit i in the mth layer, and δ_j is the delta error term back-propagated from the jth unit in layer $m + 1$ defined by:

$\delta_j = \left[d_j - o_j\right] o_j \left[1 - o_j\right],$ neuron j is in the output layer

$\delta_j = y_j \left[1 - y_j\right] \sum_k \delta_k w_{kj},$ neuron j is in a hidden layer and k is index of neurons in the layer ($m+2$), ahead of the layer of neuron j.

Choosing a small learning rate η leads to slow rate of convergence, however, too large η leads to oscillation. A simple method for increasing the rate of learning without oscillation is to include a momentum term as:

$$\Delta w_{ji}(n+1) + \eta \delta_j o_i + \alpha \Delta w_{ji}(n) \tag{15}$$

where n is the iteration number, and α is a positive constant which determines the effect of past weight changes on the current direction of movement in weight space. Detailed description of the multilayer feedforward neural networks and the backpropagation algorithm may be found in [16].

7. Results and Discussions

The variables that are found to have the most significant effect on the dehumidifier operation are: flow rates of fluids, inlet temperature of the fluids, inlet desiccant concentration, dimensionless temperature ratio, air inlet humidity ratio, and cooling water inlet temperature. The ranges of these operating variables that are used in generating the data for training are given in Table 1. Equations (1) to (9) have been used to generate about 1200 data points to be used for training the neural network model. The simulated data is used to train a feedforward neural network with 8 inputs and one output to predict the water condensation rate. After several trials, a network with 8 inputs, 10 hidden units in one layer and one output is selected. A maximum of 100 epochs were allowed, alternatively, training could stop when the Mean Square Error (MSE) for the difference between the desired output and the obtained results reaches a pre-specified limit which is on this case set to be 0.0001. When network training was successfully completed, the network was tested on a reliable set of experimental data from the literature [17] and the results are given in Table 2. It is to be noted that the experimental data set was not used during the training process. The comparison between the ANN simulated results and the experimental data are also shown in Figures 3 and 4. The R^2 value is found to be 0.9251 which indicates a reasonable agreement between the measured testing data and the neural network simulated results. As seen from the results obtained, the water condensation rate to the desiccant solution is obviously predicted within acceptable ranges.

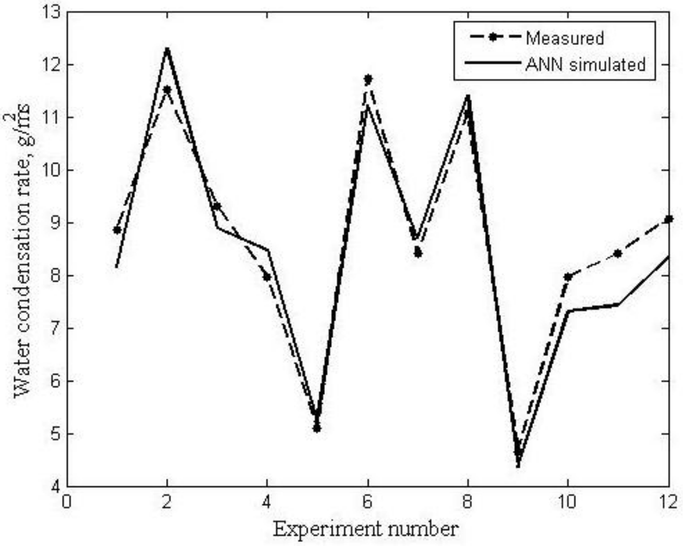


Figure 3. Performance of the dehumidifier as compared to experimental data.

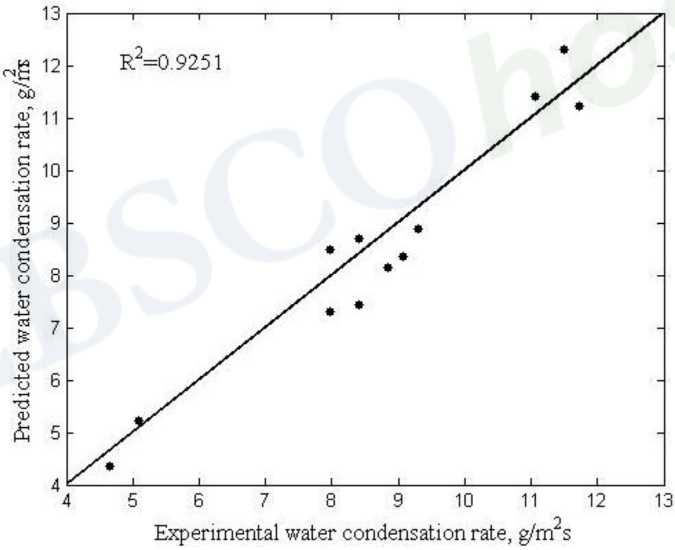


Figure 4. The ANN prediction for the water condensation rate.

Table 1. Operating parameters and their ranges used for generating data

S. No.	Operating parameters	Symbol	Unit	Operating range
1	Air inlet temperature	$T_{a,i}$	ºC	28 to 41
2	Air inlet humidity ratio	$\omega_{a,i}$	kg w/kg da	0.014 to 0.022
3	Air flow rate	G_a	kg/m^{2}s	0.85 to 1.5
4	Desiccant inlet temperature	$T_{s,i}$	ºC	25 to 36
5	Desiccant inlet concentration	$\xi_{s,i}$	% by weight	33 to 40
6	Desiccant flow rate	G_s	kg/m^{2}s	3.5 to 6.5
7	Temperature difference ratio	β	-	-10 to 26
8	Cooling water inlet temperature	$T_{c,i}$	ºC	23 to 35.2

Table 2. Comparison of trained ANN results with the experimental data

G_a	$T_{a,i}$	$\omega_{a,i}$	G_s	$T_{s,i}$	$\xi_{s,i}$	$T_{c,i}$	β	M, g/m^2s	
								ANN	Expt.
1.180	30.1	0.0181	6.227	30.3	34.7	29.6	10.50	8.14	8.85
1.513	30.2	0.0181	6.113	30.0	34.3	29.1	-10.0	12.29	11.5
1.189	35.5	0.0188	6.290	30.3	34.5	29.3	0.52	8.89	9.29
1.183	40.1	0.0180	6.287	30.5	34.4	29.5	0.73	8.47	7.97
1.214	30.3	0.0142	6.273	30.1	33.9	29.7	-4.0	5.23	5.09
1.187	29.9	0.0215	6.272	30.3	33.9	29.4	8.75	11.21	11.73
1.190	30.1	0.0180	5.019	30.2	34.4	29.4	21.0	8.69	8.4
1.198	29.9	0.0177	6.269	25.0	34.7	23.9	0.35	11.40	11.06
1.176	29.9	0.0178	6.309	35.2	34.9	35.1	1.10	4.36	4.65
1.182	29.9	0.0179	6.164	30.1	33.1	29.5	12.5	7.30	7.96
1.192	29.9	0.0179	6.267	30.2	33.8	29.6	8.67	7.43	8.41
1.176	30.0	0.0181	6.206	30.2	34.8	29.5	10.0	8.35	9.07

CONCLUSIONS

Air dehumidification is crucial for industrial applications as well as for air conditioning of all types of buildings in humid climates. This leads to a reduction in power consumption and the elimination or minimization of the CFC's. Water condensation rate is the most important characteristic for the operation of the humidity reduction system. ANN approach is used in this paper to predict the water condensation rate in a liquid desiccant dehumidifier. The ANN model is implemented and its feasibility is established. Good agreement between the output from the ANN model and the corresponding results from the experimental data has been found. This study reveals that ANN model can work well as a predictive tool to complement the experiments with a high degree of accuracy. When more experimental data becomes available, the model can be further evaluated.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of King Fahd University of Petroleum & Minerals, Dhahran-31261, Saudi Arabia.

NOMENCLATURE

a_i	internal activity level of unit *i*
C	specific heat at constant pressure, kJ/kg K
d_{sk}	*k*th element of the *p*th desired pattern vector
E	error function

G	mass flux, kg/m^2 s
k	index of elements in the output vector
M	water condensation rate, g/m^2 s
o_i	output of unit *i* in the *m* th layer
o_{sk}	*k*th element of the output vector when pattern *s* is presented as input to the network
p	vapor pressure, mm Hg
P	total pressure, mm Hg
s	index of the training pair of vectors
S	total number of training vectors
T	temperature, ºC
w_{ij}	weight of the connection from input *j* to neuron *i*
w_{io}	threshold associated with unit *i*
x_{ij}	input signal number *j* to neuron *i*
y_i	output of the neuron

Greek

β	dimensionless temperature difference ratio
δ_j	delta error term backpropagated from the *j*th unit in layer *m+1*
ε	heat exchanger effectiveness
λ	latent heat of condensation, kJ/kg
ξ	concentration of the desiccant solution by weight, %
Δw_{ji}	weight adjustment of the connection between neuron *i* in layer *m* and neuron *j* in layer *m*+1
ω	air humidity ratio, kg water vapor/kg dry air
φ	nonlinear activation function
η	learning rate

Subscripts

a	air
c	cooling water
i	inlet
o	outlet
s	desiccant solution

REFERENCES

[1] Goswami, D. Y., Kreith, F. & Kreider, J. F. (2000). Principles of Solar Engineering, 2nd Ed., Taylor & Francis, Philadelphia, PA.

[2] Jain, S. & Bansal, P. K. (2007). Performance analysis of liquid desiccant dehumidification systems, *International Journal of Refrigeration*, *30*, 861-872.

[3] Pesaran, A. A., Penney, T. R. & Czanderna, A. W. (1992). Desiccant Cooling: State-of-the-Art Assessment, National Renewable Energy Laboratory, Golden, CO, October.

[4] Dhar, P. L. & Singh, S. K. (2001). Studies on solid desiccant based hybrid air-conditioning systems, *Applied Thermal Engineering*, *21*, 119-134.

[5] Gandhidasan, P. (2004). A simplified, model for air dehumidification with liquid desiccant, *Solar Energy*, *76*, 409-416.

[6] Harriman, L. (1996). *Applications Engineering Manual for Desiccant Systems,* American Gas Cooling Center, Arlington, VA.

[7] Gandhidasan, P. & Mohandes, M. A. (2008). Predictions of vapor pressures of aqueous desiccants for cooling applications by using artificial neural networks, *Applied Thermal Engineering*, *28*, 126-135.

[8] Turner, N. C. (1979*). U.S. patent No. 4,171*,620, October 23.

[9] Peng, C. S. & Howell, J. R. (1981). Analysis and design of efficient absorbers for low temperature desiccant air conditioning, *Trans. of the ASME Journal of Solar Energy Engineering*, *103*, 67-74.

[10] Chung, T. W. & Wu, H. (1998). Dehumidification of air by aqueous triethylene glycol solution in a spray tower, *Separation Science and Technology*, *33*, 1213-1224.

[11] Wasserman, P. (1993). *Advanced methods in neural computing*. Van Nostrand Reinhold, New York.

[12] Lippmann, R. P. (1987). *An introduction to computing with neural nets*. IEEE ASSP Mag., 4-22.

[13] Bishop, C. M. (1996). *Neural networks for pattern recognition*. Oxford University Press.

[14] Pham, D. T. & Liu, X. (1995). *Neural networks for identification, predication and control* Springer-Verlag, London.

[15] Rumelhart, D. E. & McClelland, J. L. (1986). PDP Research Group, *Parallel Distributed Processing*, The MIT Press.

[16] Haykin, S. (1994). *Neural networks: A comprehensive foundation*. Macmillan College Publishing.

[17] Fumo, N. & Goswami, D. Y. (2002). Study of an aqueous lithium chloride desiccant system: air dehumidification and desiccant regeneration, *Solar Energy*, *72*, 351-361.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 6

APPLICATION OF THE ARTIFICIAL NEURAL NETWORKS ON VISUALLY GUIDED ROBOT

Emregul Ersan[1,*]***, Vedat Topuz,***[†2] ***and Ayca Gokhan Ak***[2,‡]
[1]Zeytinburnu Industrial Vocational School Zeytinburnu, Istanbul/Turkey
[2] Vocational School of Technical Sciences Marmara University,
Goztepe, Istanbul/Turkey,

ABSTRACT

In this chapter, the carrying of an object at a workspace, which was perceived by vision, to another location was realized by a robot arm with five axes. Basic image process techniques were used for object recognition and position determination. If the desired object was inside the workspace, the inverse kinematics solution was realized, and then after coordinates of the object's location was sent to the robot arm. The inverse kinematics solution of the robot arm was performed with Artificial Neural Networks (ANN) model (Multi Layer Perceptron-MLP and Radial Basis Function (RBF) Neural Network) based on the forward kinematics solution. For an inverse kinematics solution of the robot, the training data set was created in the ANN method by using the robot's forward kinematics values first and then, ANN modeling was realized. After the robot's inverse kinematics solution was realized, the determined joint angle values were directed to the robot arm and moving the object to the desired location was realized successfully. Experimental results presented in this chapter indicate that RBF is more efficient solution than MLP for inverse kinematic solution of visually guided robot.

[*] emregulersan@hotmail.com
[†] vtopuz@marmara.edu.tr
[‡] aycaak@marmara.edu.tr

1. INTRODUCTION

Large part of research in computer technology increasingly day to day to simplify the people's lives and to provide new opportunities is concerned in developing intelligent machines that can move itself without human intervention. Using in conjunction of image processing algorithms and intelligent machines such as robots has an important place among the technological developments of today. Most of these robotic manipulators require knowledge about their environment in order to perform their function. Indeed, typical robots have no eyes for tracking, detecting, or recognizing any objects. If we want a robot to catch objects in real time, it needs a camera to form a vision system as its eyes in order to measure the position of an object. The vision system provides the raw data corresponding to a coordinate frame, x and y, which appear on a CCD array. Therefore, the vision system must cooperate with the robot controllers. In addition, the robot requires a visual servo controller to assist in successfully catching a moving object. As a result, the incorporation of a vision system is an integral part of both the application of robotics and the robotics industry [1].

There are many studies for robotic image processing techniques in the literature. The mobile robot to catch a ball that had fallen had implemented by Mundhra et al. [2]. Mondi et al. have developed ping-pong playing robot arm. The coordinates of the ping-pong balls have determined with image processing techniques applied on real-time images taken from CCD video camera and the robot arm move is realized [3]. Electrical plug and its coordinates have been determined with a camera attached to the robot arm by using pattern recognition algorithms in Bustamante and Gu study. Then they have ensured recharging of the mobile itself going to designated areas and that making the necessary connections [4]. Knoeppel et al. have determined with two CMOS camera mounted on the car's rear window whether a vehicle following the car on max 150m away and between the distances from other vehicle [5]. A mobile robot has moved without hitting the wall through the sensors in a confined space such as the maze in study of Choi, Ryu and Kim. They have plotted the confined space with processing the image received from the camera with morphological image processing techniques [6].

Inverse kinematics modeling has been one of the main problems in robotics research. The most popular method for controlling robotic arms is still based on look-up tables that are usually designed in a manual manner. ANNs are alternative methods for inverse kinematics solution [7]. Various ANN approaches has been studied for inverse kinematics solution. The performance of the recurrent neural network and the dynamic neural processor was compared on inverse kinematic computations of a two-linked by robot Rao and Gupta [8]. Bingul et al. have applied an ANN using backpropagation algorithm to solve inverse kinematics problems of industrial robot manipulator [9]. Choi and Lawrence have applied MLP networks to the 3 degrees of freedom (DOF) spatial manipulator robot inverse kinematic problem [10]. Yang et al. have realized the comparison of the performance of RBF and back propagation ANN paradigms trained to learn data obtained from the kinematics model of a U.M.1 RTX robotic arm. [11]. Different neural network approaches have been studied for visual application on robots [12], [13].

In this chapter, the carrying of an object at the workspace, which was perceived by vision, to another location by a robot arm (EDUBOT) was realized. This process consists of four main stages such as object recognition, determination of the object's location, inverse kinematics solution of the robot arm and the robot arm movement. An interface was designed for image processing of the object's image and motion control of the robot arm.

Basic image processing techniques were used for object recognition and position determination. An image of the object was transferred to the system by the camera which was placed at the top of the workspace so that the robot arm would not hit it. After the captured image was taken through some image processing stages such as filter and calibration, it was determined that the object at workspace was the desired object. If the desired object was at the workspace, the inverse kinematics solution was realized, after coordinates of the object's location were sent to the robot arm. An interface was designed for image processing and motion control of the robot arm for this purpose.

The inverse kinematics solution of the robot arm was performed with ANN (MLP and RBFNN) based on the forward kinematics solution. For an inverse kinematics solution of the robot, the training data set was created in the ANN method by using the robot's forward kinematics values first and then, ANN modeling was realized.

After the robot's inverse kinematics solution was realized, the determined joint angle values were directed to the EDUBOT robot arm and then, moving the object to the desired location was realized successfully.

This chapter isarranged as follows: The next section visually guided robot arm is presented then, image processing technique, robot manipulator and ANN are introduced. ANN for calculation of the inverse kinematics is presented section 3. Section 4 contains the experimental results. Section 5 concludes the paper.

2. Visually Guided Robot

In visually guided robot system, object found in the study area, is identified with the help of image processing techniques. Than inverse kinematics solution of the robot is computed with ANNs. Finally, object is moved to the desired location by the five-axis robot. The block diagram of the system is given on Figure1.

2.1. Image Processing

Each point on images taken the projection from three-dimensional environment to two-dimensional images plane has a color value and coordinate. To create a digital image, the perceived continuous data should be converted to digital format. Digitization process includes two processes: sampling and quantization [14].

Resolution of digital image is determined as a result of sampling process (i.e.:1024x768, 640x480), the color depth of the image is determined as a result of quantization (color depth; 256 for 8 bit grey level, 65536 for 16 bit grey level). In addition, a matrix of the actual number is obtained as a result of sampling and quantization.

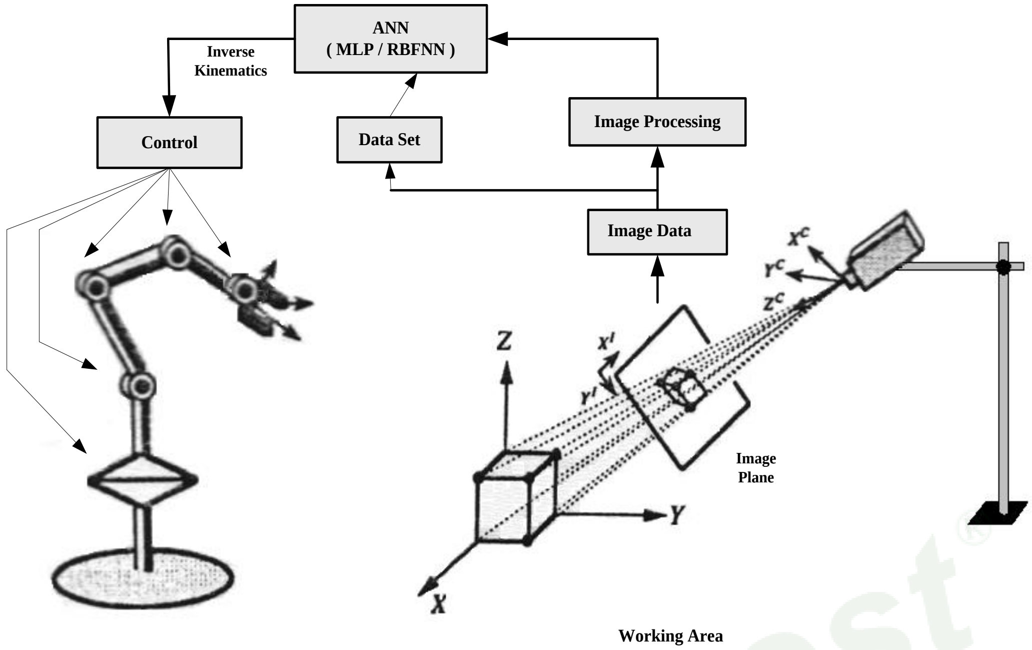


Figure 1. Block Diagram of the Visually Guided System.

MxN dimensional digital image of a digital *f(x,y)* image can be written in matrix form (1). Every element in the matrix array are called pixel.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdot & \cdot & f(0,N-1) \\ f(1,0) & f(1,1) & \cdot & \cdot & f(1,N-1) \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ f(M-1,0) & f(M-1,1) & \cdot & \cdot & f(M-1,N-1) \end{bmatrix} \tag{1}$$

Another property of a pixel in the image other than the coordinate is gray level. Color images, are formed by passing the gray value of the original image through, respectively, red, green, blue filters. Due to hardware limitations of the number of gray levels are determined mainly by 2 forces. The most commonly used is 8-bit (28), although some applications are used 16 bit. 256 gray values can be defined as a byte here.

After obtaining digital images, image processing techniques are utilized to perform many applications such as to recover them from background, to eliminate the unwanted noise caused by a lighting system, to increase image quality, to correct image distortions caused by the structure of the system, etc. Image processing techniques can be grouped as image enhancement, image filtering and restoration, image compression, object identification.

Image enhancement is done to improve image interpretation and understanding of the image. Various filters try to recreate or fix a distorted image with using the solid parts of the image. Elimination or sharp the blur in the image includes applications such as sharp, refine edge, increasing the contrast and brightness of the image, removing the noise [15].

Segmentation is a grouping to analysis and to classify of specific shapes taken from an image. Segmentation process could be used to adjustment the desired color level, discretization, clarification, cleaning the background noise and the detection of objects in different form.

Discretization is done with thresholding technique between the object and the background. Before making thresholding to the image a picture in different shades of gray can be converted into binary format into a matrix which consisted 0 and 1s, at to show 0 black and 1 white. Related pixel on the output image is set 1 for each value above the threshold value and 0 for below the threshold value. Thus, features of the objects with the background on the object can be extracted.

Tagging is the separation of objects at none neighborhood from each other in a binary-coded image with painted different colors. Tagging process is done pixel to pixel scanning of binary-coded image starting at the (0.0). In binary-coded pictures objects is white and the floor is black. During the scanning, when white pixels (object) is encountered, it is looked at neighboring pixels 3x3 matrix shape of that pixel. White pixel value is assigned the same label if a pixel already labeled between adjacent pixels is found. If there isn't pre-labeled pixel value between adjacent the white pixel, the end labeling value is given to the white pixel by an increased and the labeling is applied to all picture [16].

Brightness and contrast settings of the image are made with contrast enhancement. The limits of different colors can be sharp with contrast and brightness settings by changing color value at a pixel and the image can be made more prominent.

2.2 Robot Manipulator

Robot used in this study (EDUBOT) is a five-axis robot designed for educational purposes. All joints realize rotational motion and joint movements are provided with R/C servo motor (Fig 2).

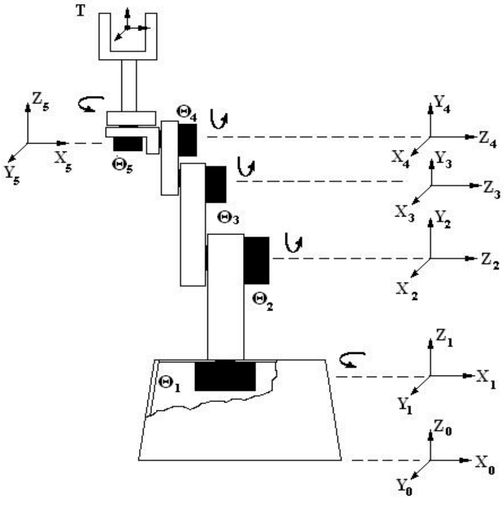


Figure 2. EDUBOT Robot and Rotation Axis.

Firstly robot's rotational axes are designated for calculation of the robot's forward kinematics and DH parameters are set out with utilizing the axes movement shown in Figure 2. DH parameters are given on Table 1.

Table 1. DH Parameters of the Robot

	1	2	3	4	5
α_i	0	-90	0	0	+ 90
a_i	0	0	0.08	0.06	0.125
θ_i	θ_1	θ_2	θ_3	θ_4	θ_5
d_i	0.18	0	0.01	0.042	0

α_i : the angle between z_{i-1} and z_i measured about x_i
a_i : distance along x_i from o_{i-1} to the intersection of the x_i and z_{i-1} axes
θ_i : the angle between x_{i-1} and x_i measured about z_{i-}
d_i : distance along z_{i-1} from o_{i-1} to the intersection of the x_i and e z_{i-1}

The transformation matrix of a joint is expressed as equation 2 [17]. Forward kinematics solutions are obtained as follows using DH parameters and equation 2.

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i * \cos\alpha_{i-1} & \cos\theta_i * \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1} * d_i \\ \sin\theta_i * \sin\alpha_{i-1} & \cos\theta_i * \sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1} * d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

The forward kinematics solution containing of the initial position and orientation of the end effector is obtained multiplied the transformation matrices of each joint, as shown in Equation 3.

$$ {}^{0}_{5}T = {}^{0}_{1}T\;{}^{1}_{2}T\;{}^{2}_{3}T\;{}^{3}_{4}T\;{}^{4}_{5}T \tag{3} $$

Forward kinematics solution of the Edubot Robot was obtained in eq. 4.

$$ {}^{0}_{5}T = \begin{bmatrix} 1 & 0 & 0 & 0.265 \\ 0 & 1 & 0 & 0.052 \\ 0 & 0 & 1 & 0.18 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

Inverse kinematics problem is conversion process of the orientation and the position of a robot manipulator end effectors from Cartesian coordinate system to the joint coordinate system [18]. This relationship between joint space and Cartesian space is illustrated in Figure 3 [4].

Inverse kinematics problem for a given robot configurations can be solved completely analytically. But the solution of the inverse kinematics problem comprised of nonlinear equation systems with two or three unknown cannot be achieved with analytical solutions in some cases. In such cases, numerical methods that joint angles iteratively solved are used to reach the solution [17]. In this study, EDUBOT robot's inverse kinematics solution was carried out with the ANN model based on forward kinematics solution.

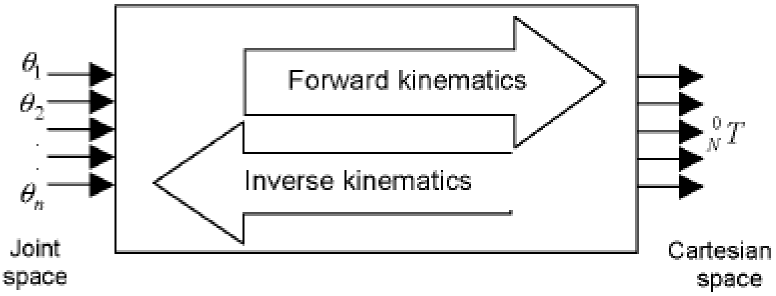


Figure 3. The Schematic Representation of Forward and Inverse Kinematics.

2.3 Artificial Neural Networl Application

ANN was designed as a simplified model of the biological neurons. The neurons are generally arranged in parallel to form “layers.” Although many combinations are possible, configuration usually follows a similar pattern. Each neuron in a layer has the same number of inputs, which is equal to the total number of inputs to that layer [19].

There exist many types of feedforward neural networks in the literature. In this study, Multi Layer Perceptron (MLP) and Radial Based Function (RBF) Neural Network model is used.

2.3.1. Multi Layer Perception

MLP consists of three-layers of neurons (input, hidden and output layer as shown in Figure 3) interconnected by weights. The MLP transforms n inputs to m outputs through some nonlinear function, are solved or trained by the error backpropagation algorithm.

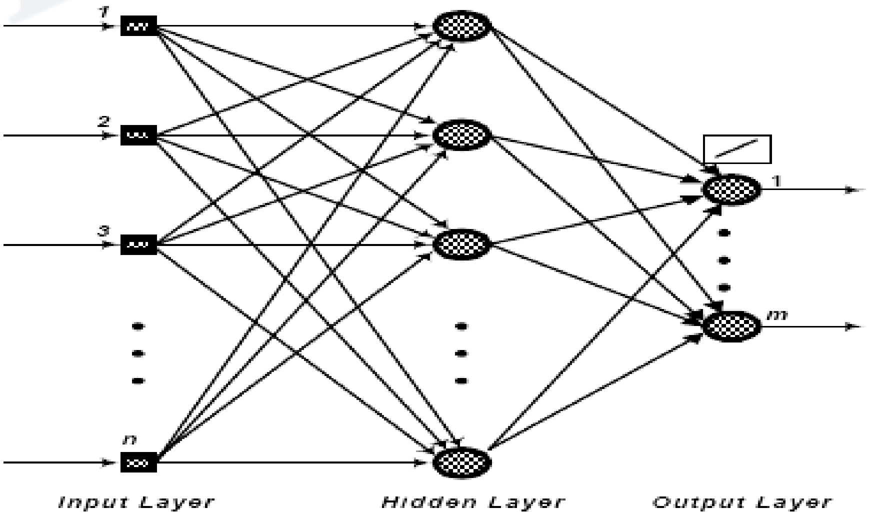


Figure 4. Multi Layer Perceptron.

The output layer neurons are pure summations of inner products between the nonlinear regression vector from the hidden layer and output weights.

MLP network starts with random initial values for its weights, and then computes a one-pass backpropagation algorithm at each time step k, which consists of a forward pass propagating the input vector through the network layer by layer, and a backward pass to update the weights by the gradient descent rule. After having trained on-line for a period of time, the training error should have converged to a value so small that if training were to stop, and the weights frozen. Also, the training of NNs is said to have reached a global minimum when, after changing the operating conditions, as well as freezing the weights, the network's response is still reasonably acceptable [20]. Steps of the algorithm as following:

- Initialize connection weights into small random values.
- Present the p th sample input vector of pattern $X_p = \left(X_{p1}, X_{p2}, \dots, X_{pN_0}\right)$ and the corresponding output target $T_p = \left(T_{p1}, T_{p2}, \dots, T_{pN_M}\right)$ to the network.
- Pass the input values to the first layer, layer 1. For every input node i in layer 0, perform:

$$Y_{0i} = X_{pi} \tag{5}$$

- For every neuron i in every layer $j = 1, 2, \dots M$, from input to output layer, find the output from the neuron:

$$Y_{ji} = f\left(\sum_{k=1}^{N_{j}=1} Y_{(j-1)k} W_{jik}\right) \tag{6}$$

where *f* is the activation function. The most widely used activation function for neurons in the hidden layer is the following sigmoidal function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{7}$$

- Obtain output values. For every output node i in layer M, perform:

$$O_{pi} = Y_{Mi} \tag{8}$$

- Calculate error value δ_{ji} for every neuron i in every layer in backward order $j = M, M-1, \dots 2, 1$, from output to input layer, followed by weight adjustments. For the output layer, the error value is:

$$\delta_{Mi} = Y_{Mi}\left(1 - Y_{Mi}\right)\left(T_{pi} - Y_{Mi}\right) \tag{9}$$

and for hidden layers:

$$\delta_{ji} = Y_{ji}\left(1 - Y_{ji}\right) \sum_{k=1}^{N_{j+1}} \delta_{(j+1)k} W_{(j+1)ki} \tag{10}$$

The weight adjustment can be done for every connection from neuron k in layer (i-1) to every neuron i in every layer i:

$$W_{jik}^{+} = W_{jik} + \beta \delta_{ji} Y_{ji} \tag{11}$$

where β represents weight adjustment factor normalized between 0 and 1.

The actions in steps 2 through 6 will be repeated for every training sample pattern *p*, and repeated for these sets until the Root Mean Square (RMS) of output errors is minimized.

RMS of the errors in the output layer defined as:

$$E_p = \tfrac{1}{2}\sum_{j=1}^{N_m}\left(T_{pj} - O_{pj}\right)^2 \tag{12}$$

for the *p* th sample pattern [21].

2.3.2 Radial Basis Function Neural Network

The basic architecture of an RBF with inputs and a single output is shown in Figure 3. The RBF network is a two-layer network, nonlinear hidden layer and linear output layer. The nodes in adjacent layers are fully connected [22]. The proposed model uses Gaussian kernel (ψ) as the hidden layer activation function.

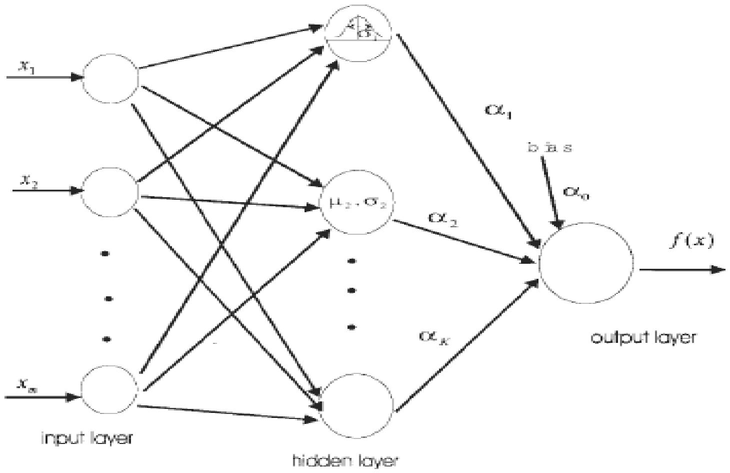


Figure 5. Radial Basis Function Neural Network.

The output layer implements a linear combiner of the basis function responses defined as;

$$f = \sum_{j=1}^{q} w_{i,j}\psi_j \tag{13}$$

Where, q is the sample size, ψ_j is response of the jth hidden neuron described as;

$$\psi_j = exp\left[-\frac{\|x-c_j\|^2}{2\sigma_j^2}\right] \tag{14}$$

Where, c_j is Gaussian function center value, and σ_j is its variance.

RBF training has two-stage procedure. In the first stage, the input data set is used to determine the center locations (c_j) using unsupervised clustering algorithm such as the K-means algorithm and choose the radii (σ_j) by the *k*-nearest neighbor rule. The second step is to update the weights (**W**) of the output layer, while keeping the (c_j) and (σ_j) are fixed [23].

3. Artificial Neural Networks for Calculation of the Inverse Kinematics

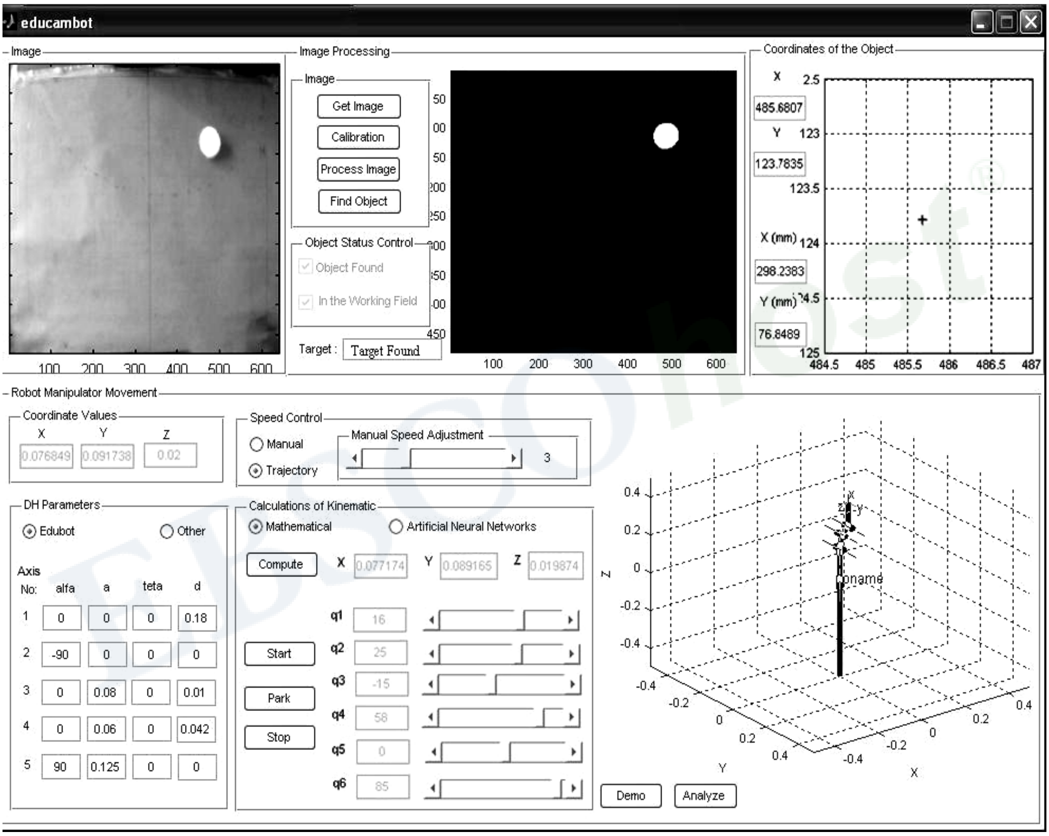
Firstly, a small sample of the ANN training data set in Table 2 was created using forward kinematics of the robot ANN for the realization of the robot's inverse kinematics solutions. While data sets was creating, uniform random variable function are used that change between boundaries values belong to each axis. Robot inverse kinematics solution was performed with the ANN modeling using this data. Two ANN model, MLP and RBF network was used. Once the MLP or RBF network model training procedure is performed, robot inverse kinematics solution is carried out in a very short time.

Table 2. Data Set Generated Randomly Used for Modeling with ANN

ANN Target Data					ANN Input Data		
Θ1	***Θ2***	***Θ3***	***Θ4***	***Θ5***	***X***	***Y***	***Z***
0,361	0,4167	-1,196	1,434	0	0,182	0,124	0,113
-0,055	-0,500	0,958	0,859	0	0,182	0,041	0,348
0,688	0,483	-1,148	1,524	0	0,121	0,166	0,085
0,687	-0,060	-0,139	0,222	0	0,170	0,207	0,193
-0,536	0,428	-0,151	1,507	0	0,116	-0,008	0,008
-0,001	-0,5703	-0,062	1,467	0	0,199	0,051	0,166
0,050	0,248	-0,228	0,6161	0	0,235	0,064	0,084

To determine the robot's joint angles values using the ANN model and the parameters the interface was created. Thus, access to parameters of ANN models used to find information of the joint angles, was conducted. Robot's forward kinematics such as equation 1 was derived taking the robot's own information with data button on the interface.

ANN used in inverse kinematics solution is should be selected after of the robot's forward kinematics was obtained. Each option contains some parameters within itself. The initial values assigned to these parameters can be changed via the interface. Interface that given overall appearance in Figure 6 was created for image processing techniques applied in study and kinematic calculations required the motion of the robot arm.



educambot

Image Processing

Object Status Control
☑ Object Found
☑ In the Working Field

Target: Target Found

Coordinates of the Object
X 485.6807
Y 123.7835
X (mm) 298.2383
Y (mm) 76.8489

Robot Manipulator Movement

Coordinate Values
X 0.076849
Y 0.091738
Z 0.02

Speed Control
○ Manual
● Trajectory
Manual Speed Adjustment 3

DH Parameters
● Edubot ○ Other

Axis No:	alfa	a	teta	d
1	0	0	0	0.18
2	-90	0	0	0
3	0	0.08	0	0.01
4	0	0.06	0	0.042
5	90	0.125	0	0

Calculations of Kinematic
● Mathematical ○ Artificial Neural Networks
Compute X 0.077174 Y 0.089165 Z 0.019874
q1 16
q2 25
q3 -15
q4 58
q5 0
q6 85

Figure 6. System Interface.

4. EXPERIMENTAL RESULTS

MATLAB R2008b was used for taking the image, detecting the location of the object inside the workspace, the robot arm kinematic calculations and simulated motion of the robot arm. Also an interface was created using MATLAB GUIDE.

In this part of application, the robot arm is directed to detected coordinate values after the object image taken from the camera is processed and passed through the necessary steps.

After ANN method for solving inverse kinematics of the robot has been selected, obtained alignment values was directed to robot arm and sync of the movement was observed.

After determined the joint angles was transferred to the interface, the arm speed adjustments were made. Then, by performing the desired motion by robot arm, carrying of object in the workspace was provided from location to another place, pressing to start. Workspace is shown in Figure 7 [24].

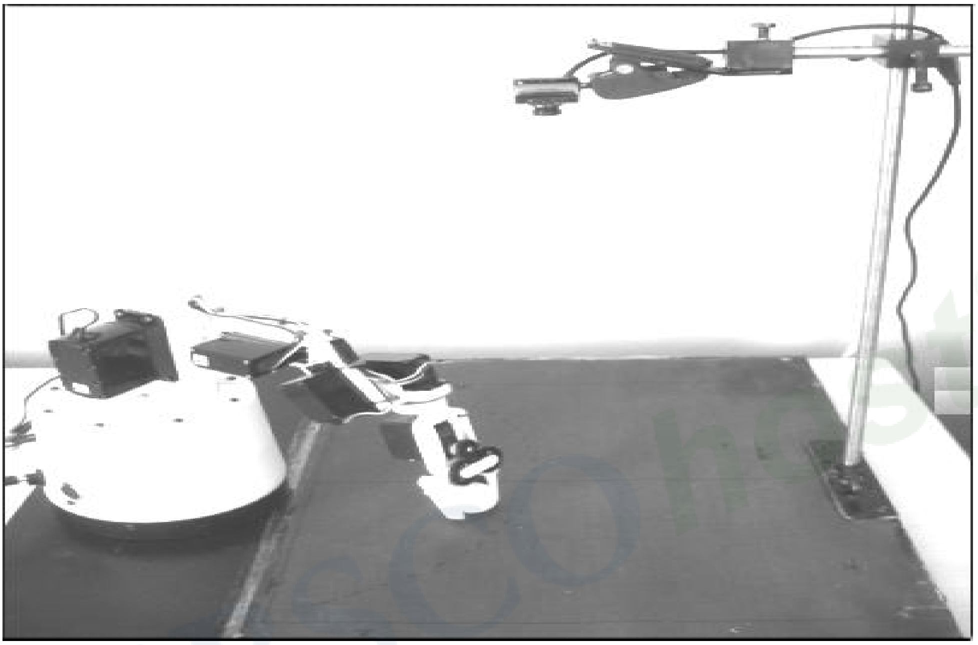
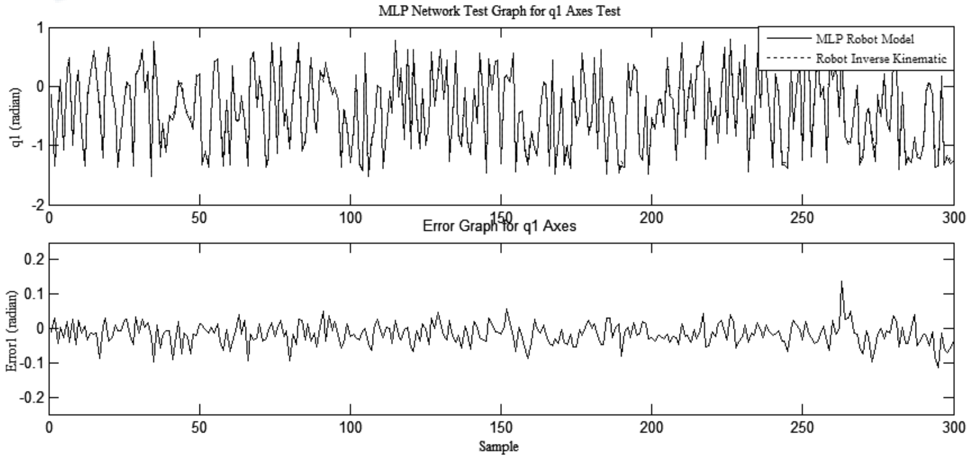


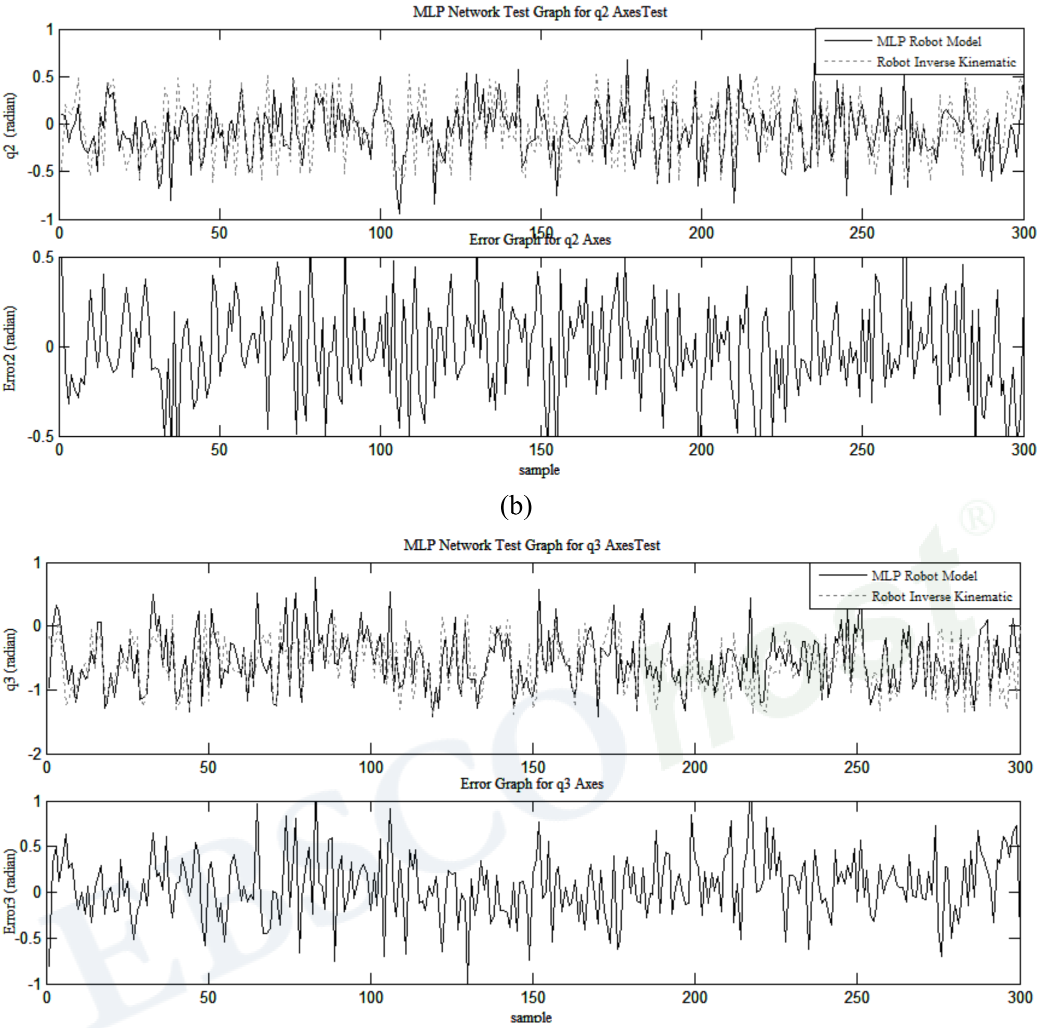
Figure 7. Workspace.

Position and position error of axis 1, 2 and 3 with MLP model are shown in Fig 8.



(a)

Figure 8. (Continued)



(b)

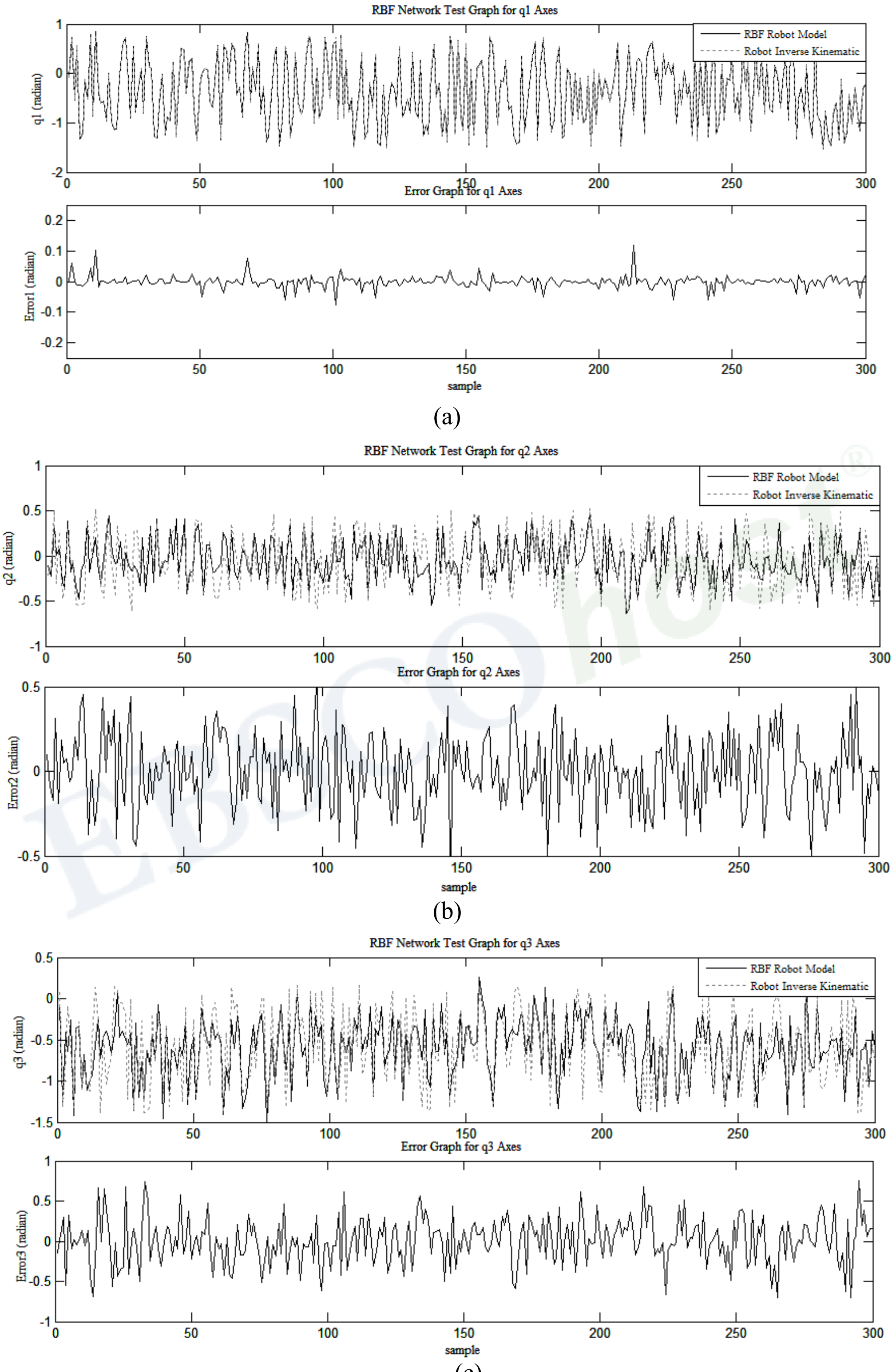
(c)

Figure 8. Position and Error of Base 3 Axes with MLP Model.

Position and position error of axis 1, 2 and 3 with RBF model are shown in Fig 9.

Errors are smaller with RBF network model than MLP model as shown Fig 8. and Figure 9. Also, error of the axis 1 is smaller than the other axes for both network models.

Many experiments were implemented for different object positions, and Table 3 was created. X, Y, Z columns in the table shows the object's actual coordinate values. X'-Y'-Z' columns identifies end effector coordinates and q1-q2-q3-q4-q5 columns shows calculation results of each joint.



(a)

(b)

(c)

Figure 9. Position and Error of Base 3 Axes with RBF Model.

Table 3. Result of Inverse Kinematics Calculation

		Desired			Calculated			Joint Angles				
		X	Y	Z	X’	Y’	Z’	q 1	q 2	q 3	q 4	q 5
MLP	tansig	0.0868	0.0509	0.02	0.1068	0.0340	0.0262	-6	19	-10	60	0
	logsig	0.0868	0.0509	0.02	0.1011	0.0373	0.0318	-5	16	-11	68	0
	purelin	0.0868	0.0509	0.02	0.0896	0.0484	0.0214	-1	20	-10	67	0
RBF		0.0868	0.0509	0.02	0.0845	0.0529	0.0176	0	22	-13	68	0
MLP	tansig	0.0422	-0.127	0.02	0.1013	-0.139	0.0455	-61	14	-14	66	0
	logsig	0.0422	-0.127	0.02	0.0938	-0.130	0.0457	-62	12	-14	74	0
	purelin	0.0422	-0.127	0.02	0.1046	-0.108	0.028	-55	20	-16	70	0
RBF		0.0422	-0.127	0.02	0.0394	-0.123	0.0197	-80	17	-6	79	0
MLP	tansig	0.0751	-0.088	0.02	0.0910	-0.089	0.0328	-49	15	-12	75	0
	logsig	0.0751	-0.088	0.02	0.0778	-0.099	0.0266	-55	19	-16	76	0
	purelin	0.0751	-0.088	0.02	0.0991	-0.067	0.0260	-42	17	-10	72	0
RBF		0.0751	-0.088	0.02	0.0736	-0.093	0.0169	-55	23	-15	75	0
MLP	tansig	0.0795	0.1253	0.02	0.0813	0.0993	0.0140	19	21	3	40	0
	logsig	0.0795	0.1253	0.02	0.0534	0.1043	0.0081	24	20	5	50	0
	purelin	0.0795	0.1253	0.02	0.0687	0.0980	0.0268	20	20	-13	66	0
RBF		0.0795	0.1253	0.02	0.0802	0.1266	0.0240	27	24	-5	36	0
MLP	tansig	0.0554	0.0754	0.02	0.0919	0.0684	0.0316	6	20	-17	70	0
	logsig	0.0554	0.0754	0.02	0.0687	0.0796	0.0301	11	15	-11	77	0
	purelin	0.0554	0.0754	0.02	0.0907	0.0690	0.0227	6	20	-10	63	0
RBF		0.0554	0.0754	0.02	0.0558	0.0742	0.0161	9	19	-8	75	0
MLP	tansig	0.0322	0.0315	0.02	0.0728	0.0294	0.0365	-9	15	-17	90	0
	logsig	0.0322	0.0315	0.02	0.0843	0.0103	0.0416	-15	15	-22	90	0
	purelin	0.0322	0.0315	0.02	0.1003	0.0241	0.0263	-9	19	-12	67	0
RBF		0.0322	0.0315	0.02	0.0422	0.0232	0.0200	-15	11	11	81	0
MLP	tansig	0.0591	0.0030	0.02	0.0856	0.0191	0.0398	-12	12	-12	84	0
	logsig	0.0591	0.0030	0.02	0.0771	-0.013	0.0379	-25	14	-17	90	0
	purelin	0.0591	0.0030	0.02	0.0961	0.0046	0.0286	-16	16	-9	71	0
RBF		0.0591	0.0030	0.02	0.0620	0.0046	0.0185	-20	17	-3	80	0
MLP	tansig	0.0361	-0.043	0.02	0.1026	-0.042	0.0502	-34	10	-17	90	0
	logsig	0.0361	-0.043	0.02	0.0872	-0.039	0.0299	-36	19	-20	90	0
	purelin	0.0361	-0.043	0.02	0.1146	-0.033	0.0274	-28	16	-9	69	0
RBF		0.0361	-0.043	0.02	0.0486	-0.047	0.0161	-52	14	10	81	0

CONCLUSION

In this study, after the object was detected, moving the object to another location was performed by five-axis robot arm within the workspace. For the realization of the robot arm motion, robot's forward kinematics solution was carried out finding DH parameters of the robot. Inverse kinematics solution of the robot was performed based on forward kinematics by ANN based approach. Two different ANN models, the MLP and RBFNN, are used for inverse kinematics calculations. According the Table 3 RBF network yielded much better results than MLP. To reach the target point with RBFNN method was realized approximately with accuracy rate between 80% to 87%.

REFERENCES

[1] Wanichnukhrox, N. *Real-Time, Visual Servo Control of a Planar Robot*, Ohio University Master Thesis, 2003.

[2] Mundhra, K.; Suluh, A.; Sugar, T.; McBeath M. *Intercepting a Falling Object: Digital Video Robot.* Proceeding of the 2002 IEEE International Conference on Robotics and Automation, 2002; USA.

[3] Modi, K.P.; Sahin, F.; Saber, E. "*An Application of Human Robot Interaction: Development of a Ping-Pong Playing Robotic Arm*", 2005 IEEE International Conference on System, Man and Cybernetics, 2005; Vol.2.

[4] Bustamante, L.; Gu, J. *Localization of Electrical Outlet for a Mobile Robot Using Visual Servoing*, Canadian Conference on Electrical and Computer Engineering, 2007.

[5] Knoeppel, C.; Schanz, A.; Michaelis, B. *Robust Vehicle Detection at Large Distance Using Low Resolution Cameras*, Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn(MI), 2000; USA.

[6] Choi, W.; Ryu, C.; Kim, H. *Navigation of a Mobile Robot Using Mono-Vision and Mono-Audition*, Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999; Vol.4.

[7] Gan, J. Q.; Oyama, E.; Rosales, E.M. and Hu, H. *A Complete Analytical Solution to the Inverse Kinematics of the Pioneer 2 Robotic Arm*, Robotica, 2005; vol. 23, pp. 123–129.

[8] Rao, D.H. and Gupta, M.M. *Performance Comparison of Dynamic Neural Networks as Applied to Robot Inverse Kinematic Computations*, Proceedings of the American Control Conference, 1994.

[9] Bingul, Z.; Ertunc, H.M. and Oysu, C. *Comparison of Inverse Kinematics Solutions Using Neural Network for 6R Robot Manipulator with Offset*, Congress on Computational Intelligence Methods and Applications, 2005.

[10] Choi, B. B. and Lawrence, C. *Inverse Kinematics Problem in Robotics Using Neural Networks*, NASA Technical Memorandum 105869, 1992.

[11] Yang, S.S.; Moghavvemi, M. and Tolman, J. D. *Modeling of Robot Inverse Kinematics Using Two ANN Paradigm*, TENCON Proceedings, 2000.

[12] Zhao, Q.; Sun, Z.; Sun, F. and Zhu, J. *Appearance-based Robot Visual Servo via a Wavelet Neural Network*, International Journal of Control, Automation, and Systems, 2008; vol. 6, pp. 607-612.

[13] Cisneros, M.A.P. *Intelligent Model Structures in Visual Servoing*, University of Manchester Institute of Science and Technology, Doctorate Thesis, 2004.

[14] Bal, H. *Grain Size Analysis From Material Images by Using Image Processing Techniques*, Master Thesis, Gazi University, Ankara, Turkey, 2006.

[15] Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3.rd Edition, Prentice Hall, 2008.

[16] Kert, M. *Optimization of the Route for a Robot Arm by Captured Coordinate From a Real Image*, Master Thesis, Mustafa Kemal University, Antakya, Turkey, 2006.

[17] Bingul, Z.; Kucuk, S. *Robot Technique-I*; Birsen Press: Istanbul, Turkey, 2005.

[18] Wang, L.C.T.; Chen, C.C. *A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators*, IEEE Transaction on Robotics and Automation, 1991; Vol. 7.

[19] Cavuto, D. J. *An Exploration and Development of Current Artificial Neural Network Theory and Applications with Emphasis on Artificial Life*, Master Thesis, The Cooper Union Albert Nerken School Of Engineering, 1997.

[20] Park, J.; Harley, G.; Venayagamoorthy, G. K. *Comparison of MLP and RBF Neural Networks Using Deviation Signals for On-Line Identification of a Synchronous Generator,* Jung- Power Engineering Society Winter Meeting, IEEE, 2002; vol.1. pp 274–279.

[21] Kawaguchi, K. (2000). http://www.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node22.html.

[22] Lee, C.; Chung, P.; Tsai, J. and Chang, C. *Robust Radial Basis Function Neural Networks,* IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, 1999; Vol. 29.

[23] Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall: NY, USA, 1999.

[24] Ersan, E. *The Vision – Based Control of A Robot Arm*, Master Thesis, Marmara University, Istanbul, Turkey, 2009.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 7

ANN-BASED APPROACHES TO STUDY THE NANOSCALE CMOS DEVICES

F. Djeffal*[*] and T. Bendib
LEA, Department of Electronics, University of Batna, Batna, Algeria

1. INTRODUCTION

Over the past three decades, the primary driver of the exponential improvements in integrated circuit performance has been the scaling of transistor dimensions. The inherent benefits of MOSFET scaling are the speed improvement and energy reduction associated with a binary-logic transition. As the MOSFET is scaled below the 100 nm technology node the advantages of MOSFET scaling are diminished by the short channel effects [1-3]. Ultra-thin film body multigate structures become to be envisaged as a possible alternative to the conventional devices, due to its enormous potentiality to push back the integration limits to which conventional bulk transistor are subjected [1-4]. The main advantage of this architecture is to offer a reinforced electrostatic coupling between the conduction channel and the gate electrode. In other terms, a multigate structure can efficiently sandwich (and thus very well control, electrostatically speaking) the semiconductor element playing the role of the transistor channel, which can be a Silicon thin layer or nanowire. Moreover, it is known for its higher drive current, improved subthreshold slope, improved short channel effect control and potential circuit design flexibility [1–4]. As shown in Figure 1, with two gates controlling the channel, short-channel effects can be greatly suppressed. Due to the fact that simulation of nanoscale CMOS circuits has been the primary factor driving improvements in integrated circuit performance and cost, which contributes to the rapid growth of the semiconductor industry, there is a need to develop a new theory and modeling techniques that capture the physics of quantum transport accurately to guide the design for nanoscale CMOS circuits.

[*] Corresponding author: Tel/Fax: 0021333805494, E-mail: faycaldzdz@hotmail.com

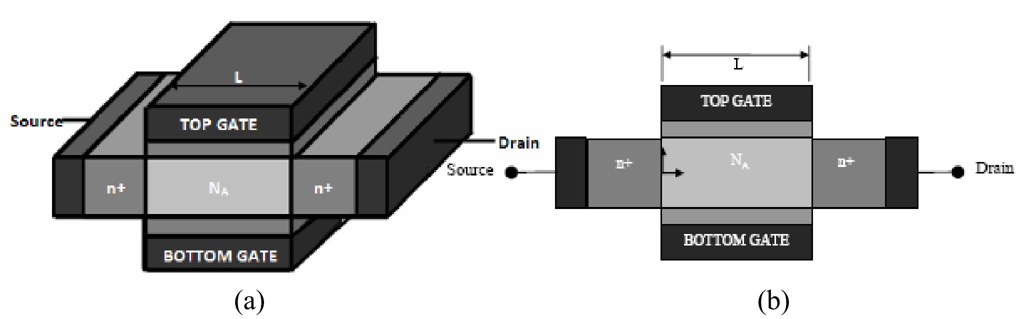


Figure 1. Multigate (Double Gate) nanoscale MOSFET considered in this study: (a) 3-D device structure and (b) 2-D cross section.

Although the operation of multigate MOSFET is similar to the conventional MOSFET, the physics of this type of MOSFET is more complicated. Moreover, physical phenomena such as quantum mechanical effects have to be considered. Therefore, simulation tools which can be applied to design nanoscale transistors in the future require new theories and modeling techniques that capture the physics of quantum transport accurately and efficiently [5-8]. But from the circuit modeling point of view even 2D solution of numerical models (non-equilibrium Green's function (NEGF) formalism with mode space representation, NEGF with full 2D space representation, Monte Carlo, fully self-consistent coupled Schrödinger and Poisson equations,…) is an overkill approach in terms of both complexity and computational cost. For analytical modeling, in general, it is difficult or almost impossible to obtain closed form analytical models for nanodevices (analytical drain current modeling, physical modeling, …) [5-8]. Thus, models are obtained by a simplification of the full physical model. The compact models allow for fast system level simulation of the nanoscale circuits. However, the accuracy of such a model can be questionable because of the simplifications made during the model development phase. Model accuracy and simplicity are important for the design of complex systems. Artificial intelligence model would be preferable and could provide practical solutions [9]. In this sense, this chapter presents the applicability of the artificial neural networks (ANN) for the study and simulation of the nanoscale (multigate) CMOS-based devices. The presented results are discussed in order to show the efficiency of ANN method for studying future nanoscale devices.

2. Modeling Constraints and Challenges

Accurate models for undoped double-gate [10-13] and surrounding-gate MOSFETs [10], [14-15] have been recently developed using the above principles, showing good agreement with three dimensional numerical simulations. These models assume that the electrostatic control of the channel is so good that short-channel effects can be neglected. The inclusion of short-channel effects in multigate MOSFETs models, using physical equations and without decreasing the order of continuity of the devices, is still a modeling challenge.

For devices with channel lengths shorter than 50 nm, the drift-diffusion mechanism may not be the dominant transport mechanism. Ballistic or quasi-ballistic transport may occur. Adequate models for nanoscale devices must consider the quantum transport regime [16-17]. On the other hand, for films smaller than 10 nm, quantum confinement in the film may not be negligible. The subband contributions should be considered in the drain current equation. The quantum effects affect the distribution of charge in the film and as a consequence, the threshold voltage (increase of the threshold voltage in an n-channel device). Some recent models which consider the band structure of silicon have been recently presented [18]. At very low temperatures, quantum confinement of the charge becomes more important, and this affects the shape of the transconductance characteristics [18].

A succession of peaks and valleys are observed, which correspond, respectively, when the bottom of the subbands cross the quasi-Fermi energy levels at the source and drain. The location of peaks and valleys has been accurately predicted. These models should be still completed with the inclusion of short channel effects.

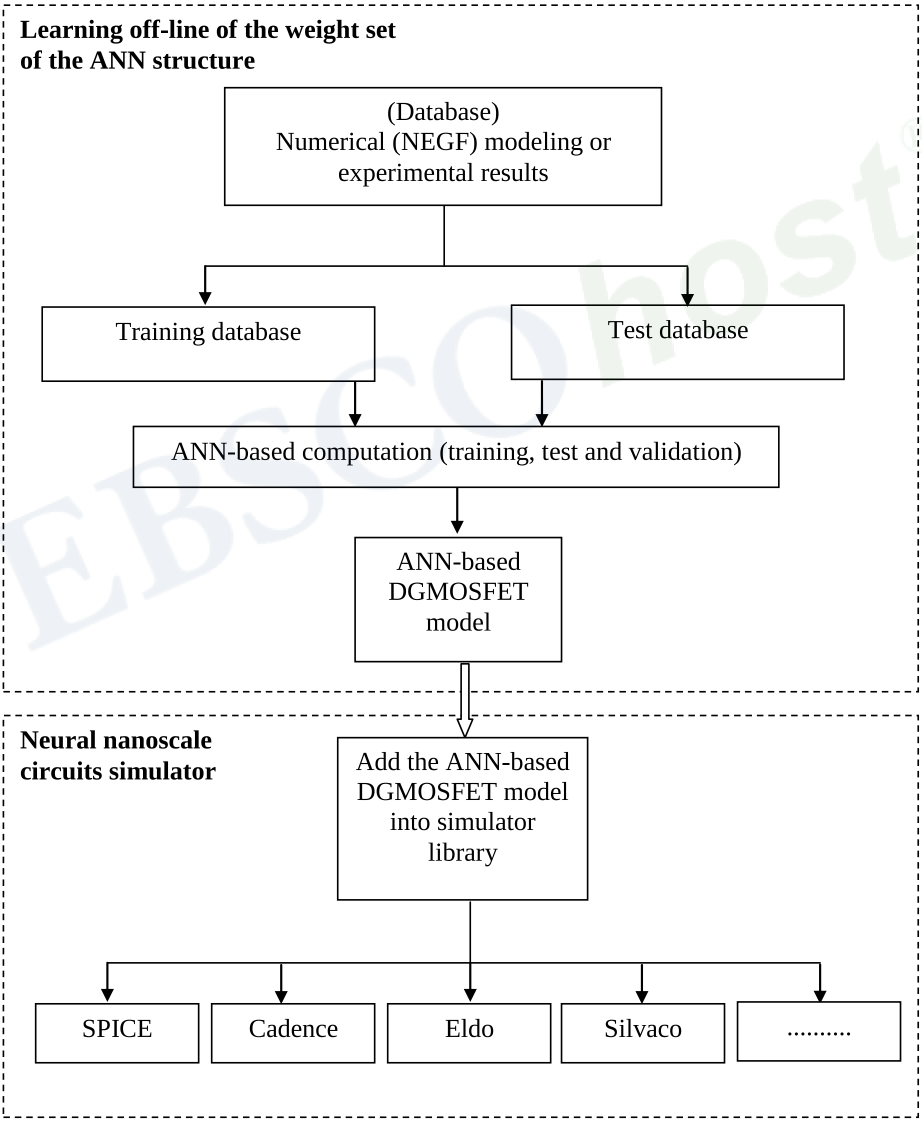

Learning off-line of the weight set of the ANN structure
(Database) Numerical (NEGF) modeling or experimental results
Training database
Test database
ANN-based computation (training, test and validation)
ANN-based DGMOSFET model
Neural nanoscale circuits simulator
Add the ANN-based DGMOSFET model into simulator library
SPICE
Cadence
Eldo
Silvaco
....

Figure 2. ANN-based approach for nanoscale CMOS circuits simulation.

In this chapter, we have simulated the nanoscale multigate (DG) MOSFET within the NEGF formalism by using nanoMos 2.0 [19, 20]. NanoMOS 2.0 is an accurate 2-D simulator for thin body, fully depleted, double gate n-type MOSFETs that has been developed by a group at Purdue university. However, because of the complexity and time consuming nature of the quantum calculations for implementation in SPICE-type circuit simulators, it has been necessary to simulate the quantum mechanical model with high speed and accuracy. The data obtained by measurements or device simulations can be used as the target data set for optimizing the ANN architecture. In this study, the database for optimizing the ANN structure has been created based on data obtained by simulation of nanoscale (DG) MOSFET within the NEGF formalism. After optimization, the simulation results of the proposed ANN-based model is compared with NEGF simulation and finally, the ANN model for nanoscale (DG) MOSFET is used as a subcircuit in SPICE software for modeling and simulation of the nanoscale CMOS-based circuits. A simplified overview of the ANN-based approach to study the nanoscale CMOS-based circuits is shown in Figure 2.

3. Modeling Methodology

3.1. Numerical Modeling

As MOSFETs scale to the nanometer regime, canonical carrier transport theories are no longer capable of describing carrier transport accurately. The canonical theories are basically derived from the Boltzmann transport equation, with more or fewer approximations being made [7]. These models focus on scattering-dominant transport, which typically occurs in long channel devices. However, nanoscale transistors operate in a quasi-ballistic transport regime [7]. Simulations using conventional models may either under-predict or over-predict the device performance [5-7].

The NEGF method [1,19] provides a sound approach for simulating the non-equilibrium nanoscale systems. Using NEGF method for solving the Schrödinger equation, the density of states and the charge on the surface of the nanoscale DG MOSFET are calculated. Subsequently, utilizing the calculated charge and solving the Poisson equation, the new electrostatic potential is obtained [1]. However, due to the complexity and time-consuming behavior of these analyses, it is being necessary to apply a fast method that can be used in SPICE-type circuit simulators. In this regard, therefore, a high computational speed is necessary if the model is to be implemented in a SPICE simulator. Based on the efficiency proven by NEGF for the modeling of nanoscale MOSFETs and the difficulty imposed at the moment by the constraints of the nanotechnology (sub-10 nm) to form an experimental databases [21], the NEGF formalism will be used to form the required databases which will be employed to optimize our Neural Networks structures that have several advantages over conventional computing methods. Those advantages are robustness to input and system noise, learning from examples, ability to memorize, handling situations of incomplete information and corrupted data, and performing in real time.

3.2. Neural Modeling

Artificial neural networks (ANNs) are powerful tools for empirical modeling that can be applied to a broad spectrum of problems, even when the underlying causal relationships are poorly understood or completely unknown. Moreover, any finite-dimensional vector function on a compact set can be approximated to arbitrary accuracy by multi-layer feed-forward (MLP) or Radial basis function (RBF) ANNs, provided that enough data and computing resources are available [22, 23]. Recently, several works have been focused to study and improve the modeling techniques of nanoscale devices using ANN-based approaches.

a. MLPANNs-based approach

MLPANNs-based methods have been widely used for modeling various complex and nonlinear processes (classification, speech recognition, and signal processing). The model based on artificial neural network [22-24] assumes that input and output patterns of a given problem are related by a set of neurons organized in hidden layers. The layers in these networks are interconnected by communication links that are associated with weights (Figure 3) that dictate the effect on the information passing through them. These weights are determined by the learning algorithm.

The network consists of three layers named as input layer, hidden layer and output layer. Each layer has its own number of neurons. The input to the node l in the hidden layer is given by

$$h_j = f\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right) \tag{1}$$

and the output of the network by

$$y = \sum_{i=1}^{k} wo_i . h_i \tag{2}$$

where w_{ij} are the weights connecting the inputs to node j in the hidden layer, b_j is the bias to the node, and w_{oi} are the weights from the hidden to the output layer.

According to the learning algorithm used to optimize the MLPANN structure, the MLPANNs can be categorized as:

- Fixed weight MLPANNs: these do not need any kind of learning.
- Unsupervised MLPANNs: these networks are trained (weights are adjusted) based on input data only. The networks learn to adapt using experience gained from previous input.
- Supervised MLPANNs: these are the most commonly used MLPANNs. In these networks, the system makes use of both input and output data.

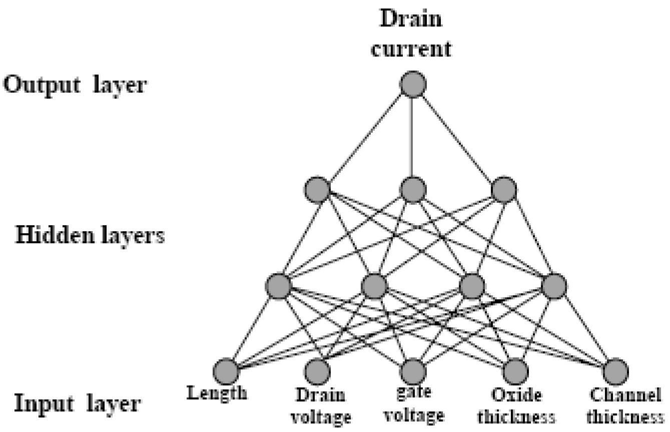


Figure 3. Optimized MLPANN structure to study the nanoscale DG MOSFET.

The activation function relates the output of a neuron to its input based on the neuron's input activity level. Some of the commonly used functions include: the threshold, piecewise linear, sigmoid, tangent hyperbolic, and the Gaussian function [22-23]. The learning process of the MLP network involves using the input–output data to determine the weights and biases. One of the techniques used to obtain these parameters is the back-propagation algorithm [22–24]. In this method, the weights and biases are adjusted iteratively to achieve a minimum mean square error between the network output and the target value.

The artificial neural network was built to relate input parameters (*L,* V_{DS}*,* V_{GS}*,* t_{ox} *and* t_{si}) to output parameter I_D, where *L* is the channel length, V_{DS} is the drain source voltage, V_{GS} is the gate source voltage, t_{si} silicon film thickness, t_{ox} gate oxide thickness and I_D is the drain current. Each of these parameters is indexed with one neuron (Figure 3). The activation function used in this ANN structure is the sigmoid function. It is important to denote that the number of input parameters of the ANN-based model can be extended for other parameters (temperature, band-to-band leakage current, gate direct tunneling current, overlap effect, cut-off frequency, . . .).

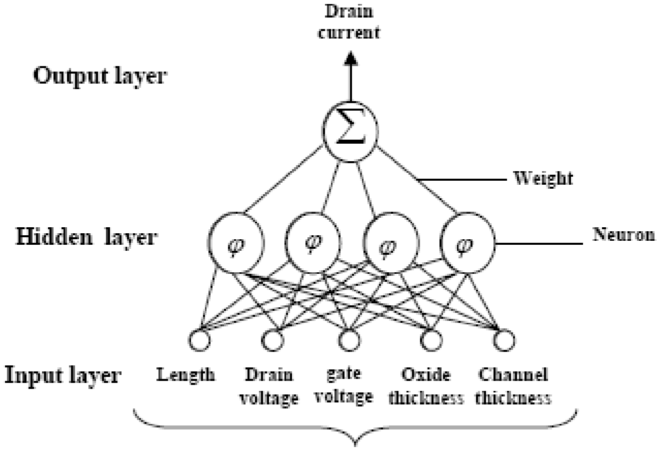


Figure 4. Optimized RBFANN structure to study the nanoscale DG MOSFET.

b. RBFANNs-based approach

Radial basis function (RBF) networks are a relatively new class of ANNs. In several investigations [1,2,23], their suitability for nonlinear system modeling has been demonstrated. They consist of an input, one hidden and an output layer which are interconnected in a feed forward way from the input to the output layer. The multiple-input/single-output (MISO) structure of an RBF network used to simulate the nanoscale DG MOSFETis shown in Figure 4, where x is the input vector given as function of gate length L, drain source voltage V_{ds}, gate source voltage V_{gs}, oxide thickness t_{ox} and silicon thickness t_{si}; drain current I_D is the output vector and φ_i^* is the radial basis function. Applying the network structure shown in Figure 4, the input/output mapping can be described by

$$I_D = \sum_{i=1}^{m} P_i \varphi_i^* (\| x(k) - \zeta_i \| \tag{3}$$

where *x* is the input vector, *m* is the number of neurons in the hidden layer, *Pi, i=1, ..., m,* are the output gains (weights) of the hidden layer and ζ_i, *i=1, ..., m,* are the centres of the basis functions.

For the optimization of our RBFNN, the RBF algorithm, developed by Bishop [25] and Nabney [26], which uses a combination of unsupervised learning in a hidden layer, and a supervised learning technique in the output layer can be used as a better algorithm to optimise the RBFNN structure. The optimized ANN configuration was selected based on the minimal residual error found from the set of the optimal structures. In the training process, the weights are corrected according to the Bishop and Nabney algorithm [25, 26].

The optimised RBFANN, which gives the behavior model of the studied nanoscale DG MOSFET, had a hidden layer with 5 units and Gaussian basis functions were trained by 480 cases and tested in a separated set of 50 cases as it is shown in Figure 4.

c. ANFIS-based approach

Adaptive Neuro-Fuzzy Inference System (ANFIS) is an adaptive network which permits the application of neural network topology together with fuzzy logic. The artificial neural networks provide effective learning methods and speed of computations whereas fuzzy set theory allows thinking and reasoning capability of the fuzzy logic. The learning usually applies to the membership function parameters of the IF and THEN part of the fuzzy rules [27,28].

At the computational level, ANFIS can be regarded as a flexible mathematical structure that can approximate a large class of complex nonlinear systems to the desired degree of accuracy. In fuzzy section, only zero or first order Sugeno inference system or Tsukamoto inference system can be used [29-30]. For simplicity, we assume that the fuzzy inference system has two inputs (*x*, *y*) and one output (*f*). For a first order Sugeno fuzzy model, a typical rule set with fuzzy based if-then rules can be expressed as follows:

Rule1: If *x* is *A1* and *y* is *B1*, then $f_1 = p_1 x + q_1 y + r_1$
Rule2: If *x* is *A2* and *y* is *B2*, then $f_2 = p_2 x + q_2 y + r_2$

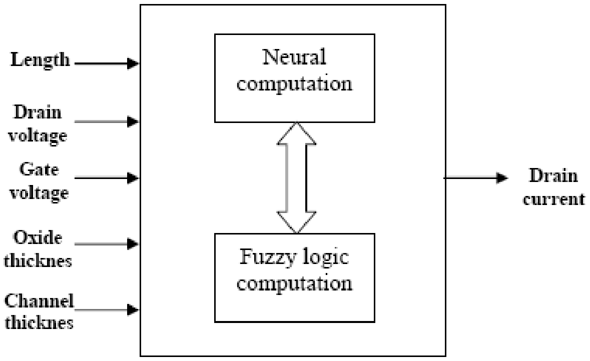


Figure 5. A simplified overview of ANFIS model.

where *pi*, *qi*, *ri* are linear output parameters (consequent parameters).

A hybrid learning algorithm [31] is used for ANFIS training and it consists of two stages: forward pass and backward pass. In the forward pass, the consequent parameters are identified by the least squares estimation and in the backward pass the premise parameters are updated by the gradient descent. Based on the advantages provided by this techniques, the ANFIS model has been built to relate input parameters (L, V_{ds},V_{gs} , t_{ox} and t_{si}) to output parameter I_d, where L is the channel length, V_{ds} is the drain source voltage, V_{gs} is the gate source voltage, t_{si} is the silicon film thickness, t_{ox} is the gate oxide thickness and I_d is the drain current in DG MOSFET. A simplified overview of the proposed ANFIS model is shown in Figure 5.

The ANFIS-based model has been proposed as an improved approach over ANNs-based models. The ANFIS model could significantly reduce the output errors to less than 2% in comparison with ANNs-based models that the output errors were less than 5%. Another aspect of superiority of ANFIS model in comparison with ANNs-based models is the lower number of epochs (training time) which is needed to reach convergence. Therefore, the training time for the ANFIS model is definitely less than required time for designing similar model using pure neural network. It means that the ANFIS model is better than ANN for redeveloping the model and increasing the input parameters. It may be noted that for an ANN model, we have to perform a trial and error process to develop the optimal network architecture, while the ANFIS model does not require such a procedure. Because one of the advantages of ANFIS as opposed to ANN is that the ANFIS is more transparent and we can obtain input-output relationship from membership functions and “If-Then” rules.

d. NSM-based approach

Neural Space Mapping is an advanced optimization concept, proposed by Bandler *et al.* [13], for modeling and design of engineering devices and systems, allowing expensive devices optimizations to be performed efficiently with the help of fast and approximate "coarse" or surrogate models [32], [33]-[35]. Space mapping intelligently links companion "coarse" (ideal, fast, or low fidelity) and "fine" (accurate, practical, or high fidelity) models of different complexities, e.g., empirical circuit-theory based simulations and nanoscale devices modeling, to accelerate iterative design optimization. Through Space mapping optimization, the surrogates of the fine models are iteratively refined to achieve the accuracy of EM simulations with the speed of circuit-theory based simulations.

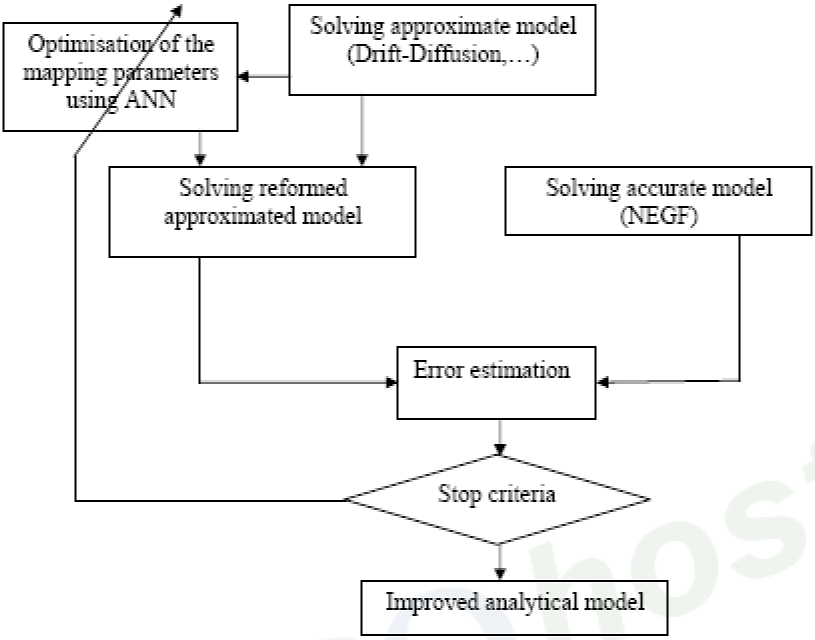

Optimisation of the mapping parameters using ANN
Solving approximate model (Drift-Diffusion,…)
Solving reformed approximated model
Solving accurate model (NEGF)
Error estimation
Stop criteria
Improved analytical model

Figure 6. NSM-based approach block diagram to study the CMOS-based devices.

Recently a space mapping based neuromodeling technique combining neural networks with space mapping [32] was developed, using neural networks to map the coarse model to the fine model. It retains the efficiency of space mapping optimization by directing the bulk of CPU intensive evaluations to the coarse model, while preserving the accuracy and confidence offered by the fine model. The coarse model is typically an empirical or equivalent circuit model, which is fast but often has a limited validity range for its parameters, beyond which the simulation results may become inaccurate. The fine model is from a detailed devices simulator or measurement, which is accurate but CPU intensive. Neural networks are used to provide the mapping between the coarse model parameter space and the fine model parameter space, subsequently establishing the mathematical link between the coarse and fine models.

In NSM-based approach to simulate the nanoscale CMOS circuits, ANN generates the optimal distribution of the mapping parameters (mobility, diffusion coefficient,…), which are inserted in the approximate model to produce simulation results close to the accurate model. Figure 6 shows the NSM-based approach block diagram to study the CMOS-based devices.

3.3. Computational Efficiency

The forgoing simulation results show that the ANN-based approach makes it feasible to include quantum effects accurately and generally in nanoCMOS-based circuits' simulation. In this section, we go further to show that the ANN-based models are in fact quite efficient in accomplishing this. In particular, we compare the computational performances of the classical (analytical or numerical) and neural-based models.

Table 1. Comparison between the various approaches of study of the DG MOSFET

Approach	Model form	Effectiveness of the approach
-NEGF	Numerical	Accurate/ very slow
-Compact models	Analytical	Less accurate/ fast
-MLPANN	Black-box	Accurate/ fast
-RBFANN	Black-box	Accurate/ fast
-ANFIS	Black-box	Accurate/ very fast
-NSM	Analytical	Accurate/ very fast

Table 1 gives a comparison of the computational time requirements for simulating nanoscale DG MOSFET with various approaches where the ANNs-based models computation time should be compared to the orders of magnitude increase in computation time for more rigorous quantum models and analytical approaches, such as those based on the real-space NEGF formalism [1,5,8], mode-space NEGF formalism and analytical approximations [5]. Obtained results can be explained by the fact that the ANNs-based approaches are characterized as computational models based on parallel distributed processing of data. It is important to note that the NSM-based approach provides a better performance in terms of computational time and flexibility to be implemented into circuits electronics simulators. Since the ANNs-based models are only moderately more computationally demanding than the associated numerical models, it can even be feasibly for to study other structures more complex than DG MOSFET for future nanoscale circuits simulators. Therefore, the neural modeling provides practical insight into quantum effects in ultra-small electronic devices without the uncertain accuracy or meticulous tuning effort that face more rigorous quantum models. The neural modeling is a step towards a new generation of simulation tools that will allow device engineers to explore new classes of electronic devices.

4. ANN-Based Approach to Design Nano-CMOS Circuits

In order to study the impact of ANNs-based modeling on the design of nano-CMOS circuits, we propose, in this section, the simulation of the nanoscale CMOS inverter (Figure 7) which is considered as the most basic element of digital VLSI circuits [36]. However, as the CMOS technology enters the nanoscale regime, quantum effects and SCE become more and more important and consequently a quantum-mechanical simulation of a DG MOSFET is necessary. In this section, using the MLPANN-based model, we have simulated an inverter gate. The purpose of this simulation is to study the evolution of nano-CMOS inverter transfer curves (V_{out}–V_{in}). Each inverter consists of two DG MOSFETs.

The I-V characteristics of each DG MOSFET were predicted using the ABM MLPANN-based model (Analog Behavioral Modeling) as it is shown in Figure 8. So, the proposed ANN-base model can be used as a subcircuit in SPICE software. A DG MOSFET can be modeled as a voltage-controlled current source between drain and source where the current value can be calculated with the neural network equations and a null current source between gate and source (because the gate current of the MOS transistor is always neglected).

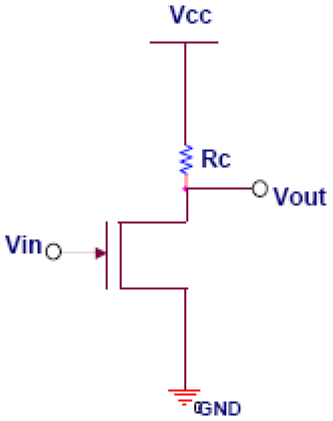


Figure 7. Nanoscale CMOS inverter gate.

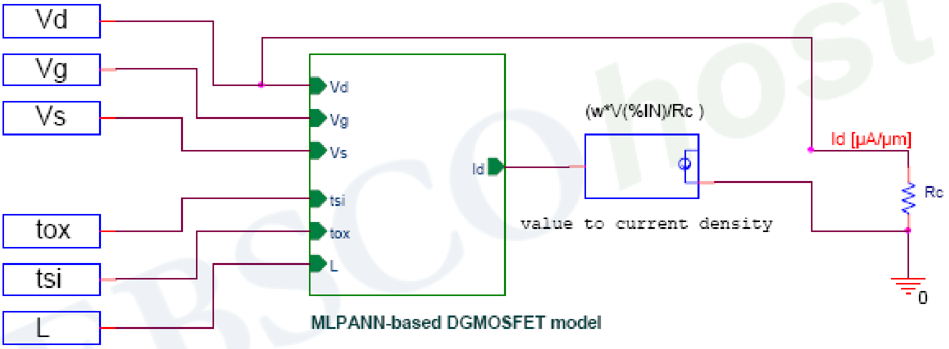


Figure 8. Detailed diagram of the ABM-based Pspice macromodel for MLPANN-based DG MOSFET model.

Figure 9 shows that a good agreement between numerical and predicted results was found for short channel length DG MOSFETs. The simulations were carried out for wide range of nanoscale channel lengths, from *L=10nm to L=50nm*, where the founded (Rout Mean Square) RMS errors are within 5**%.** This last observation shows the applicability of MPLANN technique to study the nanoscale CMOS-based circuits.

The Pspice input/output signals of our MLPANN-based inverter gate are shown in Figure 9. Figure 10 shows the transfer curves for the designed CMOS inverters for different channel lengths. It is clear that as the gate length decreases, the short-channel-effects become more and more serious. Consequently they degrade the performance of the nano-CMOS inverters. In practice, a high inverter voltage gain can provide a high transition speed of nano-CMOS inverter and better performance of digital operations.

It is important to note that our CNT-MOSFET model can be realistically extended to other practical circuits with many devices like nano-current source and memory cells.

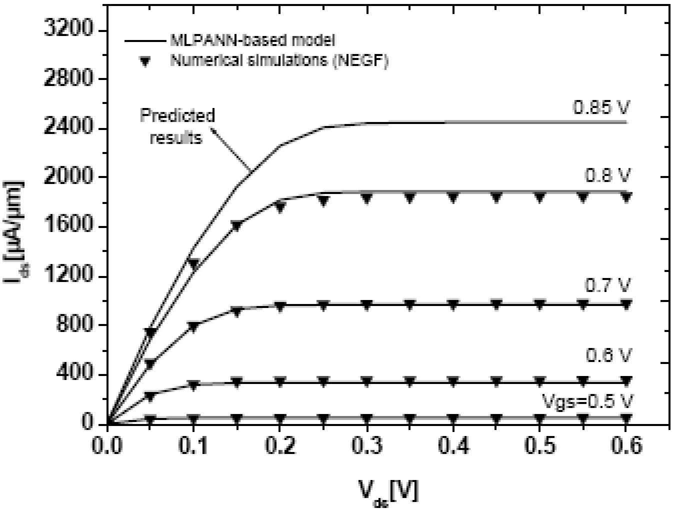


Figure 9. Current-voltage characteristics (I-V). Solid lines are from MPLANN-based model. Symbols are the results from the numerical simulations (NEGF) for L=30nm.

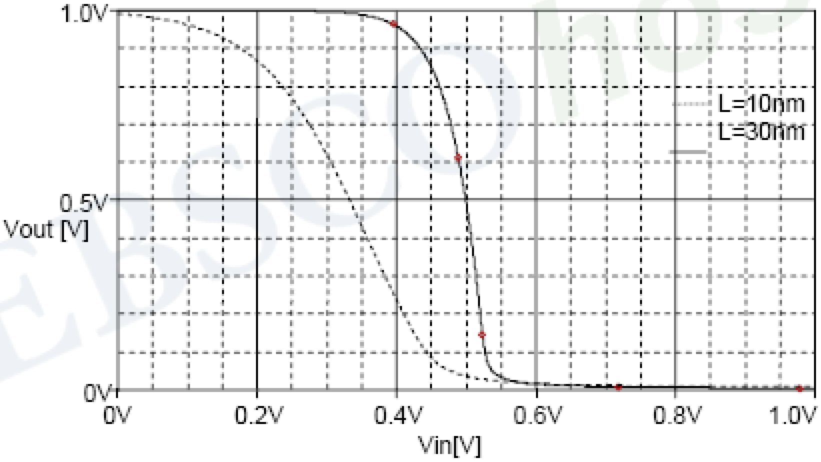


Figure 10. Transfer curves predicted with MLPANN-based model versus channel lengths.

Conclusions and Perspectives

In this chapter, we showed the applicability of the Artificial Neural Networks (ANNs) approach to the nanoscale CMOS circuits' simulation problem. The performance of state-of-the-art ANN-based models in terms of accuracy and computational time has been reviewed and compared, and their relevance to the nanoscale CMOS-based devices modeling was discussed. The use of numerical or experimental results can be used to build the required database in order to optimize the artificial neural network structure. The ANN was trained to the current–voltage characteristics off-line using a training algorithm. After optimization, the weight set of the ANN was imported into circuits' simulator where no additional optimization was required to the model. Also, it is shown that the ANN-based models are much faster than NEGF simulations. The encouraging comparisons between numerical results and ANN-based simulations have indicated that the proposed ANN-based approaches are particularly suitable to be incorporated in electronic devices simulators to study the nanoscale CMOS circuits.

REFERENCES

[1] Djeffal, F., Chahdi, M., Benhaya, A. & Hafiane, M. L. (2007). *Solid State electronics*, *Vol. 51*, 26-34.

[2] Djeffal, F., Abdi, M. A., Dibi, Z., Chahdi, M. & Benhaya, A. (2008). *Materials Sci and Eng*: B, *Vol.147*, 239-244.

[3] Kranti, A., Chung, T. M., Flandre, D. & Raskin, J. P. (2005). *Semicond Sci. Technol*, *Vol. 20*, 423–429.

[4] El-Hamid, H. A., Roig, J. & Iniguez, B. (2007). *Solid State electronics*, *Vol 51*, 414-422.

[5] Ren, Z., Venugopal, R., Datta, S., Lundstrom, M., Jovanovic, D. & Fossum, J. (2000). IEDM *Tech Dig*, 715–718.

[6] Taur, Y., Liang, X., Wang, W. & Lu, H. (2004). *IEEE Electron Dev Lett*, *25(2)*, 107-109.

[7] Baccarani, G. & Reggiani, S. (1999). *IEEE Trans Electron Dev*, *46(8)*, 1656–1666.

[8] Datta, S. (2000). *Superlattices Microstruct*, *28(4)*, 253–78.

[9] Djeffal, F., Guessasma, S., Benhaya, A. & Chahdi, M. (2005). *Semicond Sci Technol, 20*, 158–64.

[10] Benjamín Iñiguez, T. A. (2006). Fjeldly, Antonio Lázaro, François Danneville, and M. Jamal Deen, *IEEE Trans Electron Dev*, *53(9)*.

[11] Xiong, S., King, T. J. & Bokor, J. (2005). *IEEE Trans Electron Dev*, *52(8)*, 18591867.

[12] Taur, Y., Liang, X., Wang, W. & Lu, H. (2004). *IEEE Electron Device Letters*, *25(2)*, 107-109.

[13] Ortiz-Conde, A., Garcia-Sanchez, F. J. & Muci, J. (2005). *Solid-State Electronics*, *49*,.

[14] Jimenez, D., Iniguez, B., Sune, J., Marsal, L. F., Pallares, J., Roig, J. & Flores, D. (2004). IEEE *Electron Device Letters*, *251(8)*, 571-573.

[15] Jimenez, D., Saenz, J. J., Iniguez, B., Sune, J., Marsal, L. F. & Pallares, J. (2003). *Journal of Applied Physics*, *94(2)*, 1061-1068.

[16] Toriumi, A., Iwase, M. & Yoshimi, M. (1988). *IEEE Trans. Electron Devices*, *35*, 999-1003.

[17] Ge, L. & Fossum, J. (2001). *IEEE Trans Electron Dev*, *48(9)*, 2074–80.

[18] Abd Elhamid, H. (2007). Compact modeling of multiple gate MOS devices, *Phd thesis*, University of Rovira i Virgili, Tarragona-Spain.

[19] Venugopal, R., Paulsson, M., Goasguen, S., Datta, S. & Lundstrom, M. S. (2003). *J Appl Phys*, *93*, 5613–2625.

[20] A. Svizhenko, MP. Anantram, TR. Govindan, B. Biegel, R. Venugopal, *J Appl Phys*, *91(4)*, 2343–54.

[21] Djeffal, F., Dibi, Z., Hafiane, M. L., Arar, D. (2007). *Materials Science and Engineering* C, *27*, 1111–1116.

[22] Topping, B. H. V. & Bahreininejad, A. (1997). Neural computing for structural mechanics. Edinburgh (UK): *Saxe-Coburg Publications*.

[23] Taylor, J. G. (1996). *Neural networks and their applications*. West Sussex (UK): John Wiley & Sons Ltd.

[24] Gallant, A. R. & White. H. (1992). On learning the derivatives of an unknown mapping with multilayer feed forward networks, *vol. 5*, *Elsevier Science*.

[25] Bishop, C. M. (1997). *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford.

[26] Nabney, I. T. (1999). *Efficient Training of RBF Networks for Classification*, Aston University, Birmingham.

[27] Babuska, R. (1998). *Fuzzy Modeling for Control*, Kluwer Academic Publishers, Norwell, MA.

[28] Jang, J. S. R., Sun, C. T. & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*, Prentice Hall, *19*, 510-514.

[29] Takagi, T. & Sugeno, M. (1985). *IEEE Transactions on Systems*, *15*, 116-132.

[30] Chiu, S. (1994). *Intelligent and Fuzzy Systems*, *2*, 762-767.

[31] Jang, J. S. R., Sun, C. T. (1995). Proc. of the IEEE, Special Issue on *Fuzzy Logic in Engineering Applications*, *83*, 378-406.

[32] Meijer, P. (1990). IEEE Trans. *Circuits and Syst*, *37*, 335-346.

[33] Zhang, L., Xu, J., Yagoub, M. C. E., Ding, R. T. & Zhang, Q. J. (2005). *IEEE Trans. Microw. Theory Tech, 53, 2752–2767.*

[34] Wood, J., D. E. Root, & Tufillaro, N. B. (2004). IEEE Trans. *Microw. Theory Tech.*, *52*, 2274–2284.

[35] Bandler, J. W., Georgieva, N., Ismail, M. A., Rayas-Sánchez, J. E. & Zhang, Q. J. (2001). IEEE Trans. *Microw. Theory Tech.*, *49*, 67-79, 2001.

[36] Yaur, Y. (1998). TH. Ning, *Fundamentals of modern VLSI devices*, Cambridge (UK), Cambridge University Press.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 8

ARTIFICIAL NEURAL NETWORKS IN CHROMATOGRAPHY AND SPECTROSCOPY

***Iva Rezić*[1*] *and Tomislav Rolich*[2]**

[1]Laboratory of Analytical Chemistry, Department of Applied Chemistry,
Faculty of Textile Technology, University of Zagreb, Croatia
[2]Department of Fundamental Natural and Engineering Sciences, Faculty
of Textile Technology, University of Zagreb, Croatia

ABSTRACT

The purpose of this chapter is to demonstrate the application of artificial neural networks in modern chemical investigation, mainly in chromatography and spectroscopy.

The first part of the chapter presents a fast and simple procedure for estimation of steel materials corrosion in artificial sweat solution with usage of artificial neural networks (ANN). For the purpose of the mathematical modeling, optical emission spectroscopy experimental data were used to train and test the most appropriate model of artificial neural networks. Evaluation was performed by comparing the experimental data and values estimated by ANN and by calculating the correlation coefficient and mean absolute error. It was concluded that ANN can be easily applied for estimation of corrosion processes.

The second part of this chapter presents a combination of artificial neural networks and genetic algorithms (GA) in optimization of thin layer chromatographic separation of seven components from their mixture. As a goal of optimization, a resolution factor was calculated for different mixture model solutions. Afterwards the prediction of the same parameter was performed by ANN and GA, and very good correlation between predicted and calculated data was observed. Therefore it can be concluded that the developed combination of ANN and GA may be successfully used in many different chromatography investigations, like high performance liquid chromatography (HPLC), ion chromatography (IC), gas chromatography (GC) and other similar techniques.

[*] Corresponding author: Faculty of Textile Technology, Prilaz baruna Filipovića 28a, 10000 Zagreb, Croatia, e-mail: iva.rezic@ttf.hr, iva_rezic@net.hr, phone: ++385 1 3712 593, fax: ++385 1 3712 599

Based on our current results we expect that artificial neural networks and their combination with other methods (such as genetic algorithms) will be implemented in many different chemical and analytical applications in a near future.

Keywords: ANN, GA, chromatography, spectroscopy, optimization.

1. Part A. ANN in Estimation of Steel Corrosion

In this part of the chapter the application of ANN in estimation of steel corrosion in artificial sweat solution is presented.

1.1. Introduction

Artificial neural network (ANN) represents method which is inspired by biological neural system that can learn through examples [1 - 3]. The efficiency of ANN and the success in obtaining a reliable and robust network depends on chosen process variables and tested set of data [4-6]. ANN is one of many different numerical methods that can be applied in chemistry, chemical industry and in monitoring or optimizing chemical and industrial processes [7-10]. Artificial neurons mimic the function of biological neural cells in the living organisms [11]. The neuron cell consists of cell body, axon and dendrites, as it is presented in Figure 1.

Neural networks were developed in 1982 by J. Hopfield [11, 12]. After 1980s, neural networks slowly started being incorporated in different applications, while today ANN is important not only for the scientific investigations but also in diverse areas like art, law or economics.

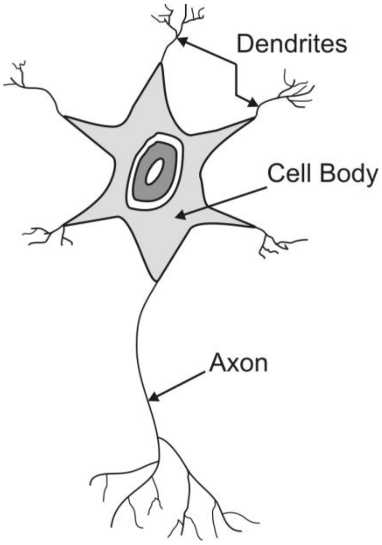


Figure 1. Schematic drawing of a neuron cell.

Table 1. Input data for training of the neural network [47]

Sample	M_1, w%	M_2, w %	M_3, w%	Time / days
1	0.50	0.43	0.33	7, 14 and 28
2	0.51	0.30	0.27	7, 14 and 28
3	0.57	0.34	0.30	7, 14 and 28
4	-	0.16	0.25	7, 21 and 28
5	-	0.16	0.28	7, 21 and 28
6	-	0.17	0.32	7, 21 and 28

Table 2. Output data for training of the neural network [47]

Sample	Time / days	$\mu g(M_1)/ cm^2$	$\mu g(M_2)/ cm^2$	$\mu g(M_3)/ cm^2$
1	7	0.85	0.14	0.35
	14	1.01	0.25	1.17
	28	1.22	0.25	1.58
2	7	0.38	0.09	0.39
	14	0.46	0.16	0.87
	28	0.71	0.36	1.54
3	7	0.39	0.08	0.34
	14	0.49	0.16	0.97
	28	0.60	0.32	2.06
4	7	-	0.07	0.70
	21	-	0.18	1.04
	28	-	0.28	1.12
5	7	-	0.04	0.38
	21	-	0.15	0.57
	28	-	0.20	0.61
6	7	-	0.06	0.88
	21	-	0.22	1.12
	28	-	0.31	1.14

The estimation of the corrosion process of steel material is important for different scientific and industrial purposes. In this part of the chapter the idea of applying the artificial neural networks for prediction of steel corrosion process in sweat solution is presented. As model samples, guitar strings were used since such processes directly influence human health [13-20]. Artificial neural networks (ANN) with one hidden layer were used for computer simulation of the behavior of steel materials. Such network was chosen since it has shown to be a promising tool for estimation of corrosion of wide variety of different materials [21-41]. Corrosion of strings on musical instruments causes leaching of metal ions that can be absorbed by the human skin. This influences human health significantly [42 - 45]. Therefore the goal of this chapter is to present the usefulness of application of ANN for prediction of steel corrosion. Since ANN have to be trained with known data collected from experiments, the data hereby presented were obtained after optical emission spectroscopic measurements of concentration of leached metal ions after immersion of steel strings in sweat solution, as it was presented in our previous research papers [46, 47].

1.2. Experimental Procedure

For the purpose of this investigation, ANN consisted of three layers: the first layer with input variables, a hidden layer and the last with output variables. Model samples investigated were electric guitar steel strings (E6, A5, D4, g3, h2 and e1). The samples were put into a sealed tube with 10 ml of sweat at 37 °C for 168 h to 672 h [47]. The amounts of three randomly chosen metal ions from steel (M_1, M_2 and M_3) released in the corrosive solutions were used as the input and the output data for training of the ANN (Tables 1, 2).

Samples were divided into two groups: one-component strings marked as samples 1 – 3, and two-component strings (marked as samples 4 – 6). For each group of samples, ANN were specially trained, tested and optimized [47].

Chemical analysis of samples was performed by optical emission spectroscopy (OES). It was shown that network which is designed for one group of samples cannot effectively be applied for another type of samples [47].

Radial basis network consisting of two layers was proposed for solving a given problem. The most appropriate networks were hidden radial basis layer with 6 neurons, and an output linear layer with 2 or 3 neurons (Figure 2).

The transfer function used in the hidden layer was:

$$f(x) = e^{-x^2} \tag{1}$$

All calculations were carried out with MATLAB software version 7.8.0.347 (R2009a) and Neural Network Toolbox version 6.0.2.

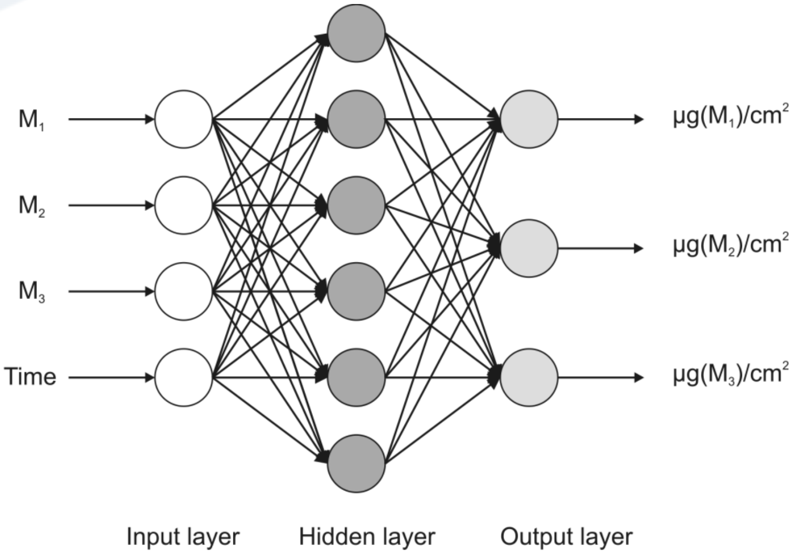


Figure 2. A conceptual diagram of applied artificial neural network.

1. 3. Results and Discussion

Artificial neural network (ANN) is a method which may be applied for solving different problems in diverse research fields related to materials corrosion. In this work, it was trained to predict the corrosion process. The network was tested by comparing the calculated data with known experimental ones.

1.3.1. Training the ANN

As was previously emphasized, the samples were divided into two separate groups according to their morphology and corrosion behavior. It will be shown that each group of samples demands specially trained ANN, otherwise good results obtained by ANN for one group of samples could not be applied for another, different kind of sample strings. Therefore the samples were grouped according to their corrosion behavior (linear or parabola corrosion rates) [46].

For prediction of corrosion of one component samples (strings 1, 2 and 3), the results from the first, the second and the fourth week were used for training the ANN. In addition, for the purpose of the ANN training, the corrosion experiments of two component samples (strings 4, 5 and 6) were performed for one, three and four weeks of immersion in the sweat solution. After collecting the results presented in Tables 1 and 2, those data were used for training the ANN. Afterwards the comparison between predicted and measured data was performed.

1.3.2. Testing the ANN

The mean squared error (MSE) was used to measure the performance of the network:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} e_i^2 = \frac{1}{n}\sum_{i=1}^{n}(y_{i,\exp} - y_{i,pred})^2 \quad (2)$$

The results of data used for testing the accuracy of the chosen network are presented in Table 3.

Table 3. Predicted and measured data for testing of the neural network [47]

	μg(M_1)/ cm^2 measured	μg(M_1)/ cm^2 predicted	μg(M_2)/ cm^2 measured	μg(M_2)/ cm^2 predicted	μg(M_3)/ cm^2 measured	μg(M_3)/ cm^2 predicted
1	1.07	1.08	0.25	0.29	1.39	1.59
2	0.55	0.59	0.25	0.26	1.39	1.37
3	0.55	0.56	0.23	0.24	1.64	1.60
4	-	-	0.13	0.12	0.93	0.90
5	-	-	0.11	0.11	0.55	0.53
6	-	-	0.17	0.15	1.11	1.04

*samples 4, 5 and 6 did not contain any metal ion one.

Table 4. Calculated errors and relative errors of trained and tested network [47]

	Δ = Predicted – Measured values			Δ, % = ((Predicted–measured) / measured values) ×100		
	M_1	M_2	M_3	M_1	M_2	M_3
1	0.01	0.04	0.20	0.93	16.00	14.39
2	0.04	0.01	-0.02	7.27	4.00	-1.44
3	0.01	0.01	-0.04	1.82	4.35	-2.44
4	-	-0.01	-0.03	-	-7.69	-3.23
5	-	0.00	-0.02	-	0.00	-3.64
6	-	-0.02	-0.07	-	-11.76	-6.31

The chosen network for two component samples (4, 5 and 6) contained 3 neurons in the input, 6 neurons in the hidden and 2 neurons in the output layer. In addition, the chosen network for one component samples (1, 2, and 3) consisted of 4 neurons in the input layer, 6 neurons in the hidden and 3 neurons in the output layer. After training of the networks, the errors and relative errors were calculated for each sample separately (Table 4).

Since estimated and experimental data were in a very good agreement for the majority of the elements, it was concluded that the ANN were properly trained. Therefore this network might be successful in estimation and prediction of corrosion processes of steel samples.

In further text the way of prediction of metal ions leaching, and prediction of steel corrosion by ANN will be described.

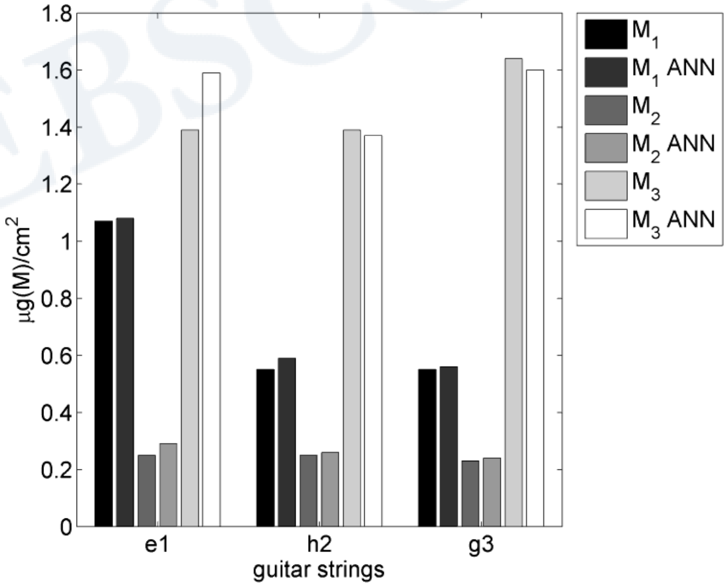


Figure 3. Very good correlation between measured data and predicted (ANN) data of metals for samples 1 to 3 obtained by properly trained network (4 neurons in the input layer, 6 neurons in the hidden and 3 neurons in the output layer).

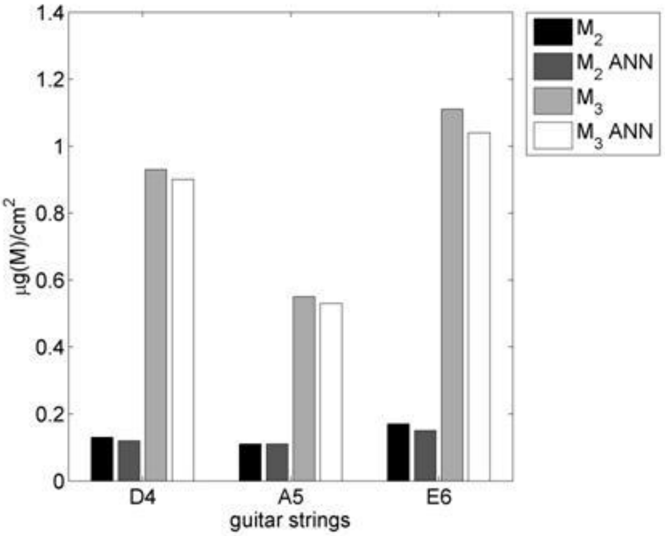


Figure 4. Very good correlation between measured data and predicted (ANN) data of metals for samples 4 to 6 obtained by properly trained network (3 neurons in the input, 6 neurons in the hidden and 2 neurons in the output layer).

1.3.3. Prediction of the corrosion process by ANN

After training and testing the best ANN model, all experimental points were predicted by developed neural network and compared to measured data. In majority of cases, correlation data between measured and predicted values were in a very good agreement, as it can be seen in Figures 3 and 4.

As it can be seen from figures 3 and 4, the values of ANN predicted data were in a very good agreement with the experimental data. Based on the results obtained, it can be concluded that it is possibly to estimate the corrosion behavior for 2 different groups of steel samples by ANN. Nevertheless, a special attention has to be given to the training and testing of the neural networks. It has to be emphasized that each group of different samples demanded specially trained and tested networks, and that the best ANN for one group of samples does not give the best results when applied on some other, different group of samples. The same network might be successfully applied only on similar samples of same characteristics.

2. Part B: Application of ANN and Genetic Algorithms in Optimization of Chromatography Separation of Amino Acids

Thin layer chromatography (TLC) is a very useful analytical method for separation and identification of different amino acids from their mixture in different samples, as well as in protein composition. Amino acids are monomer units in proteins and can be found in numerous natural products. Therefore their determination is very important for various scientific and industrial purposes. In this part of the chapter the combination of genetic algorithm and artificial neural network as a useful combination for prediction of separation in thin layer chromatography is proposed.

2.1. Introduction

Special benefit of all chromatographic procedures is that they can separate and determine components in their mixture at the same time. The problem with chromatography occurs when two or more components have similar physical and chemical properties, because then the peaks of the components will not be separated and it is not going to be possible to detect individual components. More components per sample there are, more complex situations can occur. Typical chromatogram of a mixture of components obtained after thin layer chromatography is shown in Figure 5.

Optimization in the TLC is the process of achieving the best analytical conditions with the goal of separation of particular components from their mixture. This can be achieved by changing the stationary or the mobile phases, after which the modeling of the experimental data follows. In this respect many sophisticated mathematical models can be applied: nonlinear mapping [48], partial least square analysis [49], principle component analysis [50], genetic algorithms [51] and artificial neural networks [52, 53]. Some of the models have been already tested for different chromatographic systems. S. Babić et. al. have proposed a genetic algorithms (GA) for optimization of mobile phase [54] and sample preparation method (microwave assisted extraction) in thin layer chromatography [55]. Artificial neural networks (ANN) were successfully applied in liquid chromatography [56], miscellar electro kinetic chromatography [57], and ion chromatography [58]. Nevertheless, according to our findings, the combination of genetic algorithms with artificial neural networks has not yet been investigated for any chromatographic technique. Only few attempts in the field of spectroscopic investigation have occurred recently [59].

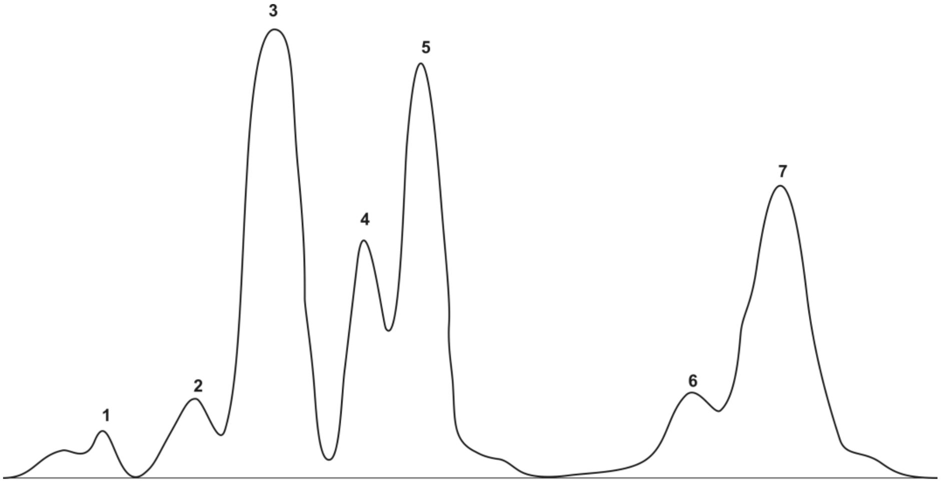


Figure 5. Typical thin layer chromatogram of seven components in their mixture, identification and visualization by Video Denzitometer.

Combination of GA and ANN is a promising tool which will probably find its application in many fields of chemical spectroscopic and chromatographic investigations in a near future [60-64]. A genetic algorithm (GA) is a search technique used in finding solutions to optimization problems. Genetic algorithms are computer simulation in which a population of candidate solutions evolves toward better solutions. The evolution starts from a population of randomly generated individuals. In each generation, the fitness of individual is evaluated, stochastically selected based on the fitness and modified to form a new population. The new population is then used in the next iteration of the algorithm.

Genetic algorithms combined with ANN can find application in bioinformatics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields [5, 65]. When a GA input is presented to the neural network, a corresponding desired or target response will be set at the output. An error is than composed from the difference between the desired response and the system output [66 - 68]. There are many criteria which can be applied for the TLC optimization: the resolution factor, the separation factor or multi-spot response function [68 - 70]. When the resolution factor (R_S) is chosen as the goal of optimization, the principle of optimization is the relationship between the resolution factor and the composition of ternary mobile phase [71, 72]. Many attempts have been made with the goal of developing the most optimal chromatographic parameters for separation of different compounds by thin layer chromatography [73- 78]. As was previously emphasized, the problems with multi component analysis occur when two or more components have chromatograms (or spectra in spectroscopy) which overlay one another. In the past, chemists had to perform hundreds of analysis in order to discover the best analytical conditions for separation of such complex samples by methods of trial and mistake. Today this is much more elegantly achieved by modern mathematical modeling, for which GA and ANN are extremely helpful.

The goal of this part of the chapter is to present a combination of genetic algorithms (GA) and artificial neural networks (ANN) in optimization of TLC. It shows big potential for analysis of complex samples. In our previous work we have developed a GA and GA-ANN mathematical methods for prediction and optimization in separation of seven amino acids from their mixture [72]. The concept of that work will be shortly described in this part of the chapter.

2.2. Experimental

Seven amino acids were analyzed by TLC by different mixtures of ternary mixtures of mobile phases (solvents A, B and C). From the chromatograms obtained, the resolution factors (marked with numbers 1 – 6 in the following text) were calculated for each par of samples according to the equation (3):

$$R_S = \frac{l_0 \cdot (R_{F2} - R_{F1})}{0.5 \cdot (w_1 + w_2)} \tag{3}$$

where the l_0 presents the constant distance of 8 cm, R_{F1} and R_{F2} are experimental R_F parameters (calculated as ration between distances of sample and the solvent) of two neighboring spots, and w_1, w_2 are their spot widths.

2.3. Results and Discussion

The goal of the optimization was to determine the optimal composition of the ternary mobile phase which consisted of three solvents (marked as A, B and C). The tested ranges of solvents were 40–100%, 0–60% and 0–60% for A, B and C solvents, respectively. The constraints were based on preliminary experimental data. The ternary diagram of the experimental design is shown in Figure 6, and the parameters calculated from experimental data are presented in table 5.

The obtained parameters presented in table 5 were obtained from experimental data and were used for mathematical modeling. For the optimization purposes the data were randomly divided into three groups: for ANN training, validation and testing. For training mixtures of solvents used were a-j, l, m, o, r - v, and for monitoring and testing mixtures k, n, p and z (Table 5). ANN model contained a network of neurons with an input layer, two hidden layers and an output layer. Data at the output layer were approximation of the experimental values presented in Table 5.

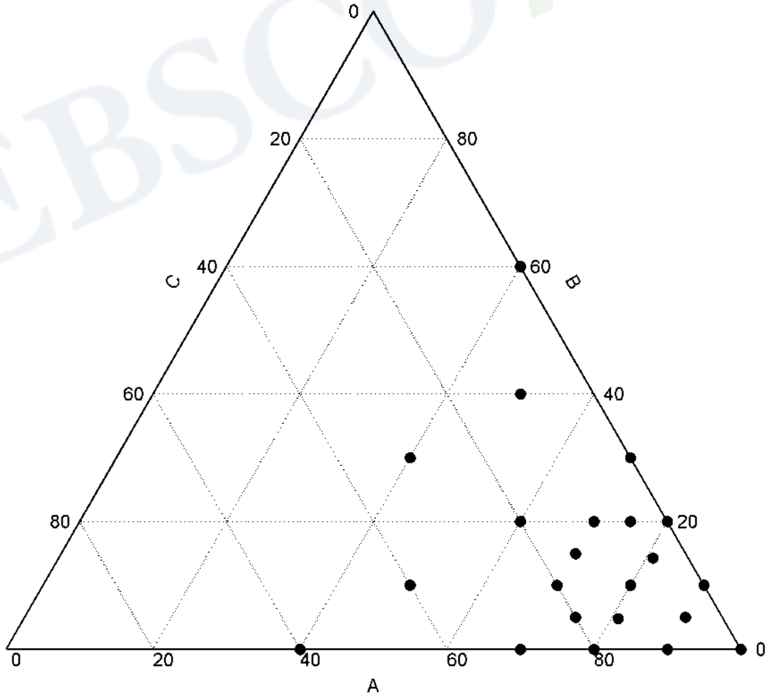


Figure 6. The ternary diagram of experimental design with solvents A, B and C.

Table 5. The experimental data calculated for 22 different ternary mixtures

Combination	Solvent A	Solvent B	Solvent C	Combination	Solvent A	Solvent B	Solvent C
a	0.05	0.03	0.03	l	0.00	0.32	0.09
b	0.00	0.00	0.00	m	0.41	0.05	0.05
c	0.09	0.00	0.05	n	0.01	0.06	0.02
d	0.18	0.09	0.09	o	0.05	0.28	0.07
e	0.09	0.24	0.06	p	0.02	0.40	0.05
f	0.00	0.07	0.03	r	0.03	0.16	0.05
g	0.03	0.10	0.01	s	0.00	0.08	0.05
h	0.18	0.08	0.67	t	0.00	0.09	0.02
i	0.03	0.34	0.94	u	0.01	0.51	0.84
j	0.10	0.94	0.64	v	0.03	0.11	0.01
k	0.06	0.13	0.02	z	0.06	0.33	0.08

The approximation of R_S parameters was performed by polynomial equation with two variables (4):

$$f(x,y) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 \tag{4}$$

For the sum of volume ratios is true:

$$A + B + C = 100 \tag{5}$$

therefore the following assumption was made:

$$x = \frac{B}{A} \tag{6}$$

$$y = \frac{C}{A} \tag{7}$$

The maximum value of function $f(x,y)$ will be for $(x_{\max}, y_{\max})$ thus from (5), (6) and (7) follows:

$$C = \frac{100 \cdot y_{\max}}{1 + x_{\max} + y_{\max}} \tag{8}$$

$$B = \frac{100 \cdot x_{\max} - C \cdot x_{\max}}{1 + x_{\max}} \tag{9}$$

$$A = 100 - B - C \tag{10}$$

which will give the best volume ratios of tested ternary mixtures of A, B and C for optimal R_S values. The calculations were carried out with program MATLAB version 7.8.0.347 (R2009a) and Neural Network Toolbox version 6.0.2. Calculated polynomial functions which approximate the calculated data for *x*, *y*, R_SI, R_SII, R_SIII, R_SIV, R_SV and R_SVI are presented in Table 6, where R_SI, R_SII, R_SIII, R_SIV, R_SV and R_SVI present resolution factors of seven components.

The graphical displays of calculated functions which approximate R_S values are presented in Figure 7.

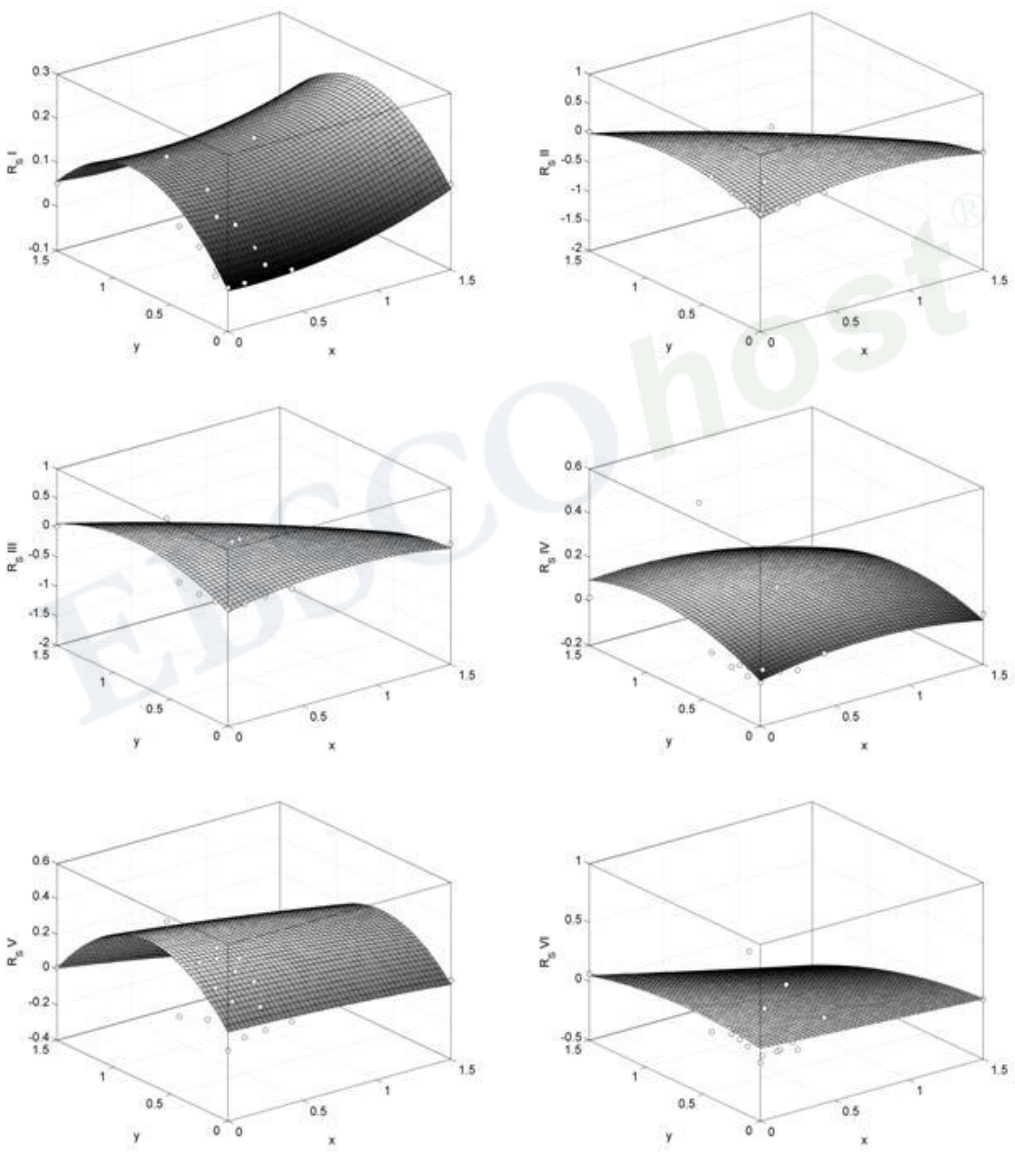


Figure 7. The graphical displays of functions calculated for six R_S parameters.

Table 6. The coefficients of polynomial functions which approximate R_S parameters

Function	a_0	a_1	a_2	a_3	a_4	a_5
$f_1(x,y)$	-0.0075	-0.0187	0.4210	-0.0193	0.0524	-0.2515
$f_2(x,y)$	-0.0732	0.6215	0.8971	-0.7435	-0.3908	-0.5693
$f_3(x,y)$	-0.0544	0.5336	1.1045	-0.7612	-0.3417	-0.6796
$f_4(x,y)$	0.0058	0.1404	0.3104	-0.0581	-0.0962	-0.1661
$f_5(x,y)$	0.1049	-0.0248	0.8708	-0.0723	-0.0194	-0.6231
$f_6(x,y)$	0.1224	-0.0814	0.1660	-0.1468	0.0056	-0.1419

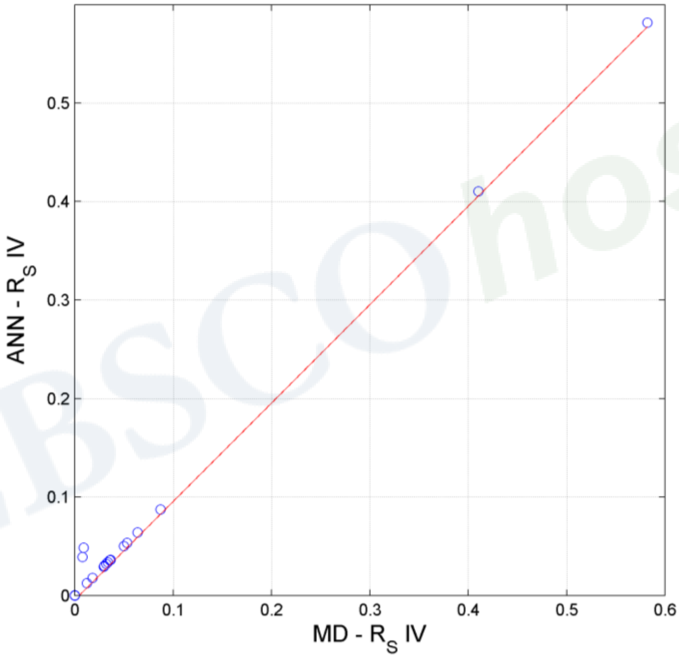


Figure 8. Correlation between experimental data and data obtained by prediction with ANN model with input layer of 3 neurons, (9-9) in two hidden layers and 6 neurons in output layer

The quality of the developed ANN procedure was estimated by comparing the estimated and the experimentally obtained data. Very good correlation was observed for two different networks: the first with 3 neurons in input layer, 9 neurons in each hidden layers and 6 neurons in output layer, and the second with 3 neurons in input layer, 18 neurons in each hidden layers and 6 neurons in output layer. An example of the correlation between ANN outputs and calculated experimental data for neural network with 3 neurons in input layer, 9 neurons in each hidden layers and 6 neurons in output layer is presented in Figure 8.

From the Figure 8 it is obvious that there is a very good relationship between the measured and the calculated data. Therefore the correlation factors between ANN outputs and calculated experimental data for neural networks 3-9-9-6 were calculated. They were in ranges between 0.90 and 0.99.

After designing the ANN model, the optimum of the separation was calculated by genetic algorithm (GA). The calculation was performed by Genetic Algorithm and Direct Search Toolbox (MATLAB version 7.8.0.347 (R2009a)). The optimal percentages of three solvents were: 55% of component A, 19% of component B and 26% of component C. From the presented results it can be concluded that coupling of GA to ANN can be applied as a very useful combination in thin layer chromatographic separation and optimization.

It is likely to be expected that the coupling of those two methods will find even bigger role in future applications in different chromatographic techniques (HPLC, LC, GC, IC and others), as well as in spectroscopy.

Conclusions

Artificial neural networks are becoming unavoidable method in different research fields. This chapter presented its implementation in spectroscopy and chromatography, both being the most frequent analytical techniques which are widely applied all over the world. The first part of the chapter presented an innovative approach of predicting the corrosion process based on the optical emission spectroscopy data. The second part was focused on combining artificial neural networks to genetic algorithms with the goal of calculating the best conditions for separation of seven components (amino acids) from their mixture by thin layer chromatography. Both of the chapter parts are evidences of applicability of artificial neural networks in wide variety of problems in modern science.

References

[1] Aleboyeh, A; Kasiri, MB; Olya, ME; Aleboyeh, H. *Dyes Pigm*, 2008, 77, 288-294.

[2] Gob, S; Oliveros, E; Bossmann, SH; Braun, AM; Guardani, R; Nascimento, CAO, *Chem Eng Proces,* 1999, 38, 373-382.

[3] Stegemann, JA; Buenfeld, NR. *J Haz Mat,* 2002, 90, 1183-1187.

[4] Despange, F; Massart, DL. *Analyst*, 1998, 123, 157-178.

[5] Massart, DL; Vandeginste, BGM; Buydens, LMC; De Jong, S; Lewi, PJ; Smeyers-Verbeke, J. *Handbook of Chemometrics and Qualimetrics: Part A;* Elsevier, 1987, Amsterdam,

[6] Massart, DL; Vandeginste, BGM; Deming, SN; Michotte, Y; Kaufmann, L; *Chemometrics: a textbook*; Elsevier, 1988, Amsterdam.

[7] Brereton, RG. *Chemometrics: Data analysis for the Laboratory and Chemical Plant,* 2003, John Wiley & Sons.

[8] Miller, JN; Miller, JC. *Statistics and Chemometrics for Analytical Chemistry*; Prentice Hall, 4th Edition, 2000, Dorchester.

[9] Eckschlager, K; Danzer, K. *Information Theory in Analytical Chemistry*; John Wiley & Sons, 1994, New York,

[10] Cox, DR; Reid, N. *The Theory of the Design of Experiments*; Chapman & Hall, 2000, Boca Raton.

[11] Zupan, J; Gasteiger, J. *Neural Networks in Chemistry and Drug Design*; Wiley- VCH, 2nd Edition, 1999, Weinheim.

[12] Hopfield, PJJ. *Natl. Acad. Sci*, USA 1982, 79, 2554-2558.

[13] Gambichler, T; Uzun, A; Boms, S; Altmeyer, P; Altenmüller, E. *Cont Derm*, 2008, 58, 217-222.

[14] Önder, M; Aksakal, AB; Öztaş, MO; Gürer, MA. *Int J Derm,* 1999, 38, 192-195.

[15] Baccouche, D; Mokni, M; Abdelaziz, AB; Osman-Dhahri, AB. *Annales de Derm Ven,* 2007, 134, 445-449.

[16] Gambichler, T; Boms, S; Freitag, M. *BMC Dermatol,* 2004, 4, 3-15.

[17] Smith, VH; Holmes, RC; Bedlow, A. *Clin Exp Derm*, 2005, 31, 129-156.

[18] Marshman, G; Kennedy, CTC; *Cont Derm,* 1992, 26, 134.

[19] Hausen, BM; Noster, U. *Cont Derm*, 1988, 14, 244-245.

[20] Rezić, I; Ćurković, L; Ujević, M. *Corr Sci*, 2009, 51, 1985-1989.

[21] Živko-Babić, J; Lisjak, D; Ćurković, L; Jakovac M. *Dent Mat*, 2008, 24, 18-27.

[22] Aliofkhazraeia, M; Sabour Rouhaghdam, A. *Mat Lett,* 2008, 62, 2192-2195.

[23] Brett, CMA; Ioanitescu, I; Trandafir, F. *Corr Sci*, 2004, 46, 2803-2816.

[24] Lind, SE. *Corr Sci*, 1972, 12, 749-755.

[25] Rao, TS; Nair, KVK. *Corr Sci*, 1998, 40, 1821-1836.

[26] Saji, VS; Choe, HC. *Corr Sci*, 2009, 51, 1658-1663.

[27] Seah, KHW; Chen, X. *Corr Sci*, 1993, 34 1841-1851.

[28] Brennenstuhl, AM; Gendron, TS; Cleland, R. *Corr Sci*, 1993, 35, 699-711.

[29] Ali, JA; Ambrose, JR. *Corr Sci*, 1991, 32, 799-814.

[30] Thair, L; Mudali, UK; Rajagopalan, S; Asokamani, R; Raj, B. *Corr Sci*, 2003, 45, 1951-1967.

[31] Pourbaix, M. *Corr Sci*, 1974, 14, 25-82.

[32] Lee, AK; Buehler, MG; Newman, DK. *Corr Sci*, 2006, 48, 165-178.

[33] Kear, G; Barker, BD; Stokes, KR; Walsh, FC. *Corr Sci*, 2005, 47, 1694-1705.

[34] Melchers, RE; Wells, T. *Corr Sci*, 2006, 48, 1791-1811.

[35] Melchers, RE. *Corr Sci*, 2003, 45, 923-940.

[36] Shifler, DA. *Corr Sci*, 2005, 47, 2335-2352.

[37] Rajasekar, A; Babu, TG; Pandian, SK; Maruthamuthu, S; Palaniswamy, N; Rajendran, A. *Corr Sci,* 2007, 49, 2694-2710.

[38] Wharton, JA; Barik, RC; Kear, G; Wood, RJK; Stokes, KR; Walsh, FC; *Corr Sci,* 2005, 47, 3336-3367.

[39] Wang, YZ; Beccaria, AM; Poggi G. *Corr Sci*, 1994, 36, 1277-1288.

[40] Song, G. *Corr Sci*, 2007, 49, 1696-1701.

[41] Hutchinson, AJ; Johnson, JB; Thompson, GE; Wood, GC; Sage, PW; Cooke, MJ. *Corr Sci* , 1993, 34, 1881-1889.

[42] Apostoli, P. *J Chrom B,* 2002, 778, 63 – 97.

[43] Elbetieha, A; Bataineh, H; Darmani, H; Al-Hamood, MH. *Tox Lett,* 2001, 119, 193-201.

[44] Blunden, S; Wallace, T. *Food Chem Tox,* 2003, 41, 1651-1662**.**

[45] Cho, H; Lee, J; Kwak, NJ; Lee, KH; Rha, S; Kim, YH; Cho, YY; Yang, KH; Kim, K; Lim, Y. *Erratum: Tox Lett* , 2003, 143, 323-330.

[46] Rezić, I; Ćurković, L; Ujević, M. *Mat Corr* doi: 10.1002/maco.200905368.

[47] Rolich, T; Rezić, I; Ćurković, L. *Corr Sci,* 2010, 52, 996-1002.

[48] Hudson, B; Livingstone DJ; Rahr, E. *J Comp Aid Mol Desc,* 1989, 3, 55-65.

[49] Cramer, RD; Paterson, DE; Bunce, JD. *J Am Chem Soc,* 1988, 110, 5959-5967.

[50] Camilleri, P; Livingstone, DJ; Murphy, JA; Manalack, DT. *J Comp Aid Mol Desc,* 1993, 7, 61-69.

[51] Leardi, R. *J Chrom A,* 2007, 1158, 226-233.

[52] Leardi, R. *J Chem*, 2001, 15, 559 – 569.

[53] McCann, MC; Defernez, M; Urbanowicz, BR; Tewari, JC; Langewisch, T; Olek, A; Wells, B; Wilson RH; Carpita, NC. *Plant Physiol*, 2007, 143, 1314-1326.

[54] Havel, J; Pena, EM; Rojas-Hernandez, A; Doucet JP; Panaye, A. *J Chrom A,* 1998, 793, 317-329.

[55] Babić, S; Horvat AJM; Kaštelan-Macan, M. *J Plan Chrom,* 2005, 18, 112-117.

[56] Babić, S; Horvat, AJM; Mutavdžić, D; Čavić D; Kaštelan-Macan, M. *J Plan Chrom,* 2007, 20, 95-99.

[57] Loukas, YL. *J Chrom A*, 2000, 904, 119-129.

[58] Baher, E; Fatemi, MH; Konoz E; Golmohammadi, H. *Microchim Acta*, 2007, 158, 117-122.

[59] Bolanča, T; Cerjan-Stefanović, Š; Regelja, M; Regelja H; Lončarić, S. *J Chrom A* , 2005, 1085, 74 – 85.

[60] Fei, Q; Wang, MLi, B; Huan, Y; Feng G; Ren, Y. *Chemom Int Lab Syst,* 2009, 97, 127-131.

[61] Niculescu, SP. *J Mol Struct,* 2003, 622, 71-83.

[62] Fatemi, MH; Jalali-Heravi M; Konuze, E. *Analyt Chim Acta*, 2003, 486, 101-108.

[63] Russell SJ; Norvig, P. *Artificial Intelligence a Modern Approach*, Pearson Education, 2003.

[64] Luke, S. *Essentials of Metaheuristics*, Zeroth Edition, Online Version 0.3, 2009, available at http://cs.gmu.edu/~sean/book/metaheuristics/.

[65] Haupt RL; Haupt, SE. *Practical genetic algorithms*, John Wiley & Sons, 2004.

[66] Haykin, S; Neural Networks, A. Comprehensive Foundation, Prentice Hall, USA, 1994.

[67] Siouffi, AM. *J Chrom*, 1991, 556, 81-94.

[68] Stahl, E. *Thin Layer Chromatography*; Springer-Verlag, New York, 1969.

[69] Sherma, J; Fried, B. *Handbook of Thin-Layer Chromatography*; Marcel Dekker, 1996.

[70] Wang QS; Yan, BW. *J Plan Chrom*, 1996, 9, 192 - 196.

[71] Issaq HJ; Seburn, KE. *J Liq Chrom*, 1989, **12,** 3121-3128.

[72] Rezić, I; Rolich, T. *J Plan Chrom, sent for publication.*

[73] Niestroj, A. *J Plan Chrom,* 2007, 20, 483-486.

[74] Sârbu1 C; Tiperciuc, B. *J Plan Chrom*, 2006, 19, 342 – 347.

[75] Djaković-Sekulić, T; Perišić-Janjić, N; Sârbu C; Lozanov-Crvenković, Z. *J Plan Chrom,* 2007, 20, 251-257.

[76] Perišić-Janjić, NU; Podunavac-Kuzmanović, SO. *J Plan Chrom*, 2008, 21, 135-141.

[77] Gumieniczek, A; Berecka, A; Matosiuk D; Hopkała, H. *J Plan Chrom*, 2007, 20, 261-265.

[78] Niestroj, A. *J Plan Chrom,* 2006, 19, 208-211.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 9

A NEW CANDIDATE LIST STRATEGY FOR ARCHITECTURE SELECTION IN ARTIFICIAL NEURAL NETWORKS

Cagdas Hakan Aladag
Department of Statistics, Hacettepe University, Ankara, Turkey

ABSTRACT

In the past decade, there have been many implementations in which artificial neural networks (ANN) successfully applied to many areas of science and engineering. One of these areas is time series forecasting. ANN method has been preferred to conventional time series forecasting models because of its easy usage and providing accurate results. In spite of the fact that ANN produces accurate forecasts in many time series implementations, there are still some problems with using this method. When ANN method is utilized to forecast time series, selection of the components of the method is a vital issue for obtaining good forecast values. These components such as architecture structure, learning algorithm and activation function have important effect on the performance of ANN. An important decision is the selection of architecture structure that consists of determining the numbers of neurons in the layers of a network. Therefore, to making a good choice for architecture selection, various approaches have been proposed in the literature. Aladag (2009a; 2009b) proposed a method based on tabu search algorithm to determine the best ANN architecture. He showed that accurate forecasts are obtained when the proposed method is employed for architecture selection. In his proposed algorithm, he utilized a candidate list strategy in which six architectures are examined. In this study, the tabu search algorithm proposed by Aladag (2009b) is tried to be improved by defining a new candidate list strategy in which only four architectures are examined. The beer consumption in Austria and the electricity consumption in Turkey time series are forecasted by ANN and the applicability of the proposed strategy is shown.

Keywords: Architecture selection; artificial neural networks; candidate list strategy; forecasting; tabu search; time series.

1. INTRODUCTION

In recent years, ANN method have been widely used for forecasting problem since the method can produce very accurate forecasts in various time series forecasting implementations (Gunay et al., 2007). Although ANN provides many advantages, there are still some problems with using this method (Aladag et al., 2010). ANN consists of some components such as architecture structure, learning algorithm and activation function and determining the right components for relevant problem is a vital issue in terms of obtaining accurate forecasts (Egrioglu et al., 2008). Determination of the best components, especially determining the best architecture and weights, remains a problem in ANN applications (Aladag, 2010b).

Many researchers from different disciplines have working on the problem of architecture selection in ANN. Architecture selection problem can be defined as determining the number of neurons in the layers of artificial neural network. In order to solve the architecture selection problem, various techniques have been proposed in the literature (Aladag, 2009). Some of them are constructive and pruning algorithm (Siestema and Dow, 1988), polynomial time algorithm (Roy et al., 1993), network information criterion (Murata et al., 1994), iterative construction algorithm (Rathbun et al., 1997), a method based on Box-Jenkins analysis (Buhamra et al., 2003), a method based on information entropy (Yuan et al., 2003), genetic algorithms (Dam and Saraf, 2006), the principle component analysis (Zeng et al., 2007), weighted information criterion (Egrioglu et al., 2008), a deletion/substitution/addition algorithm (Durbin et al., 2008), an architecture selection strategy for autoregressive seasonal time series (Aladag et al., 2008), and design of experiments (Balestrassi et al., 2009). Although these proposed methods are systematic approaches, the most preferred method is trial and error in the most of the ANN applications (Leahy et al., 2008).

In forecasting, an architecture which has the best forecasting performance criterion value is tried to be found. Forecasting performance is generally based on difference between the forecasts and the observations. Some of well known criteria such as root mean square error (RMSE) and mean absolute percentage error (MAPE) measure the difference between the forecasts and the original values. In other words, minimum value of the performance criterion points out the best architecture. Therefore, finding a good architecture can be considered as a minimization problem. In this optimization problem, each architecture is a solution and the performance criterion value produced by this solution is the objective function value. In order to solve the architecture selection problem, a systematic approach was proposed by Aladag (2009a; 2009b). In the method proposed by him, tabu search algorithm is used to solve the minimization problem.

Aladag (2009b) used a candidate list strategy including six neighbor solutions of the current solution. Thus, at each iteration, six neighbor solutions are created from the current solution. In this study, a new candidate list strategy for tabu search algorithm is proposed. In the proposed strategy, only four neighbor solutions are created from the current solution instead of six. Therefore, the tabu search algorithm proposed by Aladag (2009b) is tried to be improved by defining a new candidate list strategy in which only four architectures are examined. In the implementation, the beer consumption in Austria and the electricity consumption in Turkey time series are forecasted by utilizing both the candidate list strategy proposed by Aladag (2009b) and the strategy proposed in this study.

In the next section, ANN is briefly presented. Section 3 gives the tabu search algorithm proposed by Aladag (2009a). The proposed candidate list strategy is introduced in Section 4. The implementation is given in Section 5. Finally, Section 6 concludes the study.

2. Artificial Neural Networks

ANN method is briefly given in Aladag et al. (2010a) as follows:

> 'What is an artificial neural network?' is the first question that should be answered. Picton (1994) answered this question by separating this question into two parts. The first part is why it is called an artificial neural network. It is called an artificial neural network because it is a network of interconnected elements. These elements were inspired from studies of biological nervous systems. In other words, artificial neural networks are an attempt at creating machines that work in a similar way to the human brain by building these machines using components that behave like biological neurons. The second part is what an artificial neural network does. The function of an artificial neural network is to produce an output pattern when presented with an input pattern.

In forecasting, artificial neural networks are mathematical models that imitate biological neural networks. Researchers have used artificial neural networks methodology to forecast many nonlinear time series events in the literature (Ture and Kurt, 2006). The methodology consists of some elements. Determining the elements of the artificial neural networks issue that affect the forecasting performance of artificial neural networks should be considered carefully. Elements of the artificial neural networks are generally given as network architecture, learning algorithm and activation function.

One critical decision is to determine the appropriate architecture, that is, the number of layers, number of nodes in each layers and the number of arcs which interconnects with the nodes (Zurada, 1992). However, in the literature, there are not general rules for determining the best architecture. Therefore, many architectures should be tried for the correct results.

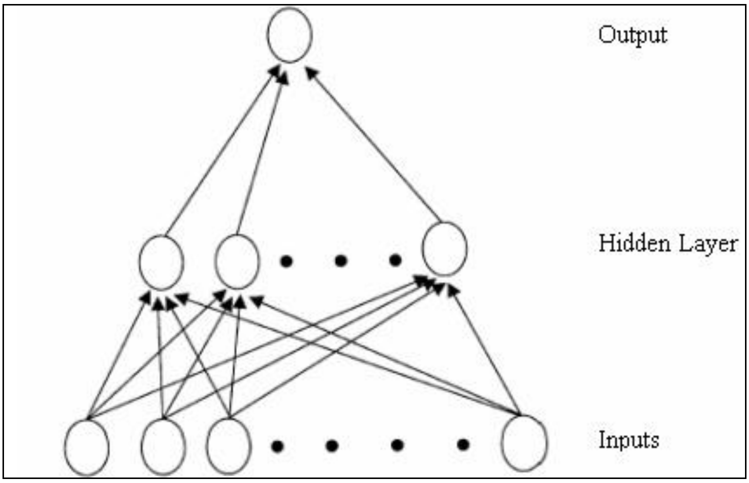


Figure 1. A broad feed forward neural network architecture.

The most preferred architecture structure in the applications is feed forward neural networks. The feed forward neural networks have been also used successfully in many studies for forecasting. In the feed forward neural networks, there are no feedback connections. Figure 1 depicts the broad feed forward neural network architecture that has single hidden layer and single output.

Learning of an artificial neural network for a specific task is equivalent to finding the values of all weights such that the desired output is generated by the corresponding input. Various training algorithms have been used for the determination of the optimal weights values. The most popularly used training method is the back propagation algorithm presented by Smith (2002). In the back propagation algorithm, learning of the artificial neural networks consists of adjusting all weights considering the error measure between the desired output and actual output (Cichocki and Unbehauen, 1993).

Another element of the artificial neural networks is the activation function. It determines the relationship between inputs and outputs of a network. In general, the activation function introduces a degree of the non-linearity that is valuable in most of ANN applications. The well known activation functions are logistic, hyperbolic tangent, sine (or cosine) and the linear functions. Among them, logistic activation function is the most popular one (Zhang et al., 1998).

3. The Used Tabu Search Algorithm

Tabu search algorithm was proposed by Glover (1989). Tabu search is a meta-heuristic method which has been used to solve many optimization problems (Glover and Laguna, 1997). The main principle of tabu search is that it has some memory of the states that has already been investigated and it does not revisit those states (Ye et al., 2007). The tabu search focuses greatly on expanding the global search area and avoiding the search of the same area. It can always get much better global solutions (Ye et al., 2007).

Starting conditions such as starting solution and initial state of the memories play important role on performance of tabu search algorithm. Determination of these conditions is a vital issue to reach better solutions. A strategy called “starting pool” proposed by Aladag (2009a) is employed in order to determine the starting conditions. In this strategy, a specified number of solutions are examined and the information gathered from this examination is used to generate starting solution and the initial state of the memories. Therefore, the number of examined solutions in the starting pool is a parameter for the proposed algorithm. Detailed information about the starting pool strategy can be found in Aladag (2009a). The used tabu search algorithm is employed to determine a feed forward neural network architecture including one hidden layer and one output neuron given in Figure 1. Therefore, the number of inputs and the number of neurons in the hidden layer will be obtained as a result of application of the used tabu search algorithm.

The elements of the used tabu search algorithm can be briefly defined as follows (Aladag, 2009b): firstly, the used tabu search algorithm begins with an initial solution generated by using starting pool strategy. Each solution x is a vector whose elements represent the numbers of neurons in the input and the hidden layers. The number of output neuron is always equal to one so there is no need to show this number in the solution vector. The solution space *X* is a set of solutions. The number of solutions in this set can be defined by the user. Thus, the size of solution space is also one of the parameters of the search algorithm. The objective function is RMSE which is computed over the test set. The objective function is as follows:

$$f(x) = \left(\frac{\sum_{i=1}^{T} (y_i - \hat{y}_i)^2}{T} \right)^{1/2}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value forecasted by using the solution *x*, and *T* is the number of test data. Since the objective function is RMSE, the used algorithm will try to minimize RMSE value. Global aspiration by objective is used as aspiration criterion in the tabu search algorithm.

For the number of inputs and neurons in the hidden layer, two frequency based memories are used separately. Occurrences in the examined current solutions are kept in these memories. Thus, for a specific number of inputs (or neurons in the hidden layer), it can be seen from the frequency based memory that how many times this number of input (or neuron) is used in the examined current solutions.

In order to generate a diversification effect, restarting strategy is employed. In intensification strategy, all examined current solutions are considered as critical events and are kept by using a critical event memory. After the stopping criterion is satisfied and the algorithm is terminated, three solutions in the critical event memory which have the best objective function values are examined again.

When the pre determined maximum iteration bound is reached, the algorithm is terminated. This maximum iteration bound is another parameter for the used algorithm.

After the elements of the used tabu search algorithm are given, the used algorithm for architecture selection can be given as follows (Aladag, 2009b):

ALGORITHM

A) Run the starting pool strategy.

B) Set the best solution obtained from the starting pool strategy as the best global solution.

C) Initialize the memories.

D) Run the tabu search algorithm.

Step 1. Generate the starting solution by using the information obtained from the starting pool strategy and begin from this solution.

Step 2. Use the neighborhood structure to generate a candidate list for the current solution.

Step 3. Set the solution with the best objective function value in the candidate list as the current neighbor.

(*i*) If the current neighbor is not tabu, set this one as the next current solution then go to Step 4. If the current neighbor is tabu, go to Step 3 ii.

(*ii*) If the tabu solution satisfies the aspiration criterion, set this solution as the next current solution and go to Step 4. Otherwise, go to Step 3 iii.

(*iii*) Set the other neighbor with the best objective function value among the remains candidate solutions as the current neighbor and go to Step 3 i. If all of the candidate solutions are examined, go to Step 4.

Step 4. If all of the examined solutions in the previous step could not be set as the next current solution, then generate a solution by using the information obtained from frequency based memory and set this solution as the next current solution and go to Step 5. Otherwise, go to Step 5 directly.

Step 5. Update the memories according to selected current solution.

(i) Update tabu list by adding the current solution.

(ii) Update frequency based memories.

(iii) Update the critical event memory by adding the current solution.

Step 6. If the objective function value of the current solution is better than the best value found so far, set this current solution as the best global solution, set diversification counter value as zero, and go to Step 7. Otherwise, increase the diversification counter by 1 and go to Step 7.

Step 7. If the diversification counter is smaller than the specified diversification parameter, then go to Step 8. Otherwise, run the diversification strategy to generate the next current solution, and go to Step 8.

Step 8. If the pre determined maximum iteration bound is reached, go to Step 9. Otherwise, go to Step 2.

Step 9. For three solutions in the critical event memory with the best objective values, run the intensification strategy. If a solution better than the best global solution found so far is obtained as a result of the intensification strategy, then set this solution as the best global solution.

4. The Candidate List Strategies

Firstly, the neighborhood structure and move types should be given. Then the candidate list strategy proposed by Aladag (2009a) and the candidate list strategy proposed in this study will be presented.

The Neighborhood N(X) and the Used Move Types

When a solution $x' \in X$, which is a neighbor of a solution $x \in X$, is generated from *x*, two move types add and remove are used in the neighborhood structure of the used tabu search algorithm. For the solution *x*=[7 11], by adding 2 to the number of inputs (x_1+2), the neighbor solution *x'* =[9 11] can be obtained. In a similar way, the neighbor solution *x'* =[7 10] can be generated by subtracting 1 from the number of neurons (x_2-1) in the hidden layer of the solution *x*=[7 11]. This example is shown in Figure 2.

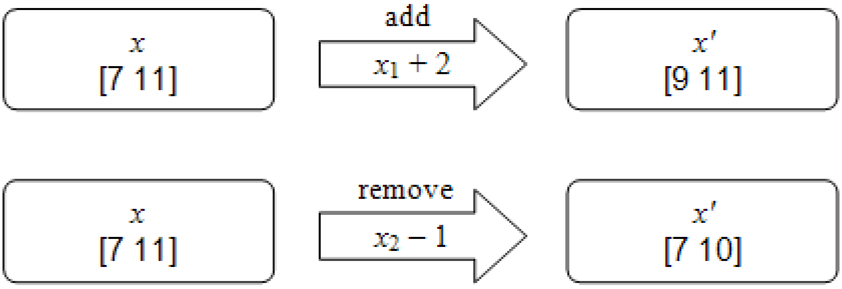


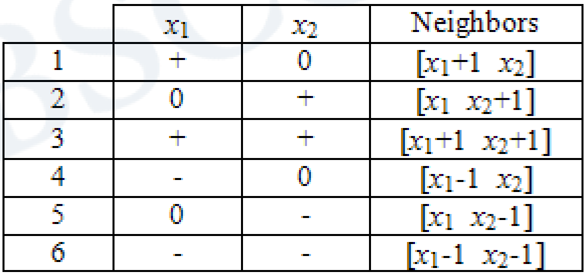
Figure 2. An example for the used move types.

4.1. Candidate List Strategy Proposed by Aladag (2009a)

Let '+', '-', and '0' represents increasing by 1, decreasing by 1, and no change, respectively. Then, for a solution x=[x_1 x_2], a candidate list $V(x)$ can be constructed as given in Table 1.

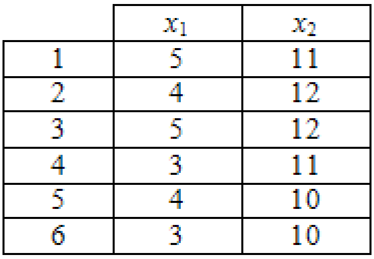
For instance, the candidate solutions obtained from the solution x=[4 11] are given in Table 2. Instead of exploring its whole neighborhood, only six solutions in Table 2 are examined for the solution x=[4 11].

Table 1. The candidate list $V(x)$ for the solution x=[x_1 x_2]



	x_1	x_2	Neighbors
1	+	0	[x_1+1 x_2]
2	0	+	[x_1 x_2+1]
3	+	+	[x_1+1 x_2+1]
4	-	0	[x_1-1 x_2]
5	0	-	[x_1 x_2-1]
6	-	-	[x_1-1 x_2-1]

Table 2. The candidate list $V(x)$ for the solution x=[4 11]



	x_1	x_2
1	5	11
2	4	12
3	5	12
4	3	11
5	4	10
6	3	10

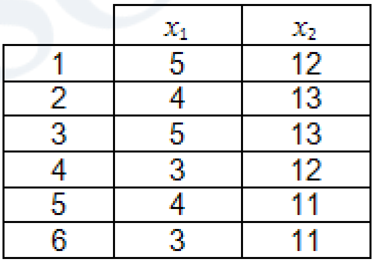
4.2. The Proposed Candidate List Strategy

In the candidate list strategy proposed by Aladag (2009a), after a neighbor solution *x'* is generated from a current solution *x*, there are two same solutions in *V*(*x*) and *V*(*x'*). For example, let the candidate solution *x'*=[4 12] is selected among the candidate solutions given in Table 2 of solution *x*=[4 11]. The candidate list *V*(*x'*) for solution *x'*=[4 12] is given in Table 3.

When the candidate lists of *x* and *x'* is examined, it is seen that the candidates solutions [5 12] and [3 11] are same in these lists. Therefore, these two examined candidate solutions for x will be examined again for the next solution *x'*. According to Aladag (2009a), this candidate list strategy causes an intensification effect in the examined region. On the other hand, at each iteration, these repeater candidate solutions can be avoided so the running time of the search algorithm can be decreased. To do this, a new candidate list strategy is proposed in this study. Let '+', '-', and '0' represents increasing by 1, decreasing by 1, and no change, respectively. Then, for a solution $x=[x_1\ x_2]$, the proposed candidate list *V*(*x*) can be generated as given in Table 4.

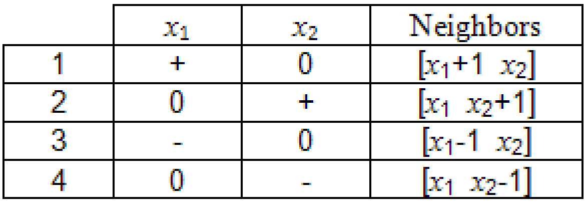
When the proposed candidate list strategy is used, the candidate lists of two consecutive current solutions do not include same candidate solutions. Also, while only four candidate solutions are examined at each iteration in the proposed strategy, six candidate solutions are examined in the strategy proposed by Aladag (2009a). Therefore, the running time will be shorter if the proposed candidate list strategy is employed.

Table 3. The candidate list *V*(*x'*) for the solution *x'*=[4 12]



	x_1	x_2
1	5	12
2	4	13
3	5	13
4	3	12
5	4	11
6	3	11

Table 4. The proposed candidate list *V*(*x*) for the solution $x=[x_1\ x_2]$



	x_1	x_2	Neighbors
1	+	0	$[x_1+1\ \ x_2]$
2	0	+	$[x_1\ \ x_2+1]$
3	-	0	$[x_1-1\ \ x_2]$
4	0	-	$[x_1\ \ x_2-1]$

5. The Application

In the implementation the beer consumption (10^6 *liter*) between first quintile of the year 1976 and fourth quintile of the year 1974 in Austria with 73 observations and the electricity consumption (10^6 *kWh*) between 1976 and 2006 in Turkey with annual 37 observations are forecasted. The graphs of these series are illustrated in Figure 3 and 4. To determine the best ANN architecture, the tabu search algorithm given in Section 3 is used. These time series are analyzed by using feed forward neural networks and when the best architectures is determined, two candidate list strategies given in previous section are used.

When the best architecture is being searched, the other elements of the ANN are fixed like in Aladag (2009b). The logistic activation function is used in all of the neurons of networks. For all of the time series, the last 5 observations are taken as the test set and the rest of the observations are used for the training set. Levenberg Marquardt algorithm is employed as training algorithm.

The parameters of the proposed tabu search algorithm are taken as in Aladag (2009b). Minimum and maximum numbers of inputs for the solution space are 1 and 18, respectively. Same values are used for the number of neurons in the hidden layer. Minimum and maximum numbers of inputs for the starting pool are 1 and 3, respectively. The same values are also used for the number of neurons in the hidden layer. In other words, both the number of inputs and the number of neurons in the hidden layer are changed from 1 to 3. Thus, 9 solutions are examined in the starting pool strategy. Tabu tenure and maximum iteration bound are taken as 20 and 50 iterations, respectively. The repeat parameter for each solution is 8 and the maximum iteration bound is 50. Finally, the diversification and the intensification parameters are chosen as 8 and 40, respectively. (See Aladag, 2009a for detailed information about the parameters of the used tabu search algorithm).

By using the used tabu search algorithm, each time series are forecasted for 200 times totally. The tabu search with Aladag's (2009a) candidate list strategy is run for 100 times. And, the tabu search with the candidate list strategy proposed in this study is run for 100 times. Aladag (2009a) showed that 100 repetitions are enough to reach general results. Also, by using trial and error method to determine the best architecture, each time series are forecasted for 100 times. The obtained average RMSE values calculated over the test sets are summarized in Table 5. In Table 5, ACLS and PCLS represents Aladag's (2009a) candidate list strategy and the proposed candidate list strategy, respectively.

According to Table 5, it is clearly seen that more accurate forecasts are obtained when ACLS or PCLS are used as candidate list strategy in the tabu search algorithm for the examined real time series. In other words, it is seen that good forecasts are obtained when the proposed candidate list strategy is used. In addition there is no big difference between the two candidate lists strategies in terms of forecasting accuracy. Therefore, when the proposed candidate list strategy is employed instead of ACLS, accurate forecasts are obtained and at the same time running time of the tabu search algorithm are decreased. While ACLS includes six candidate solutions, the proposed candidate list includes only four solutions. Theoretically, when the proposed candidate list strategy is used, running time of the tabu search algorithm will be decreased by 33%, since instead of examined 6 solutions at each iteration, only 4 solutions are examined in the proposed strategy. Especially, when the number of observations in time series is high, the proposed strategy should be used to get accurate forecasts in a reasonable time.

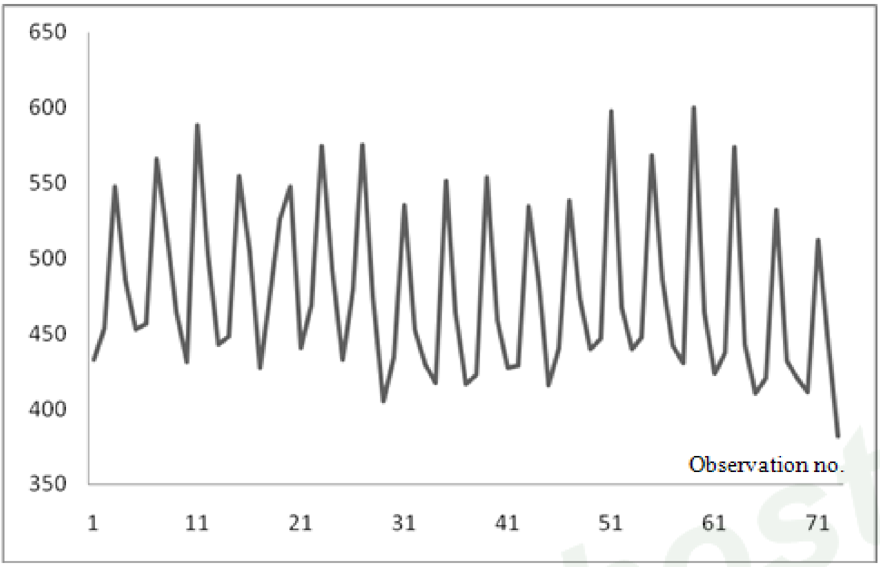


Figure 3. The beer consumption in Austria (10^6 *liter*).

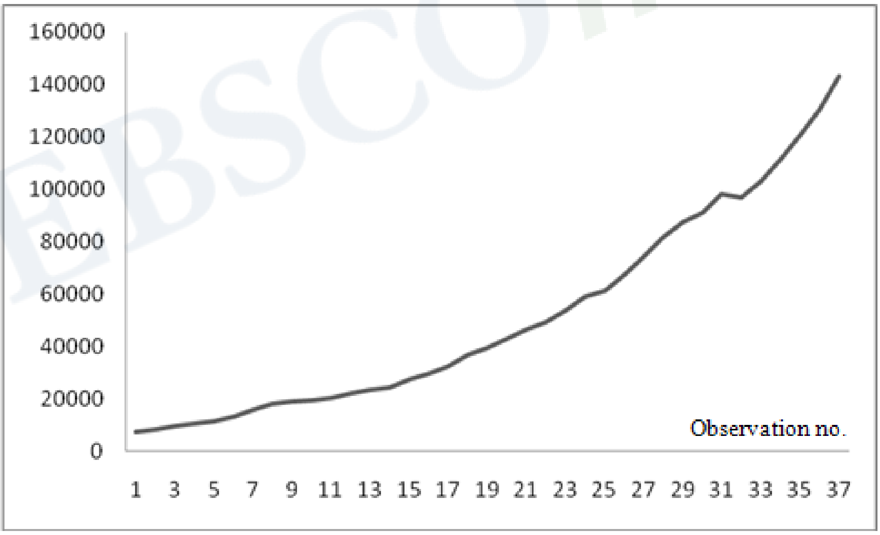


Figure 4. The electricity consumption in Turkey (10^6 *kWh*).

Table 5. The calculated average RMSE values

Time series	Trial &	ACLS	PCLS
	Error		
Beer consumption	10.20	5.11	5,80
Electricity consumption	4496.2	2354.07	2429,29

CONCLUSION

In recent years, ANN method has been employed in time series forecasting since ANN provides some advantages. On the other hand, determination of components of ANN is still a problem for practitioners. Selection of the best architecture is a vital issue since this choice plays very important role on getting accurate forecasts. Therefore, some methods have been proposed in the literature to solve architecture selection problem. The solution methods range from principle component analysis to heuristic algorithms. Among them, the most preferred one is trial and error technique because of its easy usage.

Aladag (2009a; 2009b) proposed a tabu search algorithm to determine the best architecture in forecasting. In this study, a new candidate list strategy is proposed to improve the tabu search algorithm proposed by Aladag (2009a; 2009b). The proposed strategy is used when two real time series are forecasted. Also, Aladag's (2009a) candidate list strategy is employed for these series. Then, the obtained results are compared. As a result of comparison, it is seen that accurate forecasts are obtained when the proposed strategy is used in the tabu search algorithm.

At each iteration of the tabu search algorithm, while six candidate solutions are generated and examined in Aladag's (2009a) candidate list strategy, only four solutions are generated and examined in the proposed strategy. Thus, it can be theoretically said that the running time of the tabu search algorithm will be decreased by 33% if the proposed candidate list strategy is used instead of Aladag's (2009a) strategy. For further applications, if time series include a lot of observations, the proposed strategy should be preferred in order to reduce the running time of the tabu search algorithm so that accurate forecasts can be obtained in a reasonable time.

REFERENCES

Aladag, CH; Egrioglu, E; Gunay, S. A new architecture selection strategy in solving seasonal autoregressive time series by artificial neural networks, *Hacettepe Journal of Mathematics and Statistics*, 2008, 37(2), 185-200.

Aladag, CH. Using tabu search algorithm in the selection of architecture for artificial neural networks, *Ph.D. thesis*, 2009a, Hacettepe University.

Aladag, CH. A new architecture selection method based on tabu search for artificial neural networks, *Expert Systems with Applications,* 2009b, under review.

Aladag, CH; Egrioglu, E; Kadilar, C. Modeling brain wave data by using artificial neural networks, *Hacettepe Journal of Mathematics and Statistics,* 2010a, 39 (1), 81–88.

Aladag, CH; Egrioglu, E; Gunay, S; Basaran, MA. Improving weighted information criterion by using optimization, *Journal of Computational and Applied Mathematics,* 2010b, 233, 2683-2687.

Balestrassi, PP; Popova, E; Paiva, AP; Marangon Lima, JW. Design of experiments on nn training for nonlinear time series forecasting, *Neurocomputing*, 2009, Volume 72, Issues 4-6, 1160-1178.

Buhamra, S; Smaoui, N; Gabr M. The Box-Jenkins analysis and neural networks: prediction and time series modelling, *Applied Mathematical Modeling,* 2003, 27, 805-815.

Cichocki, A; Unbehauen, R. Neural networks for optimization and signal processing, John Willey & Sons: New York, 1993.

Dam, M; Saraf, DN. Design of neural networks using genetic algorithm for on-line property estimation of crude fractionator products, *Computers and Chemical Engineering*, 2006, 30, 722–729.

Durbin, B; Dudoit, S; Van der Laan, MJ. A deletion/substitution/addition algorithm for classification neural networks, with applications to biomedical data, *Journal of Statistical Planning and Inference*, 2008, 138, 464-488.

Egrioglu, E; Aladag, CH; Gunay, S. A new model selection strategy in artificial neural network, *Applied Mathematics and Computation*, 2008, 195, 591-597.

Glover, F. Tabu search-Part I, ORSA *J. Comput*., 1989, 1(3), 190–206.

Glover, F; Laguna, M. *Tabu Search*, Kluwer Academic Publishers*:* Boston, 1997.

Gunay, S; Egrioglu, E; Aladag, C.H. *Introduction to single variable time series analysis*, Hacettepe University Press: Ankara, 2007.

Leahy, P; Kiely, G; Corcoran, G. Structural optimization and input selection of an artificial neural network for river level prediction, *Journal of Hydrology,* 2008, 355, 192– 201.

Murata, N; Yoshizawa S; Amari S. Network information criterion determining the number of hidden units for an artificial neural network model, IEEE *Transaction on Neural Networks*, 1994, 5, 865-872.

Picton, PD. *Introduction to Neural Networks*, Macmillan Press Ltd.: London, 1994.

Rathbun, TF; Rogers, SK; DeSimo, MP; Oxley, ME. MLP iterative construction algorithm, *Neurocomputing*, 1997, 17, 195-216.

Roy, A; Kim, LS; Mukhopadhyay, S. A Polynomial time algorithm for the construction and training of a class of multilayer perceptrons, *Neural Networks*, 1993, 6, 535-545.

Siestema, J; Dow, R. Neural net pruning: why and how?, Proceedings of the IEEE *International Conference on Neural Networks*, 1988, 1, 325-333.

Smith, KA. *Neural networks in business: techniques and applications*, Idea Group, 2002.

Ture, M; Kurt, I. Comparison of four different time series methods to forecast hepatitis A virus infection, *Expert Systems with Applications,* 2006, 31, 41–46.

Ye, J; Qiao, J; Li, M; Ruan, X. A tabu based neural network learning algorithm, *Neurocomputing*, 2007, 70, 875–882.

Yuan, HC; Xiong, FL; Huai, XYA. method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy, *Computers and Electronics in Agriculture*, 2003, 40, 57-64.

Zeng, J; Guo, H; Hu, Y. Artificial neural network model for identifying taxi gross emitter from remote sensing data of vehicle emission, *Journal of Environmental Sciences*, 2007, 19, 427-431.

Zhang, G; Patuwo, BE; Hu. YM. Forecasting with artificial neural networks: the state of the art, *International Journal of Forecasting*, 1998, 14, 35-62.

Zurada, JM. *Introduction of artificial neural systems, West Publishing*: New York, 1992.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 10

ARTIFICIAL NEURAL NETWORK BASED ESTIMATION OF GLOBAL AND DIFFUSE FRACTION OF SOLAR RADIATION USING METEOROLOGICAL PARAMETERS

Shafiqur Rehman*[*] *and Mohamed Mohandes
Engineering Analysis Section, Center for Engineering Research, Research Institute,
King Fahd University of Petroleum and Minerals, Saudi Arabia

ABSTRACT

Measured meteorological parameters such as air temperature and relative humidity values recorded between 1998 and 2002 for Abha city in Saudi Arabia were used for the estimation of global solar radiation (GSR) and fraction of diffuse solar radiation (DSR) in future time domain using artificial neural network method. The estimations of GSR and DSR were made using three combinations of data sets namely, (i) day of the year and daily maximum air temperature as inputs and global solar radiation as output, (ii) day of the year and daily mean air temperature as inputs and global solar radiation as output and (iii) time day of the year, daily mean air temperature and relative humidity as inputs and global solar radiation as output. The measured data between 1998 and 2001 was used for training the neural networks while the remaining 240 days' data from 2002 as testing data. The testing data was not used in training the neural networks.

Obtained results show that neural networks are well capable of estimating global and diffuse solar radiation from temperature and relative humidity. Hence the methodology can be used for estimating GSR and DSR for locations where only temperature and humidity data are available.

Keywords: Artificial neural networks, backpropagation algorithm, meteorology, air temperature, relative humidity, global solar radiation, diffuse solar radiation, prediction.

[*] Corresponding author's address: KFUPM BOX # 767, DHAHRAN-31261, SAUDI ARABIA, Telephone: 00966 3 8603802, Fax: 00966 3 8603996, E-mail: srehman@kfupm.edu.sa

INTRODUCTION

Due to increasing costs of fossil fuels, uncertainty of availability, increasing environmental pollution and general awareness amongst the common people the green sources of energy are being encouraged. The green sources of energy include solar photovoltaic, solar thermal, wind, biomass, small and big hydro, tidal, wave, ocean, etc. Among these sources of energy, wind, solar and hydro are the common ones in use. Despite being relatively costly compared to wind and other green sources of energy, solar photovoltaic technology is being used commonly for the generation electricity for both grid connected and stand alone power systems. For proper, economical and efficient development and utilization of solar energy, an accurate knowledge of the availability and variability of solar radiation intensity both in time and special domain is very critical. In this respect, Dave and Norman [1], Okeke and Aunforom [2] and Riordan et el. [3] have pointed out the importance of solar radiation data for the optimal design and efficient operation of solar energy systems. The increasing demand for information on the availability of solar radiation has highlighted the inadequacy of the current observational networks. According to Elminir et el. [4], this inadequacy occurs in three basic ways: limited spatial coverage; limited length of record; and predominance of global radiation data, while few data are available for the diffuse component.

Usually, the global and diffuse solar radiation measurements are made at few locations in a country, especially in developing countries, which may or may not be the same as the actual site of solar energy development and utilization. In order to know the behavior of solar radiation at the site of interest long term data from a nearby location along with empirical, semi-empirical, physical, neural networks, wavelets, fractals, etc. techniques is used. An extensive amount of work on Ångström type of empirical models for the estimation of global solar radiation on horizontal surface using measured sunshine duration values has been cited in the literature, for example [5-10]. In the recent years, neural network methods have been employed for the prediction of global solar radiation both in time and special domains as can be seen from [11-20].

Since the temperature and relative humidity records are more readily available around the globe, these values are being used to estimate the global solar radiations. Elizondo et. al. [11] used meteorological parameters such as air temperature, precipitation, clear-sky radiations, day length and day of the year as input into the feed-forward neural network technique for the estimation of daily global solar radiation. Al-Alawi and Al-Hinai [12] predicted total radiation to an accuracy of 93% by using meteorological parameters as input into artificial neural network models. Toĝrul and Onat, [14] used geographical and meteorological parameters along with the ANN methods for the prediction of global solar radiation for a city in Turkey. Kalogirou et. al. [19] used a recurrent neural network method to estimate the maximum solar radiation using measured values of air temperature and relative humidity as input. The authors found that the correlation coefficient varied between 98.6% and 98.8%. Sözen et. al. [21] used meteorological and geographical parameters as input into the neural networks model for the prediction solar potential for Turkey. In another study, Yang and Koike [22] utilized upper air humidity values for the estimation of solar radiation on the surface of the earth through artificial neural network method.

Elminir et el. [23] proposed an artificial neural network (ANN) model for the prediction of diffuse fraction (KD) in hourly and daily scale and compared the performance of the ANN model with that of two linear regression models. Their results hint that the ANN model is more suitable to predict diffuse fraction in hourly and daily scales than the regression models in the plain areas of Egypt. Šúri et al. [24] presented a solar radiation database of Europe developed in the geographical information system, and three interactive web applications providing an access to it. The database included monthly and yearly average values of the global irradiation on horizontal and inclined surfaces, as well as climatic parameters needed for an assessment of the potential PV electricity generation (Linke atmospheric turbidity, the ratio of diffuse to global irradiation, an optimum inclination angle of modules to maximize energy yield). An objective evaluation has shown that the diffuse solar radiation is better reproduced by neural network synthetic series than by a correlation model as reported by Soares et. al [25]. The authors used a perceptron neural-network technique to estimate hourly values of the diffuse solar-radiation at the surface in Sao Paulo City, Brazil, using as input the global solar-radiation and other meteorological parameters measured from 1998 to 2001.

Other studies on diffuse solar radiation prediction using meteorological parameters include the work of Sozen et al. [26] where the authors used artificial neural networks for the estimation of solar potential for Turkey and Sozen and Arcakhoglu [27] reported the effect of relative humidity on solar radiation. The estimation of diffuse solar radiation has been performed by various authors (Aras et el., [28]; Tarhan and Sar, [29]; Paliatsos et el., [30]; Amauri et el., [31]; Jin et el., [32]; Rensheng et el., [33]; El-Sebaii and Trabea, [34]; Roderick, [35]; and Elminir, [4]) using meteorological parameters such as ambient temperature and relative humidity for different locations around the world.

Similarly, ample work has been done on different aspects of solar radiation data modeling and prediction in Saudi Arabia as can be seen from references (Rehman, [36]; Rehman and Ghori, [37]; Rehman, [38]; Aksakal and Rehman, [39]; Rehman and Halawani, [40] and Rehman and Halawani, [41], Mohandes et al., [42 and 43], and Tasadduq et al., [44]; and Rehman and Mohandes, [45 and 46]). The present study utilized the neural network method for the estimation of the diffuse fraction of solar radiation for Abha, a city in the south west region of the Kingdom of Saudi Arabia and daily maximum or minimum or means ambient temperatures as input to the ANN model. The model was trained using daily values of measured diffuse solar radiation data along with the daily values of temperature during 1998 to 2001 and the model was tested using measured data for the year 2002.

The present study utilizes the air temperature, day of the year and relative humidity values as input in neural networks for the prediction of global and diffuse solar radiation on horizontal surfaces for Abha, a city in the western region of Saudi Arabia.

DATA DESCRIPTION

The five minutes mean data on air temperature, relative humidity, global solar radiation, etc. at Abha and 11 other locations is collected since January 1998 by King Abdulaziz City for Science and Technology (KACST). The solar radiation data collection station is situated at latitude of 18.23°N, longitude of 42.66°W and an altitude of 2039 meters above sea level. The climate of Abha is classified as moderate with long term temperature and relative humidity varying between 1.2°C and 36°C and 5% and 100%, respectively throughout the year. The global solar radiation (GSR) was found to vary between a minimum of 0 to a maximum of 9900 W/m^2. Figures 1 to 3 show the variation of air temperature, relative humidity and global solar radiation for Abha between 1998 and 2002, respectively.

The daily variation of air-temperature, relative humidity and GSR is shown in Figures 1 to 3, respectively. The daily average values of temperature were found to vary between approximately 10 ºC and 27 ºC during the years 1998 to 2001 while in 2002 the temperature varied between 12 ºC and 27 ºC, as can be seen from Figure 1. The daily total values of GSR fluctuated between, approximately, 3kWh/m^2/day and 8 kWh/m^2/day with some peaks reaching 12 to 14 kWh/m^2/day in 1998 and 2000, as shown in Figure 3. The diffuse solar radiation (DSR) was found to vary between a minimum of 0 to a maximum of 9900 W/m^2. Figures 4 show the variation of diffuse solar radiation (DSR) for Abha between 1998 and 2002. The daily total values of DSR fluctuated between, approximately, 0.315 kWh/m^2/day and 1.86 kWh/m^2/day with some peaks reaching up to 8 kWh/m^2/day in 1998 and 2000, as shown in Figure 4.

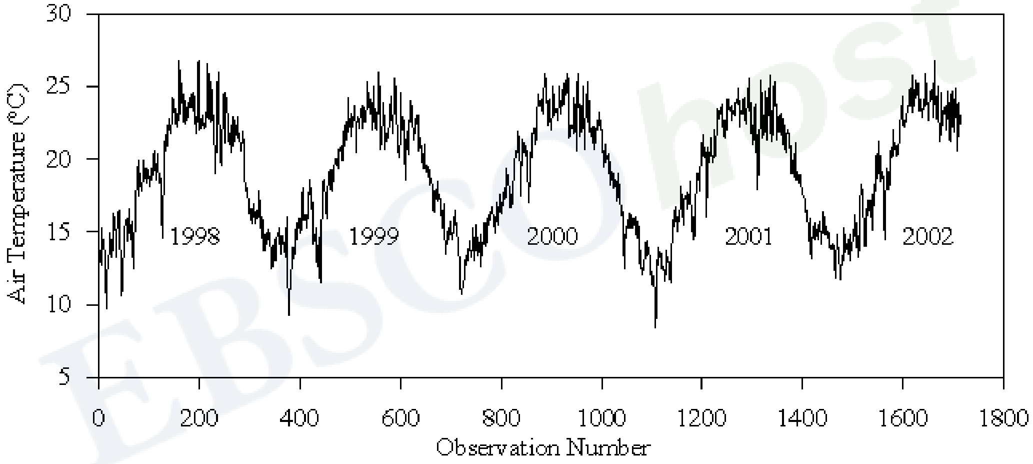


Figure 1. Daily mean air temperature at Abha.

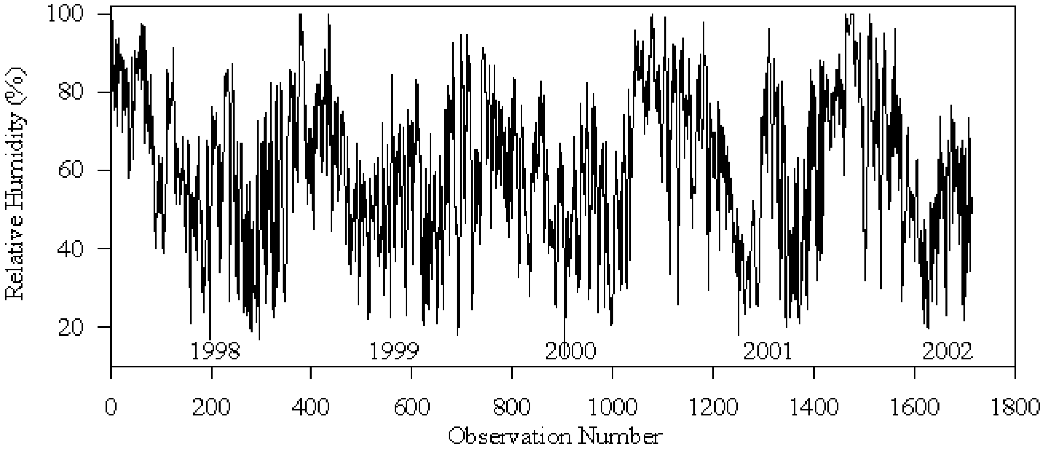


Figure 2. Daily mean relative humidity at Abha.

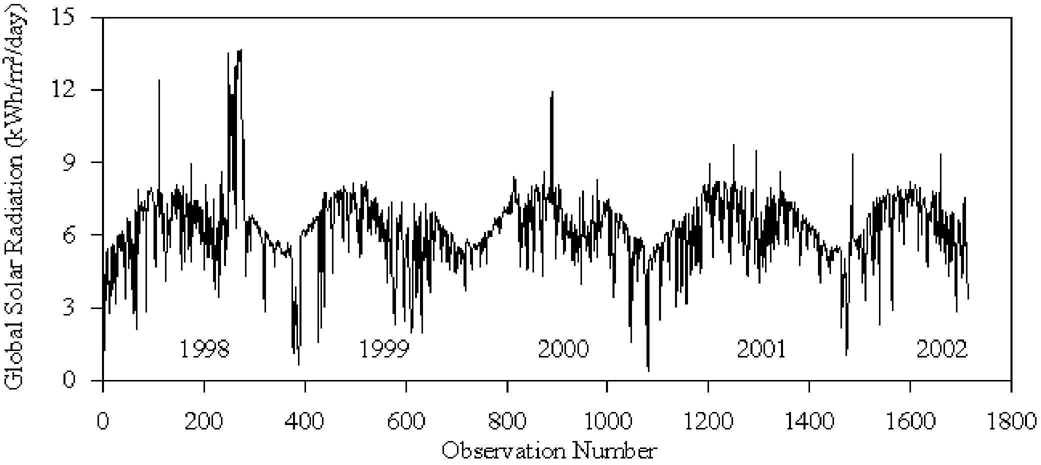


Figure 3. Daily mean global solar radiation at Abha.

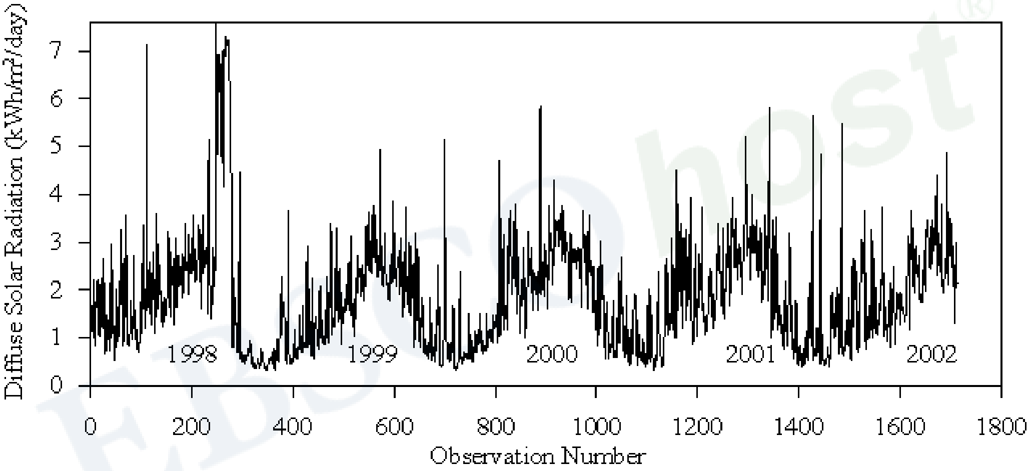


Figure 4. Daily variation of diffuse fraction of solar radiation at Abha.

METHODOLOGY USED

The interest in artificial neural networks (ANNs) is largely due to their ability to mimic natural intelligence in its learning from experience [47]. They learn from examples by constructing an input-output mapping without explicit derivation of the model equation. ANNs have been used in a broad range of applications including: pattern classification [48 and 49], function approximation, optimization, prediction and automatic control [50] and many others. Additionally, ANNs have been used extensively for meteorological applications. In this paper ANNs are used for the estimation of global solar radiation based on measured temperature and relative humidity data.

An artificial neural network consists of many interconnected identical simple processing units called neurons. Each connection to a neuron has an adjustable weight factor associated with it. Every neuron in the network sums its weighted inputs to produce an internal activity level a_i ,

$$a_i = \sum_{j=1}^{n} w_{ij} x_{ij} - w_{io} \tag{1}$$

where w_{ij} is the weight of the connection from input *j* to neuron *i*, x_{ij} is input signal number *j* to neuron *i*, and w_{io} is the threshold associated with unit *i*. The threshold is treated as a normal weight with the input clamped at -1. The internal activity is passed through a nonlinear function φ to produce the output of the neuron y_i ,

$$y_i = \varphi(a_i) \tag{2}$$

Several forms of differentiable activation functions have been used with the most popular being the logistic function of the form:

$$\varphi(a_i) = \frac{1}{1+\exp(-a_i)} \tag{3}$$

The weights of the connections are adjusted during the training process to achieve the desired input/output relation of the network. A multilayer feedforward network has its neurons organized into layers with no feedback or lateral connections. Layers of neurons other than the output layer are called hidden layers. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. Figure 5 shows a three-layer feedforward network.

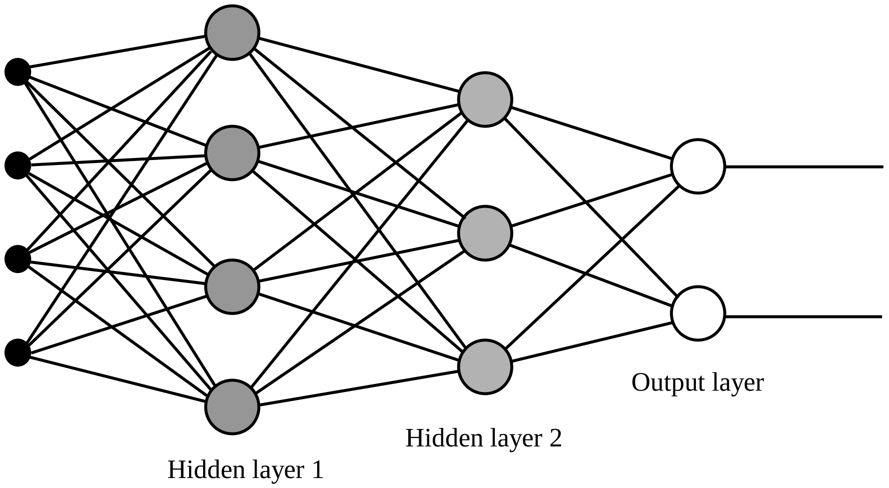


Figure. 5. Three-layer feedforward neural network.

The backpropagation algorithm [51] is a supervised iterative training method for multilayer feedforward nets with sigmoidal nonlinear threshold units. It uses training data consisting of S input-output pairs of vectors that characterizes the problem. Using a generalized Least-Mean-Square algorithm the backpropagation algorithm minimizes the mean square difference between the real network output and the desired output [52]. The error function that the backpropagation algorithm minimizes is the average of the square difference between the output of each neuron in the output layer and the desired output. The error function can be expressed as:

$$E = \frac{1}{S}\sum_{s}\sum_{k}(d_{sk} - o_{sk})^2 \tag{4}$$

where *s* is the index of the S training pair of vectors, *k* is the index of elements in the output vector, d_{sk} is the *k*th element of the *s*th desired pattern vector, and o_{sk} is the *k*th element of the output vector when pattern s is presented as input to the network. Minimizing the cost function represented in Eqn. (4) results in an updating rule to adjust the weights of the connections between neurons. The weight adjustment of the connection between neuron *i* in layer *m* and neuron *j* in layer *m*+1 can be expressed as:

$$\Delta w_{ji} = \eta\delta_j o_i \qquad 5$$

where *i* is the index of units in layer *m*, η is the learning rate, o_i is the output of unit *i* in the *m*th layer, and δ_j is the delta error term backpropagated from the *j*th unit in layer *m+1* defined by:

$\delta_j = \left[d_j - o_j\right]o_j\left[1 - o_j\right]$, neuron *j* is in the output layer

$\delta_j = y_j\left[1 - y_j\right]\sum_k \delta_k w_{kj}$, neuron *j* is in a hidden layer and *k* is index of neurons in the layer (*m+2*), ahead of the layer of neuron *j*.

Choosing a small learning rate η leads to slow rate of convergence, however, too large η leads to oscillation. A simple method for increasing the rate of learning without oscillation is to include a momentum term as:

$$\Delta w_{ji}(n+1) = \eta\delta_j o_i + \alpha\Delta w_{ji}(n) \tag{6}$$

where *n* is the iteration number, and α is a positive constant which determines the effect of past weight changes on the current direction of movement in weight space. Detailed description of the multilayer feedforward neural networks and the backpropagation algorithm may be found in [52].

RESULTS AND DISCUSSION

First, a feedforward ANN has been trained to estimate the global solar radiation based on the daily maximum temperature and day of the year. After several experiments it was found that a network with 2 inputs, 24 hidden neurons in one layer and one output unit was sufficient for such application. The data for 1462 days during 1998- 2001 were used for training purpose, while data for 240 days from the year 2002 were used for testing the performance. A maximum of 1000 iterations was allowed. The details of the results for GSR and DSR predictions are discussed in the following paragraphs.

Prediction of GSR

The estimated GSR values were compared with the measured for the testing period as shown in Figure 6. The mean squared error for this data was found to be 2.823x10^{-04}, while the absolute mean percentage error was 10.3%.

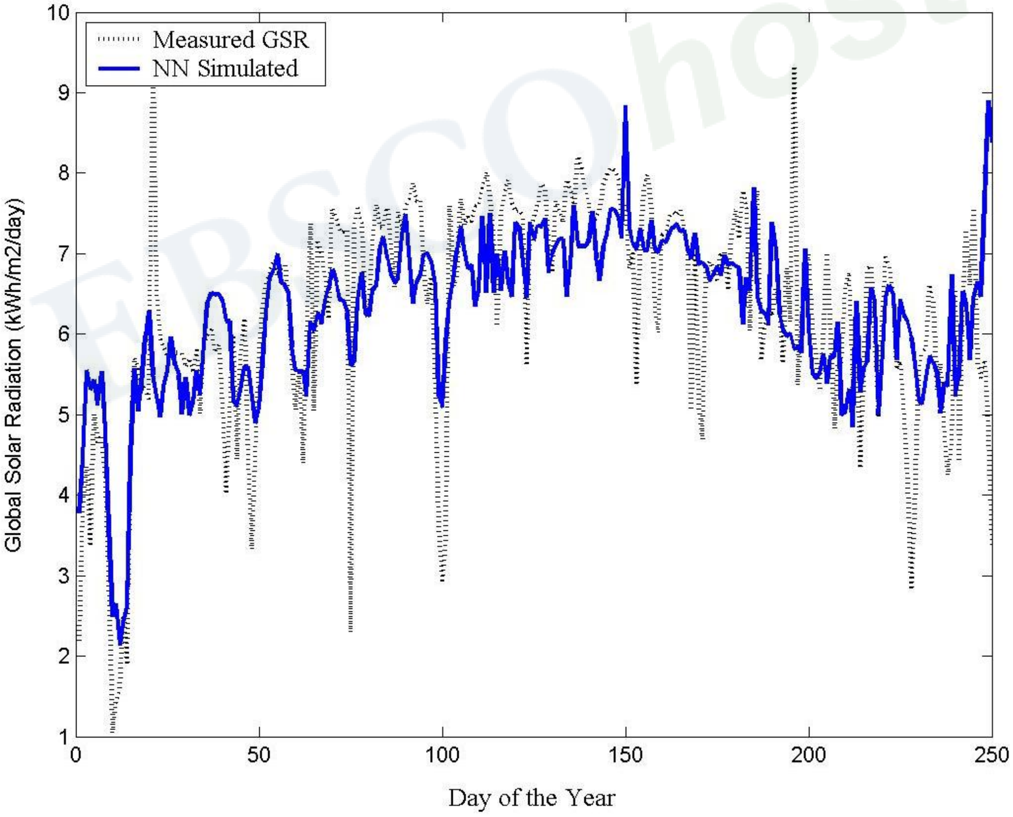


Figure. 6. Estimated and measured GSR on testing data when daily maximum temperature and day of the year are used as inputs.

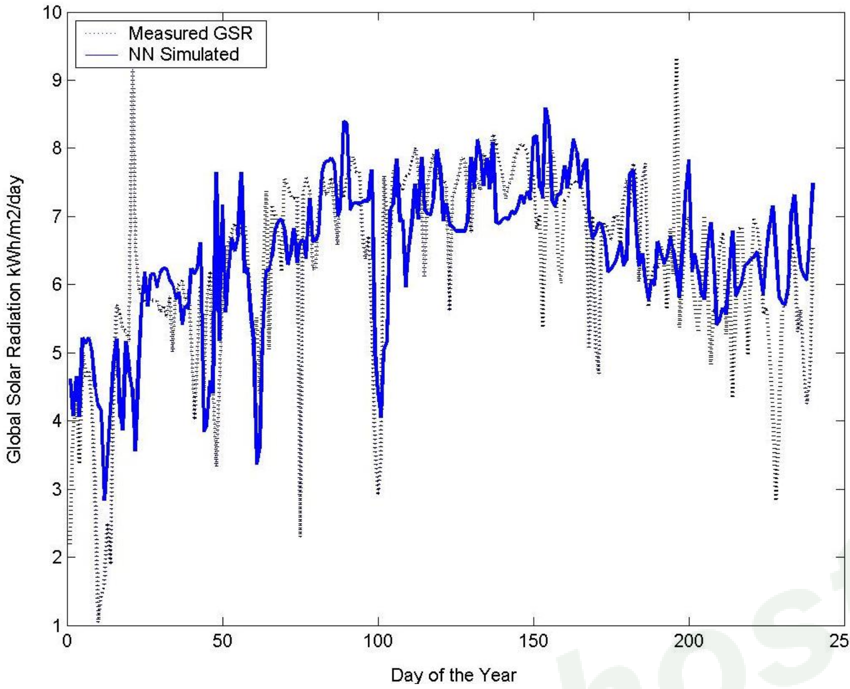


Figure. 7. Estimated and measured GSR on testing data when mean daily temperature and day of the year are used as inputs

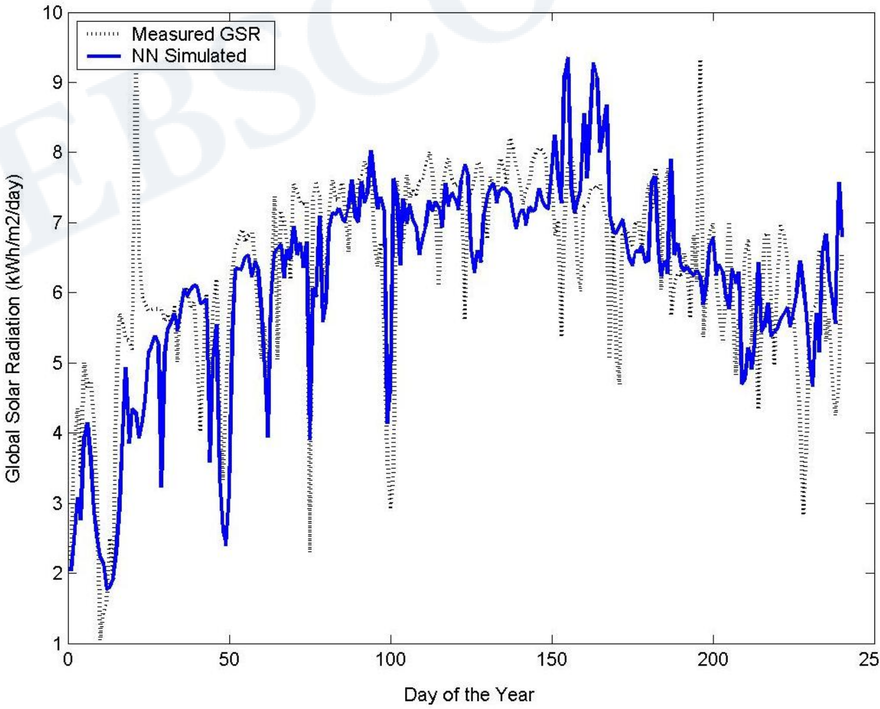


Figure. 8. Estimated and measured GSR on testing data when relative humidity, mean daily temperature and day of the year are used as inputs.

Another feedforward ANN was trained to estimate GSR based on the daily mean temperature and day of the year. A network of two inputs, 32 hidden neurons in one layer, and one output was found to perform reasonably good for this case. With same data division as done in the previous case, the obtained mean squared error was 0.0052 while the absolute mean percentage error was 11.8%. The measured and estimated values of GSR for this case are shown in Figure 7.

Lastly, a neural network with 3 inputs, 24 hidden neurons in one layer and one output unit was trained on day of the year, daily mean relative humidity and daily mean temperature to predict the GSR. The same range of training data (1462) days used for training and data for 240 days for testing. The mean squared data error for this case on testing data was 3.0148x10^{-5}, while the absolute mean percentage error was 4.49%. Figure 8 shows plot of the measured and estimated GSR for the testing data.

Prediction of DSR

The training results for DSR case are shown in Figure 9. The estimated DSR values were compared with the measured for the testing period as shown in Figure 10. The mean squared error for this data was found to be 2.06x10^{-04}. Another feedforward ANN was trained to estimate DSR based on the daily mean temperature and day of the year. A network of two inputs, 36 hidden neurons in one layer, and one output was found to perform reasonably good for this case. With same data division as done in the previous case, the obtained mean squared error was 2.36x10^{-5}. The measured and estimated values of DSR for this case are shown in Figure 11. To further explore the effect of daily minimum temperature on DSR, the ANN model was trained with daily minimum temperature along with the day of the year. A network of two inputs, 36 hidden neurons in one layer, and one output was found to perform reasonably good for this case too. With same data division as done in the previous case, the obtained mean squared error was 9.3x10^{-6}. The measured and estimated values of DSR for this case are shown in Figure 12.

Lastly, a neural network with 3 inputs, 20 hidden neurons in one layer and one output unit was trained on day of the year, daily mean relative humidity and daily mean temperature to predict the DSR. The same range of training data (1462) days used for training and data for 250 days for testing. The mean squared data error for this case on testing data was 5.18x10^{-7}. Figure 13 shows plot of the training of the ANN model and the comparison of measured and estimated DSR is shown in Figure 14 for the testing data.

CONCLUDING REMARKS

This paper presents the outcome of an attempt made to predict the global solar radiation based on measured values of temperature and relative humidity only. This is important because temperature and relative humidity are commonly available parameters, while GSR and DSR are costly to measure and requires continuous attention of skilled manpower. Data for Abha city in Saudi Arabia between 1998 and 2001 was used for training a feedforward

ANN using backpropagation algorithm. Data for 240 days in the year 2002 was used to test the performance of the ANN system.

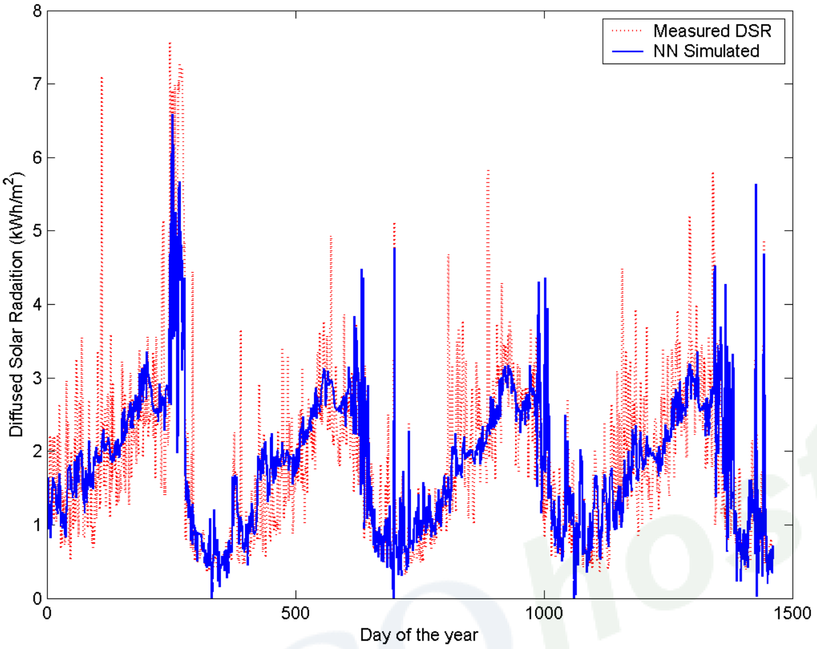


Figure 9. Training results with maximum air temperature and day of the year as inputs.

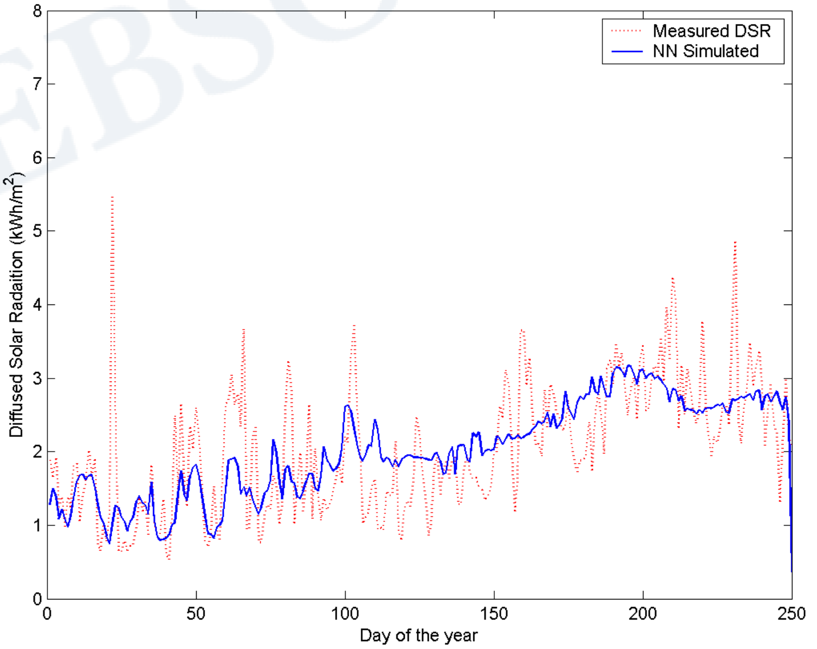


Figure 10. Comparison of measured diffuse solar radiation with estimated values obtained using maximum air temperature and day of the year as input to ANN model.

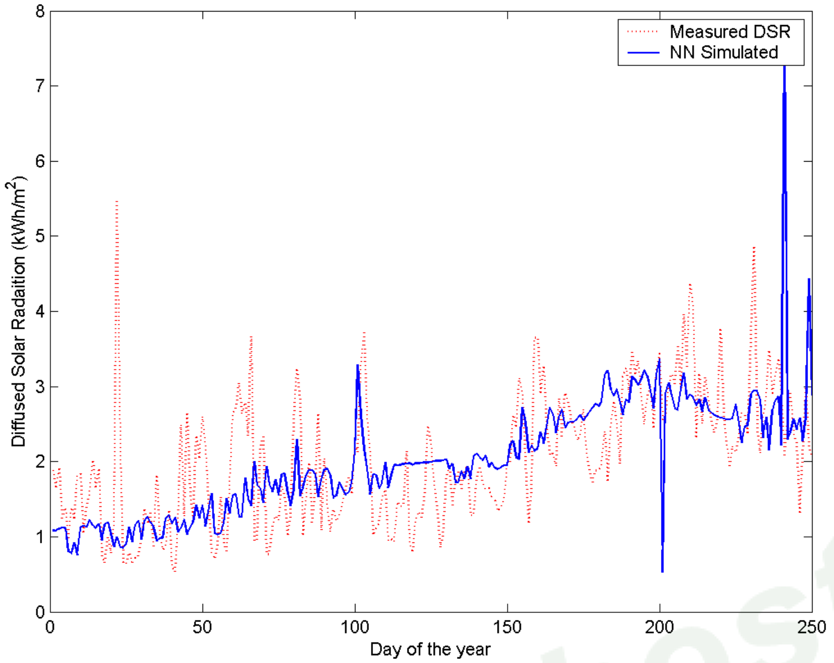


Figure 11. Comparison of measured diffuse solar radiation with estimated values obtained using mean air temperature and day of the year as input to ANN model.

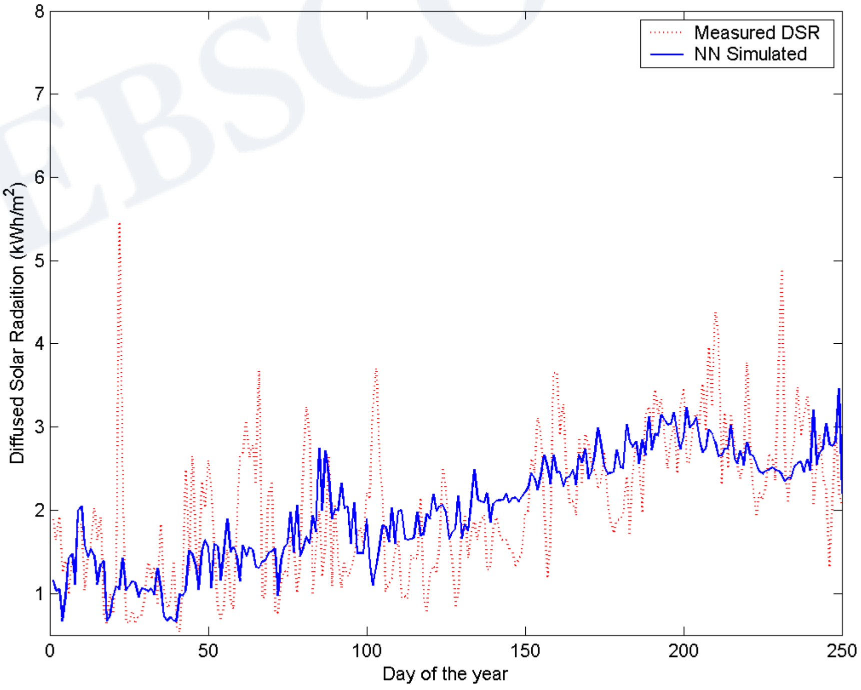


Figure 12. Comparison of measured diffuse solar radiation with estimated values obtained using minimum air temperature and day of the year as input to ANN model.

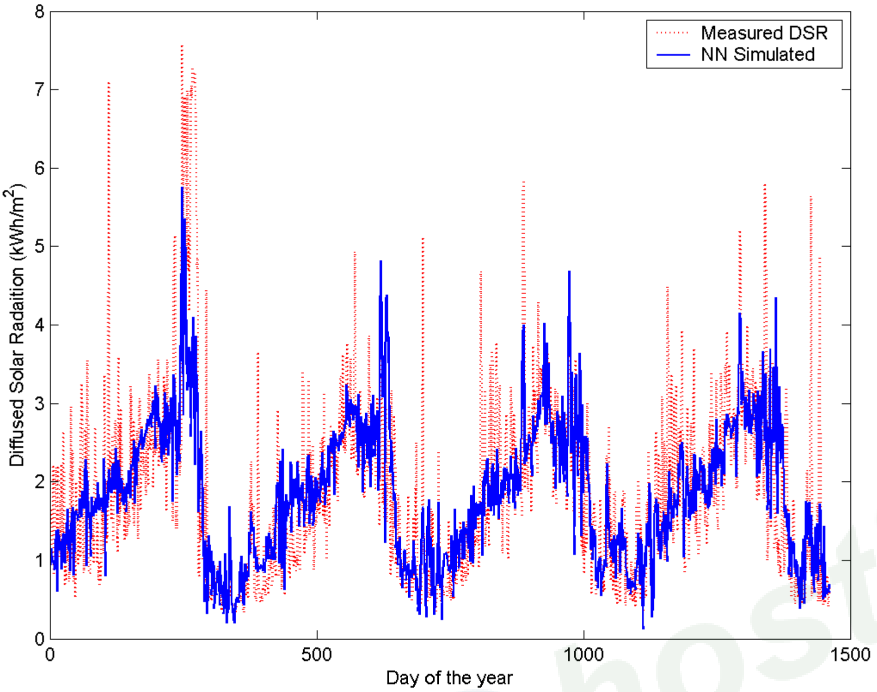


Figure 13. Training results with mean air temperature, relative humidity and day of the year as inputs.

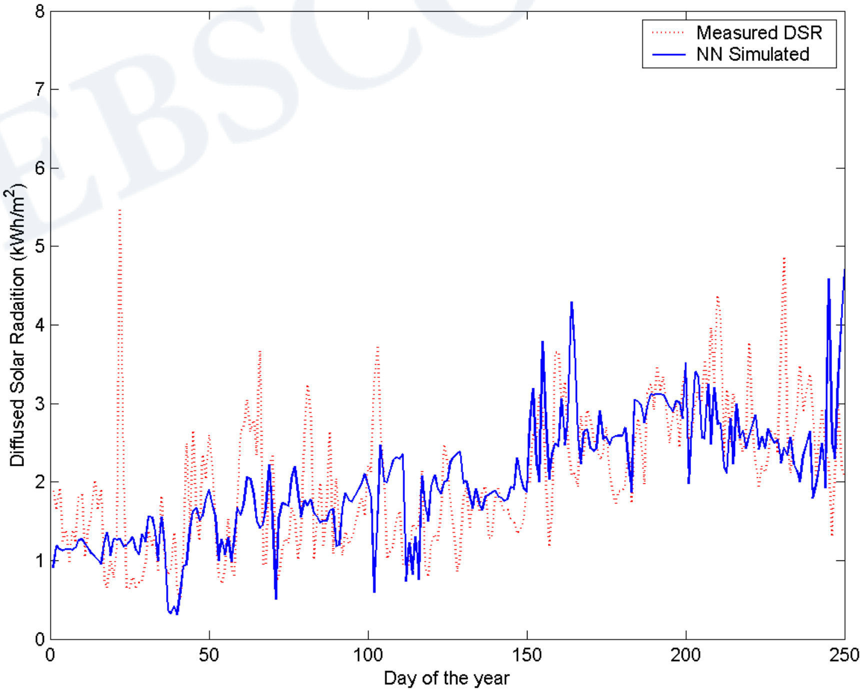


Figure 14. Comparison of measured diffuse solar radiation with estimated values obtained using mean air temperature, relative humidity and day of the year as input to ANN model.

For one case, only the day of the year and daily maximum temperature was used as inputs and GSR as output. In second case, the day of the year and daily mean value of temperature were used as inputs and GSR as output. In the last case, the day of the year, and daily average values of temperature and relative humidity were used to predict the GSR. Results show that using the relative humidity along with daily mean temperature outperforms the other cases with absolute mean percentage error of 4.49%. The absolute mean percentage error for the case when only day of the year and mean temperature were used as inputs was 11.8% while when maximum temperature is used instead of mean temperature is 10.3%.

In first case, only the day of the year and daily maximum temperature was used as inputs and DSR as output. In second case, the day of the year and daily mean value of temperature were used as inputs and DSR as output. In the third case, the daily minimum temperature along with day of the year were used as input to the ANN model and DSR as output. In the last case, the day of the year and daily mean values of temperature and relative humidity were used to predict the DSR. Results show that using the relative humidity along with daily mean temperature outperforms the other cases with mean square error (MSE) of $5.18\text{x}10^{-7}$. The MSE for the case when only day of the year and mean temperature were used as inputs was $2.36\text{x}10^{-5}$ while when maximum temperature used instead of mean temperature the MSE was $9.3\text{x}10^{-6}$. Lastly, for the case when daily minimum temperature was used as input along with day of the year, the mean square error was found to be $2.06\text{x}10^{-4}$.

For future work, an attempt will be made to split the global solar radiation in to diffused and normal incident solar radiation on horizontal surfaces based on meteorological parameters as well.

ACKNOWLEDGMENT

Authors acknowledge the support provided by the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in conducting this study.

REFERENCES

[1] Dave, J. V. & Norman, B. (1976). Importance of the diffuse sky radiation in evaluation of the performance of a solar cell. *Solar Energy, 1*, 215–23.

[2] Okeke, C. E. & Aunforom, A. C. (1987). Effect of diffuse solar radiation on the efficiency of an array of silicon solar cells. *Solar Energy*, *5*, 105–13.

[3] Riordan, C., Hulstrom, R., Cannon, T. & Myers, D. (1991). Solar radiation research for photovoltaic applications. *Solar Cells, 30*, 489–500.

[4] Elminir, H. K. (2007). Experimental and theoretical investigation of diffuse solar radiation: data and models quality tested for Egyptian sites. *Energy, 32*, 73–82.

[5] Ångström, A. (1924). Solar and terrestrial radiation. *J. Roy. Met. Soc*., *50*, 121–126.

[6] Hussain, M. (1984). Estimation of global and diffuse solar radiation from sunshine duration and atmospheric water vapor content. *Solar Energy*., *33(2)*, 217–220.

[7] Samuel, T. D. M. A. (1991). Estimation of global solar radiation for Sri Lanka. *Solar Energy*., *47(5)*, 333-337.

[8] Akinoglu, B. G. & Ecevit, A. (1990). Construction of a quadratic model using modified Angstrom coefficients to estimate global solar radiation. *Solar Energy*., *45(2)*, 85-92.

[9] Rehman, S. (1998). Solar radiation over Saudi Arabia and comparison with empirical models. *Energy*., *23(12)*, 1077-1082.

[10] Aguiar, R. & Collares-Pereira, M. (1992). A time dependent autoregressive, Gaussian model for generating synthetic hourly radiation. *Solar Energy*., *49*, 167–174.

[11] Elizondo, D., Hoogenboom, G. & McClendon, R. (1996). Development of a neural network to predict daily solar radiation. *Agric. Forest Meteorol*., *71*, 115–132.

[12] Al-Alawi, S. M. & Al-Hinai, H. A. (1998). An ANN-based approach for predicting global solar radiation in locations with no measurements. *Renewable Energy*., *14(1-4)*, 199–204.

[13] Mohandes, M., Rehman, S. & Halawani, T. O. (1998). Estimation of global solar radiation using artificial neural networks. *Renewable Energy*., *14(1-4)*, 170–184.

[14] Toğrul, I. T. & Onat, E. (1999). A study for estimating the solar radiation in Elazığ using geographical and meteorological data. *Energy Conversion and Management*., *40*, 1577–1584.

[15] Mohandes, M., Balghonaim, A., Kassas, M., Rehman, S. & Halawani, T. O. (2000). Use of radial basis functions for estimating monthly mean daily solar radiation. *Solar Energy*., *68(2)*, 161–168.

[16] Hontoria, L., Aguilera, J., Riesco, J. & Zufiria P. J. (2001). Recurrent neural supervised models for generating solar radiation. *J. Intell. Rob. Syst*., *31*, 201–221.

[17] Hontoria, L., Aguilera, J. & Zufiria, P. J. (2002). Generation of hourly irradiation synthetic series using the neural network multilayer perceptron. *Solar Energy*., *75(2)*, 441–446.

[18] Tasadduq, I., Rehman, S. & Bubshait, K. (2002). Application of neural networks for the prediction of hourly mean surface temperature in *Saudi Arabia*. *Renewable Energy*., *25*, 545–554.

[19] Kalogirou, S., Michanelides, S. & Tymbios, F. (2002). Prediction of maximum solar radiation using artificial neural networks. In: *Proceedings WREC VII*, Germany.

[20] Tymvios, F. S., Jacovides, C. P., Michaelides, S. C. & Scouteli, C. (2005). Comparative study of Ångström's and artificial neural networks' methodologies in estimating global solar radiation. *Solar Energy*., *78*, 752–762.

[21] Sözen, A., Arcaklıoğlu, E. & ÖZalp, M. (2004). Estimation of solar potential in Turkey by artificial neural networks using meteorological and geographical data. *Energy Conversion and Management*., *45*, 3033–3052.

[22] Yang, K. & Koike, T. (2002). Estimating surface solar-radiation from upper-air humidity. *Solar Energy*., *72(2)*, 177–186.

[23] Elminir, H. K., Azzam, Y. A. & Younes, F. I. (2007). Prediction of hourly and daily diffuse fraction using neural network, as compared to linear regression models. *Energy* *32*, 1513–1523.

[24] Šúri, M., Huld, T. A. & Dunlop, E. D. (2005). PV-GIS: A web-based solar radiation database for the calculation of PV potential in Europe. *International Journal of Sustainable Energy, 24(2)*, 55–67.

[25] Soares, J., Oliveira, A. P., Boznar, M. Z., Mlakar, P., Escobedo, J. F. & Machado, A. J. (2004). Modeling hourly diffuse solar-radiation in the city of Sao Paulo using a neural-network technique. *Applied Energy*, *79*, 201–214.

[26] Sozen, A., Arcakhoglu, E., Ozalp, M. & Çağlar, N. (2005). Forecasting based on neural network approach of solar potential in Turkey. *Renewable Energy, 30(7)*, 1075-1090.

[27] Sozen, A. & Arcakhoglu, E. (2005). Effect of humidity on solar potential. *Applied Energy, 82*, 345-367.

[28] Aras, H., Balli, O. & Hepbasli, A. (2006). Estimating the horizontal diffuse solar radiation over the Central Anatolia Region of Turkey. *Energy Conversion and Management, 47*, 2240–2249.

[29] Tarhan, S. & Sar, A. (2005). Model Selection for global and diffuse radiation over the Central Black Sea (CBS) Region of Turkey. *Energy Conversion and Management, 46*, 605–613.

[30] Paliatsos, A. G., Kambezidis, H. D. & Antoniou, A. (2003). Diffuse solar irradiation at a location in the Balkan Peninsula. *Renewable Energy, 28*, 2147–2156.

[31] Amauri, P. O., Escobedob, J. F., Machadoa, A. J. & Soaresa, J. (2002). Correlation models of diffuse solar-radiation applied to the city of Sao Paulo, Brazil. *Applied Energy, 71*, 59–73.

[32] Jin, Z., Yezheng, W. & Gang, Y. (2004). Estimation of daily diffuse solar radiation in China. *Renewable Energy, 29*, 1537-1548.

[33] Rensheng, C., Ersi, K., Jianping, Y., Shihua, L., Wenzhi, Z. & Yongjian, D. (2004). Estimation of horizontal diffuse solar radiation with measured daily data in China. *Renewable Energy, 29*, 717–726.

[34] El-Sebaii, A. A. & Trabea, A. A. (2003). Estimation of horizontal diffuse solar radiation in Egypt. *Energy Conversion Management, 44*, 2471–2482.

[35] Roderick, M. R. (1999). Estimating the diffuse component from daily and monthly measurements of global radiation. *Agricultural Forest Meteorology, 95*, 169–185.

[36] Rehman, S. (1998). Solar radiation over Saudi Arabia and comparison with empirical models. *Energy, 23(12)*, 1077 – 1082.

[37] Rehman, S. & Ghori, S. G. (2000). Spatial estimation of global solar radiation using Geostatistics. *Renewable Energy, 21(3-4)*, 583-605.

[38] Rehman, S. (1999). Empirical model development and comparison with existing correlations. *Applied Energy, 64(1-4)*, 369-378.

[39] Rehman, S. & Halawani, T. O. (1997). Global solar radiation estimation. *Renewable Energy, 12(4)*, 369-385.

[40] Aksakal, A. & Rehman, S. (1999). Global solar radiation in Northeastern Saudi Arabia. *Renewable Energy*, *17*, 461-472.

[41] Rehman, S. & Halawani, T. O. (1998). Development and utilization of solar energy in Saudi Arabia – A Review. *The Arabian Journal for Science and Engineering,* 23(1B), 33-46.

[42] Mohandes, M., Rehman, S. & Halawani, T. O. (1998). Estimation of global solar radiation using artificial neural networks. *Renewable Energy, 14(1-4)*, 170–184.

[43] Mohandes, M., Balghonaim, A., Kassas, M., Rehman, S. & Halawani, T. O. (2000). Use of Radial Basis Functions for Estimating Monthly *Mean Daily Solar Radiation, Solar Energy*, *68(2)*, 161-168.

[44] Tasadduq, I., Rehman, S. & Bubshait, K. (2002). Application of Neural Networks for the Prediction of Hourly Mean Surface Temperature in Saudi *Arabia, Renewable Energy*, *25*, 545-554.

[45] Rehman, S. & Mohandes, M. (2008). Artificial Neural Network Based Estimation of Global Solar Radiation using Air Temperature and Relative Humidity. *Energy Policy Journal*, *v. 36*, 571-576.

[46] Rehman, S. & Mohandes, M. (2009). Estimation of Diffuse Fraction of Solar Radiation using Artificial Neural Networks. Energy Sources Part A: Recovery, *Utilization, and Environmental Effects*, *v. 31*, 974-984.

[47] Wasserman, P. (1993). *Advanced methods in neural computing*. Van Nostrand Reinhold, New York.

[48] Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE ASSP Mag.*, 4-22.

[49] Bishop, C. M. (1996). *Neural networks for pattern recognition*. Oxford University Press.

[50] Pham, D. T. & Liu, x. (1995). *Neural networks for identification, predication and control* Springer-Verlag, London.

[51] Rumelhart, D. E., McClelland, J. L. & PDP Research Group, (1986). *Parallel distributed processing*. The MIT Press.

[52] Haykin, S. (1984). *Neural networks: A comprehensive foundation*. Macmillan College Publishing.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 11

EFFECTS OF MICRO PORE CHARACTERISTICS ON STRENGTH OF CEMENT MORTAR USING ARTIFICIAL NEURAL NETWORK

***Ali Ugur Ozturk*[1] *and Okan Onal*[2]**
[1]Department of Civil Engineering, Faculty of Engineering,
Celal Bayar University, Manisa, Turkey
[2]Department of Civil Engineering, Faculty of Engineering,
Dokuz Eylul University, Izmir, Turkey

ABSTRACT

Cementitious materials comprise a great part in construction process of structures such as buildings, bridges, roads and dams. The most expected properties of structural members prepared with cement mortar or concrete, are strength and durability. These structural members are supposed to have strength values determined in the structural analysis and to be durable against aggressive media in their service life. These characteristics are the most effective criteria in civil and material engineering. Therefore, these two parameters depend mainly on the pore structure and its characteristics of structural members. Nowadays, scientists and engineers are using new computer technologies, simulations and experimental techniques try to perform to characterize the inner structure of structural materials in order to define microstructural formations and the effects of microstructural phases such as pores on macro properties.

New image capturing tools and their improved magnification capacity induced researchers to have an expanded view on investigation of microstructures. In addition, the results of these studies are simply not enough to realize the simulation of effects of inner structure. Some numerical and statistical methods performed by computers are needed at this stage. Artificial neural network (ANN) is one of these methods. In last decades, artificial neural network applications have become more considerable issue in engineering applications. In the scope of this chapter, pore area ratio values represent total pore area amount in a polished section of cement mortars were determined. Also, some pore characteristics representing the probability of channels between pores are investigated. The pore amounts and these pore characteristics are related to compressive strength values of cement mortars in order to establish a microstructure – macro property relationship. Thus, nondestructive methodologies and artificial neural network have been used in the prediction of a macro property, which only be determined by destructive testing techniques.

INTRODUCTION

Since the first experimental study in the literature, experimental studies have become more distinctive in order to find out scientific results. Thus, only the experimental or even results obtained by theorical studies are not enough to simulate the required information. Hence scientists and researchers have needed to explain these results; some mathematical and numerical methods are required, since ever. Artificial neural network and fuzzy logic analysis can be given as examples of these methods.

Both experimental and theoretical studies on construction materials are also required these methods, currently. Nowadays, scientists or researchers perform these mathematical and numerical studies in order to simulate the tendency or behaviour of a construction material under the survey conditions. Cementitious materials such as concrete, cement mortar and cement paste are several ones of these materials.

Cementitious materials are common and almost the most used materials in the production of structural members. Thus cement is the widespread used construction material; the usage limits, the performance and the tendency under aggressive media of structural members comprised with cement are not well known. The performance in the service life of a structural member is the most important property. Indeed, the performance including durability and strength characteristics has a great role in service life. Durability characteristics indicating the resistance against aggressive media such as acidic condition, cold and hot weather, biological and physical aggressive conditions; and strength characteristics such as compressive and flexural strengths depend on water/cement ratio values of the mix, pore structure of structural members, hydration degree of cement, etc. Furthermore the knowledge of mix design properties needed for producing these structural members has a remarkable role on the performance in service life.

Concrete is among the most used cement based materials in construction. In designing a concrete structure, one of the most important properties which have to be considered, besides the ability of the structure to resist all loads, is its durability. The service life and durability of a concrete structure strongly depend on its material transport properties, such as permeability, sorptivity, and diffusivity which are controlled by the microstructural characteristics of concrete [1]. It is known that the porosity and pore characteristics are the critical components of the microstructure of hydrated cement paste that influence durability and strength. In order to achieve high strength, low permeability, and durable concrete, it is therefore necessary to reduce the porosity of cement paste [2].

Whether conventional or modern, construction process has remarkable effects on the performance of structural members in their service life. Nowadays, cement is the most used materials in these construction periods. However forming structural members comprising cementitious materials requires some procedures such as using vibrators and having proper curing technologies. Such construction processes lead side engineers to prepare durable structural members with more strength.

Thus, only these technologies are not enough to attain desired performance of structural members. Suitable designs and selection of qualified materials should be taken into account, at this manner. From this point, the mix design of cement based structural members has a remarkable effect on the performance. Not only the amounts of the ingredients, but also; the base of chemical or mineral additives, the curing conditions, transportation and placing procedures are also important at performing the best microstructure affecting on macro properties such as durability and strength. So, in order to realize and determine these performances, scientists and side engineers need to get the knowledge of mix design and properties such as mineral additives amount, w/c ratio or presence chemical admixtures of concrete members cast before.

The selection of mix proportions is the process of choosing suitable ingredients of concrete and determining their relative quantities with the object of producing as economically as possible concrete of certain minimum properties, notably strength, durability, and a required consistency [3]. Concrete mix design involves complicated issues, and the correct ways to perform this can be achieved with experts' advice and experience [4].

Simply just materials should also be considered in the view of strength and durability of cementitious materials before or during casting. Some properties of cementitious materials such as hydration heat, expansion limits, own strength, flowability have influences on the ultimate performance of structural members.

To improve the properties and the performance of materials, microstructural investigations and studies have become more featured issues in civil and materials engineering areas. Therefore new experimental technologies, improvement of computer capabilities and software systems have lead scientists to perform these studies more easily. Currently, microstructure of cementitious materials can be investigated, and so the microstructural properties resulted by these microstructural analyses can be quantified. Porosity and inner phases such as undifferentiated hydration products and unhydrated cement content can be quantified by using image capturing techniques such as scanning electron (SEM) or optical microscopy, and image analysis of micrographs of materials [5-8].

Currently, engineers and scientists have opportunities to perform some new experimental techniques such as microstructural analysis using optical or scanning electron microscopy. Results obtained by these techniques lead civil engineers and scientists to understand the microstructure of cement based materials such as concrete more realistic. Microstructural characterizations are required for detailed inner structure analysis and allow modifications of macro properties. Furthermore, microstructure of cementitious materials has become an increasing scrutiny due to the recent developments in microstructural investigations by using computational analysis techniques. Image analysis of micrographs of cementitious materials bring benefits for quantify microstructural characteristics of cement pastes such as porosity, pore structure and phases which includes undifferentiated hydration products and anhydrous cement content [5-12]. Developments of such phases and their effects on the performance (strength and durability characteristics) in the service life can be researched.

Although inner structure of cementitious and other construction materials can be explained easily and in detail, just these microscopic results are not enough to understand and realize the behavior of structural members in their service life. Therefore relationship between these microstructural properties and macro properties such as strength and durability characteristics are required by engineers and scientist to model and simulate the effects of inner structure on the performance. Currently, some analysis method such as artificial neural network and fuzzy logic are being widely used to solve a wide variety of problems in civil engineering applications [13-16]. The importance and applications of these techniques increased significantly during last years [17]. Therefore ANNs are successfully used in research studies and civil engineering applications.

The inspiration for applying ANNs to different branches of engineering sciences comes from the analysis of transmission and transformation of signals in human and animal neural system. The importance of this technique increased significantly during last years [17]. Nowadays, ANNs are successfully used in computer-aided management, system identification, modeling of different physical dynamic processes depending on several fuzzy variables. Further, many revolutionary ideas have been proposed by those authors who introduced the technique of soft computing in the field of numerical modeling of mechanics [17].

Artificial neural networks are based on the present understanding of biological nervous system, although much of the biological detail is neglected. ANN are enormously parallel systems composed of many processing elements connected by links of variable weights [18-19]. Generally, ANN are made of an input layer of neurons, sometimes referred to as nodes or processing units, one or several hidden layer of neurons and output layer of neurons. The neighboring layers are fully interconnected by weight. The input layer neurons receive information from the outside environment and transmit them to the neurons of the hidden layer without performing any calculation [20-21]. Layers between the input and output layers are called hidden layers and may contain a large number of hidden processing units [22]. All problems, which can be solved by a perception can be solved with only one hidden layer, but it is sometimes more efficient to use two or three hidden layers. Finally, the output layer neurons produce the network predictions to the outside world [20-21].

In the scope of this chapter, microstructure-macro property relationship of cement mortars has been established in order to define the effects of pore structure and its characteristics on strength. Microstructural studies have been become great issue in materials engineering. Nowadays, to characterize the microstructural phase properties and to improve and modify them are performed by scientist for forecasting and enhancing. According to this objective, cement mortars incorporating with chemical admixtures were prepared to constitute different microstructural graphs. These micro graphs were analyzed to determine the amounts of pore amount (pore area ratio) and the quantitive pore structure characteristics such as dendrite length, total pore length and average roundness of pores. Afterwards, the amounts of these microstructural properties were related to strength values of each cement mortar specimen.

The relationship was established by using artificial neural network analysis between microstructural characteristics and compressive strength values of cement mortar. Artificial neural network analysis indicated that by using ANN as non-linear statistical data modeling tool, a strong correlation between the microstructural properties of cement mortar and compressive strengths can be established. The study indicated that using a contemporary data analysis technique, which is capable of searching nonlinear relationships more thoroughly, would result more realistic relationship between strength and pore characteristics.

EXPERIMENTAL

In the scope of this study, the relationship between pore characteristics and compressive strength was established by using ANN analysis. In order to establish a relationship between pore characteristics and compressive strength values using ANN, axial compressive strength test and microstructural investigation under SEM have been performed. Cement mortars were prepared by incorporation of four different types of chemical admixtures. Two types of admixtures are lignin based modified polymer, and others are naphthalene sulphonate based and modified phosphate based admixtures. All types of chemical admixtures have retardation effect on the hydration of cement mortar and cause different pore structure formations. Compressive strength properties of cement mortars at the ages of 1, 2, 7, 28 and 90 days were found on the 50 mm cube.

On the other hand, microstructural studies require qualified expert practice. Better micrographs and results can only be obtained by careful execution of microstructural investigation procedures. To minimize the effects of cutting specimens, some of the mixtures were cast into plastic containers and compacted in two equal layers. After 1, 2, 7, 28 and 90 days, hydration processes of the mixes were delayed by submerging specimens into alcohol isopropyl alcohol for five days. Before microstructure studies, the specimens were covered by a polyester film, and then the surface of each specimen was polished. Each specimen was sanded by 600 and 1200 grid sandpapers. After sanding, each specimen was polished by 0.25, 1, 3 and 9 µm diamond paste for 120 s [23].

The phases in polished sections of cementitious materials should be segmented with accuracy and consistency. Accurate analyses lead to meaningful quantitative data that can be used for comparative studies and to characterize a relationship between microstructure and mechanical behavior. The phases in a polished section can be segmented by their gray level thresholds in the micrograph. The gray level values of phases compose separate peaks in the gray level histogram with their heights proportional to the relative fractions of each phase.

Furthermore, during imaging under SEM, brightness and contrast setting were done, and the histogram of each micrograph was conducted to be centered and stretched to span the whole dynamic range of the current gray scale. The fractions of microstructural phases (were determined by image analysis on backscattered electron image (Figure 1.) taken by SEM at 500X magnification [24]. After the suitable settings for brightness and contrast were found, all images were taken under these arrangements to avoid errors by imaging process.

Image analyses comprise investigation of pore structure and its characteristics. Total pore length and total dendrite length indicate total length of all pores and total dendrite length of all pores, respectively (Figure 2.b-c). Average roundness values depend on area and perimeter of pores. The aim to determine the pore characteristics such as total pore length, total dendrite length and average roundness is to investigate the probability of being branching between pores.

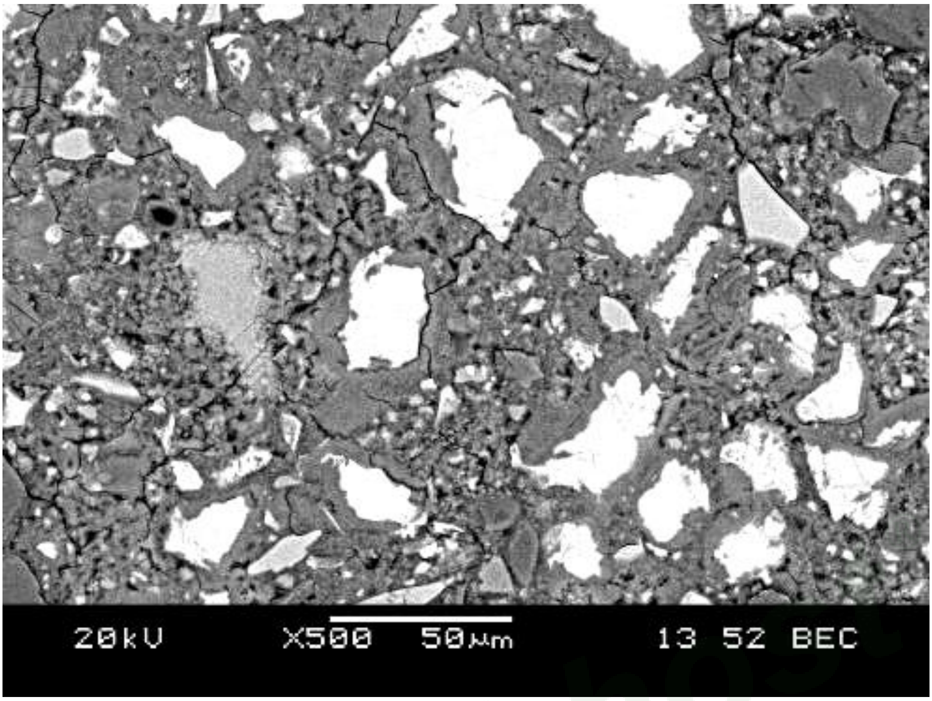
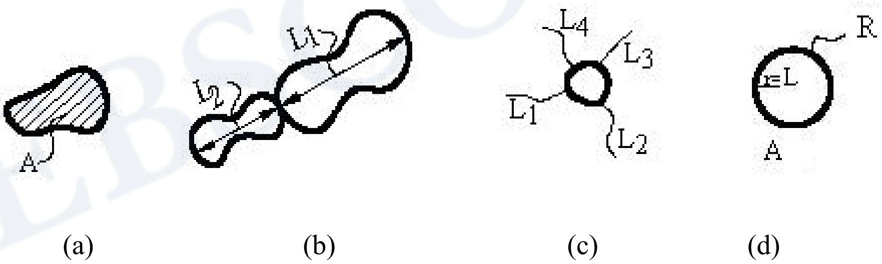


Figure 1. Micrographs of cement mortar by 500X magnification.



(a) (b) (c) (d)

Figure 2. Pore characteristics (a) pore area ratio, (b) total pore length, (c) total dendrite length, and (d) roundness.

ARTIFICIAL NEURAL NETWORK ANALYSIS

ANNs are non-linear statistical data modeling tools, which may be used to model complex relationships between inputs and outputs. This technique allows investigation of the relationship between several visual features of cement mortar and compressive strength by simulating the structure of biological neural networks. ANN consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. ANN is an adaptive system that changes its structure based on information that flows through the network during the learning phase.

In function approximation problems feed forward back propagation technique is the preferred method [25]. Therefore, algorithm of this neural network approach has been employed in the prediction of the compressive strength values of the cement mortar specimens.

The results of the compression tests and the features derived from the digital image processing operations have been used as the data set in the neural network analysis. The half of the data set (i.e. belonging to 65 test specimens) has been employed as the training set in the supervised learning process. Since the success of the network analysis is evaluated by an iterative procedure, the training set has been kept constant in the setup of the network topology as the odd numbered data rows. The remaining data have been used as the test set for the verification of the neural network. Thus, randomized data selection has been avoided in order to compare the individual runs of the algorithm. However, since the initial weight factors are assigned in a randomized manner, each individual run of the training algorithm shows slightly varying results.

On the other hand researchers are developing different data selection methodologies in order to improve the success of the ANN applications. The performances of the well known data selection methodologies including holdout, random sub-sampling, k-fold cross validation, leave-one-out and bootstrap are investigated by researchers [26].

Once the most appropriate training data has been selected, it must be preprocessed; otherwise, the neural network will not produce accurate predictions. The decisions made in this phase of development are critical to the performance of a network.

Normalization and principal component analysis are two widely used preprocessing methods. Knowledge of the domain is important in choosing preprocessing methods to highlight underlying features in the data, which can increase the network's ability to learn the association between inputs and outputs. Normalization involves a transformation performed on a single data input to distribute the data evenly and scale it into an acceptable range for the network. Since the input data set consists of input values that range between different upper and lower limits, normalization process has been applied to the input data so that its mean and standard deviation are set as 0 and 1, respectively. This preprocessing operation maps the data in a new form, more suitable to train the network.

The success of the neural network analysis has been increased by performing principle component analysis, which is a common technique for finding patterns in data of high dimension by capturing the variance in a data set in terms of principle components. The goal of the principal component analysis is to find an orthogonal set of vectors that maximize the variance of the projected data. Principal component analysis is a linear transformation of the data into another frame of reference with as little error as possible in order to extract relevant information from high dimensional data sets.

The preprocessed data set has been used in the back propagation algorithm offered by the neural network toolbox of the MatLab Technical Computing Language. The topology of the network has been organized to consist of one hidden layer with ten neurons, having tangent sigmoid function, and one output layer, having linear function as in the general function approximation applications.

The training has been conducted until maximum epoch, predefined as 250, was reached. The maximum epoch has been determined as 250, because, a decrease in errors over the training data does not mean a decrease over novel data and in fact may lead to less generalization, which is called as over-learning. Over-learning results in an over fitted function, which has good approximation for the training set, but poor approximation for the test set.

After the completion of the network training process, test data set have been normalized using the pre-calculated mean and standard deviation of the training set. The normalized test data set have been preprocessed by applying the same principal component transformation that was previously computed for the training data. Hence, the test data has been preprocessed in order to use in the trained network.

The simulation of the network has been conducted by using the preprocessed test data. However, the performance of the network can only be evaluated by post processing the output of the network. Therefore, the same transformation matrices used in the preprocessing of the train and test data (i.e. principle component analysis and then normalization), have been used in the post processing operations. Hence, the output of the network has been converted to the same quantity, as the raw data before the preprocessing of the data set (Figure 3).

The performance of the network has been evaluated calculating the coefficient of determination for the predicted and original values of the data set (Figure 4). Please notice that the training set data has been also simulated by using the established network and outputs have been marked in the same figure.

ANN analysis indicates that a good agreement between microstructural pore characteristics and compressive strengths of cement mortars can be establish by using backpropagation algorithm. The coefficient of correlation between the measured and predicted compressive strength values is calculated as R^2=0,9948 representing a strong relationship for the investigated parameters.

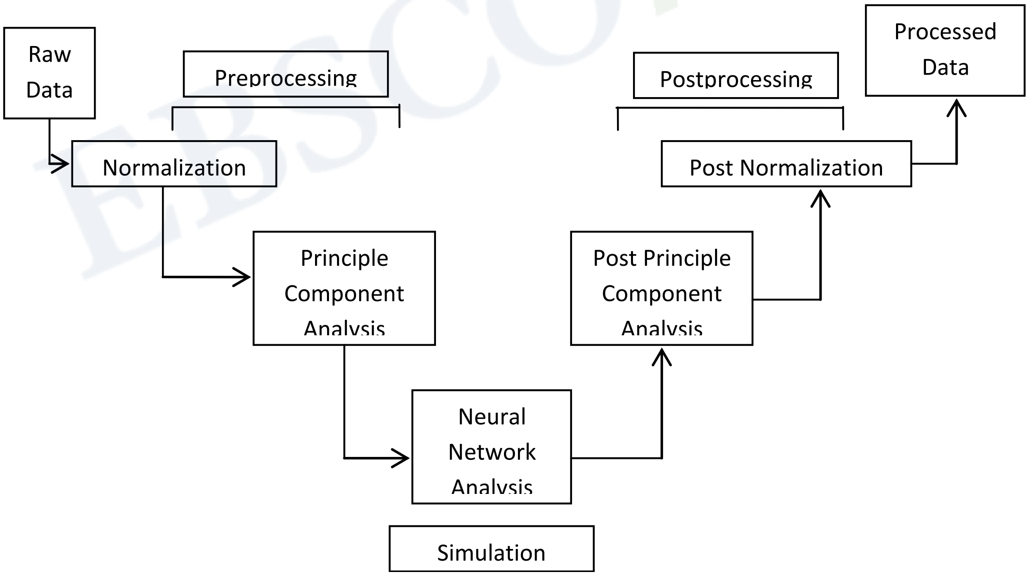


Figure 3. Schematic presentation of the data analysis process.

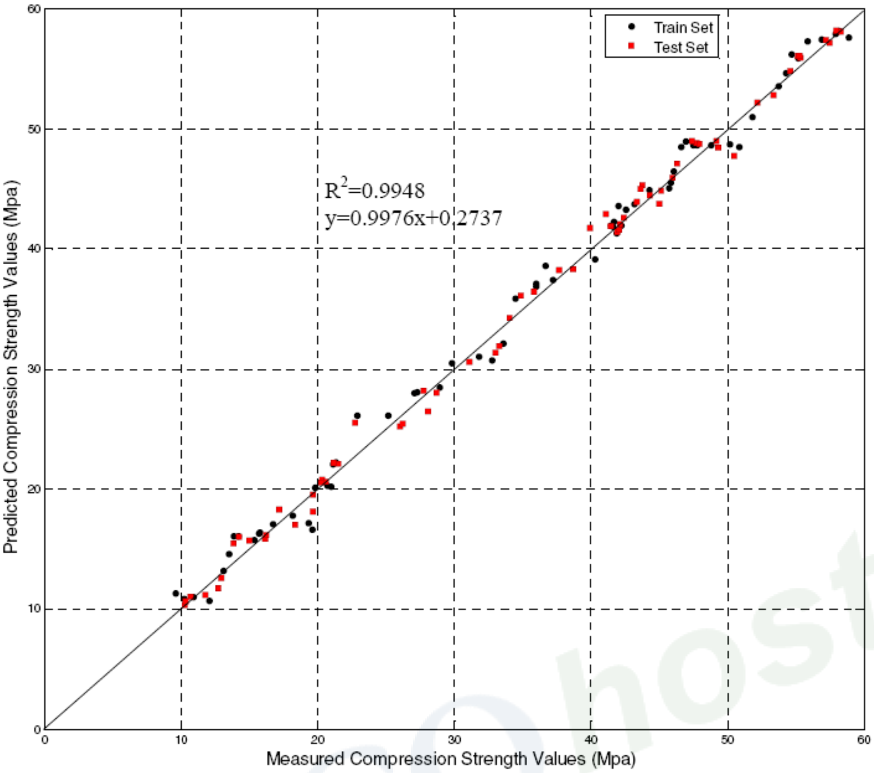


Figure 4. Relationship between predicted and measured values of compressive strength of cement mortar.

CONCLUSION

Compressive strength of cementitious materials is highly affected by pore structure and hydration process of cementitious materials. However, pore structure and pore characteristics are deserved more attention compared to hydration degree of a cement based structural member. Likewise it is obvious that cement based materials having less pore structure may have more strength compared to more porous cement based materials.

According to this view, pore structure and its characteristics have a remarkable role in the effects of microstructural phases on strength. This pore structure must be investigated more in detail. Nowadays, researchers and scientists have a great potential on performing the required microstructural analyses due to the improved experimental tools and technologies. They have capacity to improve microstructural characteristic for modifying the macro properties such as compressive strength.

In the last decade, image analysis techniques have been improved by using new image capture technologies such as SEM and computer capabilities. Results obtained by these techniques lead civil engineers and scientists to understand the microstructure of cement based materials such as concrete more realistic. Current image analysis techniques performed on micrographs of cement based materials have a great capacity to simulate the microstructure.

Total pore length, total dendrite length, average roundness and pore area ratio being microstructural properties of cement mortars have been investigated by using digital image analysis methods. These microstructural properties have been related to compressive strength being macro property. The artificial neural network analysis technique has been employed at the analysis stage of the database, which features are extracted by using digital image analysis techniques.

It has been concluded that by using ANN as non-linear statistical data modeling tool, a good agreement between the microstructural pore properties of cement mortar and compressive strengths can be established. Thus a macro property, which only be determined by destructive testing techniques, may be estimated with a great accuracy (R^2=0.9948) by using pore micro properties of cement mortar, obtained by nondestructive methodologies.

REFERENCES

[1] Martys, NS. Survey of concrete transport properties and their measurement. *NISTIR* 5592, US Department of Commerce, 1996, 1–40.

[2] Prinya, C; Chai, J; Theerawat, S. Effect of fly ash fineness on compressive strength and pore size of blended cement paste. *Cement & Concrete Composites*, 2005, 27, 425–428

[3] Neville, AM. *Properties of concrete*, fourth ed., Longman Group Limited, Essex, 1995.

[4] Bai, Y; Amirkhanian, SN. *J. Cons. Eng. Manage*., ASCE, 1994, 120(2) , 357–373.

[5] Scrivener, KL; Patel, HH; Pratt, PL; Parrott, LJ. Analysis of phases in cement paste using backscattered electron images. In: LJ; Struble, PW; Brown, editors. Microstructural development during hydration of cement. *Mater Res Soc Symp Proc*, 1987, 85, 67–76.

[6] Zhao, H; Darwin, D. Quantitative backscattered electron analysis for cement paste. *Cem Concr Res,* 1992, 22, 695–706.

[7] Kjeilsen, KO; Detwiler, RJ; Gjùrv, OE. Backscattered electron image analysis of cement paste specimens: Specimen preparation and analytical methods. *Cem Concr Res*, 1991, 21, 388–90.

[8] Lange, DA; Jennings, HM; Shah, SP. Image analysis techniques for characterization of pore structure of cement based materials. *Cem Concr Res,* 1994, 24, 841–53.

[9] Wang, Y; Diamond, S. An approach to quantitative image analysis for cement pastes. *Mater Res Soc Symp Proc*, 1995, 370, 23–32.

[10] Darwin, D; Abou-Zeid, MN. Application of automated image analysis to the study of cement paste microstructure. *Mater Res Soc Symp Proc,* 1995, 370, 3–12.

[11] Diamond, S; Leeman, ME. Pore size distributions in hardened cement paste by SEM image analysis. *Mater Res Soc Symp Proc,* 1995, 370, 217–26.

[12] Barrioulet, M; Saada, R; Ringot, E. A quantitative structural study of fresh cement paste by image analysis: part 1. *Image processing. Cem Concr Res*, 1991, 21, 835–43.

[13] Wong, HS; Buenfeld, NR. Determining the water-cement ratio, cement content, water content and degree of hydration of hardened cement paste: Method develop and validation on paste samples. *Cem. Concr. Res*., 2009, 39(10), 957-965.

[14] Specht, DF. A general regression neural network. *IEEE Transactions on Neural Networks*, 1991, 2(6), 568–576.

[15] Kisi, O; Cigizoglu, HK. Comparison of different ANN techniques in river flow prediction. *Civil Engineering & Environmental Systems*, 2007, 24(3), 211-231.

[16] Cigizoglu, HK. Application of the Generalized Regression Neural networks to Intermittent Flow Forecasting and Estimation, *ASCE J Hydrol Eng*, 2005, 10(4), 336-41.

[17] Lefik, M; Boso, DP; Schrefler, BA. Artificial neural networks in numerical modelling of composites. *Comput Methods Appl Mech Eng*, 2009, 198, 1785–804.

[18] Lefik, M; Boso, DP; Schrefler, BA. Artificial neural networks in numerical modelling of composites. *Comput Methods Appl Mech Eng*, 2009, 198, 1785–804.

[19] Demirbas_ AE. Modelling of seismically excited structures using ANN, *Master Thesis,* METU, Ankara; 1998.

[20] Dias, WPS; Pooliyadda, SP. Neural networks for predicting properties of concretes with admixtures. *Constr Build Mater*, 2001, 15, 371–9.

[21] Caglar, N. Artificial neural networks in dynamic analysis of buildings. Ph.D. *Thesis*, Sakarya University, 2002.

[22] Bai, J; Wild, S; Ware, JA; Sabir, BB. Using neural networks to predict workability of concrete incorporating metakaolin and fly ash. *Adv Eng Softw*, 2003, 34, 663–9.

[23] Stutzman, PE; Clifton, JR. Sample preparation for scanning electron microscopy. In: Proceedings of 21st international conference. Cement microscopy. *Las Vegas*, 1999, 10-22.

[24] Scrivener, KL. Backscattered electron imaging of cementitious microstructures: understanding and quantification, *Cem. Concr. Compos*., 2004, 26(8), 935–945.

[25] Werbos, PJ. *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*. New York: John Wiley and Sons, 1994.

[26] Alper, M; Emre, S; Okan, A; Olcay, OA. "Predicting Uniaxial Compressive Strengths of Brecciated Rock Specimens using Neural Networks and Different Learning Models", 23rd International Symposium on *Computer and Information Sciences, ISCIS' 08,* October 27-29, Istanbul, Page(s): 1 - 4 , 2008.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 12

TRACING THE DRAINAGE DIVIDE: THE FUTURE CHALLENGE FOR PREDICTIVE MEDICINE

Enzo Grossi[*]
Medical Department, Bracco Spa Milan, Italy

ABSTRACT

The concept of the drainage divide in a flat country can be taken as a metaphor of health vs disease in a medical setting. In the medical landscape we could consider spring location as the equivalent of genetic background at birth. Rivers become people's life trajectories. The seas the rivers ultimately flow into are the outcomes, for example healthy aging vs. premature death due to chronic disease. Depending on the genetic background (starting point), the influence of life events or particular life styles (hills, peaks, ridges) on the fate of the person (river) would be negligible or determinant in inducing a particular trajectory to a specific outcome.

If the principal aim of predictive medicine is to predict the future direction of a person's life in terms of health on the basis of available data, then in this metaphor the problem is to predict what direction the river will take, given its spring location in a particular country and the environmental data available. Two rivers whose "springs" are very close to each other can easily flow in opposite directions.

At present, despite the incredible development of genetic testing technologies, defining the role of genetic predisposition of a subject in determining future outcome still is an elusive target.

In other words, at present, with the exception of extreme situations, we are not able to translate efficiently the precise location of individual springs – in the prediction of the river path . The essay discuss how the use of newer mathematical approaches like those inherent to artificial neural networks environment the next future could really offer the chance to trace the health – disease drainage divide. This is the future challenge for predictive medicine.

[*] Corresponding author: Bracco Spa Medical Department, XXV Aprile 4 20097 San Donato Milanese, Italy, Phone: +39-02-21772274, E-mail address: enzo.grossi@bracco.com

Introduction

A drainage divide, water divide, divide or (outside North America) watershed, is the line separating neighboring drainage basins (catchments). In hilly country, the divide always lies along topographical peaks and ridges, but in flat country (especially where the ground is marshy) the divide may be invisible – just a more or less notional line on the ground, on either side of which falling raindrops will start a journey to different rivers, and even to different sides of a region or continent.

The American continental divide is known worldwide. However, nobody ever considers the presence of a European continental divide. In America the subdivision is obvious (Atlantic and Pacific coasts, though nobody ever talks about the Pacific and Arctic divide!), whereas in Europe it is not. The European subdivision might be between the two largest water bodies bordering the subcontinent: the Atlantic Ocean and Mediterranean sea.

This physical (hence, objective) subdivision is interesting per se, since some countries considered Mediterranean actually lie within the Atlantic basin (Portugal fully and Spain mostly), whereas others considered central European actually lie within the Mediterranean basin (Hungary fully, Slovenia and Austria mostly). Even Germany has a big fraction of its area within the Mediterranean watershed. Figure 1 depicts the European drainage divides.

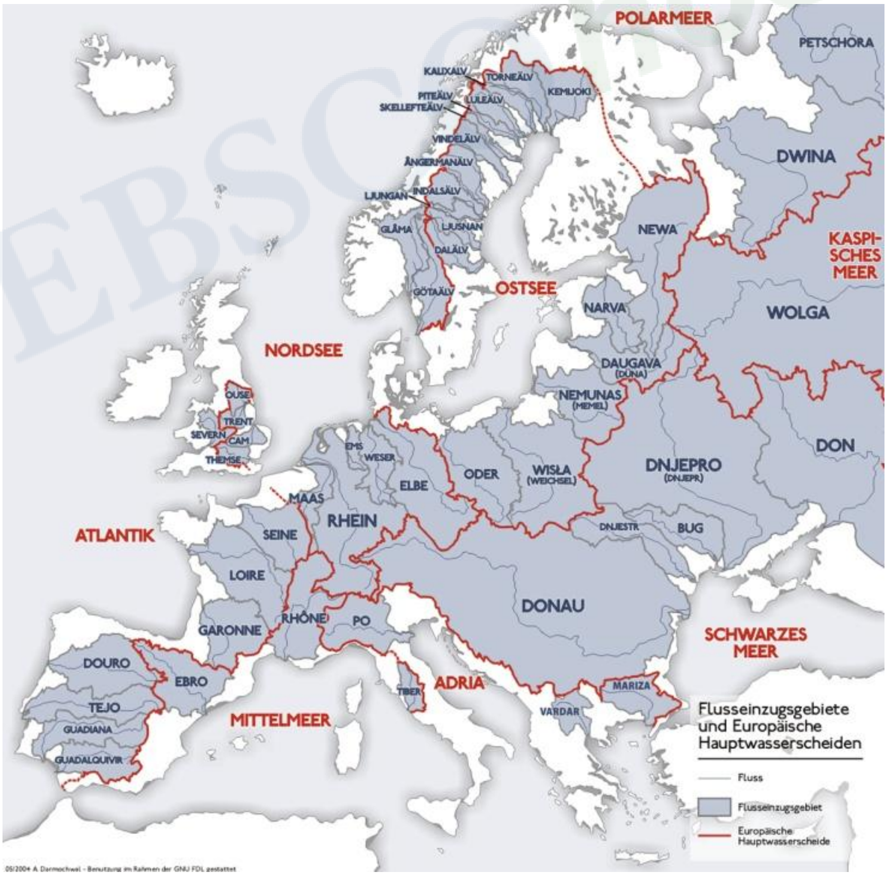


Figure 1. European drainage divides.

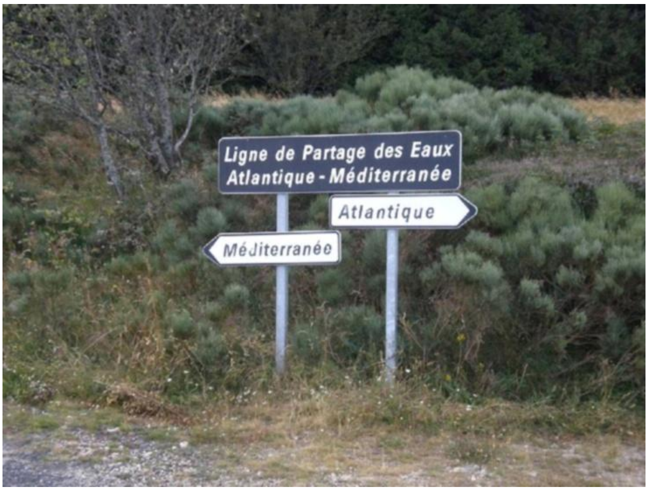


Figure 2. Ligne de partage des eaux.

Let’s take, for instance, the French divide. “ La ligne de partage des eaux” crosses the country from North to South. On the left of the divide all the water goes towards the Atlantic Ocean, while on the right of the divide the water flows towards the Mediterranean sea or again towards the Atlantic Ocean according to the crossing of another divide involving Germany (figure 2). A similar scheme applies to Spain .

Not knowing the existence of this line, it would be very difficult to predict the direction of water from two springs close to the divide, one a few meters to the left and the other a few meters to the right, even taking into account the detailed topography of the country in 3D obtained with sophisticated GIS technologies. Springs placed at a distance of just a few meters from each other could give origin to rivers with an opposite fate: one flowing into the Atlantic Ocean and the other into the Mediterranean sea.

The Drainage Divide Metaphor in Medical Settings

The concept of the drainage divide in a flat country can be taken as a metaphor of health vs disease in a medical setting. In the medical landscape we could consider spring location as the equivalent of genetic background at birth. Rivers become people’s life trajectories or, in physical terms, their non linear “universe lines”, i.e. the continuous unidimensional set of events in the history of a subject. The seas the rivers ultimately flow into are the outcomes, for example healthy aging vs. premature death due to chronic disease.

If the principal aim of predictive medicine is to predict the future direction of a person’s life in terms of health on the basis of available data, then in this metaphor the problem is to predict what direction the river will take, given its spring location in a particular country and the environmental data available. Two rivers whose “springs” are very close to each other can easily flow in opposite directions (figure 3). One river can flow towards the Mediterranean sea (positive outcome; healthy ageing) while the other may flow towards the Atlantic Ocean (negative outcome: e.g. cancer).

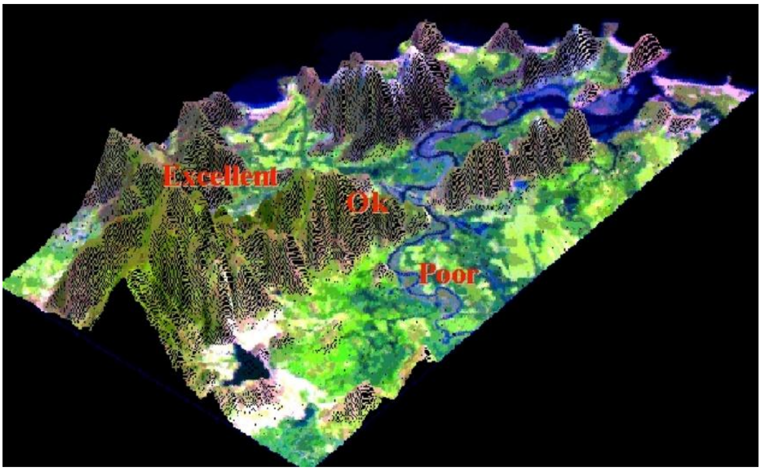


Figure 3. Fitness landscape in which two life trajectories at a certain point take two opposite directions, one towards excellent fate, and the other towards poor fate.

Considering springs as genetic predisposition, one could assume that a spring very near the Mediterranean sea would be associated with a high probability of a Mediterranean outcome. Given that starting point, the influence of life events or particular life styles (hills, peaks, ridges) on the fate of the person (river) would be negligible. Take, for instance, Winston Churchill: despite the fact that he was a legendary lazy glutton with a BMI of 39 and a heavy smoker, he died peacefully in his sleep at the age of 90.

The same applies to a spring very close to the Atlantic ocean (poor genetic predisposition); no matter what preventive measures the person takes, his/her destiny is inevitable. Take, for instance, Jim Fix, the prophet of jogging, a marathon runner, a promoter of healthy lifestyle, a non smoker with a BMI of 23, who died of myocardial infarction at 52 while he was running.

These are examples of extreme situations where evident genetic backgrounds belonging to distribution ties were in place.

GENETIC PREDISPOSITION AND RISK FACTOR PREDICTIVE VALUE

At present, despite the incredible development of genetic testing technologies, defining the genetic predisposition of a subject still is an elusive target.

Scientific and technical progress has led to the rapid increase in the number of genome wide association (GWA) studies and the identification of multiple gene-disease associations related to common diseases. This has also resulted in the commercial availability of tests designed to determine the presence or absence of gene variants that have proved to be associated with a particular disease. Proponents of personalized genomics believe that information about genetic risk of common diseases is a valuable tool for health.

Unfortunately, common multi-factorial diseases, such as depression, cancer and cardiovascular disease, typically involve many highly complex gene-gene and gene-environment interactions. Consequently, even when there is robust evidence supporting a link between a particular genetic variant and the risk of developing a given disease, it is only one factor in a very complex and as yet poorly understood pathological process. There is also the concern that such information may confuse individuals, who may not be able to fully understand and interpret risk information.

Confronted with profiles not lying in “distribution ties”, paradigmatic for healthy aging or chronic degenerative diseases, but rather in the inner part of the distribution, the physician or even the expert geneticist would not be able to establish the possible future outcome of a particular individual with a degree of certainty.

In other words, at present, with the exception of extreme situations (such as the cases of Churchill and Fix) we are not able to translate genetic predisposition testing efficiently into the precise location of individual springs – essential information for the prediction of the river path. This means that from a probabilistic point of view we are forced to assume that locations are broadly distributed around the divide in a sort of “grey area”. In the case of two rivers originating very close to the divide, genetic predisposition would be very sensitive to incidental factors: a hill (e.g. exposure to environmental toxins) could divert the river path from the Mediterranean sea, while a ridge (e.g. healthy diet, avoid smoking) could divert the fate of another river originally flowing towards the Atlantic Ocean. Therapies can be viewed as the possibility to change the geographical features of a country by intervening mechanically on the ground. Specific and early treatment could influence the river, diverting it in the opposite direction. Introducing treatment late would make it very hard, if not impossible to reverse the direction of the river.

Nowadays the location of springs in the grey area should be complemented by the geographical features of the country (risk factors). This enables the collection of meaningful information for the prediction of the future outcome of a given person. However, in the grey area the presence of risk factors is very difficult to interpret. Accurate predictions related to the individual trajectory of the river and, consequently, to the outcome of an individual, are feasible only in the presence of highly visible peaks and ridges (well defined risk factors). As a matter of fact, in clinical medicine predictive algorithms work very well only in extreme situations. Take, for instance, cardiovascular risk assessment: ten years related to exposure to a risk factor equivalent, for example, to 2% in a given person and 80% in a another person tend to be quite reliable. Unfortunately, 70-80% of the population have probability indices lying in-between very low and very high risk values.

PREDICTING THE RIVER PATH

In a geographical setting knowing only where the springs of the rivers are located in a flat country without knowing their paths makes it very difficult to draw the water divide: we would have only points without any information related to direction. We would need to know all the geographical features of the country exactly, with a very high spatial resolution and in 3D , in order to hopefully trace the divide.

Theoretically, given a certain point position of a river source in a specific environment and given a certain number of constraints related to types of physical characteristics of the territory, there is an optimal solution that minimizes the distance between the point and the sea.

However, computationally, the definition of the future path of the river is an almost impossible task, requiring the enumeration of every possible combination. Consequently, in practice, approximate, though sub-optimal, solutions are obtained through a variety of methods.

This resembles the situation in clinical medicine when we have only the baseline status of a group of subjects. Despite the abundance of features and variables available, the complexity of the environment would make it very hard to compute the divide with linear approaches, given the amount of possible interactions among the physical factors. Only observing the entire path of these rivers, following up subjects all their life, could one succeed in tracing the water divide by induction.

This is the fundamental contribution of epidemiological surveys, such as the Framingham study, which collected prospective, life-long data in a given cohort of subjects. Unfortunately, these surveys are very difficult to perform as they are time-consuming and very expensive.

The future challenge is to draw the water divide of the flat country (the grey area of health and disease) knowing only the spring location of each person and their current interaction with the environment. If we could trace this complex and non linear divide with a high degree of accuracy, computing hundreds of thousands of starting points (new cases) then we could easily predict the future outcome of a new person right from the start, just knowing his/ her location in the health and disease landscape without waiting for a long follow-up. This is the challenge for artificial intelligence in medicine: drawing the divide from snapshots of many single cases i.e. developing a film from many different photos taken randomly.

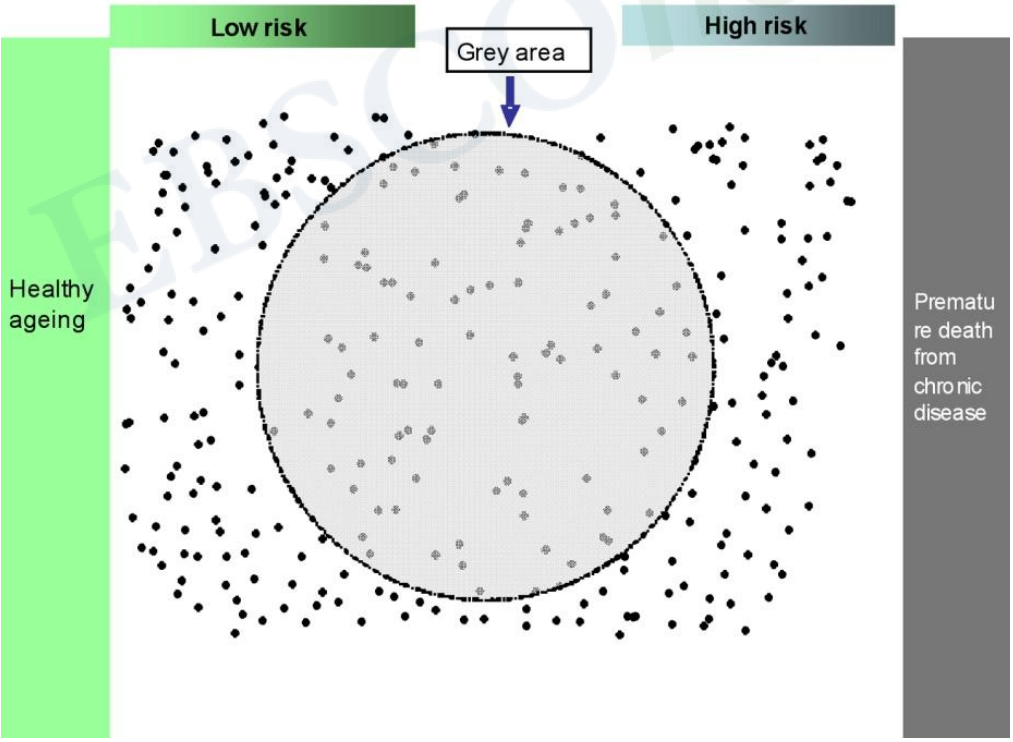


Figure 4. The concept of water drainage divide applied to health and disease. The points correspond to individual genetic backgrounds. It is very difficult to trace the Water drainage divide in the grey area unless life trajectories are followed for a long time.

In other words, predicting the path of a river originating in the middle of a flat country resembles the effort in understanding the rules underlying the complex interaction among biological, genetic and environmental factors, ultimately responsible for a certain type of evolution of the health status of a single individual. (Figure 4-6)

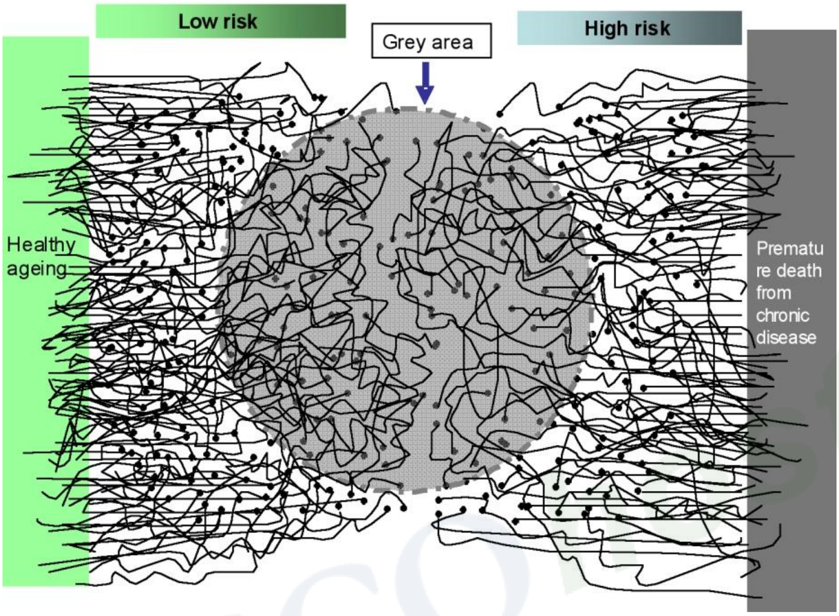


Figure 5. The water drainage divide can easily be figured out from life trajectories.

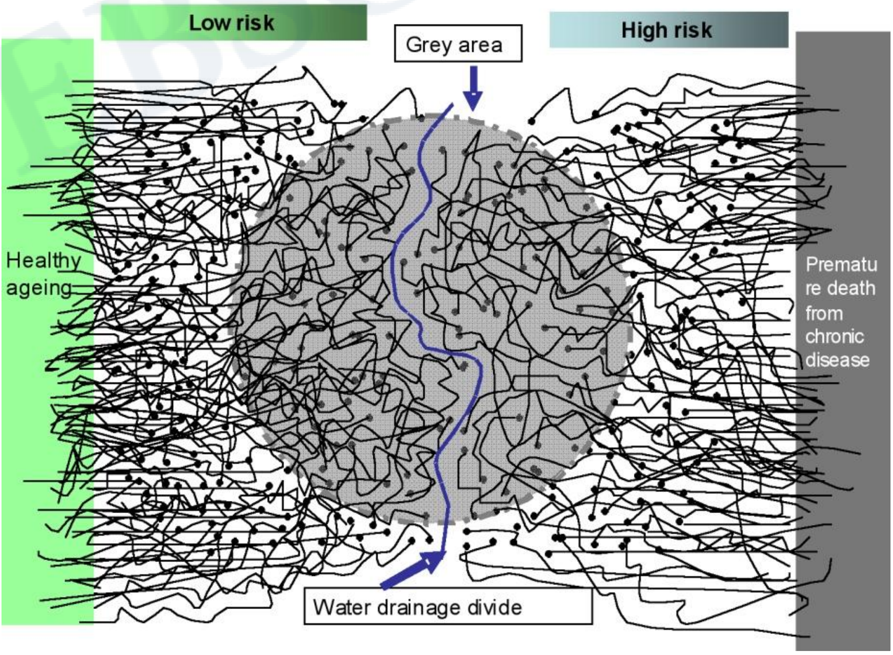


Figure 6. Tracing the water drainage divide. The real challenge is to draw the line from figure 4a.

UNDERSTANDING THE GOVERNING RULES AND UNIVERSAL LINES THROUGH ARTIFICIAL NEURAL NETWORKS

The static observation of baseline data through a classical statistical approach provides an incomplete vision of the evolution of a living organism and of the underlying biological phenomena. It is like trying to understand the rules governing American football by examining a thousand snapshots taken at different times during one thousand matches. The rules of the game would almost certainly remain obscure. Artificial neural networks, taking advantage of mathematics able to decode highly non linear dynamics, enable the reconstruction and simulation of the interaction among multiple variables, once the complex function linking the available snapshots together has been established i.e. a well trained Artificial Neural Network can re-construct the film from different snapshots belonging to different matches. We could say that they can be used also to analyze the semantics of a set of points defined in a high-dimensional space.

The efforts of new medical journals dealing with the detailed description of single cases go in this direction. The journals could create a reliable giant database of what happens to individual patients rather than to populations, gathering also a huge amount of variables. This database, provided that very sophisticated search engines powered by artificial intelligence tools are used, could really offer the chance to trace the health – disease drainage divide. This is the future challenge for predictive medicine.

Physicians every day try to accomplish their mission: to give the better care and assistance to their individual patients. They know perfectly that an individual subject is typically “unique”. He or she has specific characteristics, qualities and features, different perceptions of risk value, different values scales, different familiar environments and social roles, all together interacting in a complex way. In other words no subject is equivalent to another, nor even mono-ovular twins.

Current medical practice has created an explosion of information, which per se represent a burden for a busy medical doctor. It is not unusual to have at hand, especially when faced with treatment planning for a chronic degenerative disorder, hundreds of different variables, consisting of clinical history data, objective findings, symptomatology, multi-item scales of different meanings, laboratory examinations, imaging procedures etc. With the increased availability and use of functional genomics and digital imaging we now tentatively have at our disposition thousands of data per subject.

This new de facto reality has created yet another paradox: in comparison with 10-20 years ago we are now able to collect more data per subject than subjects per study.

More features imply more information and potentially higher accuracy. Unfortunately an important paradox is that more features we have, the more difficult information extraction is. In this high dimensional space, the hyper points corresponding to single individuals are sparse and the notion of proximity fails to retain its meaningfulness. For this reason clustering become extremely hard to be performed. In this situation we are dealing with flat, rectangular data set, a sort of telescope data set. This kind of data set are intractable from a traditional statistics point of view due to the fact that the excessive amount of degrees of freedom allows any kind of data interpolation most of the time meaningless.

Table 1. Paradigm shift introduced by AANs in medicine

No limitation in the amount of data processed
No limitation in the different nature of data processed
No limitation in the degree of complexity of data processed
Bottom - up computation: models are data driven
Interactions among different factors are easily picked-up
Inference takes place at individual level
Internal validity of modelling ensured with validation protocols
Fuzzy logic allows to escape from the probability theory trap

A part from quantitative features, non linearity, complexity, fuzzy interaction are new emerging qualitative features of chronic degenerative diseases which account for most morbidity and mortality in western world. Unfortunately even the most powerful and well established statistical methods were developed in the first half of the past century when the scenario was dominated by acute infective diseases and the available information was much more simple, or at maximum “complicated” rather than “complex”. comparison with today.

Complexity is based on small elementary units working together in small populations of synchronous processes. In a complex system each component changes, over time, losing its identity outside of the system. Complexity needs a different kind of mathematics, able to handle chaotic behavior, non linear dynamics, and fractal geometry [1-2]. There are a number of different reasons to apply complex systems mathematics on predictive medicine and some of them are listed in the table 1.

The use of computers has opened the floodgates to methods of data collection that were impossible just a decade ago, solving the quantitative problem of information load, but computers are also responsible for permitting computationally intensive medical analyses with newer numerical algorithms addressing the qualitative challenge; this is sometimes called computational and mathematical medicine.

Newer statistical approaches, base on new mathematical and logic assumptions broadly belonging to artificial adaptive system family and complex theory setting allow to tame these intractable data sets. Seen in this perspective computer science is now playing the role which mathematics did from the seventeenth through the twentieth centuries: providing an orderly, formal framework and exploratory apparatus for knowledge progress.

Actually the coupling of computer science and these new theoretical bases coming from complex systems mathematics allows the creation of “intelligent” agents able to adapt themselves dynamically to problem of high complexity: the Artificial Adaptive Systems, which include Artificial Neural Networks(ANNs.) For a detailed description of these models and tools we refer to recent reviews. [3-4]

The Paradigm Shifts of Artificial Neural Nertworks in Predictive Medicine

The basic principle which is proposed in ANNs is very simple: all the biological signals from all the sources available are analyzed together -and not individually- both in time and space. The reason for such an approach is quite simple and self-explaining: the instant value of the system in any recording source depends, in fact, upon its previous and following values (how many, and in which amount for each previous state?), upon the previous and following values of all the other recording sources (how many, and in which amount for each previous state?).

In summary, the aim of the "analyzer" is not to analyze the language of each individual variable, but to evaluate the meta-language which considers the holistic contribution of all the recorded variables. We, in fact, believe that the equilibrium of each individual subject is defined by a specific background signal model, distributed in time and in the space. Such a model is a set of background invariant features able to specify the quality of the immune activity for example. The system that we propose to apply in this research context completely ignores the subject's contingent characteristics. It utilizes a recurrent procedure which squeezes at progressive steps the significant signal and progressively eliminates the non-significant noise.

The use of Artificial Neural Networks are already emerging as new trends in medical statistics. Although these methods are not yet in widespread use, they have already had a clinical impact in specific areas, notably cervical cytology, x-Ray mammography and early detection of acute myocardial infarction where large-scale prospective multicenter studies have been carried out. An extensive review on this subject has been published by Lisboa [5].

There are in the literature many examples of successful application of ANNs in outcome research.

Our group has proved the usefulness of the added predictive value gained with the use of advanced artificial neural networks in a number of medical fields, ranging from heart diseases, gastroenterology and neurology with special regard to Alzheimer disease, stroke and Amyotrophic Lateral Sclerosis. [6-13]

ANNs bring a number of revolutionary paradigm shifts which will have a strong impact in predictive medicine. They are listed in the table 1.

ANNs are able to reproduce the dynamical interaction of multiple factors simultaneously, allowing the study of complexity; this is very important for the researcher interested to deep the knowledge of a specific disease or to better understand the possible implications relative to strange associations among variables. This has to do to what is called "intelligent data mining". But one the other hand ANNs can also help medical doctors in making decisions under extreme uncertainty and to draw conclusions on individual basis and not as average trends. The modern patient wants to be treated as an individual person and not just as a statistics. Patients want to know their own risk, not just a parameter regarding a class of people similar to them just for some aspects. ANNs are very powerful in modelling at single individual level, and by combining several parallel AANs trained on the same data set is possible to make multiple statistics on a single subject, allowing in this way the calculation of the confidence interval of the prediction estimate. Finally ANNs make possible to treat huge amount of information without squeezing arbitrarily the data and without loosing complexity.

This contributes to a new holistic vision of the human subject contrasting the statistical reductionism, which tends to squeeze or even delete the single subject sacrificing him to his group of belongingness. A remarkable contribution to this individual approach comes from Fuzzy Logic, according to which there are no sharp limits between opposite things, like health and disease. This approach allows to partially escape from probability theory trap in situations where is fundamental to express a judgment based on a single case and favors a novel humanism directed to the management of the patient as individual subject.

COMPETING INTERESTS

The author declares no competing interests

REFERENCES

[1] Kaplan, D. & Glass, L. (1995). *Understanding nonlinear dynamics*. New York, NY, Springer-Verlag.

[2] Goldberger, A. L. (1996). Nonlinear dynamics for clinicians: chaos theory, fractals, and complexity at the bedside. *Lancet*, *347(9011)*, 1312-1314.

[3] Grossi, E. & Buscema, M. (2007). Introduction to artificial neural networks. Eur J *Gastroenterology & Hepatology*, *19(12)*, 1046-1054.

[4] Grossi, E. & Buscema, M. (2006). Artificial Intelligence and Outcome Research. *Drug Development Research*, *67(3)*, 227-244

[5] Lisboa, P. J. G. (2002). A review of evidence of health benefit from artificial neural networks in medical intervention. *Neural Network*, *15(1)*, 11-39.

[6] Street, M. E., Grossi, E., Volta, C., Faleschini, E. & Bernasconi, S. (2008). Placental determinants of fetal growth: identification of key factors in the insulin-like growth factor and cytokine systems using artificial neural networks. *BMC Pediatrics*, *8(1)*, 24.

[7] Penco, S., Buscema, M., Patrosso, M. C., Marocchi, A. & Grossi, E. (2008). New application of intelligent agents in sporadic amyotrophic lateral sclerosis identifies unexpected specific genetic background. *BMC Bioinformatics*, *9(1)*, 254.

[8] Rossini, P. M., Buscema, M., Capriotti, M., Grossi, E., Rodriguez, G., Del Percio, C. & Babiloni, C. (2008). Is it possible to automatically distinguish resting EEG data of normal elderly vs. mild cognitive impairment subjects with high degree of accuracy? *Clinical Neurophysiol. J.*, *119(7)*, 1534-45.

[9] Licastro, F., Porcellini, E., Chiappelli, M., Forti, P., Buscema, M., Ravaglia, G. & Grossi, E. (2008). Multivariable network associated with cognitive decline and dementia. Neurobiology of Aging, *Elsiever.*, *31(2)*, 257-269.

[10] Helgason, C. M., Grossi, E., Pandey, D., Valika, A., Cursio, J. & Brace, L. D. (2008). Platelet aggregation and recruitment with aspirin-clopidogrel therapy. *Cerebrovasc Dis.*, *25(5)*, 392-400.

[11] Lahner, E., Intraligi, M., Buscema, M., Centanni, M., Vannella, L., Grossi, E., Annibale, B. (2008). Artificial neural networks in the recognition of the presence of

thyroid disease in patients with atrophic body gastritis. *World J Gastroenterology*, *14(4)*, 563-8.

[12] Grossi, E., Mancini, A. & Buscema, M. (2007). International experience on the use of artificial neural networks in gastroenterology. *Dig Liver Dis*., *39(3)*, 278-85.

[13] Grossi, E. (2006). How artificial intelligence tools can be used to assess individual patient risk in cardiovascular disease: problems with the current methods. *BMC Cardiovasc Disord*., *3*, 6:20.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 13

ARTIFICIAL NEURAL NETWORK (ANN) PREDICTION OF A FULL-SCALE MUNICIPAL WASTEWATER TREATMENT PLANT PERFORMANCE

Erkan Sahinkaya[*] and Sibel Pulcu Yıldız
Harran University, Environmental Engineering Department,
Osmanbey Campus, Sanliurfa, Turkey

ABSTRACT

Performance of a treatment plant highly depends on plant's operation and it is generally difficult to predict the performance due to several uncertainties. Also, many of activated sludge processes produce poor effluent quality due to escape of sludge from secondary clarifier as a result of excess filamentous growth. In this context, this study aims at modeling a real scale activated sludge process using a popular artificial neural network (ANN) to make its management easier. Effluent COD, effluent BOD_5 and SVI were used as model output parameters. The developed ANN model was very successful as an excellent to reasonable match was obtained between the measured and the predicted parameters of effluent COD (R = 0.90), effluent BOD_5 (R = 0.83) and SVI (R=0.84). Hence, the ANN based model can be used to predict a full scale wastewater treatment plant performance and to control the operational conditions for improved process performance.

Keywords: Artificial neural network, activated sludge, sludge settling, modeling.

[*] Corresponding author: E-mail: erkansahinkaya@yahoo.com, Tel: +90 414 344 00 20/1474, Fax: +90 414 344 00 31

INTRODUCTION

Activated sludge is the most commonly used secondary treatment process all over the world. Activated sludge process efficiency depends on many operational and design parameters. Lack of expert staff and the locating far away from the settlements makes the operation and management of domestic wastewater treatment plant more difficult. However, stringent discharge standards require better operation and management of the treatment plants [1]. In order to design and predict the performance of activated sludge treatment processes a number of models, such as ASM 1, ASM2 and ASM3, were developed [2,3]. However, it is generally difficult to apply these models for a real treatment plants due to the complexity of physical, biological and chemical processes involved in wastewater treatment process. Additionally, the incoming wastewater flow rate and composition may show great variation. These complexities create non-linear behaviors difficult to model [4]. Also, calibration of determinitistic models has to be performed for each specific treatment system, which cause application of these models to real systems cumbersome and problematic [1]. When circumstances or processes are not understood well enough or parameter determination is unpractical, there is a distinctive advantage for black-box modeling [5].

Most of the existing activated sludge processes have sludge settling problem due to excess growth of filamentous microorganisms, called sludge bulking. The inability of removing biomass from treated effluent due to sludge bulking results in poor effluent quality. Although sludge bulking has been extensively studied, the uncertainties about the factors triggering the growth of filamentous organism makes the sludge bulking control difficult [6]. Although several models have been developed to predict activated sludge process performance, generally these models do not consider sludge settling process. Hence, most of the models may not predict the activated sludge process performance well enough due to sludge bulking problem in existing activated sludge treatment plants. Hence, more powerful tools should be used to predict sludge settleability and to take some precautions before sludge bulking problem starts to create poor effluent quality.

Black-box models, like the artificial neural networks (ANNs), are very attractive as they do not require prior knowledge about the structure and relationships that exist between important variables. Moreover, their learning abilities make them adaptive to system changes. Up to now, there are a number of ANN applications in environmental engineering [5]. They have already been used to predict nitrate and SAR values in groundwater [7], to determine the leachate amount from municipal solid waste landfill [8], to model the performance of sulfidogenic fluidized bed reactor [9], to predict the performance of wetlands used for municipal wastewater treatment [10].

In this context, this study aims at ANN modeling of performance and sludge settleability of a real wastewater treatment plant. With the use ANN modeling, the operation and management of wastewater treatment plants would be much easier.

MATERIALS AND METHODS

Description of Wastewater Treatment Plant

Denizli wastewater treatment plant (Denizli WWTP) is composed of coarse and fine screens, grid removal, primary sedimentation tanks, activated sludge aeration tanks, and secondary sedimentation tanks. The excess sludge produced in the WWTP is anaerobically digested in two stages. The current capacity of the treatment plant is 1692 L/s and it will be increased to 2806 L/s at the end of 2025. The treatment plant performance was evaluated for around one year between 2007 and 2008. Regular 3-h, 8-h and 24-h composite samples were collected for the measurements of influent and effluent chemical oxygen demand (COD), biochemical oxygen demand (BOD_5), suspended solids (SS), total nitrogen and total phosphorous. Sludge volume index (SVI) and suspended solids were also analyzed on composite samples drawn from aeration tanks and recycle line. Temperature, pH, conductivity and dissolved oxygen concentrations at various locations of the treatment plant was on-line measured. All the analyses were conducted according to Standard Methods [11]. All measurements were done at least in duplicate. Mean values and standard deviations were presented.

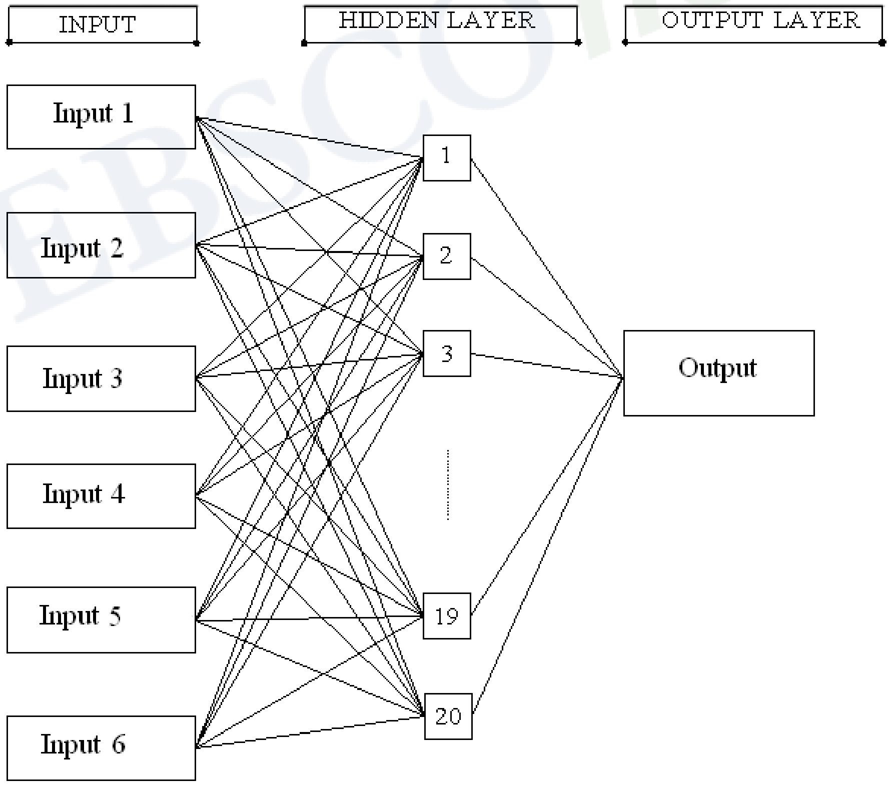


Figure 1. Artificial neural network structure for the prediction of treatment plant performance.

Modelling

A neural network is defined as a system of simple processing elements, called neurons, which are connected to a network by a set of weights (Figure 1). The network is determined by the architecture of the network, the magnitude of the weights and the processing element's mode of operation. The neuron is a processing element that takes a number of inputs, weights them, sums them up, adds a bias and uses the result as the argument for a singular valued function, the transfer function, which results in the neuron`s output [5].

Back-propagation (BP) algorithms use input vectors and corresponding target vectors to train ANN. The standard BP algorithm is a gradient descent algorithm, in which the network weights are changed along the negative of the gradient of the performance function [12,13]. There are a number of variations in the basic BP algorithm that is based on other optimization techniques such as conjugate gradient and Newton methods. For properly trained BP networks, a new input leads to an output similar to the correct output. This ANN property enables training of a network on a representative set of input/target pairs and achieves sound forecasting results.

Although some researchers suggest that one hidden layer is usually sufficient [14], the introduction of additional hidden layers allows the fit of a larger variety of target functions and enables approximations of complex functions with fewer connection weights [15]. In this work, a two-layer neural network with a tan-sigmoid transfer function for the hidden layer and a linear transfer function for the output layer were used. The data were divided into training, validation and test subsets. Half of the data were used for training and one forth of the data was used for validation and tests.

Selection of Back-Propagation Algorithm and Determining Neuron Number

In the study, the treatment plant effluent COD, effluent BOD_5 and sludge settleability (using SVI) were modeled. For all modeled parameters, 12 BP algorithms were compared to select the best fitting one. In the selection of BP algorithm, the number of neurons was kept constant at 20. After selecting best BP algorithm, the prediction results were compared for different neuron numbers to optimize neuron number keeping all other parameters constant.

Results and Discussion

Before ANN prediction of wastewater treatment plant performance, linear correlations between target parameters and the various plant parameters were investigated. The correlation coefficients between target parameters (effluent COD, effluent BOD_5 and SVI) and various operational parameters were less than 0.40 (data not shown). Hence, the real wastewater treatment plant is so complex process that it is irrelevant to use conventional regression techniques in the prediction of real wastewater treatment plant performance. A more powerful technique, like ANN, is necessary to predict the performance of the treatment plant performance.

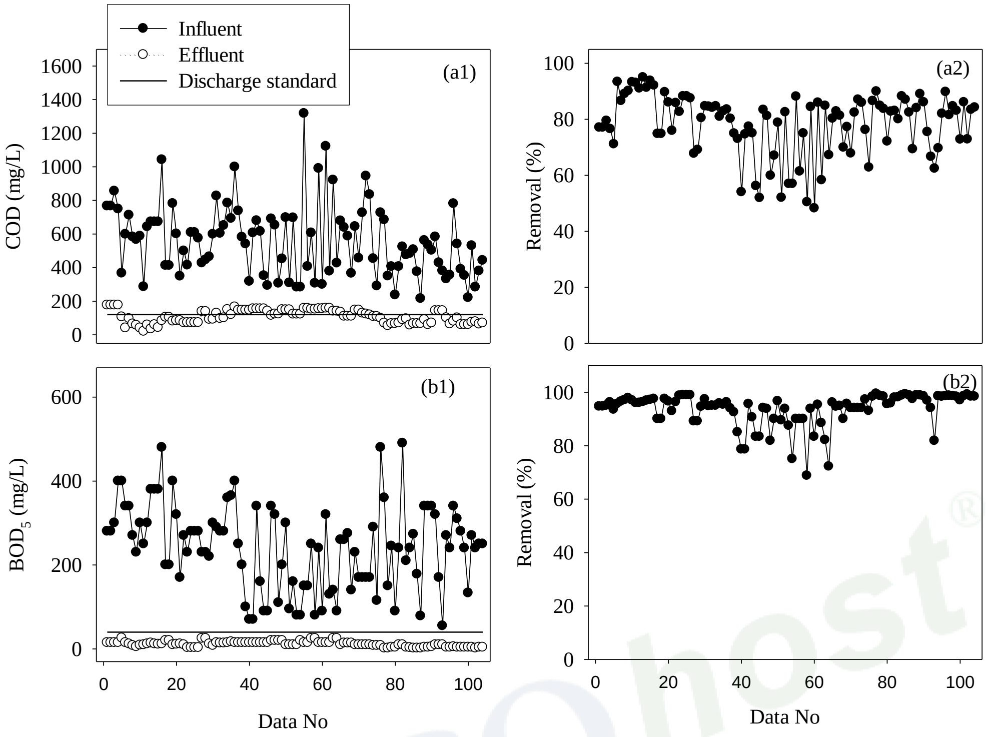


Figure 2. Influent and effluent COD (a1), BOD$_5$ (b1) and removal efficiencies (a2, b2).

ANN Prediction of Effluent COD

The variation of influent and effluent COD in the treatment plant was depicted in Figure 2. The influent COD concentrations varied between 215 and 1317 mg/L and it averaged 555±200 mg/L. The effluent COD concentration was between 176 and 20 mg/L and it averaged 107±40 mg/L. Sometimes, observing high effluent COD concentrations was due to increasing influent COD concentration as a result of industrial discharge. There are lots of textile industries in the vicinity of Denizli and untreated or not properly treated industrial discharges to the sewer system may result in increasing influent COD concentration. Additionally, industrial discharge to the sewer system decreased the biodegradability of incoming wastewater, which was reflected as increasing COD/BOD$_5$ ratio (data not shown). Most of the time, the treatment plant effluent COD concentration was below the discharge standard value (120 mg/L) given in national standard of Turkish Water Pollution Control Regulation. The national standard value and the treatment plant performance were also given in Figure 2 for the sake of comparison.

In the modeling study, the first step was the selection of input parameters. For this purpose, various trials were evaluated (Table 1) and the trial giving the best result was selected to optimize the model structure (determining the best propagation algorithm and the number of neuron). Trial-2 gave the best result with the R value of around 0.88. In the trails, the back propagation algorithm and the neuron number were kept constant at trainlm and 20, respectively. After determining ANN input parameters for COD prediction, the model was run with 12 different algorithm (Table 2) to determine the best fitting one. The performance of the BP algorithms was evaluated with the root mean square error (MSE) and determination coefficient (R) between the modeled output and measured data set. The best BP algorithm with minimum training error and maximum R was the Levenberg-Marquardt (trainlm) algorithm. Finally, the neuron number was optimized using trainlm for the Trail-2 (Table 3). The neuron number of 25 gave the best prediction with R value of 0.90 (Figure 3a). The measured and the predicted COD values were also provided in Figure 3b for comparison. Results showed that ANN model gave good fit to the measured COD values.

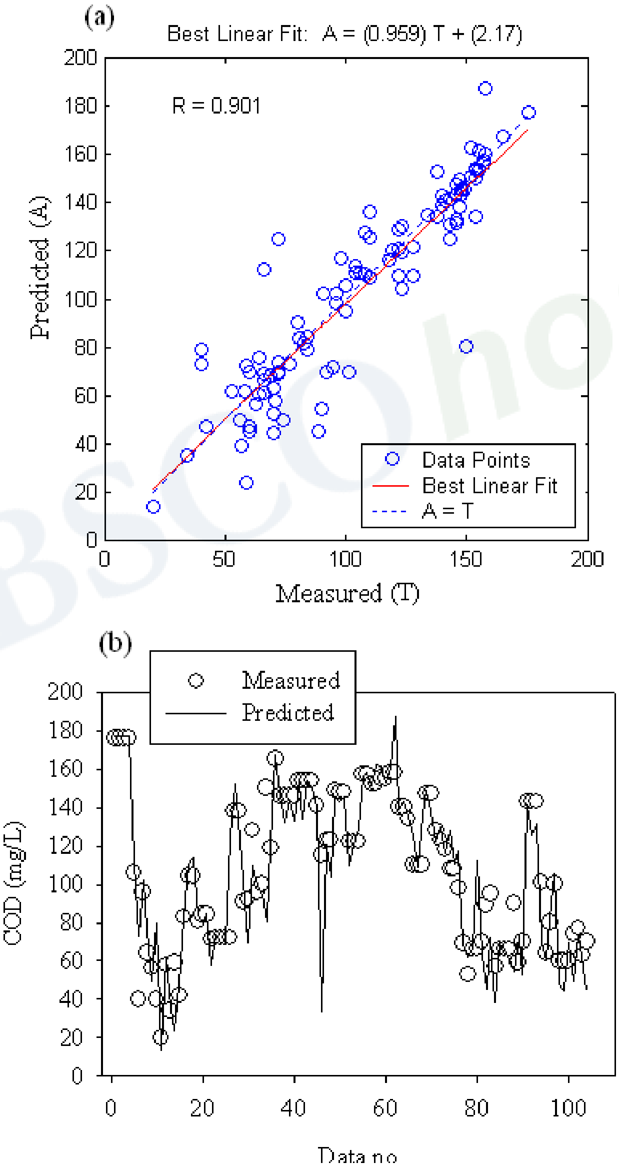


Figure 3. Linear regression between network outputs and the corresponding targets (a), and the comparison of measured and predicted COD values (b).

Table 1. Comparison of different trials to determine the ANN input parameters for effluent COD prediction. "X" shows the parameter selected as ANN input parameter for the corresponding trial

Trial-no	1	2	3	4	5
R	0,813	0,878	0,78	0,779	0,643
Influent wastewater					
pH	X	X	X		
Conductivity	X	X	X		
Temperature	X	X	X		
Suspended solid					
COD			X		
BOD_5					
Total Nitrogen					
Total Phosphate					
Treatment plant effluent					
pH	X	X	X	X	X
Conductivity	X	X	X		
Temperature	X	X	X	X	X
Suspended solid		X	X	X	X
COD					
BOD_5					
Aeration tank					
Suspended solid		X	X	X	X
Suspended solid in recycle line			X		X
SVI					X

Table 2. Comparison of back-propagation algorithms for predicting effluent COD, effluent BOD_5 and SVI

Training algorithm	COD		BOD_5		SVI	
	R	MSE	R	MSE	R	MSE
Trainlm	0.878	1.66e-28	0.832	8.04e-25	0.84	4.42e-31
Traincgp	0.826	0.172	0.73	0.279	0.795	0.065
Traingd	0.742	0.422	0.673	0.410	0.601	0.341
Traingda	0.83	0.243	0.783	0.250	0.76	0.123
Traingdx	0.831	0.190	0.794	0.261	0.803	0.052
Trainrp	0.768	0.132	0.812	0.137	0.778	0.011
Trainscg	0.813	0.175	0.691	0.238	0.745	0.024
Trainoss	0.81	0.043	0.806	0.212	0.802	0.071
Traincgf	0.809	0.220	0.738	0.223	0.719	0.043
Trainbfg	0.82	0.056	0.783	0.185	0.787	0.005
Traingdm	0.791	0.380	0.633	0.455	0.618	0.291
Traincgb	0.853	0.117132	0.649	0.267273	0.801	0.040577

Table 3. R values and mean square errors at different neuron numbers for predicting effluent COD, effluent BOD_5 and SVI

Neuron Number	COD		BOD_5		SVI	
	R	MSE	R	MSE	R	MSE
3	0.747	0.323	0.774	0.222	0.59	0.402
5	0.837	0.033	0.777	0.188	0.514	0.359
10	0.745	0.003	0.794	0.048	0.608	0.135
15	0.824	0.005	0.641	0.008	0.797	5.4e-30
20	0.878	$1.662e^{-28}$	0.832	8.04e-25	0.84	4.4e-31
25	0.901	4.9e-31	0.753	1.13e-30	0.77	6.24e-31
30	0.839	1.43e-24	0.715	1.31e-30	0.791	4.3e-30
40	0.768	4.25e-05	0.692	3.01e-25	0.804	3.0e-27

Table 4. Comparison of different trials to determine the ANN input parameters for effluent BOD_5 prediction. "X" shows the parameter selected as ANN input parameter for the corresponding trial

Trial-no	1	2	3	4
R	0,765	0,75	0,818	0,832
Influent wastewater				
pH	X			
Conductivity	X	X	X	
Temperature	X	X	X	
Suspended solid		X	X	
COD				
BOD_5			X	
Total Nitrogen				
Total Phosphate				
Treatment plant effluent				
pH	X			
Conductivity	X	X	X	X
Temperature	X	X	X	
Suspended solid		X	X	X
COD		X	X	X
BOD_5				
Aeration tank				
Suspended solid				
Suspended solid in recycle line		X	X	X
SVI			X	X

ANN Prediction of Effluent BOD_5

The influent BOD_5 concentration varied between 55 and 500 mg/L and it averaged 242±100 mg/L (Figure 2). Similar to COD, influent BOD_5 concentration showed great variation due to industrial discharges to the sewer system. The effluent BOD_5 concentrations were between 2 and 26 mg/L and it averaged 11±6 mg/L. BOD_5 removal efficiency in the treatment plant varied between 66 and 99% and the average value was 94±6% (Figure 2). According to Turkish Water Pollution Control Regulation, the treatment plant effluent BOD_5 concentration should not exceed 40 mg/L. Hence, the effluent BOD_5 value was always below the national discharge standards (Figure 2).

As mentioned previously, the conventional way of effluent BOD_5 prediction is a complicated job due to process complexity, time dependent variation of wastewater flow rate and composition. Hence, similar to COD prediction, ANN was used to predict the effluent BOD_5 concentration. The similar way used to predict effluent COD concentration was followed. First of all several trials were done to determine the ANN input parameters (Table 4). Trail-4 was determined to be the best fitting one with R value of around 0.83 (Table 4). In the Trail-4, effluent conductivity, effluent suspended solids, effluent COD, suspended solids at recycle line and sludge were used as input parameters. After determining the ANN input parameters, the train algorithm and the neuron number was optimized, similar to the COD prediction. The results were summarized in Tables 2 and 3. Results showed that the optimum training algorithm and the neuron number were trainlm and 20, respectively. The ANN predictions were shown in Figure 4a. Additionally, the variation of measured and predicted values were presented in Figure 4b for the sake of comparison. Results showed that the predictions followed the measured values closely and the ANN can be effectively used for modeling municipal wastewater treatment plant performance and better management.

ANN Prediction of Activated Sludge Settleability

Most of the problems of poor activated sludge effluent quality result from the inability of the removing the suspended biomass from the treated water. When the biomass is strongly colonized by long filamentous bacteria, called bulking, the amount of total suspended solids at the outflow of the plant increases seriously [6]. There are several design and operation related reasons for the bulking phenomenon and it is sometimes difficult to obtain well settling sludge. Hence, predicting bulking phenomena using ANN modeling may allow the better operation of activated sludge process and obtaining good performance. The sludge settleability is measured by SVI and it is generally accepted that the good settling sludge should have the SVI value between 50 and 150 mL/g [16].

The SVI values of activated sludge from The Denizli WWTP varied between 81 and 686 mL/g and averaged 197±116 mL/g. Assuming that a well settling sludge has the SVI value below 150 mL/g, it could be concluded that Denizli WWTP has a moderate sludge bulking problem. The sludge bulking may cause escape of suspended solids from the secondary effluent and deteriorate the effluent quality. As it can be seen from Figures 5b and 5c, when SVI value exceed 600 mL/g, the effluent SS value increased from its average value of 20 to over than 80 mg/L. Hence, predicting sludge settling problem before being too late may allow the operation engineer to take some precautions.

Similar to COD and BOD_5 predictions, ANN prediction of SVI started with the determination of model input parameters. The Trail-3 gave the best prediction with R value of 0.84 (Table 5). After determining the model input parameter, the model structure was optimized. Firstly, the neuron number was kept constant at 20 and back-propagation algorithms were changed. Out of 12 algorithms, trainlm gave the best result (Table 2). Secondly, the neuron number was changed using trainlm as back-propagation algorithm (Train 3). The results showed that the optimum neuron number was 20 for the present study. The prediction results of the developed ANN is given in Figure 5a and the comparison of measured and predicted values was also depicted in Figure 5b to make the comparison easier. Results showed that ANN can be used to predict the sludge settleability of a full scale activated sludge treatment plant. Therefore, ANN modeling may make management and operation of a wastewater treatment plant easier.

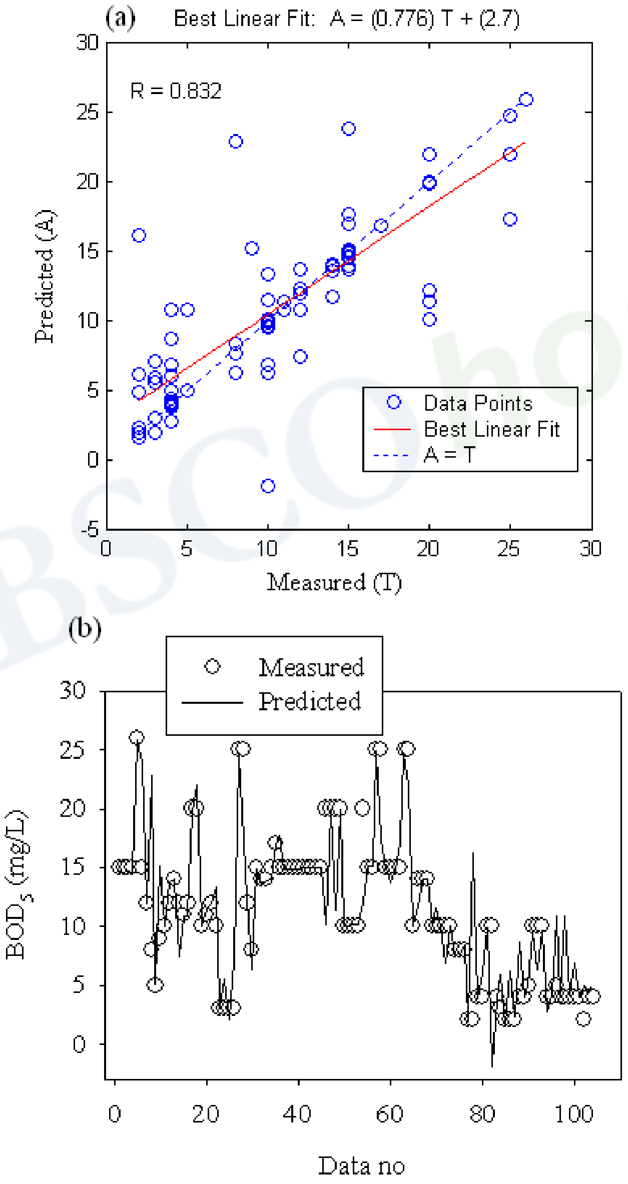


Figure 4. Linear regression between network outputs and the corresponding targets (a), and the comparison of measured and predicted BOD$_5$ values (b).

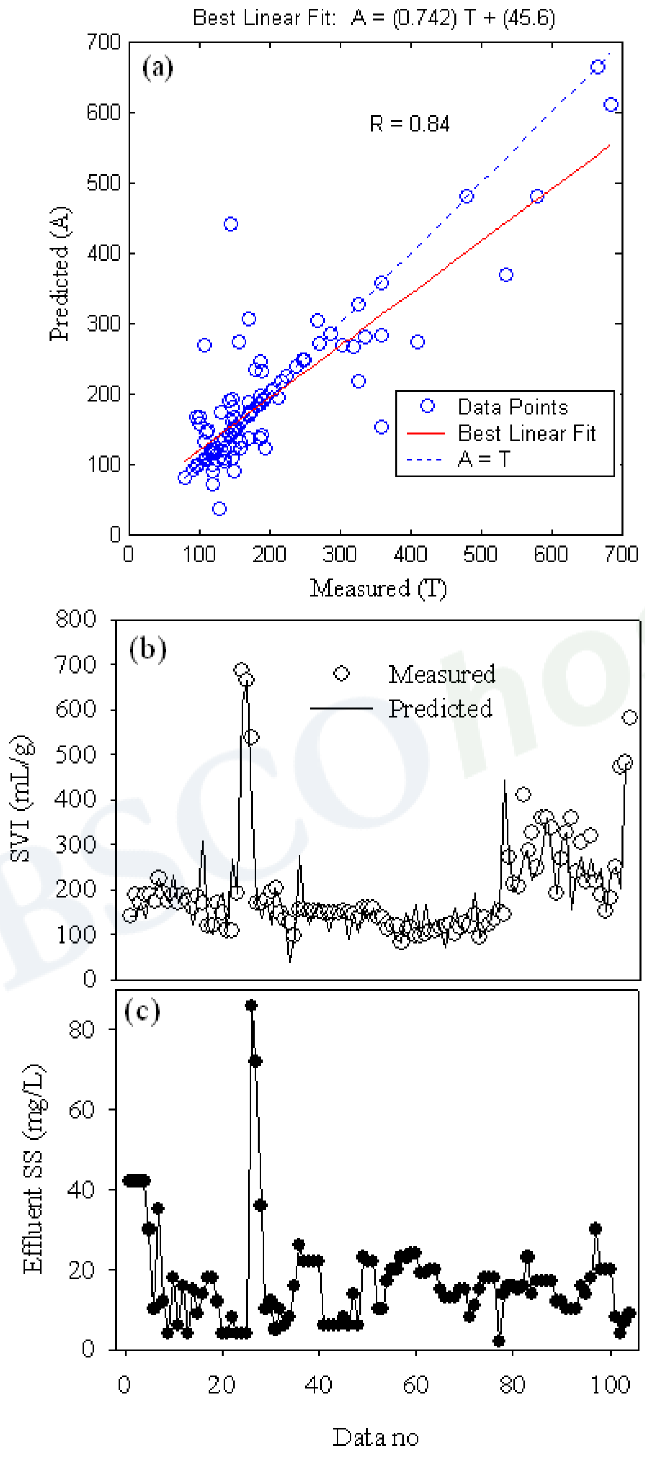


Figure 5. Linear regression between network outputs and the corresponding targets for SVI (a), the comparison of measured and predicted SVI values (b), and effluent SS concentrations (c).

Table 5. Comparison of different trials to determine the ANN input parameters for SVI prediction. "X" shows the parameter selected as ANN input parameter for the corresponding trial

Trial-no	1	2	3	4	5
R	0,762	0,799	0,84	0,769	0,714
Influent wastewater					
pH	X	X	X		
Conductivity	X	X	X		
Temperature	X	X	X		
Suspended solid					
COD			X		
BOD_5					
Total Nitrogen					
Total Phosphate					
Treatment plant effluent					
pH	X	X	X	X	X
Conductivity	X	X	X		
Temperature	X	X	X	X	X
Suspended solid		X	X	X	X
COD					
BOD_5					
Aeration tank					
Suspended solid		X	X	X	X
Suspended solid in recycle line			X		X
SVI					

CONCLUSIONS

A real wastewater treatment plant is so complex process that it is irrelevant to use conventional regression techniques in the prediction of real wastewater treatment plant performance. A more powerful technique, like ANN, is necessary to predict the performance of the treatment plant performance. In this context, this study aims at modeling a real scale activated sludge process using popular artificial neural network (ANN) to make management of the treatment plants easier. The developed ANN model was very successful as an excellent to reasonable match was obtained between the measured and the predicted parameters of effluent COD (R = 0.90), effluent BOD_5 (R = 0.83) and SVI (R=0.84). Hence, the ANN based model can be used to predict a full scale wastewater treatment plant performance, to control the operational conditions for improved process performance and to take proper precautions for better management.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to the Harran University Research Fund which supported this work.

REFERENCES

[1] Moral, H., Aksoy, A. & Gokcay, C. F. (2008). Modeling of the activated sludge process by using artificial neural networks with automated architecture screening. *Comput. Chem. Eng.*, *32*, 2471-2478.

[2] Henze, M., Gujer, W., Mino, T., Matsuo, T., Wentzel, M. C. & Marais, G. V. R. (1995). Waste-water and biomass characterization for the Activated Sludge Model No. 2: Biological phosphorus removal. *Water Sci. Technol.*, *31*, 13-23.

[3] Gujer, W., Henze, M., Mino, T. & Loosdrecht, M. V. (1999). Activated Sludge Model No. 3. *Water Sci. Technol.*, *39*, 183–193.

[4] Hamed, M. M., Khalafallah, M. G. & Hassanien, E. A. (2004). Prediction of wastewater treatment plant performance using artificial neural networks. *Environ Modell Softw, 19*, 919-928.

[5] Strik, D. P. B. T. B., Domnanovich, A. M., Zani, L., Braun, R. & Holubar, P. (2005). Prediction of trace compounds in biogas from anaerobic digestion using the MATLAB neural network toolbox. *Environ Modell. Softw.*, *20*, 803-810.

[6] Balanche, L., Valdes, J. J., Comas, J., Roda, I. R. & Poch, M. (2000). Prediction of the bulking phenomenon in wastewater treatment plants. *Artif. Intell. Eng.*, *14*, 307–317.

[7] Yesilnacar, M. I., Sahinkaya, E., Naz, M. & Ozkaya, B. (2008). Neural network prediction of nitrate in groundwater of Harran Plain, Turkey. *Environ Geol*, *56*, 19-25.

[8] Karaca, F. & Ozkaya, B. (2006). NN-LEAP: A neural network-based model for controlling leachate flow-rate in a municipal solid waste landfill site. *Environ Modell Softw, 21*, 1190-1197.

[9] Sahinkaya, E., Ozkaya, B., Kaksonen, A. H. & Puhakka, J. A. (2007). Neural network prediction of thermophilic (65°C) sulfidogenic fludized-bed reactor performance for the treatment of metal-containing wastewater. *Biotechnol Bioeng, 97*, 780-787.

[10] Naz, M., Uyanik, S., Yesilnacar, M. I. & Sahinkaya, E. (2009). Side-by-side comparison of horizontal subsurface flow and free water surface flow constructed wetlands and artificial neural network (ANN) modelling approach. *Ecol Eng*, *35*, 1255-1263.

[11] APHA, (1999). *Standard methods for the examination of water and wastewater*, 20th ed. American Public Health Association/American Water Works Association/Water Environment Federation, Washington, DC, USA.

[12] Abdi, H., Valentin, D., Edelman, B. & O'Toole, A. J. (1996). A Widrow-Hoff learning rule for a generalization of the linear auto-associator. *J. Math. Psychol.*, *40*, 175-182.

[13] Nguyen, D. & Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In: Proceedings of the *International Joint Conference on Neural Networks*, *3*, 21-26.

[14] El-Din, A. G. & Smith, D.W. (2002). A neural network model to predict the wastewater inflow incorporating rainfall events. *Water Res.*, *36*, 1115-1126.

[15] Toth, E., Brath, A. & Montanari, A. (2000). Comparison of short-term rainfall prediction models for real-time flood forecasting. *J. Hydrol.*, *239*, 132-147.

[16] Bitton, G. (2005). *Wastewater Microbiology, Third Edition*, Wiley.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 14

POSITIONING CONTROL OF AN UNDER-ACTUATED ROBOT MANIPULATOR USING ARTIFICIAL NEURAL NETWORK INVERSION TECHNIQUE

***Ahmad Azlan Mat Isa*[1], *H. M. A.A. Al-Assadi*[1] *and Ali T. Hasan*[2]**

[1]Faculty of Mechanical Engineering, Universiti Teknologi MARA
(UiTM), Shah Alam, Selangor, Malaysia
[2]Ministery of Higher Education and Scientific Research, Baghdad, Iraq

ABSTRACT

This chapter is devoted to solve the positioning control problem of under-actuated robot manipulator. Artificial Neural Networks Inversion technique was used where a network representing the forward dynamics of the system was trained to learn the position of the passive joint over the working space of a 2R under-actuated robot. The obtained weights from the learning process were fixed and the network was inverted to represent the inverse dynamics of the system, and then used in the estimation phase to estimate the position of the passive joint for a new set of data the network was not previously trained for in order to show the success of the control strategy.

Data used in this research are recorded experimentally from sensors fixed on the robot joints in order to overcome whichever uncertainties presence in the real world such as ill-defined linkage parameters, links flexibility and backlashes in gear trains.

The technique was implemented in two phases, the first phase was the forward learning phase that used to obtain the training weights which are used in the second phase which is the inverse estimation phase that is used to estimate the passive joint's position for any set of data the network was not trained for. The results were verified experimentally to show the ability of the proposed technique to solve the problem efficiently.

1. INTRODUCTION

Under-actuated robot manipulator posses fewer actuators than degrees of freedom (DOF). Complex internal dynamics, nonholonomic behavior and lack of feedback linearizability are often exhibited in such systems, making that class of robots a challenging one for synthesis of control schemes. Due to their advantages over fully actuated robots, this type of manipulators has gained the interest of several researchers [1-16]. Saving in weight and cost is an advantage, where low cost automation and space robots require this feature. Another advantage is that under-actuated robots can easily overcome actuator failure due to unexpected accident. Such fault-tolerant control is highly desirable for robots in remote or hazardous environments [1,2].

The difficulty of the control problem for under-actuated mechanisms is obviously due to the reduced dimension of the input space. In particular it has been shown that this system is highly nonlinear and it is impossible to stabilize asymptotically with a smooth feedback [3]. Sørdalen et al, [4] have designed an *n* joint robot controlled by just two motors using nonholonomic gears. Other researchers have tried controlling an under-actuated robot in a gravity field, such as the Acrobot [5-7]. The control of a high-bar robot was investigated by Takashima [8] while Saito et al, [9] have investigated the control of a brachiation robot. Neglecting joint friction which is not easy to achieve in real world as it involves high manufacturing cost, Luca et al. [10, 11] have studied the control of two-link manipulator moving in a horizontal plane with a single actuator at the first joint. In practice, joint friction cannot be neglected especially at the passive joint as friction at the active joint can be directly compensated; the same is not true for the passive joint. Some researchers have tried to overcome this problem by implementing additional equipments such as breaks at the passive joint [21-15]. The dynamic characteristics of a two-link manipulator including joint friction by proposing a mathematical model have been studied by Yu et al. [1] as a result of this research; they have found that the manipulator can be controlled if the friction acts on the passive joint. Later on, Mahindrakar et al. [16] have presented a mathematical model for a two-link under-actuated manipulator wherein the motion of the system was confined to a horizontal plane; their proposed dynamic model takes into account the frictional forces acting on the joints. In this case, any additional equipment such as brakes is not needed in positioning all the joints to desired position

While some interesting techniques and results have been presented in the previously mentioned publications, the control of such systems still remains an open problem. Most of the control schemes mentioned above either failed to provide a thorough analysis of the overall system stability or assumed that friction forces do not act on the passive joints. Furthermore, the precise knowledge of the dynamic model is generally required. In real world application, no physical property such as the friction coefficient can be exactly derived. Besides, there are always kinematics uncertainties presence in the real world such as ill-defined linkage parameters, links flexibility and backlashes in gear train [17, 18].

In the last few years much of the research interest shown in the control of non-linear systems has focused on methods where classical methodologies valid for linear systems, do not give satisfactory performance [19,20]. Among of which, Artificial Neural Networks (ANN) technique has gained a great deal of interest for their extreme flexibility due to its learning ability and the capability of non-linear function approximation. They can learn from examples, are fault tolerant in the sense that they are able to handle noisy and incomplete data, are able to deal with nonlinear problems and, once trained, can perform prediction and generalization at high speed. They are particularly useful in system modeling such as in implementing complex mappings and system identification [21,22]. During the past decade, many researchers have justified using ANNs for various applications of robotics system control [23-25].

Recently, several neural network approaches have been studied, and the network inversion approach was one of the justified approaches for inverse problems solution [26-28]. In this chapter, the network inversion technique was implemented; the input was estimated from the given output using the learned network inversely. Data used in this research was collected experimentally to overcome any uncertainty presents in the real world. The efficiency of the proposed method is shown experimentally using 2R under-actuated robot manipulator.

2. Equations of Motion with Friction Effect

As Figure 1 show, the space coordinate of the manipulator is parameterized by q. The coordinate q_i, $i = 1,2$ are the joint angles. The Euler–Lagrange equation of motion is [16]:

$$M(q)\ddot{q} + h(q,\dot{q}) = \tau, \tag{1}$$

Where $\dot{q}$ and $\ddot{q}$ are the generalized velocities and accelerations respectively. $M(q)$ is the inertia matrix, which is symmetric and positive definite.

The centripetal and Coriolis terms are collected in the vector $h(q,\dot{q})$. The vector h contains terms purely quadratic in the velocities; gravity terms are absent since it assumed that the manipulator moves in a horizontal plane.

Define the following constants:

$$c_1 = m_1 r_1^2 + m_2 l_1^2 + I_1,\ c_2 = m_2 r_2^2 + I_2,\ c_3 = m_2 l_1 r_2.$$

The equations of motion accounting for the Coulomb plus viscous friction at the joints become:

$$m_{11}\ddot{q}_1 + m_{12}\ddot{q}_2 + h_1 = \tau - SGN(\dot{q}_1)F_1 - b_1\dot{q}_1, \tag{2}$$

$$m_{21}\ddot{q}_1 + m_{22}\ddot{q}_2 + h_2 = -SGN(\dot{q}_2)F_2 - b_2\dot{q}_2, \tag{3}$$

Where,

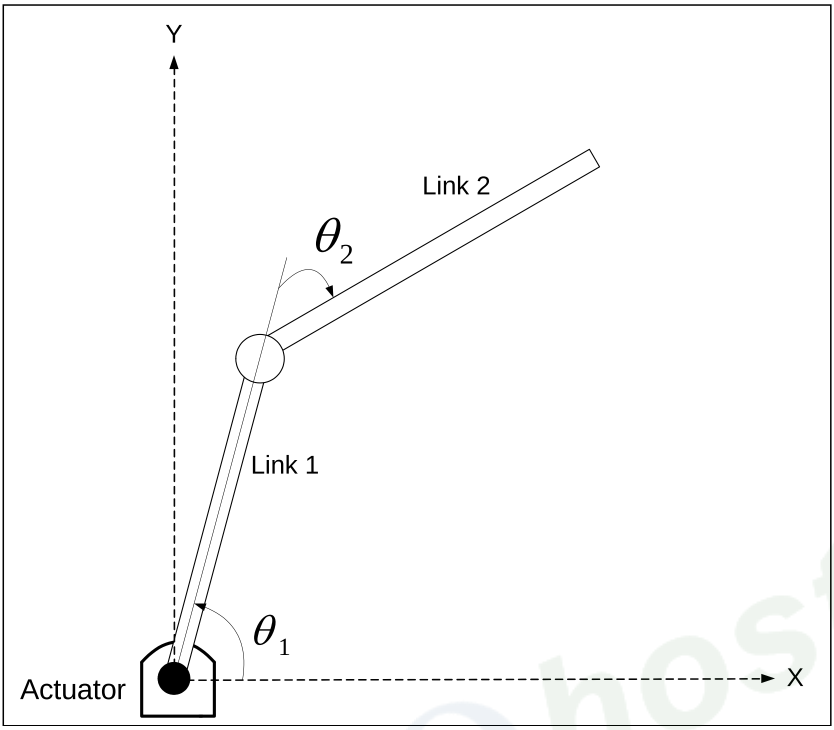


Figure 1. Schematic diagram of the manipulator.

$$m_{11} = c_1 + c_2 + 2c_3 Cosq_2,\; m_{12} = c_2 + c_3 Cosq_2,$$

$$m_{21} = m_{12},\; m_{22} = c_2,$$

$$h_1 = -c_3(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2)Sinq_2,\; h_2 = c_3\dot{q}_1^2\,Sinq_2.$$

The $F_i, b_i\dot{q}_i, i = 1,2$ represent the Coulomb and viscous friction forces respectively. The set-valued signum function is defined as:

$$SGN(x)\begin{cases} \{1\} & if \quad x > 0, \\ \{-1\} & if \quad x < 0, \\ [-1,1] & if \quad x = 0. \end{cases} \tag{4}$$

The above shown function suffers the fact that the solution does not give a clear indication on how to select an appropriate solution from the several possible solutions for a particular arm configuration.

3. The Robot Setup

In order to analyze the system's behavior, a 2R under-actuated manipulator was manufactured, as can be seen in Figure 2. The manipulator is actuated only at the first joint. The actuator used is a DC motor connected to the first link through a gearbox with a reduction ratio of 100:1, while the second joint is passive. Each of the joint has attached encoders in order to measure the rotation angle and there are torque sensors between the motor output shaft and the robot joint to measure the torque being supplied by the motor. The robot arms were made of an aluminum square section beam to ensure a resisting to bending lightweight arm. Lengths of arms are $l_1 = 30$ cm and $l_2 = 30$ cm respectively.

The control circuit is made of computer with MATLAB software connected to the robot through a data acquisition board that acquires the motion data of the two links. Input signal is generated by the MATLAB software and transferred to the motor using the electrical board, and the robot response is recorded using the MATLAB software.

Different methods for collecting data have been found in the literature. Using a pre-specified model, using a trajectory planning method or using a simulation program for this purpose are examples for some of these methods. However, there are always kinematics uncertainties presences in the real world such as ill-defined linkage parameters, links flexibility and backlashes in gear train, in this chapter, data were measured directly from sensors fixed on each joint, so every uncertainty in the dynamics of the system will be counted for.

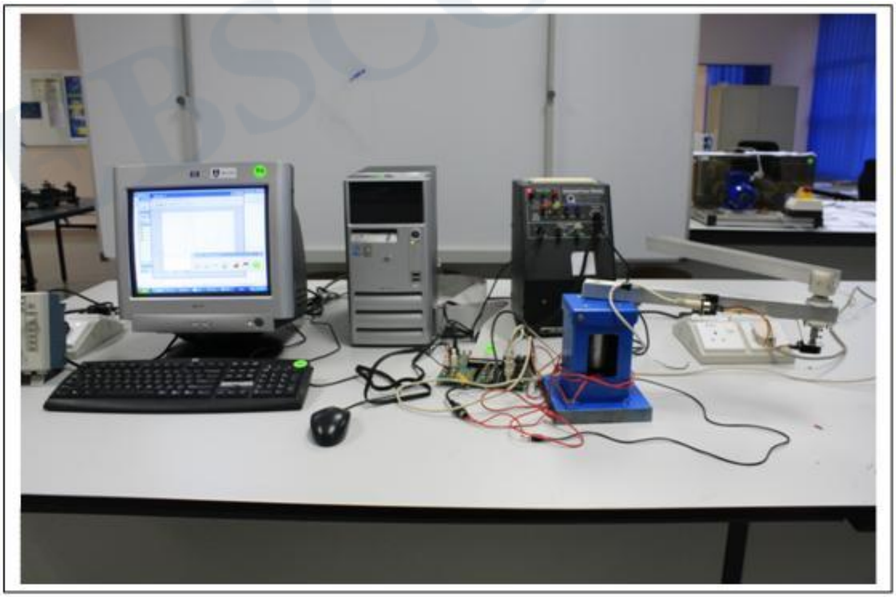


Figure 2. The robot system used.

4. Network Inversion Technique

The method of the network inversion was first proposed by Linden and Kindermann [30] where a conventional multilayer neural network is used to solve the forward problem, fixing the weights obtained during the training process after that in order to be used to estimate the inverse problem parameters.

In the usual multilayer network whose learning has completed, the input/output relation is given by:

$$y = f(w, x) \tag{5}$$

Where x, y and f are the input vector, the output vector and the function defined by the interlayer weights w of the network, respectively. Given the input vector x, the network calculates the output vector y.

In this method, after finding forward relation f by learning, the output data can be estimated using the fixed weights obtained from the learning process. Then, the input x can be updated according to the calculated input correction signal, based on the duality of the weights and input in equation (5). Actually, the input is estimated from the output by correcting input based on the output error repeatedly, as shown in Figure 3. By this way, the inverse problem for estimating input x from output y is solved using the multilayer neural network by using forward relation inversely [28].

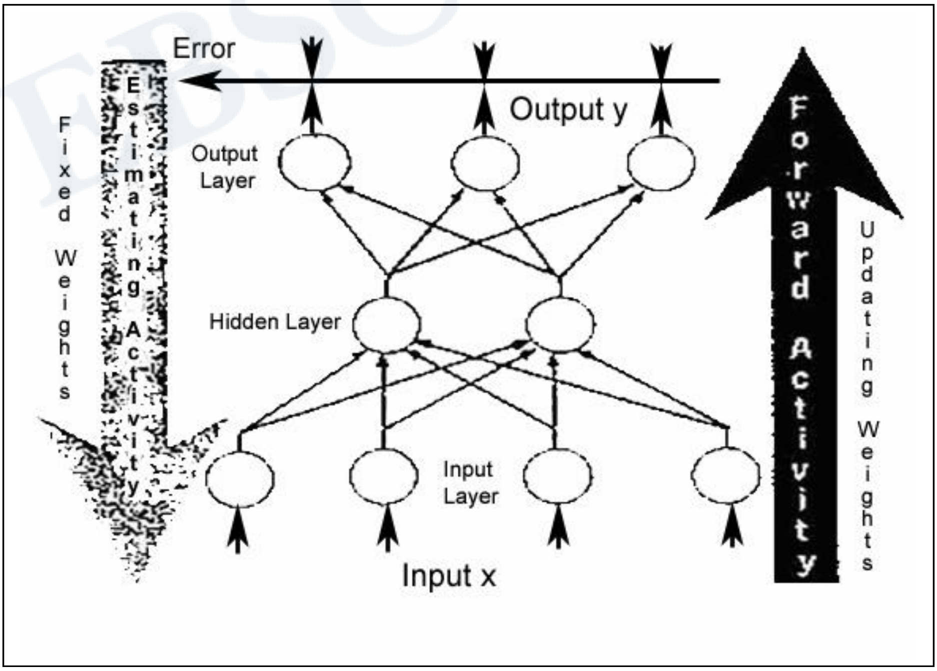


Figure 3. Inverse estimation by network inversion.

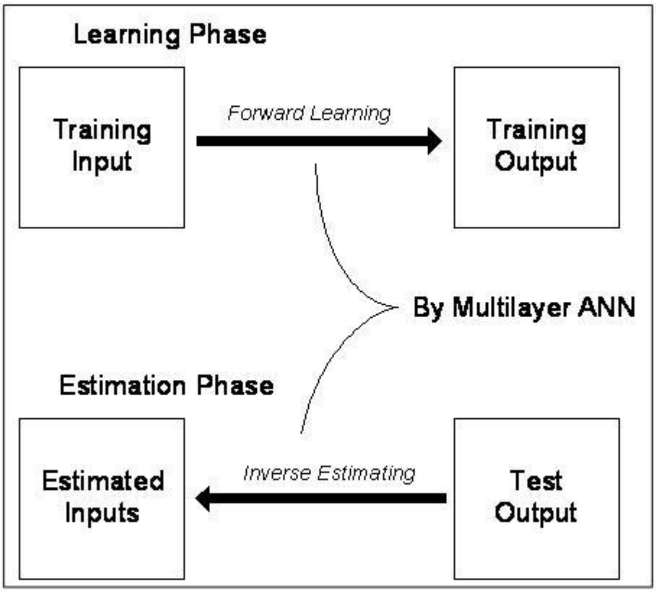


Figure 4. Two-step procedure to solve the inverse problem by network inversion.

The technique is implemented in the two phases: *forward learning* and *inverse estimating* to solve the inverse problem by the network inversion. The procedure is shown in Figure 4. In the learning phase, the learning input x is provided with the learning output y and calculate the output error E. Then, the weight is updated until the output produced by the network is the same as (or sufficiently close to) the desired output, this a procedure based on the usual backpropagation method.

In the estimation phase, the relation obtained in the learning is fixed, provide the random input x and the test output y and calculate the output error E. Then, the input is updated. By using this procedure, the input is estimated from the output [30].

5. ANN IMPLEMENTATION

The network inversion technique was implemented in two phases. The first phase was the forward learning phase in order to find the weights; the weight is updated until for each input the output produced by the network is the same as (or sufficiently close to) the desired output. Then, the weight is going to be fixed in the second phase, which is the inverse estimating phase in order to estimate the passive joint's path.

5.1. Forward Learning Phase

Sinusoidal excitation signal was applied to the actuator and the dynamic coupling effect was moving the passive joint. Figures 5 and 6 are graphically showing both active and passive joins response.

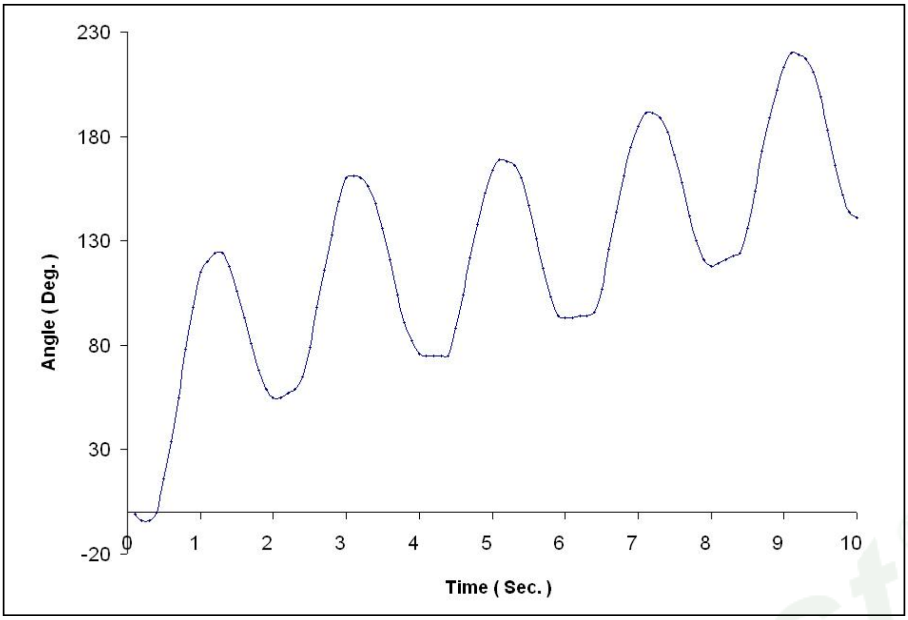


Figure 5. Trajectory of the active joint.

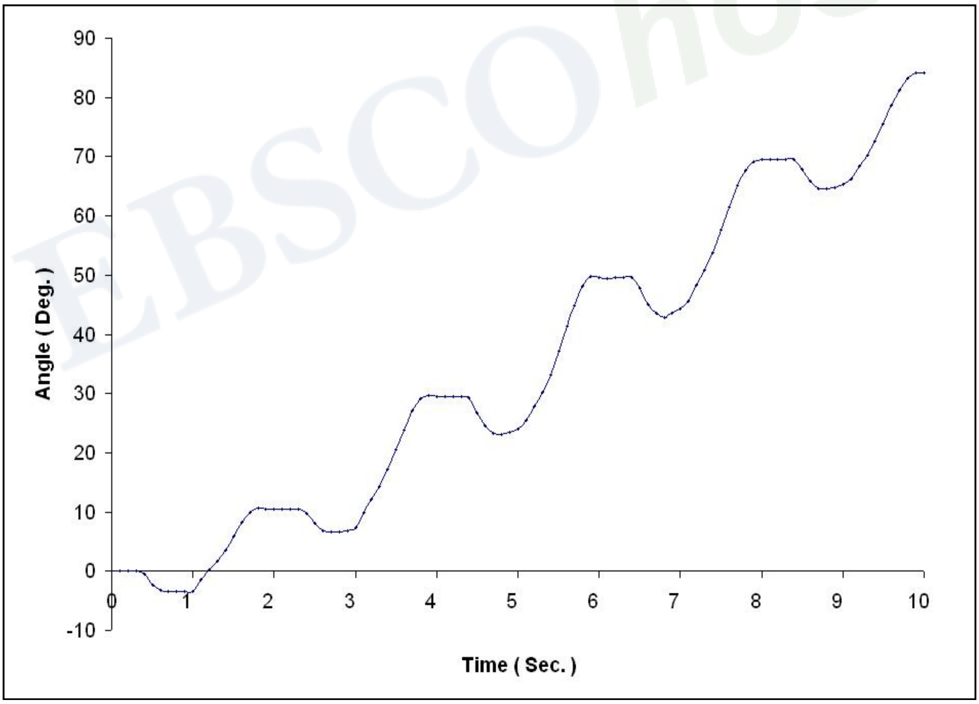


Figure 6. Trajectory of the passive joint.

A supervised feed forward ANN was designed using C programming language to learn the system behavior over its workspace. The network consists of input, output and one hidden layer, the input vector for the network consists of the angular displacement, the torque applied at the active joint (first joint) and the time interval, while the output vector was the angular position of the passive joint (second joint). As can be seen in Figure 7, every neuron in the network is fully connected with each other, sigmoid transfer function was used to be the activation function, and generalized backpropagation delta learning rule (GDR) algorithm was used in the training process.

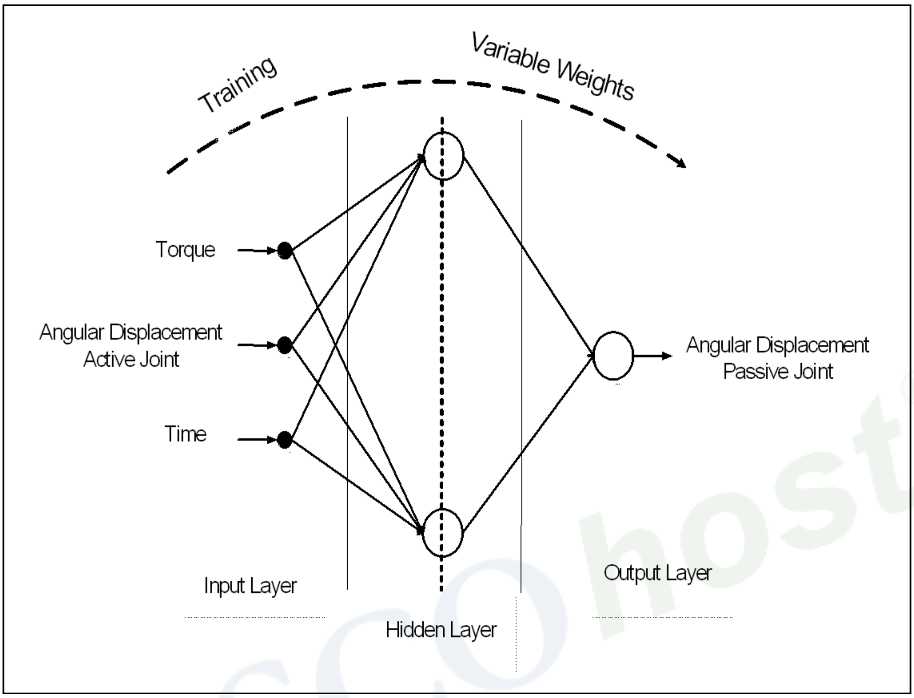


Figure 7. The Artificial Neural network used in the first phase (Forward learning).

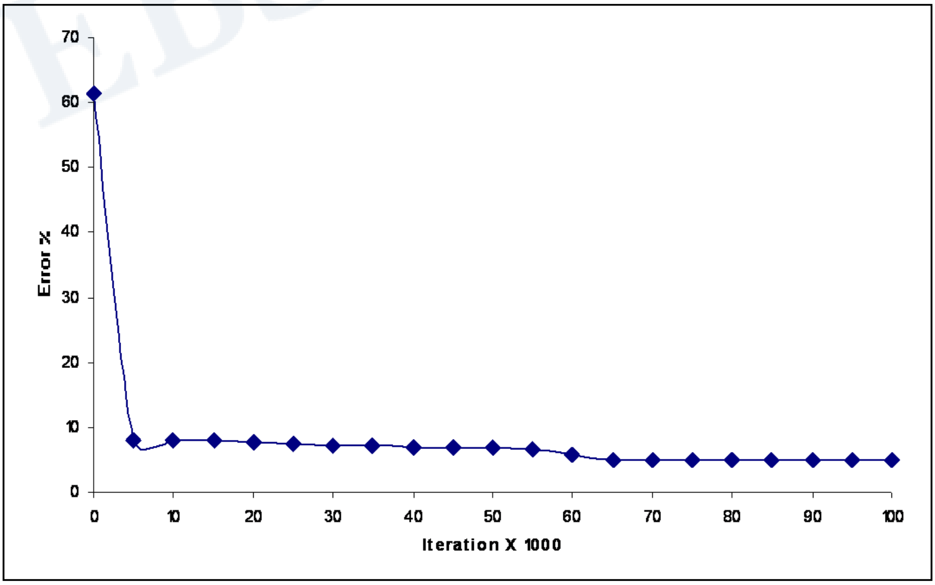


Figure 8. The learning curve of the forward learning phase.

All control datasets values had been scaled individually so that the overall difference in the dataset was maximized. Training data were divided into 50 input–output sets, which covered the entire work cell of the manipulator. To build the control knowledge, a training process was carried out using the experimentally obtained data. The network was trained by presenting several target points that the network had to learn, number of neurons in the hidden layer was set to 25 with a constant learning factor of 0.9 by trial and error. Figure 8 shows the building knowledge process for the system. The average absolute error was 4.9% after 100,000 Iterations.

5.2. Inverse Estimating Phase

To verify the success of the control algorithm, a new data set that represents new path, a path that the network was not previously trained for; was introduced to the inverted network that can be seen in Figure 9. The weights from the previous stage were fixed and the desired path for the passive joint was applied as an excitation signal to the inverted network In order to estimate the necessary input parameters to drive the passive joint to follow the desired trajectory.

The estimated input parameters were then used as excitation signal to the network used in the first phase to obtain the estimated path of the passive joint. To show the success of this control approach, the experimental trajectory tracking of the estimated passive joint is shown in Figure 10. As this figure clearly shows, the estimated trajectory was closely corresponding to the target trajectory.

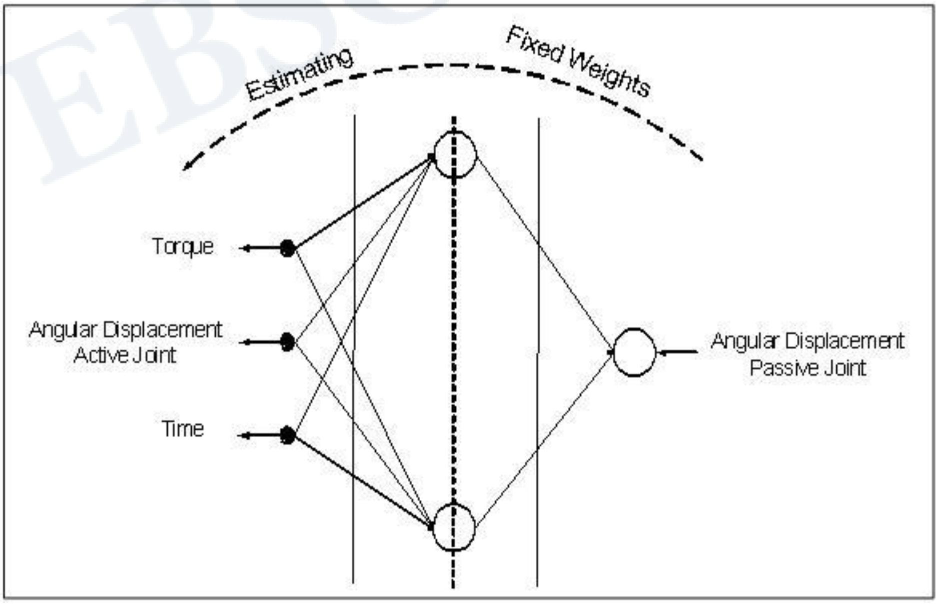

Estimating
Fixed Weights
Torque
Angular Displacement Active Joint
Time
Angular Displacement Passive Joint

Figure 9. The Artificial Neural network used in the second phase (estimating).

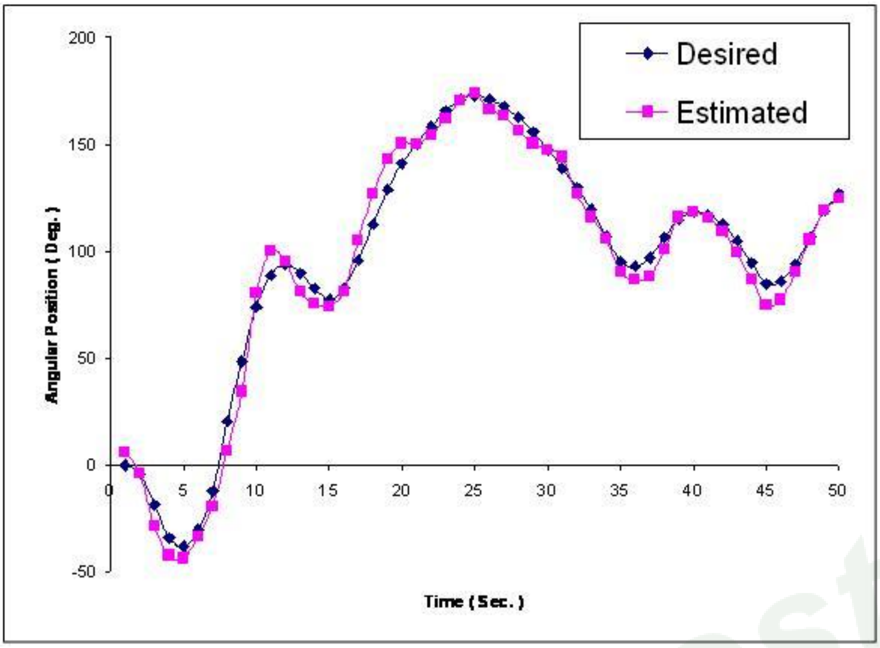


Figure 10. Estimated trajectory tracking of the passive joint.

Conclusions

In this research, we proposed to introduce the network inversion technique to the problem of under-actuated robot control that estimating the input parameters necessary to drive the passive joint to follow a desired trajectory from the output parameters of the network. The effect of the network inversion was confirmed experimentally using 2R under-actuated robot arm.

As a conclusion, it was shown that using the network inversion technique to estimate the input parameters from the output parameters of the robot arm was a successful technique. Consequently, it was shown that the network inversion could solve the problem of under-actuated robot control.

References

[1] Yu, KH; Shito, Y; Inooka, H. Position Control of an Underactuated Manipulator Using Joint Friction. *International Journal of Non-Linear Mechanics*, 1998, 33(4), 607-614.

[2] Lynch, KM; Shiroma, N; Arai, H; Tanie, K. Collision-Free Trajectory Planning for a 3-DOF Robot with a Passive Joint. *International Journal of Robotics Research*, 2000, 19(12), 1171-1184.

[3] Oriolo, G; Nakamura, Y. Control of Mechanical Systems With Second Order Nonholonomic Constraints: Underactuated Manipulator, Proceedings of the 30th Conference on Decision and Control, Brighton, UK, December 1991.

[4] Sørdalen, OJ; Nakamura,Y; Chung,WJ. Design of a nonholonomic manipulator. Proceedings of the *IEEE International Conference on Robotics and Automation*, 1994, 8–13.

[5] Hauser, J; Murray, RM. Nonlinear controllers for non-integrable systems: The Acrobot example. *Proceedings of the American Control Conference*., 1990, 669–671.

[6] Spong, MW. Swing up control of the Acrobot. Proceedings of the IEEE International *Conference on Robotics and Automation*, 1994, 2356–2361.

[7] Berkemeier, MD; Fearing, RS. Tracking fast inverted trajectories of the underactuated acrobot. Proceedings of the IEEE *Transactions on Robotics and Automation*, 1999, 15(4), 740-750.

[8] Takashima, S. Control of gymnast on a high bar. Proceedings of the IEEE/RSJ International *Conference on Intelligent Robots and Systems, Osaka,* Japan, 1991, 1424–1429.

[9] Saito, F; Fukuda, T; Arai, F. Swing and locomotion control for a two-link brachiation robot. *Proceedings of the IEEE Control Systems Magazine*, 1994, 14(1), 5–12.

[10] Luca, AD; Mattone, R; Oriolo, G; Stabilization of an Underactuated Planar 2R Manipulator. *International Journal of Robust and Nonlinear Control*, 2000, 181-198.

[11] Luca, AD; Oriolo, G. Trajectory Planning and Control for Planar Robots With Passive Last Joint. *International Journal of Robotics Research*, 2002, 21, 575-590.

[12] Arai, H; Tachi, S. Position Control of a Manipulator With Passive Joints Using Dynamic Coupling. IEEE Trans. *On Robotics and Automation*, 1991,7(4), 528-534.

[13] Mukherjee, R; Chen, D. Control of Free-Flying Underactuated Space Manipulators To Equilibrium Manifolds. IEEE Trans. *On Robotics and Automation*, 1993, 9(5), 561-570.

[14] Yu, KH; Takahashi, T; Inooka, H. Dynamics and Motion Control of a Two-Link Robot Manipulator With a Passive Joint. Proc. 1995 IEEE/RSJ Int. Conf. On Intelligent *Robots and Systems*, 1995, 2, 311-316.

[15] Bergerman, M; Lee, C; Xu, Y. Experimental Study of an Underactuated Manipulator. Proc. 1995 IEEE/RSJ Int. Conf. *On Intelligent Robotics and Systems*, 1995, 2, 317-322.

[16] Mahindrakar, AD; Rao, S; Banavar, RN. Point-to Point Control of a 2R Planar Horizontal Underactuated Manipulator. *International Journal of Mechanism and Machine Theory*, 2006, 41, 838-844.

[17] Hasan, AT; Hamouda, AMS; Ismail, N; Aris, I; Marhaban, MH. Trajectory Tracking for a Serial Robot Manipulator Passing Through Singular Configurations Based on the Adaptive Kinematics Jacobian Method. Proceedings of the IMechE Part I: *Journal of Systems and Control Engineering*, 2009, 223(3), 393-415.

[18] Hasan, AT; Ismail, N; Hamouda, AMS; Aris, I; Marhaban, MH; Al-Assadi, HMAA. Artificial Neural Network-Based Kinematics Jacobian Solution for Serial Manipulator Passing Through Singular Configurations. *International Journal of Advanced in Engineering Software*, *International Journal of Advances in Engineering Software,* 2010, 41, 359–367.

[19] Begovich, O; Sanchez, ED; Maldonado, M. Takagi-Sugeno Fuzzy Scheme for Real-Time Trajectory Tracking of an Underactuated Robot. Proceedings of the IEEE *Transactions on Control Systems Technology*, 2002, 10(1), 14-20.

[20] Al-Assadi, HMAA; Hamouda, AMS; Ismail, N; Aris, I. An adaptive learning algorithm for controlling a two-degree-of-freedom serial balland- socket actuator. Proc. IMechE, Part I: *Journal of Systems and Control Engineering*, 2007, 221(7), 1001–1006.

[21] Kalorgirou, SA. Artificial neural networks in renewable energy systems applications: a review. *Journal of Renewable and sustainable energy reviews*, 2001, 5, 373-401.

[22] Hasan, AT; Hamouda, AMS; Ismail, N; Al-Assadi, HMAA. An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator. *International Journal Advances in Engineering Software*, 2006, 37, 432–438.

[23] Karilk, B; Aydin, S. An improved approach to the solution of inverse kinematics problems for robot manipulators. *Engineering applications of artificial intelligence*, 2000, 13, 159-164.

[24] Köker, R. Reliability-based approach to the inverse kinematics solution of robots using Elman's networks. *International Journal of Engineering Applications of Artificial Intelligence*, 2005, 18, 685-693.

[25] Hasan, AT; Hamouda, AMS; Ismail, N; Al-Assadi, HMAA. A new adaptive learning algorithm for robot manipulator control. Proc. IMechE, Part I: J. *System and Control Engineering*, 2007, 221(4), 663–672.

[26] Kuroe, Y; Nakai, Y; Mori, T. A new neural network learning on inverse kinematics of robot manipulators. International conference on neural networks, *IEEE world congress on computational Intelligence,* 1994, 5, 2819-2824.

[27] Martín, P; Millán, J. Robot arm reaching through neural inversion and reinforcement learning. *Journal of robotics and autonomous systems*, 2000, 31, 227-246.

[28] Ogawa, T; Matsuura, H; Kanada, H. A Solution of Inverse Kinematics of Robot Arm Using Network Inversion. Proceedings of the International conference on computational Intelligence for modelling, control and automation, 2005.

[29] Fu, KS; Gonzalez, RC; Lee, CSG. *Robotics control, sensing, vision and intelligence.* McGraw-Hill, New York. 1987.

[30] Linden, A. and Kindermann, J; Inversion of Multilayer Networks. Proceedings of the *International Joint conference on Neural Networks*, 1993, 3, 188-1993.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 15

NONLINEAR FORECASTING WITH A HYBRID APPROACH COMBINING SARIMA, ARCH AND ANN

***Erol Egrioglu*[1]*, Cagdas Hakan Aladag*[2] *and Cem Kadilar*[2]**

[1]Department of Statistics, Ondokuz Mayis University, Samsun, Turkey
[2]Department of Statistics, Hacettepe University, Ankara, Turkey

ABSTRACT

Time series forecasting is a vital issue for many institutions. In the literature, many researchers from various disciplines have tried to improve forecasting models to reach more accurate forecasts. It is known that real life time series has a nonlinear structure in general. Therefore, conventional linear methods are insufficient for real life time series. Some methods such as autoregressive conditional heteroskedastiacity (ARCH) and artificial neural networks (ANN) have been employed to forecast nonlinear time series. ANN has been successfully used for forecasting nonlinear time series in many implementations since ANN can model both the linear and nonlinear parts of the time series. In this study, a novel hybrid forecasting model combining seasonal autoregressive integrated moving average (SARIMA), ARCH and ANN methods is proposed to reach high accuracy level for nonlinear time series. It is presented how the proposed hybrid method works and in the implementation, the proposed method is applied to the weekly rates of TL/USD series between the period January 3, 2005 and January 28, 2008. This time series is also forecasted by using other approaches available in the literature for comparison. Finally, it is seen that the proposed hybrid approach has better forecasts than those calculated from other methods.

Keywords: ARCH models; artificial neural networks; exchange rates; forecasting; nonlinearity; time series.

1. INTRODUCTION

Real life time series generally have both linear and nonlinear structures. Using linear approaches to analyze these time series is not a wise choice. Instead of using linear models, nonlinear time series approaches should be preferred to obtain more accurate results (Aladag et al., 2010a). In the literature, nonlinear time series models such as ARCH can be used to analyze these series. On the other hand, these nonlinear time series models cannot be used for every time series since they are useful for time series which have specific structures. For instance, utilizing ARCH models can lead to wrong results if there is no linear relationship between the lagged variables of square error obtained from autoregressive integrated moving average (ARIMA) models.

In recent years, feed forward artificial neural networks (FANN) have been widely used to analyze nonlinear time series (Egrioglu et al., 2008). One of the advantages of FANN is that it is a method based on data so it can be used to analyze any time series. When FANN method is employed, there is no need to examine the structure of time series. The detailed information about the usage of FANN in time series analysis can be found in Gunay et al. (2007) and Aladag et al. (2008).

FANN method has proved its success on forecasting time series. At the same time, both theoretical and empirical findings in literature show that combining different methods can be an effective and efficient way to improve forecasts (Aladag et al., 2009). In the literature, therefore, there are many studies in which linear and nonlinear time series methods are combined to get more accurate forecasts. Zhang (2003) proposed a hybrid method combining ARIMA and FANN. Faruk (2010) also used a hybrid neural network and ARIMA model for time series forecasting. Buhamra et al. (2003) proposed another hybrid approach that uses Box-Jenkins method to determine inputs of FANN. Pai and Lin (2005) proposed a hybrid ARIMA and support vector machines model. Aladag et al. (2009) suggested a hybrid approach that combines ARIMA and recurrent neural networks. Erilli et al. (2010) improved a hybrid method by combining FANN and recurrent neural networks.

In this study, a novel hybrid approach is proposed to analyze time series having both linear and nonlinear structures. To reach high accuracy level in forecasting, ARCH, SARIMA and FANN methods are combined in the proposed method. In the implementation, the proposed hybrid approach is applied to Turkish lira / US dollar (TL/USD) exchange rate series. For a comparison, this time series is also forecasted by using other methods available in the literature. As a result of the comparison, it is seen that more accurate forecasts are obtained when the proposed hybrid method is employed.

2. ARTIFICIAL NEURAL NETWORKS

ANN can be defined as a method which mimics biological neural networks. ANN is also known as a universal function approximator. Different ANN types have been used in various fields such as optimization, forecasting, classification, and engineering. ANN method has proved its success in forecasting applications (Zhang et al., 1998). In the literature, FANN has been widely used for forecasting problem (Aladag et al., 2010b). A multilayer FANN includes three layers which are input, hidden, and output layers. Each layer consists of neurons and there are connections between neurons in different layers. Degree of each connection is represented by a corresponding weight. In forecasting problem, the input values are the lagged variables of the time series. The number of input is determined due to the number of lagged variables. Although various approaches have been proposed for determining the number of hidden layer's neuron, the most preferred method is trial and error. One neuron is generally used for output layer in forecasting problems. A broad multilayer FANN architecture is illustrated in Figure 1.

An important decision is to determine the activation function when FANN is used for forecasting. Activation function is the nonlinear part of ANN method. Tangent hyperbolic, logistic, and linear functions are the most preferred activation functions in the literature. In order to provide nonlinear mapping, nonlinear functions such as tangent hyperbolic and logistic activation functions are mostly preferred in hidden layer. For output layer, both linear and nonlinear activation functions can be used.

After the architecture structure and activation function are selected, determined FANN model is trained using a learning algorithm. Back propagation and Levenberg Marquardt learning algorithms are well known methods for training of a network. Detailed information about the components of ANN, such as architecture, activation function, and learning algorithm, can be found in Zurada (1992), Cichocki and Unbehauen (1993), and Gunay et al. (2007).

3. The Proposed Method

Using a method which can model either linear or nonlinear part of time series can lead to wrong results since real life time series can have both linear and nonlinear structures at the same time. Time series, which have different structures such as linear and nonlinear, can be successfully modeled by using hybrid approaches. Therefore, various hybrid approaches have been proposed in the literature to reach more accurate results. In this study, a new hybrid approach based on SARIMA, ARCH, and FANN methods is proposed. The proposed hybrid method can be summarized in three steps as follows:

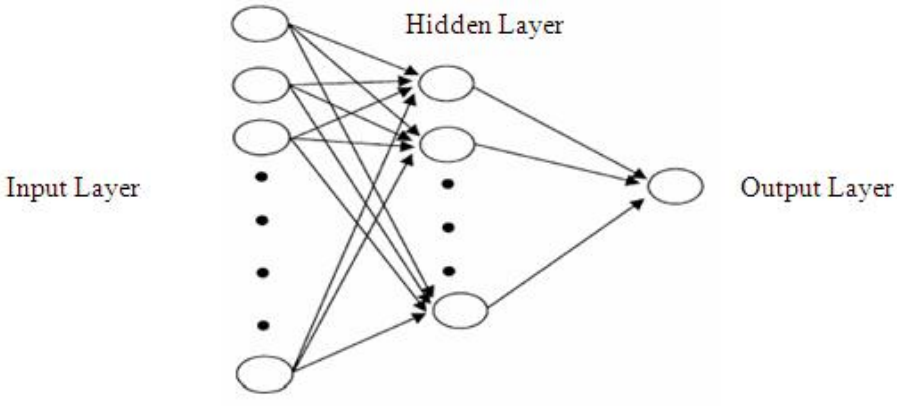


Figure 1. Multilayer feed forward artificial neural networks with one output neuron.

- The linear part of time series can be modeled by using SARIMA model.
- The heteroskedasticity problem in time series is solved by using ARCH model.
- After using SARIMA and ARCH models, obtained errors are analyzed by using FANN so the nonlinear part of time series, which cannot be modeled by SARIMA and ARCH models, can be modeled.

Any time series can be written as given below:

$$y_t = Z_t + W_t,$$

where y_t denotes original time series, Z_t denotes linear and nonlinear components of the time series that can be modeled by SARIMA and ARCH, W_t denotes the nonlinear part that can be modeled by FANN. According to mentioned representations, the proposed hybrid approach can be given as follows:

Step 1. By applying SARIMA and ARCH models to the time series y_t, predictions $\hat{Z}_t$ and errors e_t ($e_t = y_t - \hat{Z}_t = W_t$) are calculated. The errors obtained from SARIMA and ARCH models are considered as equivalent to series W_t.

Step 2. By analyzing series e_t with FANN, predictions $\hat{e}_t$ ($\hat{e}_t = \hat{W}_t$) is calculated.

Step 3. Predictions obtained from SARIMA, ARCH, and FANN methods are summed then the predictions of the proposed hybrid method are calculated. The predictions can be calculated using the formula given as follows:

$$\hat{y}_t = \hat{Z}_t + \hat{W}_t$$

4. APPLICATION

For the exchange rate series, we take the weekly rates of Turkish lira / US dollar (TL/USD) series between the period January, 3 2005 and January, 28 2008. We obtain this data set having total 160 observations from Central Bank of Turkey. In this study, we examine the log differences of the levels of this series. In other words, we are interested in the series calculated by

$$y_t = \log(r_t) - \log(r_{t-1}),$$

where r_t is the exchange rate series. Here, the series, y_t, can be defined as the change of the exchange rate series in the logarithmic transformation from the period to period. The graph of this transformed series is given in Figure 2.

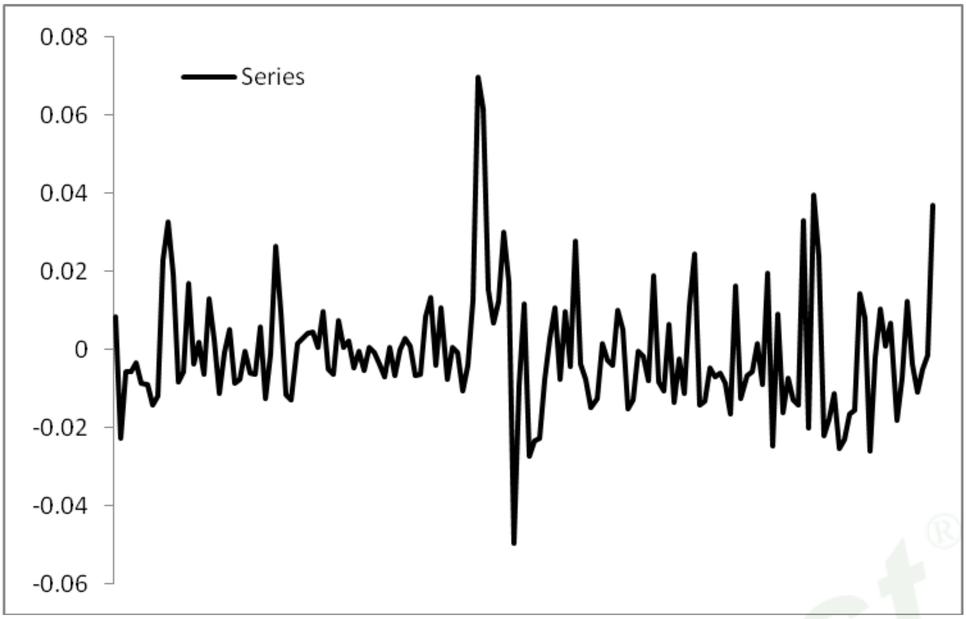


Figure 2. The graph of the y_t series.

As we keep the first 144 observations as in-sample data (training data for neural network analysis) and the last 16 observations as out-of-sample data (testing data for neural network analysis), the methods in this section are performed to only the first 144 data. Kadilar et al. (2009) analyzed the same time series. The application of the proposed hybrid method is given below.

Step 1.

When SARIMA is applied to the series, SARIMA(0,0,1) $(0,0,1)_6$ is found as a best model in terms of Akaike information criterion (AIC). The model SARIMA(0,0,1) $(0,0,1)_6$ can be given as follows:

$$y_t = -0.001 - 0.219e_{t-1} + 0.187e_{t-6} + e_t$$

When autoregressive conditional heteroskedasticity – Lagrange Multiplier test is applied to errors calculated from the model determined in the previous step, it is seen that homoskedasticity assumption is not satisfied. Therefore, SARIMA-ARCH model is applied to the series. The used model SARIMA(0,0,1) $(0,0,1)_6$-ARCH(1) can be written as follows:

$$y_t = -0.001 - 0.021e_{t-1} + 0.086e_{t-6} + e_t\,,$$

$$e_t^2 = 0.00008 - 0.90081e_{t-1}^2 + u_t\,.$$

Step 2

In this step, the errors u_t calculated from the model SARIMA(0,0,1) $(0,0,1)_6$-ARCH(1) are analyzed by utilizing FANN. The number of inputs and neurons in the hidden layer are varied between 1 and 12. One neuron is used in the output layer so 144 architectures are totally examined. The best architecture among these examined ones is determined by using root of mean square error (RMSE). RMSE is calculated using the formula given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_t - \hat{y}_t)^2}{n}}$$

where y_i is the actual value; $\hat{y}_i$ is the predicted value; n is the size of the test set. Levenberg Marquardt algorithm is used as learning algorithm. In all of neurons, logistic function given below is employed as activation function:

$$f(x) = \frac{1}{1+\exp(-x)}.$$

After 144 RMSE values are examined, it is seen that the architecture 3-4-1, which has three inputs, 4 hidden layer neurons, and one output neuron, has the smallest RMSE value. In other words, the architecture 3-4-1 is determined as the best architecture.

Step 3

In the final step, by summing the forecasts obtained from SARIMA (0,0,1) $(0,0,1)_6$-ARCH(1) and FFNN (3-4-1), forecasts of the proposed hybrid method are calculated.

After the time series is forecasted by the proposed hybrid method, the series is also analyzed by using SARIMA model and SARIMA-ARCH model. All calculations are performed by using Matlab 2009a version. The results obtained from SARIMA, SARIMA-ARCH, and the proposed hybrid approaches are summarized in Table 1. For all methods, RMSE values calculated over test set are shown in Table 1.

According to Table 1, the proposed hybrid approach has the smallest RMSE value. In other words, the proposed hybrid method produces the most accurate forecasts in terms of RMSE criterion. To show the forecasting performance of the proposed method visually, the graph of the forecasts obtained from the proposed method with the observations in test set, is given in Figure 3. When Figure 3 is examined, it is clearly seen that the forecasts produced by the proposed hybrid method are good.

Table 1. RMSE and MAPE values of the models for out-of-sample data

	SARIMA	SARIMA-ARCH	The proposed method
RMSE	0,01500	0,01551	0,01361

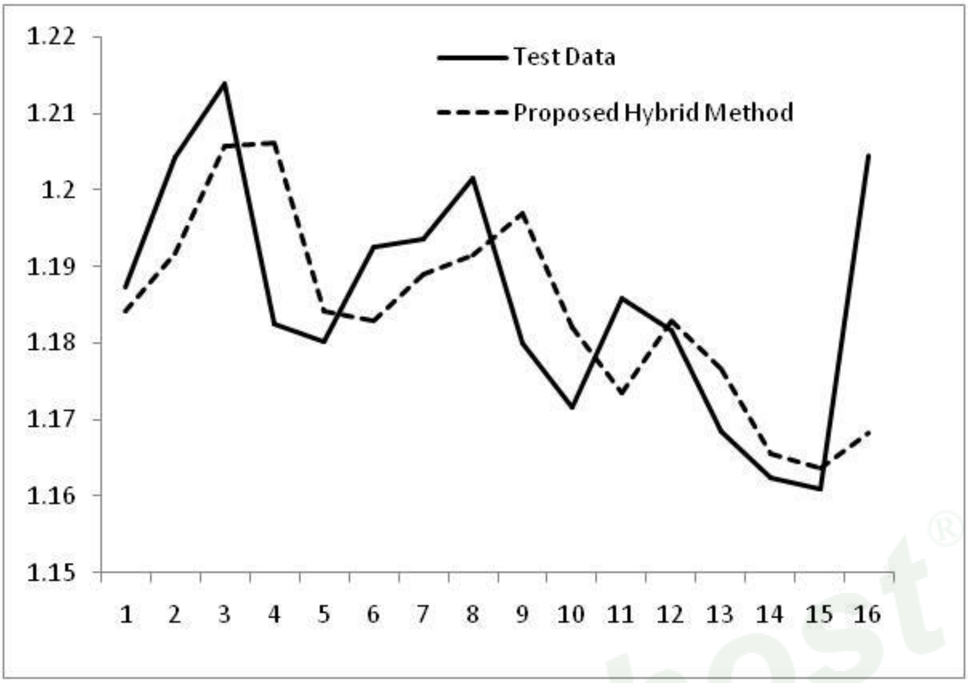


Figure 3. The graph of the proposed hybrid method forecast and test data.

CONCLUSION

Real life time series includes both linear and nonlinear components at the same time. To analyze these series accurately, hybrid approaches have been proposed in the forecasting literature. In this study, a new hybrid approach combining SARIMA, ARCH, and FANN methods is suggested to get more accurate forecasts. TL/USD exchange rate series is forecasted by using the proposed method to show the applicability of the method. This time series is also analyzed by SARIMA and SARIMA-ARCH models for the aim of comparison. It is seen that the proposed hybrid approach produces better forecasts than those obtained from other models in terms of RMSE values calculated over the data set.

It is known that most of economic time series includes heteroskedasticity problem. When the proposed hybrid approach is used, this problem is solved since the method includes ARCH model. In addition, the proposed hybrid method can analyze all real life time series since the method can model linear and nonlinear parts of time series. In future studies, it is possible to improve the proposed hybrid approach by trying different ANN types or components.

REFERENCES

Aladag, CH; Egrioglu, E; Gunay, S. A new architecture selection strategy in solving seasonal autoregressive time series by artificial neural networks, *Hacettepe Journal of Mathematics and Statistics,* 2008, 37(2), 185-200.

Aladag, CH; Egrioglu, E; Kadilar, C. Forecasting nonlinear time series with a hybrid methodology, *Applied Mathematics Letters*, 2009, 22, 1467-1470.

Aladag, CH; Egrioglu, E; Gunay, S; Basaran, MA. Improving weighted information criterion by using optimization, *Journal of Computational and Applied Mathematics*, 2010a, 233, 2683-2687.

Aladag, CH; Egrioglu, E; Kadilar, C. Modeling brain wave data by using artificial neural networks, *Hacettepe Journal of Mathematics and Statistics*, 2010b, 39 (1), 81–88.

Buhamra, S; Smaoui, N; Gabr M. The Box-Jenkins analysis and neural networks: prediction and time series modelling, *Applied Mathematical Modeling*, 2003, 27, 805-815.

Cichocki, A; Unbehauen, R. Neural networks for optimization and signal processing, *John* Willey & Sons: New York, 1993.

Egrioglu, E; Aladag, CH; Gunay, S. A new model selection strategy in artificial neural network, *Applied Mathematics and Computation*, 2008, 195, 591-597.

Erilli, NA; E. Eğrioğlu, E; Yolcu, U; Aladag, CH; Uslu, VR. Forecasting of turkey inflation with hybrid of feed forward and recurrent artificial neural networks, *Doğuş University Journal*, 2010, 11(1), 42-55.

Faruk, DO. A hybrid neural network and ARIMA model for water quality time series prediction, *Engineering Applications of Artificial Intelligence*, 2010, 23, 586–594.

Gunay, S; Egrioglu, E; Aladag, CH. Introduction to single variable time series analysis, Hacettepe University Press: *Ankara*, 2007.

Kadilar, C; Simsek, M; Aladag, CH. Forecasting The Exchange Rate Series with ANN: The Case of Turkey, *Istanbul University Journal of Econometrics and Statistics,* 2009, 9, 17-29.

Pai, PF; Lin, CS. A Hybrid ARIMA and support vector machines model in stock price forecasting, *The International Journal of Management Science*, 2005, 33, 497-505.

Zhang, G; Patuwo, BE; Hu. YM. Forecasting with artificial neural networks: the state of the art, *International Journal of Forecasting*, 1998, 14, 35-62.

Zhang, G. Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing*, 2003, 50, 159-175.

Zurada, JM. Introduction of artificial neural systems, West Publishing: New York, 1992.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 16

TOWARDS A FORMALISATION OF EVOLVING CONNECTIONIST SYSTEMS

Michael J. Watts[*]
School of Earth and Environmental Sciences,
University of Adelaide, Australia

Abstract

Evolving Connectionist Systems (ECoS) are a class of constructive artificial neural networks that grow their structure as they learn. They have been widely applied to many problems. Among their advantages are fast, efficient training and a resistance to catastrophic forgetting. An attempt has previously been made to formally describe their operation and behaviour. This formalisation has some objections associated with them. This chapter describes the basic algorithms behind ECoS networks, critiques the previous formalisation and presents a new theory of ECoS networks that overcomes the objections associated with the previous work. Finally, the formalisation is tested on two well-known benchmark data sets.

1 Introduction

Evolving Connectionist Systems (ECoS) [7, 10, 21] are a class of constructive artificial neural networks that are similar in the way in which neurons are added to their structures, and in the way in which their connection weights are modified. The Evolving Fuzzy Neural Network EFuNN [8] was the first ECoS network, from which a generalised constructive ANN architecture and training algorithm was derived. Other ECoS networks include the Simple Evolving Connectionist System SECoS [19, 22]. For a review of existing ECoS algorithms and applications, see [21].

While it seems that for many the term "evolving" evokes thoughts of evolutionary computation, ECoS are not evolutionary algorithms. ECoS networks do not use the mechanisms of evolutionary computation, such as fitness-based selection, reproduction and mutation. Instead, as far as ECoS networks are concerned, the word "evolving" has the much broader meaning of change through time. ECoS was designed around the following principles [7]:

[*] Email address: e:mjwatts@ieee.org

1. The ECoS training algorithm is designed with the intention that all the algorithm can learn from the data is learned in the first training pass (one-pass learning). Additional exposure to the training data is not necessary.

2. ECoS are intended to be used in an on-line learning application. This means that new data will be constantly and continuously coming into the system, and that this data must be learned by the network without forgetting the old.

3. The general ECoS architecture and learning algorithm allows an ECoS network to accommodate new data without a catastrophic forgetting of the old.

4. The manner in which neurons are added to an ECoS means that some training examples are stored, initially, verbatim within the structure of the network. These examples are then either modified (refined) by exposure to further examples, or, depending upon the training parameters used, remain the same and can be later retrieved.

The advantages of ECoS are that they avoid the problems associated with traditional connectionist structures such as MLP [7, 10]: They are hard to over-train, due to the constructive nature of their learning algorithm; They learn quickly, as the learning algorithm is a one-pass algorithm, that is, it requires only a single presentation of the data set; They are far more resistant to catastrophic forgetting than most other models, as new training data is represented by adding new neurons, rather than accommodating the additional data in the existing neurons.

ECoS networks also have several advantages over other constructive algorithms: Firstly, they are not limited to a particular application domain, they can be applied to both classification and function approximation; Secondly, they do not require multiple presentations of the training data set, as is the case with some of the constructive algorithms in existence, such as RAN [18] and GAL [1]; Finally, they are able to continue learning and are not restricted to learning a single training set as some other constructive algorithms such as RAN and Cascade Correlation [4] are.

Traditional ANN are supported by a large body of theory [12, 16, 3, 13]. This body of theory describes:

- How the ANN training algorithms behave, given the settings of their training parameters.

- How the training algorithms allow the network to capture knowledge.

- How this knowledge is represented by the ANN.

This theory assists the neural network practitioner in both applying these algorithms and in optimising and extending them. A theoretical basis is also useful in assisting the acceptance of a new algorithm: other researchers are more likely to utilise a new algorithm if its theoretical grounding is known.

It is for these reasons that a theoretical basis to ECoS is desirable. Any theory, or formalisation, that describes the ECoS algorithm must cover two distinct aspects.:

- The behaviour, or state, of the network at any time t.

- The way in which the state of the network changes as it trains, which includes the effect each training parameter has on the changes made to the ECoS by the training algorithm.

The chapter is arranged as follows: Section 2 describes ECoS algorithms. Section 3 describes the existing theory and describes objections to it. Section 4 introduces the foundations of the new ECoS formalisation. Section 5 builds on this foundation to describe the effect of the two training parameters that control the addition of neurons to ECoS, the sensitivity threshold and error threshold. Section 6 investigates the conjectures made about the sensitivity threshold parameters in Section 5, while Section 7 investigates the conjectures made about the error threshold parameter in Section 5. In Section 8 the learning rate one parameter is investigated, while the effect of the learning rate two parameter is investigated in Section 9. Differences in the behaviour of ECoS networks over classification and function approximation problems are discussed in Section 10. The predictions are tested in Section 11 across two well-known benchmark data sets. These results are discussed in Section 12. Finally, conclusions and future work are presented in Section 13.

2 ECoS Algorithms

An ECoS network is a multiple neuron layer, constructive artificial neural network. An ECoS network will always have at least one ‘evolving’ neuron layer. This is the constructive layer, the layer that will grow and adapt itself to the incoming data, and is the layer with which the learning algorithm is most concerned. The meaning of the connections leading into this layer, the activation of this layer’s neurons and the forward propagation algorithms of the evolving layer all differ from those of classical connectionist systems such as MLP. For the purposes of this chapter, the term ‘input layer’ refers to the neuron layer immediately preceding the evolving layer, while the term ‘output layer’ means the neuron layer immediately following the evolving layer. This is irrespective of whether or not these layers are the actual input or output layers of the network proper. For example, in Figure 1, which shows a generic ECoS structure, the “input layer” could be the input layer proper of the network (as with SECoS networks) or it could be a neuron layer that processes the actual input values for presentation to the evolving layer (as with EFuNN networks). By the same token, the “output layer” could be the actual output layer of the network (as it is with SECoS) or a neuron layer that further processes the outputs of the evolving layer (as with EFuNN). The connection layers from the input neuron layer to the evolving neuron layer and from the evolving layer to the output neuron layer, are fully connected.

The activation A_n of an evolving layer neuron n is determined by Equation 1.

$$A_n = 1 - D_n \tag{1}$$

where D_n is the distance between the input vector and the incoming weight vector for that neuron. Since ECoS networks are fully connected, it is possible to measure the distance between the current input vector and the incoming weight vector of each evolving-layer neuron. Although the distance can be measured in any way that is appropriate for the inputs, this distance function must return a value in the range of zero to unity. For this reason, most ECoS algorithms assume that the input data will be normalised, as it is far

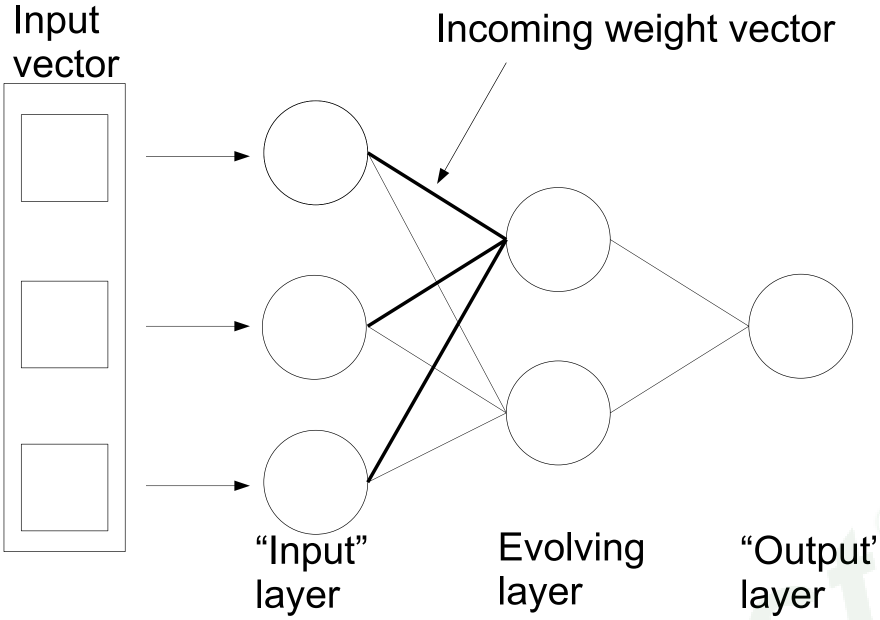


Figure 1. General ECoS architecture.

easier to formulate a distance function that produces output in the desired range if it is normalised to the range zero to unity.

Whereas most ANN propagate the activation of each neuron from one layer to the next, ECoS evolving layers propagate their activation by one of two alternative strategies. The first of these strategies, entitled *OneOfN* propagation, involves only propagating the activation of the most highly activated ("winning") neuron. The second strategy, *ManyOfN* propagates the activation values of those neurons with an activation value greater than the activation threshold A_{thr}.

The ECoS learning algorithm is based on accommodating new training examples within the evolving layer, by either modifying the weight values of the connections attached to the evolving layer neurons, or by adding a new neuron to that layer. The algorithm employed is described in Figure 2. The addition of neurons to the evolving layer is driven by the novelty of the current training example: if the current example is particularly novel (it is not adequately represented by the existing neurons) then a new neuron will be added. Four parameters are involved in this algorithm: the sensitivity threshold S_{thr}, the error threshold E_{thr}, and the two learning rates η_1 and η_2. The sensitivity threshold and error threshold both control the addition of neurons and when a neuron is added, its incoming connection weight vector is set to the input vector $\mathbf{I}$, and its outgoing weight vector is set to the desired output vector $\mathbf{O}_d$. The sensitivity and error thresholds are measures of the novelty of the current example. As can be seen in Figure 2, if the current example causes a low activation (that is, it is novel with respect to the existing neurons) then the sensitivity threshold will cause a neuron to be added that represents that example. If the example does not trigger the addition of a neuron via the sensitivity threshold, but the output generated by that example

```
for each input vector $\mathbf{I}$ and its associated desired output vector $\mathbf{O}_d$ do
  Propagate $\mathbf{I}$ through the network
  Find the most activated evolving layer neuron $j$ and its activation $A_j$
  if $A_j < S_{thr}$ then
    Add a neuron
  else
    Find the errors between $\mathbf{O_d}$ and the output activations $A_o$
    if $|\mathbf{O}_d - A_o| > E_{thr}$ then
      Add a neuron
    else
      Update the connections to the winning evolving layer neuron $j$
    end if
  end if
end for
```

Figure 2. ECoS learning algorithm.

results in an output error that is greater than the error threshold (that is, it had a novel output), then a neuron will be added.

The weights of the connections from each input i to the winning neuron j are modified according to Equation 2.

$$W_{i,j}(t+1) = W_{i,j}(t) + \eta_1(I_i - W_{i,j}(t)) \tag{2}$$

where:
$W_{i,j}(t)$ is the connection weight from input i to j at time t
I_i is the ith component of the input vector $\mathbf{I}$

The weights from neuron j to output o are modified according to Equation 3.

$$W_{j,o}(t+1) = W_{j,o}(t) + \eta_2 A_j E_o \tag{3}$$

where:
$W_{j,o}(t)$ is the connection weight from j to output o at time t
A_j is the activation of j
E_o is the signed error at o, as measured according to Equation 4.

$$E_o = O_o - A_o \tag{4}$$

where:
O_o is the desired activation value of output o
A_o is the actual activation of o.

This is essentially the perceptron learning rule. From this it becomes apparent that in [8] and subsequent publications [11, 23] the terms O_d and A_o above were incorrectly reversed.

3 Previous Formalisation

The following theory was proposed by Kasabov in [7, 9, 10] to describe the state and training of an ECoS network (specifically an EFuNN). Note that the notation used for the training parameters has been altered to bring it into line with those used in this chapter:

> Each rule neuron, e.g. r_j, represents an association between a hyper-sphere from the fuzzy input space and a hyper-sphere from the fuzzy output space, the $W1(r_j)$ connection weights representing the co-ordinates of the center of the sphere in the fuzzy input space, and the $W2(r_j)$-the co-ordinates in the fuzzy output space. The radius of an input hyper-sphere of a rule neuron is defined as $(1 - S_{thr})$ …For example, two pairs of fuzzy input-output data vectors $d1 = (Xd1, Yd1)$ and $d2 = (Xd2, Yd2)$ will be allocated to the first rule neuron r_1 if they fall into the r_1 input sphere and in the r_1 output sphere, i.e. the local normalised fuzzy distance between $Xd1$ and $Xd2$ is smaller than the radius r and the local normalised fuzzy difference between $Yd1$ and $Yd2$ is smaller than an error threshold E_{thr}. [9, pg 304]

On the topic of adaptation of the existing neurons, Kasabov goes on to say:

> Through the process of associating (learning) of new data point to a rule neuron, the centers of this neuron *(sic)* hyper-spheres adjust in the fuzzy input space depending on a learning rate η_1, and in the fuzzy output space depending on a learning rate η_2. [9, pg 304]

There are several objections to this theory. Firstly, the radius of the hyper-spheres is defined as $1 - S_{thr}$. The sensitivity threshold, however, is a property of the training algorithm, not of the evolving layer neurons. Although the example that caused the addition of that neuron will be within that radius, subsequent examples that cause the neuron to fire will not be. Also, if the radius of the hyper-spheres were dependent upon S_{thr}, then the radius of all hyper-spheres would be identical and only the learning rate parameters would have any effect upon training. Experimental results (Section 11) show that this is not the case. Secondly, the suggestion that the region defined by a neuron is a hyper-sphere is not supported by the canonical ECoS algorithm. ECoS evolving layer neurons are unthresholded, thus a neuron will activate if it is the closest in the input space to the current example, *no matter how distant the example actually is*. As long as the neuron is the closest, it will activate for that example, even if the distance is greater than $1 - S_{thr}$. Which neuron is closest to the example, however, depends on the co-ordinates of the other neurons in the evolving layer. Since these may be distributed in any manner within the input space, the boundaries between the regions defined by each neuron are not regular, that is, the polygons defined by the points represented by each neuron in the evolving layer of an ECoS network are not regular polygons, nor are they necessarily of regular size. Plainly, then, neither the description of the regions defined by neurons as hyper-spheres, nor the definition of these hyper-spheres, is appropriate for this purpose.

Another problem with this theory is that it does not describe the effect of the training parameters. Although experimental results [20] show that the parameters have different effects upon the behaviour of an ECoS network, the theory above does not describe this in any way.

Finally, the theory is untestable: it makes no predictions about the behaviour of the network or training algorithm as parameters are altered.

Some elements of the theory are satisfactory, however: each neuron in the evolving layer does provide a mapping, or association, from a region of input space to a region of output space, and the coordinates of these regions in the input space are defined by the connection weights of the neuron. These elements will be retained in the improved theory proposed in this chapter.

4 A New Formalisation of ECoS

With the existing theory unsatisfactory, it becomes necessary to formulate a new theory that overcomes the shortcomings elucidated above. Three assumptions are made:

1. That the ECoS network has one layer of connections coming into the evolving neuron layer, and one layer of connections going out of the evolving layer.

2. That the distance between two vectors will be measured so that the distance is in the range of zero to unity.

3. That the evolving layer neurons are not thresholded.

Within this chapter, the term *region* is intended to mean a set of points in n-dimensional space that is defined by specific boundaries. The term *volume* means the amount of unit space occupied by a region.

This theory will be in two parts: the state of the network at a time t; and the behaviour of the network in relation to the training parameters, that is, the way in which each training parameter effects the training process. The first set of theory here describes the state of the network.

4.1 Axioms of State

The following axioms are the basis of this theory. They are partially derived from an examination of the forward propagation algorithm of ECoS: some are also retained from the previous theoretical work described above. These axioms describe the way in which an ECoS network encapsulates what it has learned about the input space, that is, how an ECoS network represents knowledge.

Axiom 1 *each neuron n in the evolving layer of an ECoS network defines a single point in the input space.*

This is self evident from the ECoS algorithm. Since the activation of any neuron is based on the distance between the current input example and its incoming weights, each neuron therefore represents a single point in the input space.

Axiom 2 *the activation of a neuron n for an example $\mathbf{I}$ is proportional only to the distance of $\mathbf{I}$ from n.*

This also is self evident from the ECoS algorithm, specifically from Equation 1.

Axiom 3 *a neuron n will activate (win) iff its activation is greater than all other neurons in the evolving layer.*

Assuming the case of one-of-n activation, a neuron may propagate its activation to following layers only if it is the most highly activated. To be the most highly activated with a distance-based activation function, the point defined by the neuron in Axiom 1 must be the closest to **I**.

Axiom 4 *for every neuron j there is a region R_j with volume V_j in input space within which each point is closer to n than to any other neuron.*

This is self evident: in any plot with multiple points, there will be a region around each point that is closer to that point than any other. In effect, each neuron in the evolving layer corresponds to a Voronoi polygon in the input space [17]. This is consistent with the GAL algorithm [1], which is the constructive algorithm most similar to ECoS. In [1, pg 399], it is stated that:

> The input space is divided in the form of a Voronoi tessellation where exemplar units' domination regions are bounded by hyperplanes that pass through the medians of the two closest exemplar units.

Thus, an ECoS network with m neurons will partition the input space into m Voronoi regions, where a neuron will activate if an input example is at a point within the Voronoi region for that neuron. Figure 3 shows the Voronoi regions defined in two dimensional input space by a hypothetical ECoS network with three evolving layer neurons. The Voronoi region defined by the winning neuron j is shaded.

In Figure 4 the Voronoi regions of a SECoS network trained on the two spirals data set [14] is presented, where regions for neurons that represent the second of the two classes have been filled in. A basic spiral shape has become apparent, but is slightly coarser than for other constructive algorithms (see, for example, [6]).

From the above axioms and results, it is possible to infer some properties of ECoS networks. Assume that the task at hand is a classification problem of c classes, using an ECoS network with m neurons in the evolving layer, and the examples being presented to the network are uniformly distributed in input space.

The probability of an unknown example **I** of class C being correctly classified is determined by the probability of **I** falling in a region 'owned' by a neuron in the set of neurons M_C that represents C. In the general case this is determined by Equation 5.

$$P_t = \frac{\sum V_i}{\sum V_m} \tag{5}$$

where:
P_t is the probability of correctly classifying **I** as being a member of class C
V_i is the volume of region i, where $i \in M_C$
V_m is the volume of region m, where m is the set of all neurons in the evolving layer of the

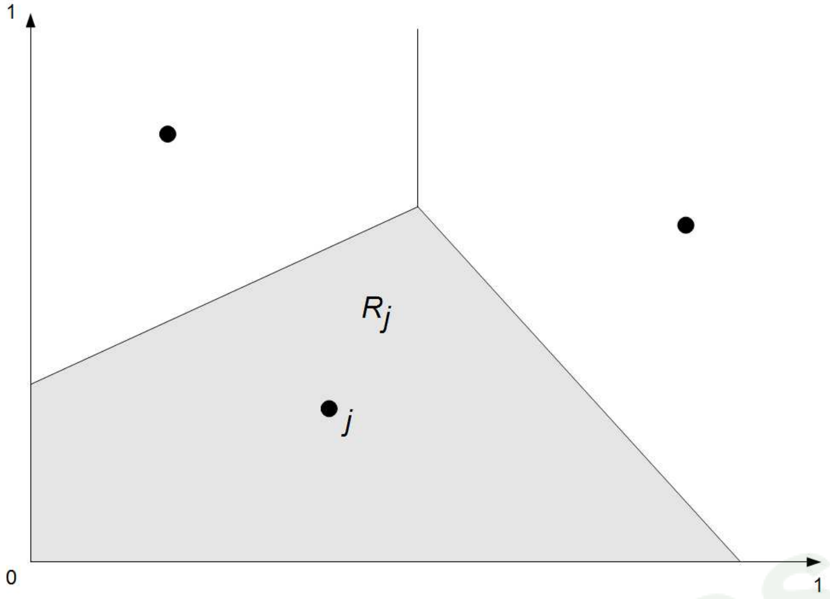


Figure 3. Voronoi regions defined by the evolving layer neurons of an ECoS network.

network. Since the distance measures used in ECoS must return values between zero and unity, the sum of all neurons regions volumes must be unity. Thus, Equation 5 simplifies to:

$$P_t = \sum V_i \tag{6}$$

With uniformly distributed examples, the volume of each of the m regions will be equal. Thus P_t is proportional to the number of neurons allocated for each class. This implies that in the case of an unbalanced training set, that is, a training set with a large number of examples of one class and smaller numbers of other classes, a network will be produced that is less likely to generalise well to the under-represented class.

For non-uniformly distributed examples, multiple classes that are tightly clustered together will cause each neuron to have a very small region associated with it. This will require a larger number of neurons to model the problem. That is, more complex problems will require more neurons.

5 Influence of the Neuron Addition Parameters

Following this line of reasoning, it is apparent that there are two regions to be considered. The first is defined by the sensitivity threshold S_{thr} and will be denoted by R_s, which has the volume V_s. The second is defined by the error threshold E_{thr}, which will be denoted by R_e and has the volume V_e. The region R_a is therefore the intersection of the region defined

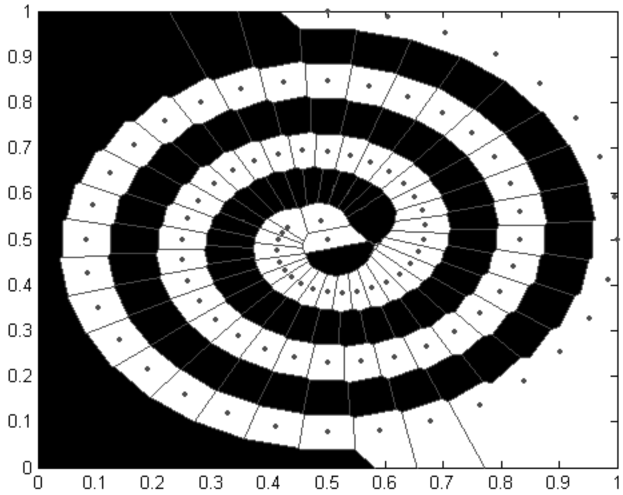


Figure 4. Voronoi regions of an ECoS network trained on the two spirals data set.

by these two regions and region R_j, that is:

$$R_a = R_j \cap (R_s \cap R_e) \tag{7}$$

If it is assumed that the training examples are uniformly distributed through the input space, and that any training input vectors that lie outside of R_a will cause the addition of a neuron, then the probability of any training example within the neurons Voronoi region R_j causing the addition of a neuron, is equivalent to the ratio between the volume V_j of R_j and the volume V_a of R_a:

$$P_a = \frac{V_j - V_a}{V_j} \tag{8}$$

If the examples are non-uniformly distributed in input space, then the relationship between P_a and V_j will be non-linear. In either case, however, it is apparent that the following relationship is true [1]:

$$V_a \to 0, P_a \to 1$$

Since V_a is defined as the volume of R_a, and R_a is defined as the intersection of R_s and R_e, it can be deduced that:

$$\begin{aligned} V_s &\to 0, V_a \to 0 \\ V_e &\to 0, V_a \to 0 \\ \Rightarrow V_s &\to 0, P_a \to 1 \\ \Rightarrow V_e &\to 0, P_a \to 1 \end{aligned}$$

[1] It can also be conjectured from this equation that an ECoS network will gain neurons rapidly during the early phases of training, but more slowly later on: as the network has few neurons initially, the volume assigned to each neuron is very large, thus the difference between V_j and V_a is very large and P_a is very high.

Since V_s and V_e are respectively determined by S_{thr} and E_{thr} it is conjectured that:

$$S_{thr} \rightarrow 1, P_a \rightarrow 1 \tag{9}$$

and:

$$E_{thr} \rightarrow 0, P_a \rightarrow 1 \tag{10}$$

No assumptions are made that these relations are linear. The relationship between each parameter and the rate at which neurons are added to the evolving layer will be influenced by the distribution of the training data in the input space, as well as the other training parameters.

6 Influence of the Sensitivity Threshold Parameter

The hyper-sphere defined by S_{thr} is defined by a single distance, denoted here as D_s. The value of D_s can be derived simply from the activation equation of the neuron, $A_j = 1 - D_j$, where D_j is the distance between j and the example **I**, by replacing D_j with D_s and rearranging to make D_s the dependent variable:

$$D_s = 1 - S_{thr}$$

Since V_a is a function of the distance D_s, $V_a = f(D_s)$, P_a is therefore also a function of D_s. Thus:

$$\begin{aligned} &S_{thr} \rightarrow 1, D_s \rightarrow 0 \\ \Rightarrow &S_{thr} \rightarrow 1, V_a \rightarrow 0 \\ \Rightarrow &S_{thr} \rightarrow 1, P_a \rightarrow 1 \end{aligned}$$

In other words, as the sensitivity threshold increases, so too does the probability of a neuron being added to the network. This proves the conjecture in Equation 9 and is consistent with experimental results (Section 11).

The region R_s as defined by D_s around j is displayed as the shaded region in Figure 5. This figure shows the Voronoi regions defined in a two dimensional input space by a hypothetical ECoS network with three neurons in the evolving layer.

7 Influence of the Error Threshold Parameter

Before considering the influence of the error threshold parameter, it is necessary to consider the activation A_o of an output neuron o. There are three cases to consider for the activation of an output neuron:

$$A_o = \begin{cases} 0, & \mathbf{W}_{j,o} = 0 \\ \mathbf{W}_{j,o} A_j, & 0 < A_j \mathbf{W}_{j,o} < 1 \\ 1, & A_j \mathbf{W}_{j,o} \geq 1 \end{cases} \tag{11}$$

Proving the conjecture in Equation 10 therefore requires a proof for each of these three cases.

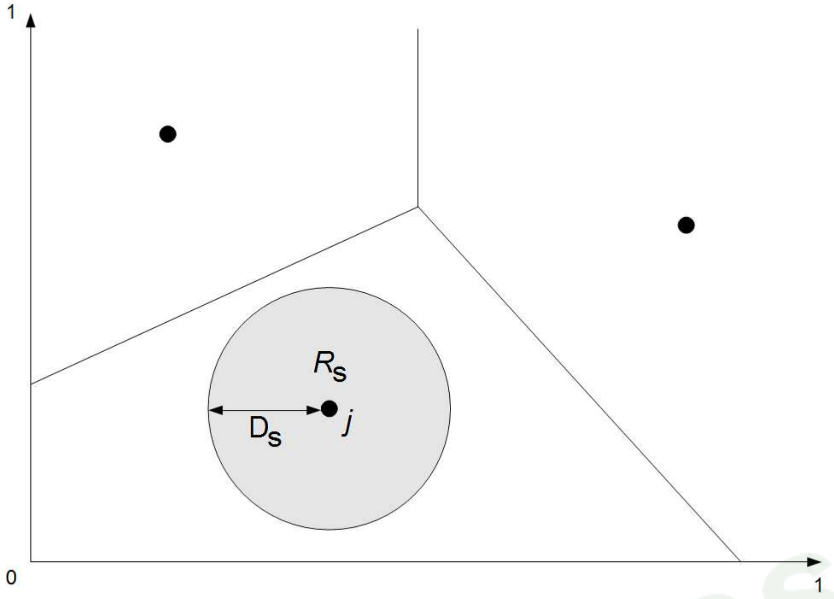


Figure 5. Distance and region R_s (shaded region) defined by the sensitivity threshold training parameter. Training examples that lie within R_s will not cause a neuron to be added.

For the case of $\mathbf{W}_{j,o} = 0$, then A_o will be zero, no matter the activation A_j. Thus, the distance between $\mathbf{I}$ and j is indeterminable from this relation. Instead, P_a can be deduced directly. For an activation $A_o = 0$, the error threshold will trigger the addition of a neuron if the desired output $\mathbf{O}_d$ is greater than the error threshold. Therefore:

$$P_a = \begin{cases} 1, & \mathbf{O}_d > E_{thr} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

This is consistent with the conjecture in Equation 10.

For the case of $0 < A_j \mathbf{W}_{j,o} < 1$, the way in which the error is calculated must be examined. The error E_o across output o between the desired output value $\mathbf{O}_d$ and the actual output value A_o is defined as:

$$E_o = |\mathbf{O}_d - A_o| \tag{13}$$

which can be expanded as:

$$E_o = \begin{cases} \mathbf{O}_d - A_o, & \mathbf{O}_d > A_o \\ A_o - \mathbf{O}_d, & \mathbf{O}_d < A_o \end{cases} \tag{14}$$

Thus, there are two situations where E_o could exceed E_{thr}: when A_o is too low, that is, the example $\mathbf{I}$ is too far from n; or when A_o is too high, that is, $\mathbf{I}$ is too close to n. There are therefore two activations to consider: A_o^{max}, that is too high, and A_o^{min} that is too low. These activations can be determined by substituting E_{thr} for E_o and rearranging Equation 14 to make A_o the dependent variable, as follows:

$$\begin{aligned} A_o^{min} &= \mathbf{O}_d - E_{thr} \\ A_o^{max} &= \mathbf{O}_d + E_{thr} \end{aligned} \tag{15}$$

For the case that $0 < A_j \mathbf{W}_{j,o} < 1$, then the terms of Equation 15 above can be expanded as:

$$\begin{aligned} A_j^{min} \mathbf{W}_{j,o} &= \mathbf{O}_d - E_{thr} \\ A_j^{max} \mathbf{W}_{j,o} &= \mathbf{O}_d + E_{thr} \end{aligned}$$

where A_j^{max} is the maximum activation of winning evolving layer neuron j that will cause the maximum output activation A_o^{max} and A_j^{min} is the minimum activation of j that will cause the minimum output A_o^{min}

These can be rearranged to make A_j^{max} and A_j^{min} the dependent variables, as follows:

$$\begin{aligned} A_j^{max} &= \frac{\mathbf{O}_d + E_{thr}}{\mathbf{W}_{j,o}} \\ A_j^{min} &= \frac{\mathbf{O}_d - E_{thr}}{\mathbf{W}_{j,o}} \end{aligned}$$

Given that Equation 1 describes the activation of j with respect to the distance D, then expanding A_j and rearranging to make D^{min} and D^{max} the dependent variables yields:

$$D^{min} = 1 - \frac{\mathbf{O}_d + E_{thr}}{\mathbf{W}_{j,o}} \tag{16}$$

$$D^{max} = 1 - \frac{\mathbf{O}_d - E_{thr}}{\mathbf{W}_{j,o}} \tag{17}$$

Thus, points that lie at a distance between D^{min} and D^{max} will not cause a neuron to be added. Figure 6 shows this. This figure shows the same three Voronoi regions from Figure 5, where the shaded region is R_e.

While it may seem that the terms $\mathbf{O}_d + E_{thr}$ and $\mathbf{O}_d - E_{thr}$ in Equations 16 and 17 could yield distances of less than zero or greater than one, from Equation 14 it is apparent that, provided that $\mathbf{O}_d$ and E_{thr} are both constrained to the range zero to unity, this cannot happen. From these equations the following constraints can be derived:

$$\begin{aligned} \text{if } D = D^{min} \text{ then } \mathbf{O}_d \leq 1 - E_{thr} \\ \text{if } D = D^{max} \text{ then } \mathbf{O}_d \geq E_{thr} \end{aligned}$$

It is apparent from Equations 16 and 17 that:

$$\begin{aligned} E_{thr} = 0, D^{min} = D^{max} = 1 - \frac{\mathbf{O}_d}{\mathbf{W}_{j,o}} \\ \Rightarrow E_{thr} \rightarrow 0, D_e^{min} \rightarrow D_e^{max} \end{aligned}$$

Since the volume V_e of the region R_e is given by:

$$V_e = f(D_e^{max}) - f(D_e^{min})$$

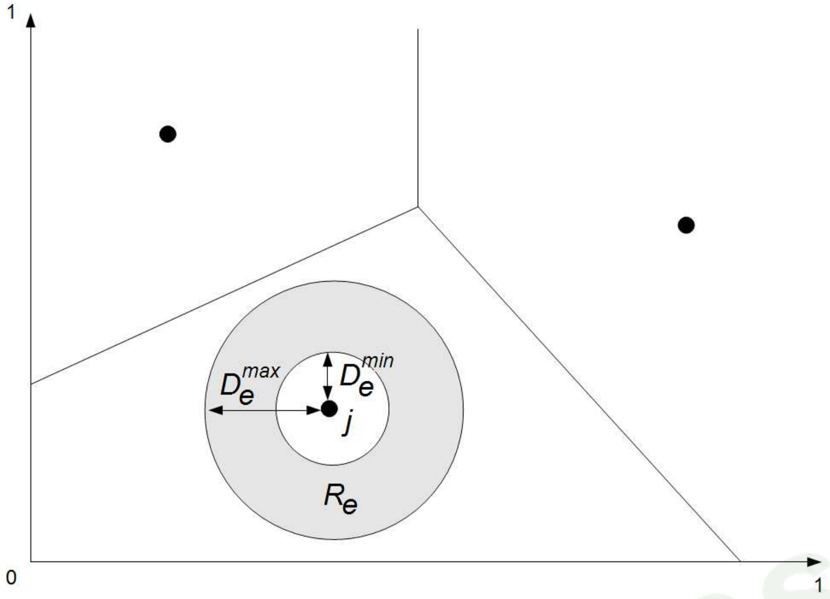


Figure 6. Distances and region R_e defined by error threshold parameter. Training examples that lie within R_e will not cause a neuron to be added.

where f is the volume function, then the following is implied:

$$\begin{aligned} E_{thr} \to 0, D_e^{min} &\to D_e^{max} \\ \Rightarrow E_{thr} \to 0, V_e &\to 0 \\ \Rightarrow E_{thr} \to 0, V_a &\to 0 \\ \Rightarrow E_{thr} \to 0, P_a &\to 1 \end{aligned}$$

In other words, as the error threshold decreases, the probability of a neuron being added increases. This proves the conjecture in Equation 10 for this case.

Additionally, if the distance D_s is less than the minimum distance D_e^{min}, then a neuron will be added for every training example. This is verified by Equations 7 and 8. If the intersection between the two regions is zero, then $P_a = 1$.

The final case is when $A_j w_{j,o} \geq 1$. Note that this requires $\mathbf{W}_{j,o} \geq 1$, although not every output will be unity whenever $\mathbf{W}_{j,o} \geq 1$. In this case, the error will exceed the error threshold only when $\mathbf{O}_d$ is less then $1 - E_{thr}$, thus:

$$P_a = \begin{cases} 1, & \mathbf{O}_d < 1 - E_{thr} \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

The conjecture in Equation 10 again holds true.

The distance D_e can be derived by rearranging $A_j \mathbf{W}_{j,o} \geq 1$ to make A_j the dependent variable and substituting D_e, thus:

$$D_e \leq 1 - \frac{1}{\mathbf{W}_{j,o}} \tag{19}$$

Therefore, the maximum distance that an example can be at for this case is determined solely by the connection weight. If the following is the case

$$D_e > 1 - \frac{1}{\mathbf{W}_{j,o}}$$

then the neuron activation will not saturate and Equations 16 and 17 will apply.

8 Influence of the Learning Rate One Parameter

The learning rate parameters also have an effect upon training, although each parameter has a different effect. This is due to the different mechanisms by which they function. The weight update rule for the input to evolving layer of connections is intended to reduce the difference between the current weight vector and the current input vector. The weight update rule for the evolving to output layer is a variant of the perceptron learning rule, and is based on the idea of reducing errors for the outputs.

Intuitively, the higher the η_1 parameter is, the higher the activation of j will be the next time the current vector **I** is presented to it. The conjecture for this parameter is thus:

$$\begin{aligned} \eta_1 \rightarrow 1, A_j^{t+1} \rightarrow 1 \\ \eta_1 \rightarrow 0, A_j^{t+1} \rightarrow A_j^t \end{aligned} \tag{20}$$

where A_j^{t+1} is the activation of neuron j at time $t + 1$ for input vector **I**. This conjecture can be proven as follows:

The weight update rule for the input to evolving layer weights can be expressed as:

$$\mathbf{W}_{i,j}(t+1) = \mathbf{W}_{i,j}(t) + \Delta\mathbf{W}_{i,j}$$

where:

$$\Delta\mathbf{W}_{i,j} = \eta_1(\mathbf{I}_i - \mathbf{W}_{i,j}(t))$$

This can be viewed as a change in distance between **W** and **I**. The goal is thus to prove that when $\eta_1 = 1$ the change in distance between the two vectors **W** and **I** is such that the distance at time $t + 1$, D_j^{t+1}, is zero.

Given the following:

$$A_j^t = 1 - D_j^t \tag{21}$$

$$A_j^{t+1} = 1 - D_j^{t+1} \tag{22}$$

$$D_j^{t+1} = D_j^t - \Delta D_j^t \tag{23}$$

From the weight update rule above, it can be seen that:

$$\Delta D_j^t = \eta_1 D_j^t \tag{24}$$

Rearranging Equation 21 to make D_j^t the dependent variable, and substituting for D_j^t in Equation 24 yields:

$$\Delta D_j^t = \eta_1(1 - A_j^t) \tag{25}$$

while substituting for ΔD_j^t in Equation 23 gives the following:

$$D_j^{t+1} = D_j^t - \eta_1(1 - A_j^t) \tag{26}$$

Rearranging Equation 22 to make D_j^{t+1} the dependent variable yields:

$$D_j^{t+1} = 1 - A_j^{t+1} \tag{27}$$

Replacing D_j^{t+1} in Equation 26 with Equation 27 yields:

$$1 - A_j^{t+1} = (1 - A_j^t) - \eta_1(1 - A_j^t)$$

Finally, solving for A_j^{t+1} and simplifying yields:

$$A_j^{t+1} = A_j^t + \eta_1(1 - A_j^t) \tag{28}$$

This holds true for any monotonic linear distance measure.

It can be seen that when $\eta_1 = 1$ Equation 28 becomes:

$$\begin{aligned} &A_j^{t+1} = A_j^t + (1 - A_j^t) \\ &\Rightarrow A_j^{t+1} = 1 \end{aligned}$$

When $\eta_1 = 0$, Equation 28 becomes:

$$\begin{aligned} &A_j^{t+1} = A_j^t + 0(1 - A_j^t) \\ &\Rightarrow A_j^{t+1} = A_j^t \end{aligned}$$

This is also true for non-monotonic distance measures, but is not proven. This objection aside, the conjecture in Equation 20 above is proven

From Equation 25 above, it is also possible to calculate the maximum distance that a neuron will move. A neuron moves the maximum distance when its activation is the minimum allowed. The minimum allowed activation is set by the sensitivity threshold parameter, S_{thr}. Thus, by substituting S_{thr} into Equation 25 above, we get:

$$\Delta D^{max} = \eta_1(1 - S_{thr}) \tag{29}$$

Thus, the effect of the learning rate one parameter is determined by the sensitivity threshold parameter.

9 Influence of the Learning Rate Two Parameter

The change to the connection weight $\mathbf{W}_{j,o}$, $\Delta\mathbf{W}_{j,o}$, is determined by the following equation:

$$\Delta\mathbf{W}_{j,o} = \eta_2 E_o A_o \tag{30}$$

where:
η_2 is the learning rate two parameter,
E_o is the absolute error over output neuron o and
A_o is the activation of neuron o

From this equation, a relationship with the error threshold becomes immediately apparent: since the weights will only be updated if a neuron is *not* added to the network, then E_o in Equation 30 above will always be less than the error threshold E_{thr}. Thus, a limit on the value of $\Delta\mathbf{W}_{j,o}$, $\Delta\mathbf{W}^{max}_{j,o}$ can be calculated:

$$\Delta\mathbf{W}^{max}_{j,o} = \eta_2 E_{thr} A_o$$

This shows that the error threshold parameter and the performance of the η_2 parameters are closely coupled together. A high error threshold will reduce the number of neurons added by two different mechanisms: firstly, by reducing the sensitivity of the network to the error over the current example, and secondly, by reducing its sensitivity to error for later examples. Error threshold is thus a very significant training parameter.

The sensitivity threshold is also relevant. The activation of A_o is a function of A_j and $W_{j,o}$, and the minimum activation of j is determined by S_{thr}, the lower limit of $\Delta\mathbf{W}_{j,o}$, $\Delta\mathbf{W}^{min}_{j,o}$ can be calculated:

$$\Delta\mathbf{W}^{min}_{j,o} = \eta_2 E_o w_{j,o} S_{thr}$$

Since ECoS networks are intended to learn for the duration of their existence, it is entirely possible that the weights in this layer of connections will become very large: with an unending (or infinite) stream of training data, it is entirely possible that the weights themselves will approach infinity, especially with a low sensitivity threshold and a high error threshold.

An interesting implication of the unbounded growth of weights can be described using Equations 16, 17 and 19. As the weights continue to grow, the distance values in these equations tend ever closer to unity, that is:

$$\mathbf{W}_{j,o} \to \infty, D_e \to 1$$

Given the following:

$$D_e \to 1, P_a \to 0$$

Since the rate at which the weights grow is directly determined by the η_2 parameter, then it is clear that as the η_2 parameter increases, the probability of adding a neuron later decreases. Over a complete training set, then, an ECoS network will be expected to add fewer neurons during training with a high learning rate. This has been experimentally verified (Section 11). Note that this applies to classification problems only. Function approximation problems behave differently, and are discussed in Section 10.

10 Function Approximation

The analysis of the effect of the sensitivity threshold and error threshold apply to all types of problems, no matter whether they are classification or function approximation problems. To explain the effect of the learning rate parameters for function approximation problems requires a slight reworking of parts of the formalisation.

Assume that an evolving layer neuron activates with a value of unity. The activation of the output neuron o will therefore be equal to the value of the connection weight $\mathbf{W}_{j,o}$. If $\mathbf{W}_{j,o}$ is greater than unity, then the output neuron o will saturate at unity. This will not cause a problem if this occurs for a classification problem, as the output values will be either zero or unity. If a neuron activates at less than unity, and the connection weight outgoing from j is equal to or greater than unity, then the error will still be low. Thus, a neuron that is not close to the current example can still correctly classify it, if its outgoing weight is sufficiently large. This means that for classification problems, a smaller number of neurons is possible, especially if E_{thr} and S_{thr} are low.

For function approximation problems, it is not possible for the evolving to output layer connection weights to exceed unity. This is because these weights represent the desired output values. Thus, an error that is less than the error threshold, will cause the weight to move towards the desired output value. This means that each outgoing connection weight has a region around it on the number line, where the bounds are defined by the error threshold. The neuron then represents a cluster of examples, with similar output values. Any input vector that causes that particular evolving layer neuron to activate, and that has a desired output value within the range $\mathbf{W}_{j,o} \pm E_{thr}$ will not cause a neuron to be added.

A high η_2, however, will cause the output values to move away from the centre of the cluster of output values. If the data is self-consistent, that is, the values are periodic, then additional neurons will be added, driven by the error threshold parameter. However, if η_2 is too low, then the neuron will not be able to find the centre of the cluster of output values. This will also cause more neurons to be added.

11 Experiments

The formalisations above made several predictions about the effect of the training parameters, insofar as they relate to the addition of neurons to the ECoS evolving layer. Firstly, that as the sensitivity threshold increases, the number of neurons added increases; secondly, that as the error threshold decreases, the number of neurons added increases; thirdly, for classification problems, as the learning rate two parameter increases, the number of neurons added decreases, while for function approximation problems, as the learning rate two parameter increases, the number of neurons will first decrease, then increase. The ways in which these three parameters behave has been discussed from a theoretical viewpoint (Sections 6, 7, 9 and 10). The purpose of the work reported in this section was to experimentally investigate these predictions.

Two well-known benchmark data sets were used in these experiments. The first was Fisher's iris classification data set [5]. The second was the Gas Furnace function approximation problem [2]. There were 150 examples in the iris data set, and 289 examples in the gas furnace set.

Two techniques were used for investigating the effects of the parameters. The effects of the sensitivity threshold and error threshold were investigated using Latin hypercube sampling [15]. Parameters were randomly drawn from one thousand partitions, and a network (either EFuNN or SECoS) trained over the test data set. The number of neurons in the evolving layer was then determined. Since the effects of error threshold and sensitivity threshold tended to overwhelm the effects of the η_2 parameter, a slightly different method was used to investigate the parameter. For these experiments, the values of the sensitivity threshold, error threshold and η_1 were held constant at 0.5, 0.1 and 0.5, respectively. Only the η_2 parameter was randomly sampled across the range $[0, 1]$. Again, one thousand random samples were drawn.

The results are presented and discussed in the following subsections.

11.1 Sensitivity Threshold

The analysis in Section 6 predicted that the number of neurons added would increase as the sensitivity threshold approached unity. The results presented here show that this is the case for both SECoS and EfuNN across both data sets.

The results across the iris data set are presented in Figure 7. Here, there is a very strong trend apparent for both SECoS and EFuNN, with a minimal rise in the number of evolving layer neurons between zero and 0.9, and a sharp rise after 0.9, quickly approaching 150 neurons in total, or one for each example in the data set. There are a relatively small number of samples where the number of neurons is very much greater than most for that parameter value, that is, there are a number of points well above the rest. These are from runs where the error threshold was randomly set to such a low value that the effects overwhelmed the effects of the sensitivity threshold. The number of neurons added to the EFuNN tended to be slightly larger than the number added to the SECoS.

The predictions were again verified over the gas furnace data, with the upward swing in numbers occurring at almost the same point as the iris data set. Again, values of the sensitivity threshold near unity caused a neuron to be added for every example. The majority of the samples were lower in the plot than for the iris data set, with more of the samples tightly grouped together near the x-axis. That is, there was much less variation in the size of the evolving layer than there was for the iris data set. Again, some samples were much higher than others about that parameter value, which was again due to the overwhelming effect of the error threshold for that sample. The differences in size between SECoS and EFuNN are more apparent here than with the iris data: once again, EFuNN had more neurons added during training.

11.2 Error Threshold

The analysis in Section 7 predicted that the number of neurons added to an ECoS network would increase as the error threshold approached zero. The results in this subsection show that this was the case for both SECoS and EFuNN across both data sets.

The results for the iris classification benchmark are presented in Figure 9. Near zero, a neuron was added for almost every example, and the numbers added dropped rapidly as the error threshold approached 0.1, then decreased much more slowly as the threshold

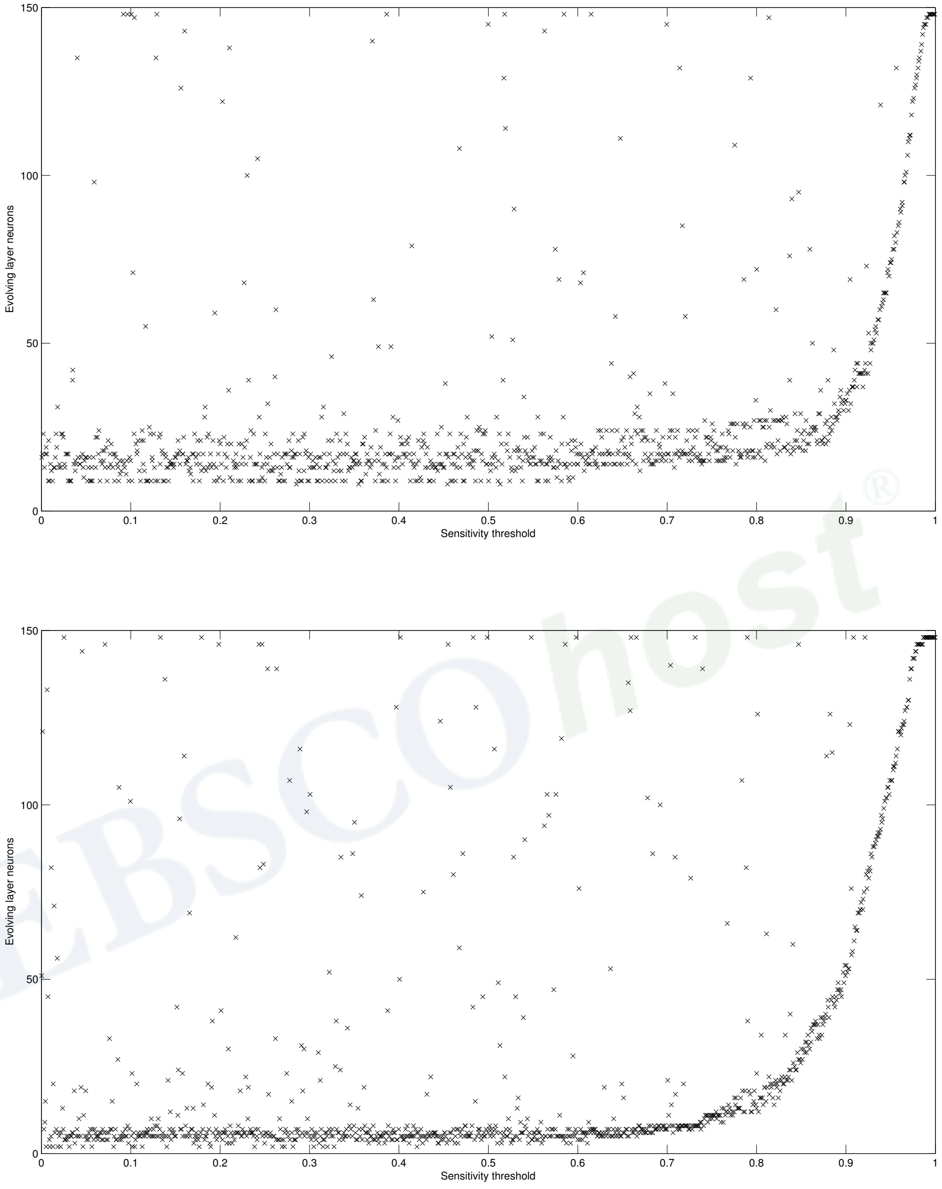


Figure 7. Number of evolving layer neurons versus sensitivity threshold parameter for SECoS (top) and EFuNN (bottom) networks trained on the iris classification data set.

approached unity. As before, there were a relatively small number of samples where the randomly-set value of the sensitivity threshold was large enough to overwhelm the effect of the error threshold. Again, EFuNN were slightly larger than SECoS.

The results across the gas furnace set are presented in Figure 10, where the networks again behaved as predicted. The curves appear to be much smoother than those for the

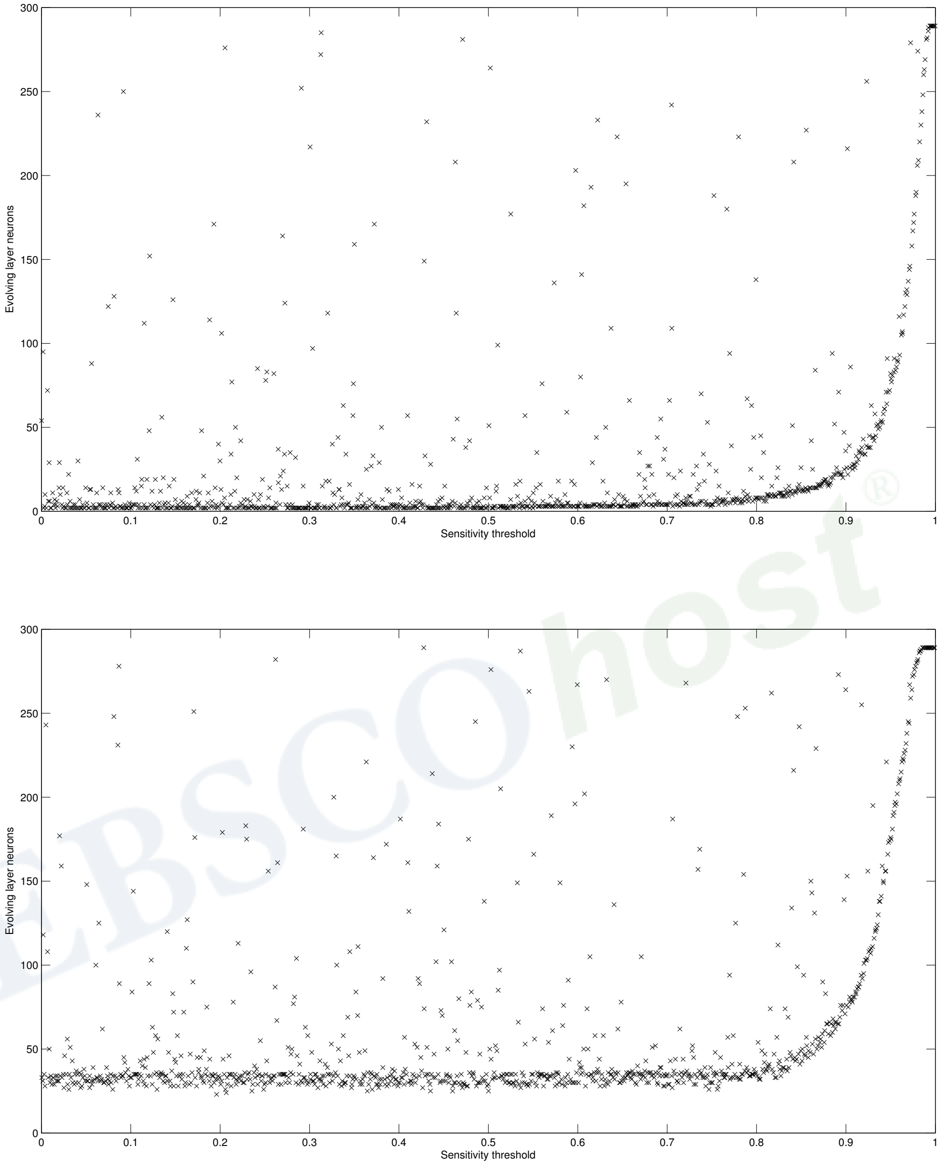


Figure 8. Number of evolving layer neurons versus sensitivity threshold parameter for SECoS (top) and EFuNN (bottom) networks trained on the gas furnace data set.

iris classification problem. The plateau in the curve occurred slightly later than for iris classification, at an error threshold of approximately 0.2, as opposed to 0.1 for iris. As before, the number of evolving layer neurons were on the whole greater for EFuNN than for SECoS.

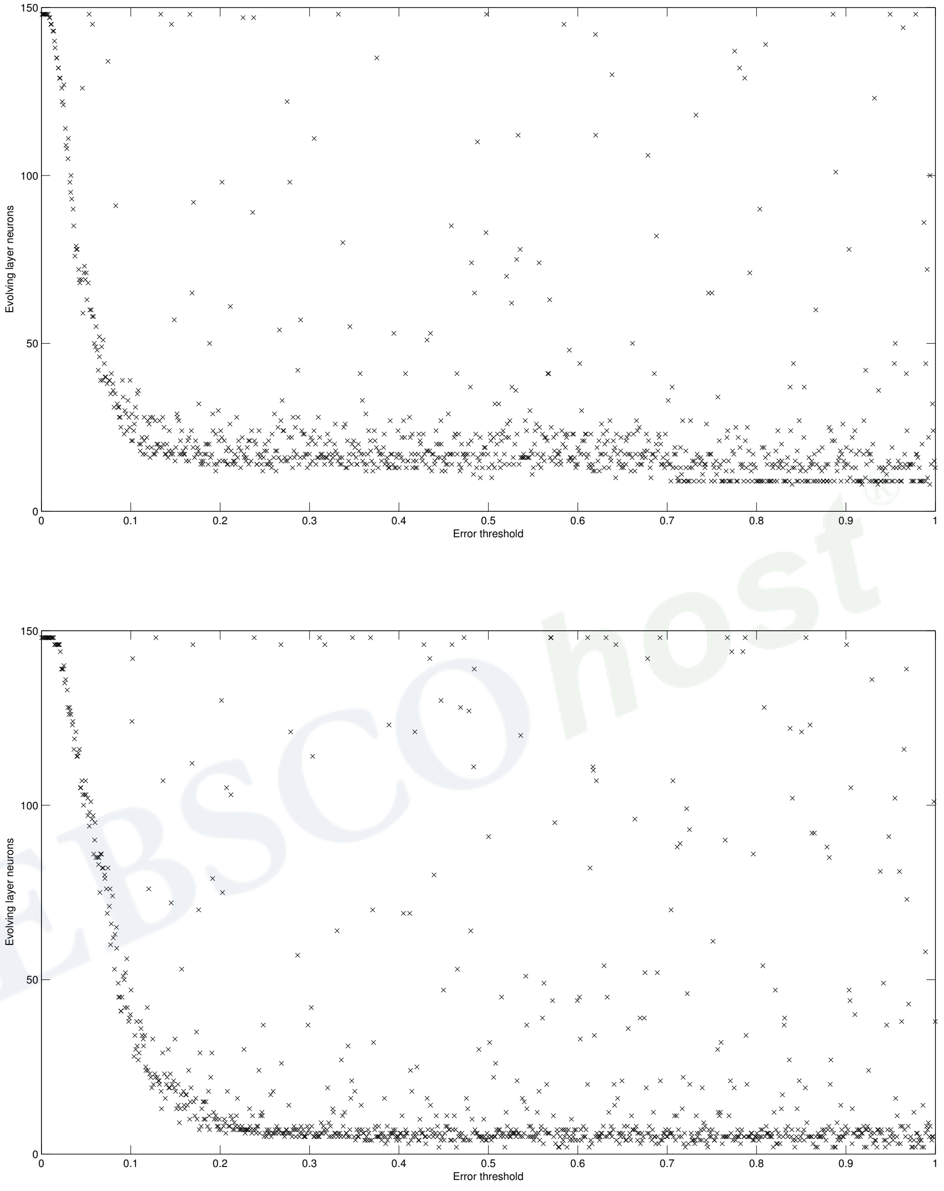


Figure 9. Number of evolving layer neurons versus error threshold parameter for SECoS (top) and EFuNN (bottom) networks trained on the iris classification data set.

11.3 Learning Rate Two

Two predictions were made about the effect of the learning rate two parameter. The first was that for classification problems, the number of neurons added will decrease as the learning rate two parameter increases. For function approximation problems, the number of neurons

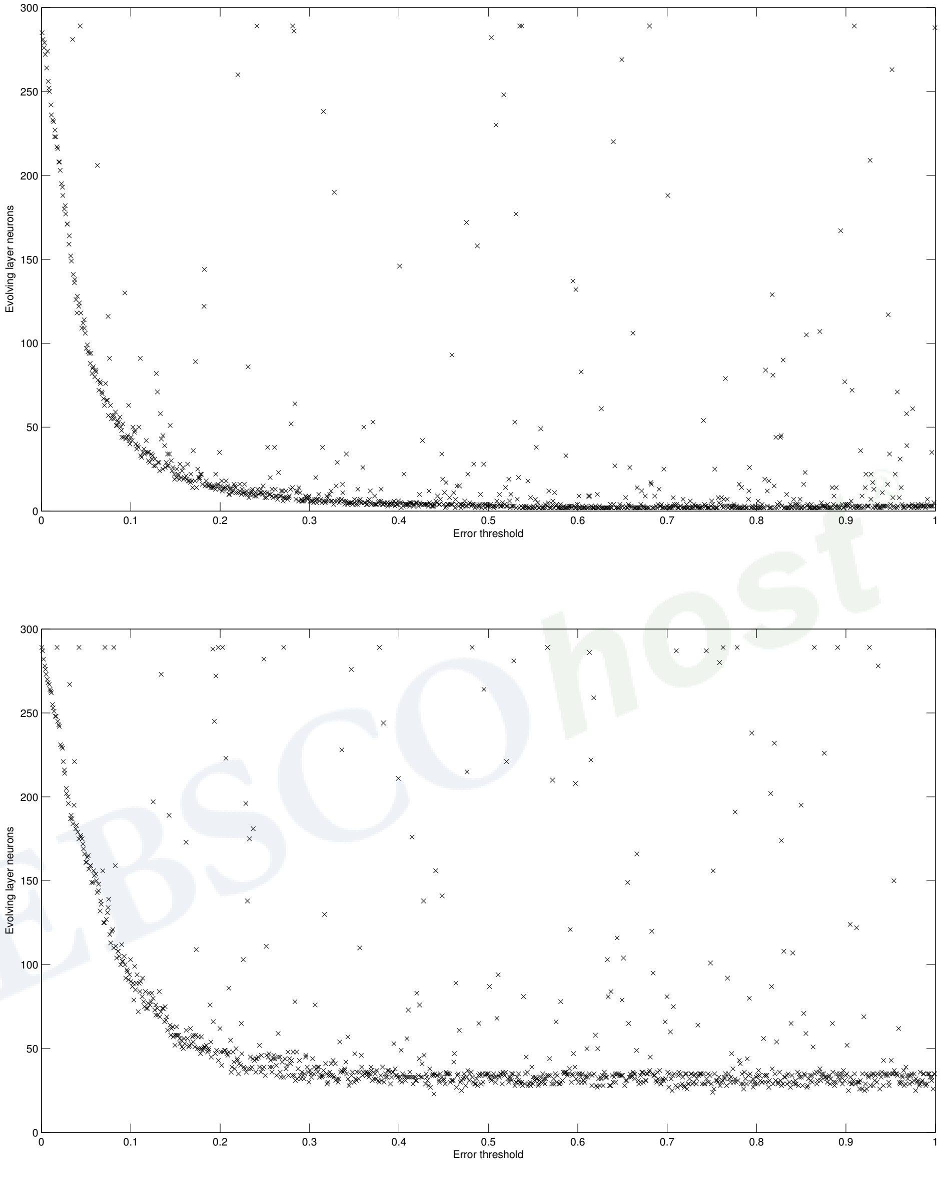


Figure 10. Number of evolving layer neurons versus error threshold parameter for SECoS (top) and EFuNN (bottom) networks trained on the Gas Furnace data set.

will decrease up until a certain point, then start to increase again.

The results for the iris classification data set are presented in Figure 11. As predicted, there is a downward trend to the curve, as the number of neurons decreased with the increasing learning rate two parameter. The decrease for SECoS ended at a lower error threshold than for EFuNN, at approximately 0.4 as opposed to 0.5.

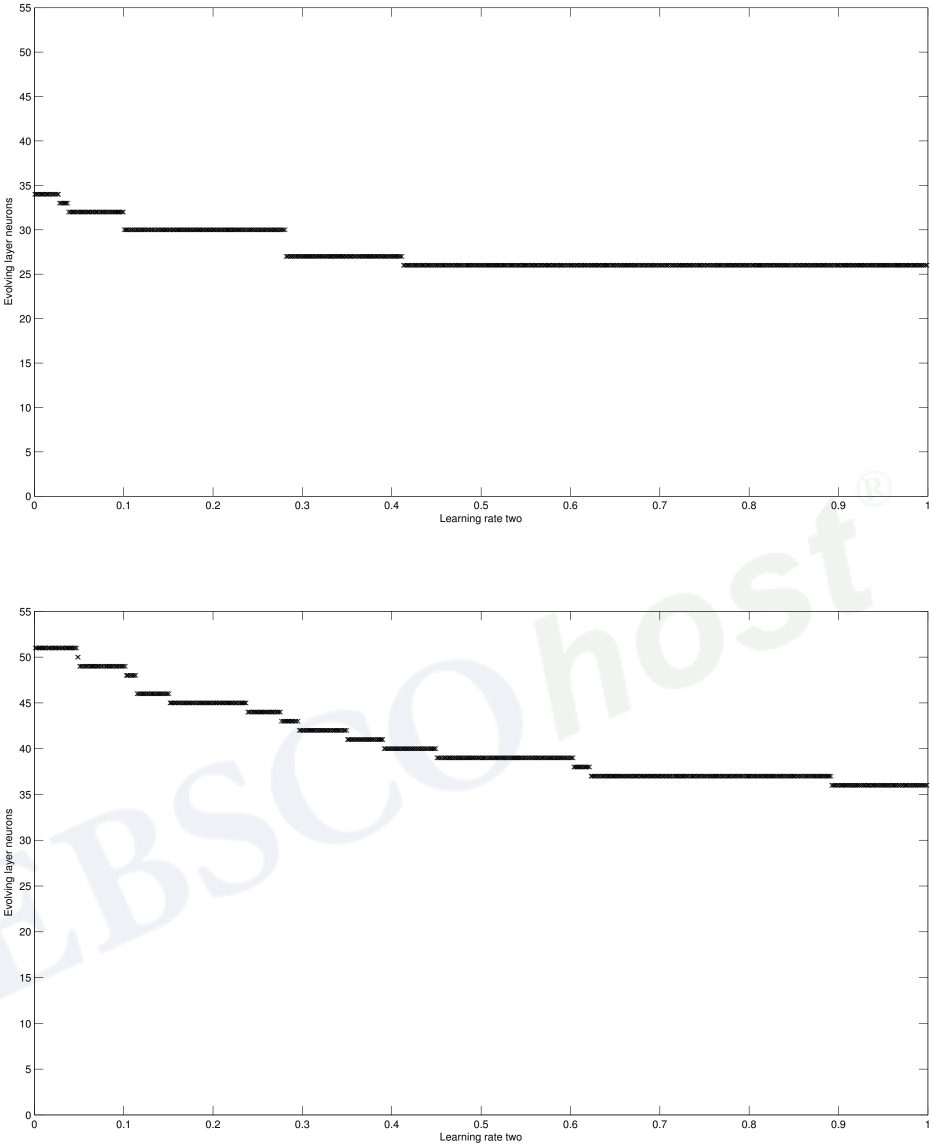


Figure 11. Number of evolving layer neurons versus learning rate two parameter for SECoS (top) and EFuNN (bottom) networks trained on the iris classification data set.

The results for the gas furnace data set are presented in Figure 12. For SECoS, a clear decrease in the number of neurons is visible from zero to 0.4, then an increase from 0.6 to unity. Although the corresponding plot for EFuNN is much less smooth than for SECoS, a definite decrease in the number of neurons is visible between 0.1 and 0.5, followed by an increase from 0.6 to unity.

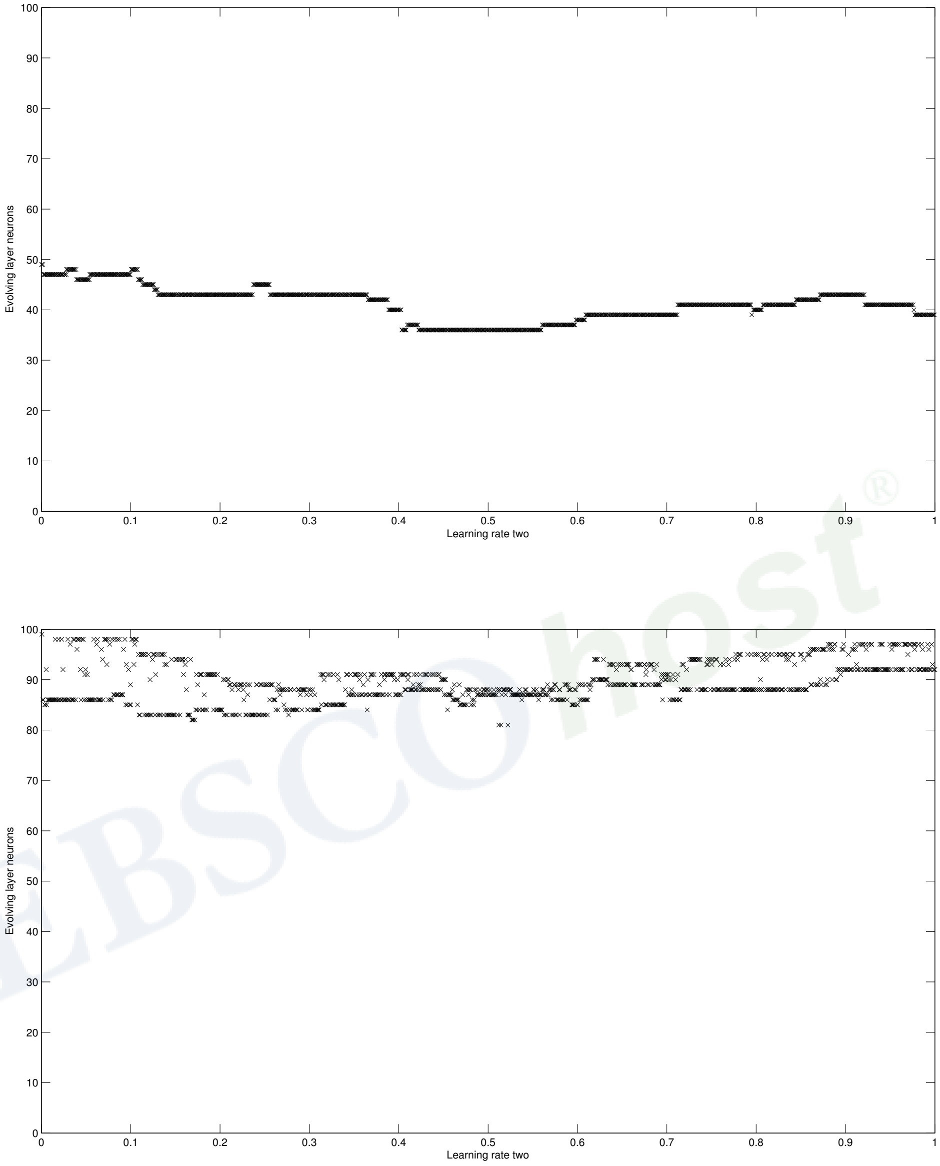


Figure 12. Number of evolving layer neurons versus learning rate two parameter for SECoS (top) and EFuNN (bottom) networks trained on the gas furnace data set.

12 Discussion

The analysis in Section 6 predicts that as the sensitivity threshold approaches unity, so does the probability of adding a neuron to the network. The results in section 11 show that this was the case across both of the benchmark data sets experimented with. The results all have a great deal of similarity about them, that is, the curves are highly similar to one another, despite the great differences in the data sets.

The points at which the curves start to increase were quite similar for both benchmark data sets. This indicates that the point at which the sensitivity threshold dominates (that is, the effect of the sensitivity threshold overcomes the effect of the other parameters) is in the region of $0.85 - 0.95$. Below that value, the curves were quite flat, which implies that the other parameters such as error threshold are responsible for adding most of the neurons.

The difference in size between SECoS and EFuNN was apparent throughout. EFuNN, as predicted, required more neurons than a SECoS to model the same data set.

The predictions made in Section 7 have been validated by these results. The prediction made was that as the error threshold approaches unity, the probability of adding a neuron to the network decreases. For both data sets, as the error threshold increased, the number of neurons added decreased. One difference observed between the classification and function approximation problems was that the decrease in network size was much smoother for the gas furnace function approximation data set. This was due to the greater range of error values that are possible during the training of function approximation networks, as discussed in Section 10.

The predictions made in Sections 9 and 10 have been validated by these results. For classification problems, the number of neurons added during training trend downwards as the learning rate two parameter increases. For function approximation problems, the number of neurons trend downwards to a point, then trend upwards again.

13 Conclusion

This chapter has presented a formalisation of the Evolving Connectionist System (ECoS) constructive ANN. The formalisation describes the internal workings of an ECoS network as a system of Voronoi regions in the input space that map to regions in output space. The formalisation also described the effect of each of the training parameters, that is, it predicted how an ECoS network will behave in relation to the training parameter settings. The formalisation has shown that the parameters are related to one another, that is, one parameter will affect another. This has implications for algorithms that attempt to optimise the training of ECoS.

The formalisation was investigated experimentally across two different data sets, and found to be supported.

It is possible that this formalisation could be used to develop methods of automatically optimising ECoS training parameters as training is underway. This will be the focus of future research.

References

[1] E. Alpaydin. GAL: Networks that grow when they learn and shrink when they forget. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):391–414, 1994.

[2] G.E.P. Box and G.M. Jenkins. *Time Series Analysis forecasting and control*. Holden-Day, 1970.

[3] G. Cybenko. Approximation by superpositions of sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[4] S.E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 524–532. Morgan Kaufman Publishers, 1990.

[5] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[6] B. Fritzke. Kohonen feature maps and growing cell structures - a performance comparison. In C.L. Giles, S.J. Hanson, and J.D. Cowan, editors, *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, 1993.

[7] N.K. Kasabov. The ECOS framework and the ECO learning method for evolving connectionist systems. *Journal of Advanced Computational Intelligence*, 2(6):195–202, 1998.

[8] N.K. Kasabov. Evolving fuzzy neural networks - algorithms, applications and biological motivation. In Takeshi Yamakawa and Gen Matsumoto, editors, *Methodologies for the Conception, Design and Application of Soft Computing*, volume 1, pages 271–274. World Scientific, 1998.

[9] N.K. Kasabov. Evolving connectionist systems: A theory and a case study on adaptive speech recognition. In *International Joint Conference on Neural Networks (IJCNN), July 10-16*, 1999.

[10] N.K. Kasabov. *Evolving Connectionist Systems: Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines*. Springer, 2003.

[11] N.K. Kasabov and B.J. Woodford. Rule insertion and rule extraction from evolving fuzzy neural networks: Algorithms and applications for building adaptive, intelligent expert systems. In *IEEE International Fuzzy Systems Conference*, pages 1406–1411, 1999.

[12] A.N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk. USSR*, 114:953–956, 1957. (in Russian).

[13] B. Kosko. *Neural Networks and Expert Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[14] K.J. Lang and M.J. Witbrock. Learning to tell two spirals apart. In David Touretzky, Geoffrey Hinton, and Terrence Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 52–57, June 17-26 1988.

[15] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, May 1979.

[16] M.L. Minsky and S.A. Papert. *Perceptrons*. MIT Press, 1969.

[17] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley and Sons, Ltd, 1992.

[18] J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.

[19] M.J. Watts. Evolving connectionist systems for biochemical applications. In N. Kasabov and K. Ko, editors, *Emerging Knowledge Engineering and Connectionist-based Systems (Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop "Future directions for intelligent systems and information sciences", Dunedin, 22-23 November 1999)*, pages 147–151. University of Otago Press, 1999.

[20] M.J. Watts. An investigation of the properties of evolving fuzzy neural networks. In *Proceedings of ICONIP'99, November 1999, Perth, Australia*, pages 217–221, 1999.

[21] M.J. Watts. A decade of Kasabov's evolving connectionist systems: A review. *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, 39(3):253–269, 2009.

[22] M.J. Watts and N.K. Kasabov. Simple evolving connectionist systems and experiments on isolated phoneme recognition. In *Proceedings of the first IEEE conference on evolutionary computation and neural networks, San Antonio, May 2000*, pages 232–239. IEEE Press, 2000.

[23] B.J. Woodford and N.K. Kasabov. Ensembles of EFuNNs: An architecture for a multi module classifier. In *The proceedings of FUZZ-IEEE'2001. The 10th IEEE International Conference on Fuzzy Systems, December 2-5 2001, Melbourne, Australia*, pages 441–445, 2001.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 17

LEARNING WITH HETEROGENEOUS NEURAL NETWORKS

Lluís A. Belanche-Muñoz[*]
Department of Software, Technical University of Catalonia
Jordi Girona, 1-3. 08034 Barcelona, Spain

Abstract

This chapter studies a class of neuron models that computes a user-defined *similarity function* between inputs and weights. The neuron transfer function is formed by composition of an adapted logistic function with the quasi-linear mean of the partial input-weight similarities. The neuron model is capable of dealing directly with mixtures of continuous as well as discrete quantities, among other data types and there is provision for missing values. An artificial neural network using these neuron models is trained using a *breeder genetic algorithm* until convergence. A number of experiments are carried out in several real-world problems in very different application domains described by mixtures of variales of distinct types and eventually showing missing values. This heterogeneous network is compared to a standard radial basis function network and to a multi-layer perceptron and shown to learn with superior generalization ability at a comparable computational cost. A further important advantage is the greatly enhanced interpretability of the resulting neural solutions, because the networks compute a weighted similarity to a set of prototypes.

Keywords: Artificial neural networks; similarity measures; data heterogeneity; missing values; evolutionary algorithms.

1. Introduction

In many important domains from the real world, objects are described by a mixture of continuous and discrete variables, usually containing missing information and characterized by an underlying uncertainty or imprecision. For example, in the well-known UCI repository [3] over half of the problems contain explicitly declared nominal attributes, let alone other

[*] E-mail address: belanche@lsi.upc.edu

data types, usually unreported. In the case of artificial neural networks (ANN), this *heterogeneous* information has to be encoded in the form of real-valued quantities, although in many cases there is enough domain knowledge to characterize the nature of the variables. The strong points of ANNs are their appealing capacity to learn from examples, their distributed computation —which helps them tolerate partial failures to a certain extent— and the possibility, so often exploited, to use them as black-box models. This last characteristic is paradoxically one of the major weaknesses, given that in practice this autonomy of functioning involves no transfer of knowledge from or to the designer. With the exception of very specific architectures, the networks are forced to learn from scratch most of the times. The network *works* (in the sense that solves a problem to a certain satisfaction), but the weights convey information as high-dimensional real vectors whose meaning about the solution can be intricate.

In this chapter a general framework is presented for dealing with data heterogeneity in ANNs under the conceptual cover of similarity, in which a class of neuron models accepting *heterogeneous inputs* and *heterogeneous weights* computes a user-defined *similarity function* between these inputs and weights. The similarity function is defined by composition of a specific power (quasi-linear) mean of the partial input-weight similarities plus a logistic function. The resulting neuron model then accepts mixtures of continuous and discrete quantities, with explicit provision for missing information. Other data types are possible by direct extension of the model.

An artificial neural network using these neuron models is trained using a *breeder genetic algorithm* until convergence. A number of experiments are carried out that illustrate the validity of the approach, using several benchmarking problems (classification and regression), selected as representatives because of the diversity in the richness and kind of data heterogeneity. The network is compared to a standard radial basis function neural network (RBF) and to a multi-layer perceptron (MLP), and shown to learn from non-trivial data sets with a generalization ability that is superior in most cases to both networks. A further advantage of the approach is the interpretability of the learned weights.

The chapter is organized as follows. The second section motivates the basis of the approach and develops some widespread ways of coping with data heterogeneity in ANNs; the third section introduces material on similarity measures; sections four and five build the heterogeneous neural network and develop a practical training algorithm, while the sixth section proposes specific similarity measures. The last two sections present experimental work and review relevant related literature, respectively.

2. Background

The action of a feed-forward ANN may be viewed as a mapping through which points in the input space are transformed into corresponding points in an output space. A significant part of the task is devoted to find structure in the data and transform it to a new *hidden* space (or space spanned by the hidden units) in such a way that the problem becomes easier (almost linearly separable in the case of the output layer). The internal workings of a neuron are thus obscure, because the weights have not been set to shape a previously defined (and considered adequate) similarity measure, but rather to adapt a general *physical measure* (like scalar product or Euclidean distance) to the problem at hand. This processing is repeated for all the subsequent hidden layers. It seems clear that better transformations will require less units and therefore a hidden space of less dimensions. This in turn leads to simpler mappings which are less likely to overfit.

The integration of heterogeneous data sources in information processing systems has been advocated elsewhere [14]. In this sense, a shortcoming of the existent neuron models is the difficulty of adding prior knowledge to the model. Current practice assumes that an input may be faithfully represented as a point in $\mathbb{R}^n$, and the geometry of this space is meant to capture the meaningful relations in input space. There is no particular reason why this should be the case. In practice the network has to discover the relations in the structure induced by the chosen coding scheme and find ways to *accommodate* the underlying similarity relationship (inherent in the training examples) to a fixed similarity computation. During the learning process, patterns seen as physically similar may have to be told apart and vice versa. In consequence, several layers may be needed for complex transformations, or a large amount of neurons per layer if we restrict the number of hidden layers to one or two, as is common proceeding. An increase in neurons leads to a corresponding growth in the number of free parameters, and these are less likely to be properly constrained by a limited size data set [13].

Knowledge representation influences the practical success of a learning task [15], but has received little attention in the ANN community. The form in which data are presented to the network is of primary relevance. In essence, the task of the hidden layer(s) is to find a new, more convenient representation for the problem *given* the data representation chosen, a crucial factor for a successful learning process that can have a great impact on generalization ability [13]. As Anderson and Rosenfeld put it, "*prior information and invariances should be built into the design of a ANN, thereby simplifying the network design by not having to learn them*" [16]. Further, the activity of the units should have a well defined meaning in terms of the input patterns [17]. Without the aim of being exhaustive, commonly used methods (see, e.g. [12, 13]) are the following:

Ordinal variables correspond to discrete and usually finite sets of values wherein an ordering has been defined. They are more than often treated as real-valued, mapped equidistantly on an arbitrary real interval. This imposes a continuum where there is not (e.g., stating that there are infinite possibilities between having two or three children). Further, since these variables need not represent numerical entities, this coding is assuring that "Wednesday" is three times "Monday", or that "D" plus "E" is "J".

A second possibility is to code them using a *thermometer*: being k the number of ordered values, k new *binary* inputs are then created. To represent value i, for $1 \leq i \leq k$, the leftmost $1, \ldots, i$ units are on, and the remaining $i+1, \ldots, k$ off. If the number of possibilities to be represented is not small the increase in variables may be simply not affordable. Moreover, the "order" imposed by this coding is only apparent: an arbitrary permutation of the inputs and training data columns will result in an identical training set for the network.

Nominal variables are coded using a 1-out-of-k representation, being k the number of values. This introduces the rows of the $I_{k\times k}$ identity matrix in the training set, leading to an *structured* increase of model input variables, which translates in a notable increment in the number of parameters. Part of the network task is to discover that all these inputs are strongly related and, as a matter of fact, that they represent a single one.

Missing information is difficult to handle, specially when the lost parts are of significant size. Simply discarding the involved data can not be considered as a "method" and is also frustrating because of the lost effort in collecting the information. Alternatively, the missing values can be "filled in" with the mean, median, nearest neighbour (ignoring or underestimating the covariance in the data) or by adding another input equal to one only if the value is absent. Statistical approaches need to model the input distribution itself [13], or are computationally intensive [18].

Vagueness and uncertainty are considerations usually put aside. However, many variables in learning processes are likely to bear some form of uncertainty. For example, consider the value `fairly tall` for the variable *height*. Fuzzy Systems are comfortable with this, but for ANNs this is real trouble.

These encodings being intuitive, their precise effect on network performance is not clear, because of the change in input distribution, the increase (sometimes acute) in input dimension and other subtler mathematical effects derived from imposing an order or a continuum where there was none. This issue can dramatically affect generalization for the worse, due to the curse of dimensionality. In this scenario, the use of hidden units which can adequately capture the desired mapping and do not add input dimensions is thus of great practical importance.

3. Similarity Measures

Let us represent patterns belonging to a space $X \neq \emptyset$ as a vector $\mathbf{x}_i$ of n components, where each component x_{ik} represents the value of a particular feature k. A *similarity measure* is a unique number expressing how "like" two patterns are, given these features. A similarity measure may be defined as a function $s : X \times X \longrightarrow I_s \subset \mathbb{R}$. Assume that s is upper bounded, surjective and total. This implies that I_s is upper bounded and also that $\sup I_s$ exists. Define $s_{max} \equiv \sup I_s$ and define $s_{min} \equiv \inf I_s$ if it exists. Let us denote by s_{ij} the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, that is, $s_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$. A similarity measure may be required to satisfy the following axioms, for any $\mathbf{x}_i, \mathbf{x}_j \in X$:

Axiom S1 (Reflexivity) $s(\mathbf{x}_i, \mathbf{x}_i) = s_{max}$. This implies $\sup I_s \in I_s$.

Axiom S2 (Consistency) $s(\mathbf{x}_i, \mathbf{x}_j) = s_{max} \Longrightarrow \mathbf{x}_i = \mathbf{x}_j$.

Axiom S3 (Symmetry) $s(\mathbf{x}_j, \mathbf{x}_i) = s(\mathbf{x}_i, \mathbf{x}_j)$.

Axiom S4 (Boundedness) s is *lower bounded* when $\exists a \in \mathbb{R}$ such that $s(\mathbf{x}_i, \mathbf{x}_j) \geq a$, for all $\mathbf{x}_i, \mathbf{x}_j \in X$. This is equivalent to ask that $\inf I_s$ exists.

Axiom S5 (Closedness) a lower bounded function s is closed if there exist $\mathbf{x}_i, \mathbf{x}_j \in X$ such that $s(\mathbf{x}_i, \mathbf{x}_j) = s_{min}$. This is equivalent to ask that $\inf I_s \in I_s$.

The similarity axioms enumerated above ensure a consistent definition of such functions, but should be taken as *desiderata*. Some similarity relations may fulfill part or all of them [4]. Other properties (like transitivity) may be of great interest in some contexts, but are not relevant for this work.

The only requirement expressed so far about the set X is the existence of an equivalence relation. However, elements in this set can be simple or complex (i.e. composed by one or more variables). For atomic elements the similarity can be trivially computed, but for complex elements, we need to define a way to combine the *partial* similarities s_k for each variable k to get a meaningful value $s_{ijk} = s_k(x_{ik}, x_{jk})$. This combination has an important semantic role and it is not a trivial choice. We use in this work the concept of an *A-average* [5], as follows. Let $[a, b]$ be a non-empty real interval. Call $A(x_1, \ldots, x_n)$ the A-average of $x_1, ..., x_n \in [a, b]$ to every n-place real function A fulfilling:

Axiom A1. A is continuous, symmetric and strictly increasing in each x_i.

Axiom A2. $A(x, \ldots, x) = x$.

Axiom A3. For any positive integer $k \leq n$:

$$A(x_1, \ldots, x_n) = A(\underbrace{y_k, \ldots, y_k}_{k \text{ times}}, x_{i_{k+1}}, \ldots, x_{i_n})$$

where $y_k = A(x_{i_1}, , \ldots, x_{i_k})$ and $(i_1, ..., i_n)$ is a permutation of $(1, \ldots, n)$.

The means defined by these axioms fulfill Cauchy's property of means, namely, that $\min x_i \leq A(x_1, \ldots, x_n) \leq \max x_i$ with equality only if $x_1 = x_2 = \ldots = x_n$. The proof is straightforward using axioms A1 and A2. This result is even stronger than idempotency; thus, the previous bounds actually impose idempotency. A further interesting property of these means is that we can add averaged elements to an A-average without changing the overall result. Formally, $A(x_1, ..., x_n) = A(z_1, ..., z_m, x_1, ..., x_n)$ if and only if $A(z_1, ..., z_m) = A(x_1, ..., x_n)$. As a consequence, if $y = A(x_1, ..., x_n)$, then $A(x_1, ..., x_n, A(x_1, ..., x_n)) = y$.

Theorem. Let $f : [a, b] \longrightarrow \mathbb{R}$ be a continuous, strictly monotone mapping. Let g be the inverse function of f. Then,

$$A(x_1, ..., x_n) \equiv g\left(\frac{1}{n}\sum_{i=1}^{n} f(x_i)\right)$$

is a well-defined A-average for all $n \in \mathbb{N}$ and $x_i \in [a, b]$ [5]. An important class of A-averages is formed by choosing $f(z) = z^q, q \in \mathbb{R}$, to obtain:

$$M_q(x_1, ..., x_n) = \begin{cases} \left(\frac{1}{n}\sum_{i=1}^{n}(x_i)^q\right)^{\frac{1}{q}} & \text{if } q \neq 0 \\ \max_{i=1,\cdots,n} x_i & \text{if } q = \infty \\ \min_{i=1,\cdots,n} x_i & \text{if } q = -\infty \\ \sqrt[n]{\prod_{i=1}^{n} x_i} & \text{if } q = 0 \end{cases} \tag{1}$$

Several well-known means are found by choosing particular values of q. Specifically, the *arithmetic mean* for $q = 1$, the *geometric mean* for $q = 0$, the *root-square* or *quadratic mean* for $q = 2$ and the *harmonic mean* for $q = -1$. A property of the means M_q is that

$M_q(x_1,...,x_n) \geq M_{q'}(x_1,...,x_n)$ if and only if $q > q'$, with equality only if $x_1 = x_2 = \ldots = x_n$. We shall refer to these as the *power* or *quasi-linear* means.

It can be seen that larger x_i receive a larger weight in the overall outcome as q increases and smaller weight as q decreases. This behaviour can be used to assign a desired *semantics* to the aggregation. In particular, if all the x_i are meaningful independent dimensions, then the natural choice should be $q = 1$. For instance, a difference of 1 in one variable and of 3 in another one will be the same as a difference of two in each. In symbols, $M_1(x_1,...,x_i + 1, x_{i+1} + 3, ..., x_n) = M_1(x_1,...,x_i + 2, x_{i+1} + 2, ..., x_n)$. On the other hand, if we would like to penalise the fact that $x_1,...,x_n$ are very *dissimilar* values with respect to the case $x_1 = x_2 ... = x_n$, then a better choice would be $q = -1$.

4. Heterogeneous Neural Networks

4.1. The Heterogeneous Neuron Model

We develop in this section neuron models allowing for heterogeneous and imprecise inputs, defined as a mapping $h : \hat{\mathcal{H}}^n \to \mathcal{R}_{out} \subseteq \mathbb{R}$. Here $\mathbb{R}$ denotes the reals and $\hat{\mathcal{H}}^n$ is a cartesian product of an arbitrary number n of *source sets* $\hat{\mathcal{H}}^{(k)}, k = 1\ldots, n$. These source sets may include extended reals $\hat{\mathcal{R}}_k = \mathbb{R}_k \cup \{\mathcal{X}\}$ (where each $\mathbb{R}_k$ is a real interval), extended families of fuzzy sets $\hat{\mathcal{F}}_k = \mathcal{F}_k \cup \{\mathcal{X}\}$, and extended finite sets of the form $\hat{\mathcal{O}}_k = \mathcal{O}_k \cup \{\mathcal{X}\}$, $\hat{\mathcal{M}}_k = \mathcal{M}_k \cup \{\mathcal{X}\}$, where the $\mathcal{O}_k$ have a full order relation, while the $\mathcal{M}_k$ have not. The special symbol $\mathcal{X}$ extends the source sets and denotes the unknown element (missing information), behaving as an *incomparable* element w.r.t. any ordering relation. According to this definition, neuron inputs are vectors composed of n elements among which there might be reals, fuzzy sets, ordinals, categorical and missing values.

Consider now a function $s : \hat{\mathcal{H}}^n \times \hat{\mathcal{H}}^n \to I_s$ a similarity function in $\hat{\mathcal{H}}^n$, where we take $I_s \subseteq [0, 1]$ for simplicity (recall that s is surjective). This function is formed by combination of n partial similarities $s_k : \hat{\mathcal{H}}^{(k)} \times \hat{\mathcal{H}}^{(k)} \to I_s$. The s_k measures are normalized to a common real interval ($[0, 1]$ in this case) and computed according to different formulas for different variables (possibly but not necessaryly determined by variable type alone). The desired *neuron model* can be devised that is both similarity-based and handles data heterogeneity and missing values, as follows. Let $\Gamma_i(\mathbf{x})$ the function computed by the i-th neuron, where $\mathbf{x} \in \hat{\mathcal{H}}^n$ having a weight vector $\mu_i \in \hat{\mathcal{H}}^n$ and smoothing parameter γ_i. Then

$$\Gamma_i(\mathbf{x}) = \check{s}(\gamma_i M_q([s_k(x_k, \mu_{i,k})]_{k=1}^n)),\ q \in \mathbb{R}, \gamma_i \in (0, 1]. \tag{2}$$

The *activation* function $\check{s}$ is any sigmoid-like automorphism (a monotonic bijection) in $[0, 1]$. In particular, the widespread *logistic* function can be used by adapting it to map the real interval $[0, 1]$ on $(0, 1)$. Computationally cheap families of sigmoidal functions can be especially designed to operate in the $[0, 1]$ interval, such as $\check{s}(\cdot) = f(\cdot, k)$:

$$f(x,k) = \begin{cases} \frac{-k}{(x-0.5)-a(k)} - a(k) & \text{if } x \leq 0.5 \\ \frac{-k}{(x-0.5)+a(k)} + a(k) + 1 & \text{if } x \geq 0.5 \end{cases} \tag{3}$$

where $k > 0 \in \mathbb{R}$ is a parameter controlling the shape. This family of functions (Fig. 1) are all continuous bijections in $[0, 1]$, fulfilling $\forall k \in \mathbb{R}^+,\ f(0, k) = 0,\ f(1, k) =$

1, $\lim_{k\to\infty} f(x,k) = x$ and $f(x,0) = H(x-0.5)$, being H the Heaviside function. The expression $a(k)$ is solution of $a(k)^2 + \frac{a(k)}{2} - k = 0$, obtained by imposing the first two equalities.

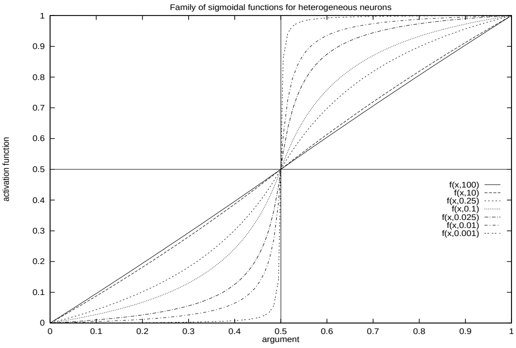


Figure 1. The family of sigmoidal functions $f(x,k)$, for different values of k.

The eventually *missing similarities* (because one or both of their arguments is missing) are regarded as *ignorance* and must not contribute in favour nor against the overall measure. In order to do this, let $C = \{k_1, \ldots, k_m\}$ be the set of indexes of the similarities that could not be computed. Then define $s_{\mathcal{X}}(\mathbf{x}, \mu_i) = M_q([s_k(x_k, \mu_{i,k})]_{k=1, k\notin C}^n)$, that is the mean of the successfully computed similarities. Thanks to the previous properties of A-averages,

$$M_q([s_k(x_k, \mu_{i,k})]_{k=1}^n) = M_q([s_k(x_k, \mu_{i,k})]_{k=1, k\notin C}^n, \underbrace{s_{\mathcal{X}}(\mathbf{x}, \mu_i), \ldots, s_{\mathcal{X}}(\mathbf{x}, \mu_i)}_{m \text{ times}})$$

4.2. Heterogeneous Neural Networks

Heterogeneous neural networks are neural architectures built out of the previously defined neuron models, thus allowing for heterogeneous or missing inputs. A feed-forward architecture, with a hidden layer composed of heterogeneous neurons and a linear output layer is a straightforward choice, thus conforming a *hybrid* structure. The heterogeneous neural network computes the function:

$$f_{\text{hnn}}(\mathbf{x}) = \sum_{i=1}^{h} \alpha_i \Gamma_i(\mathbf{x}) \tag{4}$$

where $h > 0$ is the number of (heterogeneous) hidden neurons and $\{\alpha_i\}$ is the set of mixing coefficients. The HNN thus keeps linearity in the output layer and interpretability in the hidden layer. It can be naturally seen as a generalization of the RBF. This is so because the response of hidden neurons is localized: centered at a given object (the neuron weight, where response is maximum), falling down as the input is less and less similar to that center.

5. Network Training

The general training procedure for the heterogeneous neural network (HNN for short) is based on evolutionary algorithms, due to the presence of missing information and the likely non-differentiability of the similarity measure. Specifically, the training procedure for the HNN used in this work is based on a *Breeder Genetic Algorithm* (BGA) [6], a method in mid position between Genetic Algorithms (GA) and Evolution Strategies. While in GA selection is stochastic and meant to mimic Darwinian evolution, BGA selection is driven by a deterministic *breeding* mechanism, an artificial method in which only the best individuals —usually a fixed percentage τ of total population size μ— are selected to be recombined and mutated, as the basis to form a new generation. The used BGA does not need any coding scheme, being able to cope with heterogeneous or missing genes [7].

Evolutionary methods are immediate candidates for neural network optimization, and they may alleviate the problem of local minima. However, when coding an ANN into a chromosome, highly complex interactions develop, due to the influence that a given weight on a hidden unit has on the whole network computation. The usual binary representation of the real-valued weights carries with it extra interactions between non-neighbouring genes, thus inducing strong epistatic effects in GA processing. In these conditions, it is at least doubtful that the *building block hypothesis* can hold.

5.1. The Breeder Genetic Algorithm

The Breeder Genetic Algorithm [6] is in midway between GAs and ESs. While in GA selection is stochastic and loosely inspired in natural selection, the BGA uses *truncation* selection, in which only the best individuals (usually a fixed percentage τ of the population size μ) are to be recombined and mutated. Genetic operators are applied by randomly picking two parents until the number of offspring equals $\mu - q$. Then, the former q best elements are re-inserted into the population, forming a new generation of μ individuals that replaces the previous one. The BGA selection mechanism is then deterministic (there are no probabilities), extinctive (the best elements are guaranteed to be selected and the worst are guaranteed *not* to be selected) and q-elitist (the best q elements always survive from a generation to the next). For the BGA, the typical value is $q = 1$. This is a form of the comma strategy (μ, λ) since the parents are not included in the replacement process, with the exception of the q previous best. Note that, given that q is fixed, only μ needs to be specified, since $\lambda = \mu - q$.

The BGA uses a *direct* representation, that is, a gene is a decision variable (not a way of coding it) and its allele is the value of the variable. An immediate consequence is that, in the absence of other conditionings as constraint handling, the fitness function equals the function to be optimized. In addition, the algorithm does not self-optimize any of its own parameters, as is done in ES and in some meta-GAs [8]. Chromosomes are thus potential solution vectors $\mathbf{x}$ of n components, where n is the problem size. This is of crucial importance since:

1. It eliminates the need of choosing a coding function for real numbers

2. It opens the way to the direct manipulation of different kinds of variables, other than

real numbers (e.g., fuzzy, discrete) as *single* genes.

3. It permits the design of data-dependent genetic operators.

The BGA is mainly driven by recombination (very much as an ordinary GA), with mutation regarded as an important but background operator intended to reintroduce some of the alleles lost in the population. This view is conceptually right for GAs, because alphabet cardinality is usually very small (two, in most cases). However, for those algorithms that make use of real-valued alleles (like the BGA) mutation has to be seen in the double role of solution fine-tuner (for very small mutations) and as the main discovery force (for moderate ones). In fact, the initial BGA formulation remarked the *synergistic* effect of the combined and iterated application of recombination and mutation to extract the most from an EA [6]. We now briefly describe different possibilities for the genetic operators.

5.2. Recombination

Any operator mixing the genetic material of the parents is called a recombination operator. In a BGA, recombination is applied unconditionally. Let $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$ be two selected individuals $\mathbf{x}, \mathbf{y}$ such that $\mathbf{x} \neq \mathbf{y}$. Let $\mathbf{z} = (z_1, \ldots, z_n)$ be the result of recombination and $1 \leq i \leq n$. The following are some of the more common possibilities to obtain an offspring $\mathbf{z}$:

5.2.1. Discrete Recombination (DR).

Set $z_i \in \{x_i, y_i\}$ (with equal probability).

5.2.2. Line Recombination (LR).

Set $z_i = x_i + \alpha(y_i - x_i)$, with a fixed $\alpha \in [0, 1]$.

5.2.3. Extended Intermediate Recombination (EIR).

Set $z_i = x_i + \alpha_i(y_i - x_i)$, with $\alpha_i \in [-\delta, 1+\delta]$ chosen with uniform probability. The $\delta > 0$ parameter expresses the degree to which offspring can be generated out of the parents's scope, an imaginary line that joins them in $\mathbb{R}$. More precisely, $\delta|y_i - x_i|$ is the maximum fraction of the distance between parents where the offspring can be placed, either left to the leftmost parent or right to the rightmost parent. Reasonable values should not exceed $\delta = 0.5$, since the bigger the δ, the more the effect of the parents is diminished in creating offspring.

5.2.4. Fuzzy Recombination (FR).

This operator replaces the uniform *pdf* (probability distribution function) by a bimodal one, with the two modes located at x_i, y_i. It thus favours offspring values close to either of the parents, and not in any intermediate point with equal probability, as with previous operators. The label “fuzzy” comes from the fact that the two parts of the *pdf* resemble fuzzy numbers (triangular in the original formulation [22]), fulfilling the conditions:

$$x_i - e|y_i - x_i| \leq t \ \leq x_i + e|y_i - x_i| \qquad y_i - e|y_i - x_i| \leq t \ \leq y_i + e|y_i - x_i|$$

stating that offspring t lies in one (or both) of the intervals, being $e > 0$ the fuzzy number's spread, the same for both parts. The favour for offspring values near the parents is thus stronger the closer the parents are. In the simplest case ($e = 0.5$) the two parts meet at the median and this point has zero probability.

5.3. Mutation

A mutation operator is applied to each gene with probability n^{-1} so that, on average, one gene is mutated for each individual. Let $\mathbf{z} = (z_1, \ldots, z_n)$ denote the result of mutation of an individual $\mathbf{x}$. The elements of $\mathbf{z}$ are formed as follows:

5.3.1. Discrete Mutation (DM).

Set $z_i = x_i + sign \cdot range_i \cdot \delta$ with $sign \in \{-1, +1\}$ chosen with equal probability, $range_i = \rho(r_i^+ - r_i^-)$, $\rho \in [0.1, 0.5]$ and

$$\delta = \sum_{i=0}^{k-1} \varphi_i 2^{-i}$$

where $\varphi_i \in \{0, 1\}$ taken from a Bernouilli probability distribution such that $\Pr(\varphi_i = 1) = \frac{1}{k}$. In this setting $k \in \mathbb{N}^+$ is a parameter originally related to the *precision* with which the optimum was to be located. In practice, the value of k is related to the *expected* value of mutation steps: the higher k is, the more fine-grained the resultant mutation operator is. The factor ρ sets the *maximum* step that mutation is allowed to produce w.r.t. the range $[r_i^-, r_i^+]$ of variable i.

5.3.2. Continuous Mutation (CM).

Same as DM with $\delta = 2^{-k\beta}$, where $\beta \in [0, 1]$ with uniform probability.

5.4. Extension of the BGA to Heterogeneous Problems

We consider basic data peculiarities most likely to be found in real applications. The BGA representation as well as the workings of the corresponding genetic operators are described. The algorithm manipulates the involved variables as a unique entity at all levels. Obviously, **real-valued** variables are directly treated as such, initialized at random within a predeclared range, and recombined and mutated with the operators described in (§5.2.) and (§5.3.).

5.4.1. Ordinal

Ordinal m-valued variables are represented as positive natural numbers in the interval $[1, m]$ an initialized at random within the interval. For recombination, there are three possibilities, which mimic the real-valued operators: DR (generally valid but ignores the order), LR (respects the order), and EIR (idem, needs an δ parameter). Some preliminary investigations lead to the choice of LR ($\alpha = 0.5$), that is, the *median* of the parents. Mutation involves an *increase* (to the immediately following value w.r.t. the linear order) or a *decrease* (idem, but in the opposite sense), and the decision is taken with equal probability.

5.4.2. Nominal

Nominal m-valued variables are also represented as an interval $[1, m]$, but no order relation is assumed. The clear choice for recombination is DR, being the only one explicitly assuming no underlying order. Mutation is realized by *switching* to a new value in the interval, with equal probability.

5.4.3. Fuzzy quantities.

The extension to handle *fuzzy numbers* is given by a tuple of reals (three in the general case, two if the chosen representation is symmetric). *Linguistic variables* are described by their anchor points on the abscissa axis (e.g., four in the case of trapezoidal membership functions).

Recombination of fuzzy numbers is taken as the corresponding extension of the operators for real-valued quantities. In particular, for EIR the mode is obtained following its formula (involving the selection of δ), and the spread is computed using the formula with the *same* α. This makes sense since the spread is usually proportional to the mode. Fig. (2) provides an example.

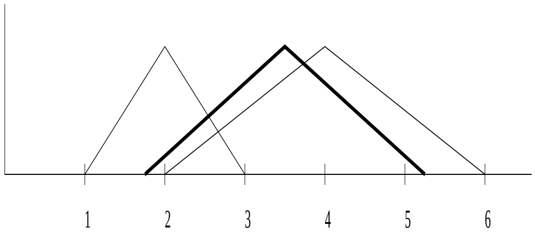


Figure 2. EIR recombination for fuzzy numbers with $\delta = 0.25$, and $\alpha = 0.75$ uniformly chosen in $[-0.25, 1.25]$. Mode and spread for the two parents are 2.0, 1.0 and 4.0, 2.0. The thicker number is the result of recombination. As for real numbers, the value of α makes offspring resemble its bigger parent more (a factor of $\frac{3}{4}$) than its smaller one. The mode is 3.5 and the spread 1.75.

Mutation of fuzzy numbers is also developed as an extension of the real-valued operators, by taking into account that mode and spread are collectively expressing a single (fuzzy) number. Both continuous and discrete operators can be used, as follows. The change on the mode is determined using the respective formulas. The change on the spread uses the *same* sign and δ (which are the terms depending on probabilities) as used for the mode.

5.4.4. Missing values

These values are dealt with in a specially meaningful way. They are initially generated according to the estimated probability of a missing value in the variable. This makes sense since for variables containing high numbers of missing values, the probability of placing one in the corresponding gene increases. If this probability is zero a missing value could still be introduced by mutation (signaling the temporal loss of a gene or trait). A mutation operator sets a missing value in the allele with a certain probability (usually very low). If this change leads to improved performance, it will be retained. A missing value cannot be mutated back to a non-missing one. A definite value can only be recovered by recombination to the (non-missing) gene of another individual.

Recombination is treated as *discrete* (DR) whenever at least one of the parents have a missing trait. This is coherent with the philosophy of EA: recombination stands for the *transmission* of the parents's genetic material to their offspring. If a parent is lacking a gene, this characteristic has to be given the chance to be passed on. Besides, if the trait or gene is lacking for both parents, it will be so for the offspring, since nothing can be "invented from scratch" (this is the role of mutation). In summary, given Ω a recombination operator (possibly heterogeneous), it is extended to a $\Omega_{\mathcal{X}}$ (where $\mathcal{X}$ denotes the missing value) as:

$$\Omega_{\mathcal{X}}(x_i, y_i) = \begin{cases} \Omega(x_i, y_i) & \text{if } x_i \neq \mathcal{X} \;\;\&\;\; y_i \neq \mathcal{X} \\ DR(x_i, y_i) & \text{if } x_i = \mathcal{X} \perp y_i = \mathcal{X} \\ \mathcal{X} & \text{otherwise} \end{cases}$$

where $\perp$ denotes exclusive-or. All this manipulation for missing values differs from the one that results by treating it as any other value, for its generation and propagation would be carried out blindly. The proposed treatment has the added appeal of being simple, and natural from the point of view of an EA in the sense that it is taken as a missing *gene* and is independent of the data type.

6. Heterogeneous Similarity Measures

We present specific measures defined to belong to the interval $[0, 1]$ for the sake of illustration, but there are many variations that may be superior by making better use of available expert knowledge.

6.0.5. Ordinal variables

It is assumed that the values of the variable form a linearly ordered space $(\mathcal{O}, \preceq)$. Let $x_{ik}, x_{jk} \in \mathcal{O}$, $x_{ik} \preceq x_{jk}$, P_{lk} the fraction of values of variable k that take on the value x_{lk} and the summation run through all the ordinal values x_{lk} such that $x_{ik} \preceq x_{lk}$ and $x_{lk} \preceq x_{jk}$ [9].

$$s_{ijk} = \frac{2\log(P_{ik} + \ldots + P_{jk})}{\log P_{ik} + \log P_{jk}} \tag{5}$$

6.0.6. Nominal variables

It is assumed that no partial order exists among these values and the only possible comparison is equality. The basic similarity measure for these variables is the *overlap*. Let $\mathcal{N}$ be a *nominal* space and $x_{ik}, x_{jk} \in \mathcal{N}$. Then $s_{ijk} = 1$ if $x_{ik} = x_{jk}$ and 0 otherwise.

6.0.7. Continuous variables

Let $x, y \in \Gamma = [r^-, r^+] \subset \mathbb{R}, r^+ > r^-$. The standard metric in $\mathbb{R}$ is a metric in Γ. Therefore, for any two values $x_{ik}, x_{jk} \in \mathbb{R}$:

$$s_{ijk} = \hat{s}_{\mathbb{R}} \left(\frac{|x_{ik} - x_{jk}|}{\sup\limits_{x,y\in\Gamma} |x - y|} \right) \tag{6}$$

where $\hat{s}_{\mathbb{R}} : [0,1] \longrightarrow [0,1]$ is a decreasing continuous function. The family $\hat{s}_{\mathbb{R}}(z) = (1 - z^{\beta})^{\alpha}, 0 < \beta \leq 1, \alpha \geq 1$ is used here, with $\alpha = 2, \beta = 1$.

6.0.8. Integer variables

Given that $\mathbb{N} \subset \mathbb{R}$, any distance-based similarity in $\mathbb{R}$ is also valid in $\mathbb{N}$. A flexible choice does not limit the range of integer values (assumed positive for convenience). In this case, a self-normalizing distance-based similarity $s_{ijk} = \hat{s}_{\mathbb{N}}(|x_{ik} - x_{jk}|)$ is indicated, where $\hat{s}_{\mathbb{N}} : [0, \infty) \longrightarrow (0, 1]$ is a decreasing continuous function. In particular, the function $\hat{s}_{\mathbb{N}}(z) = \frac{1}{1+z}$ can be used.

6.0.9. Binary variables

In the data analysis literature there are many similarity measures defined on collections of binary variables (see e.g. [10]). This is mostly due to the uncertainty over how to accommodate negative (i.e. false-false) matches. The present situation is that of comparison of *a single* binary variable rather than a whole vector. In general, one should know which of the two matches is the really relevant (true-true or false-false). For these reasons, treating the variable as purely nominal can result in a loss of relevant information. The difference between a two-valued nominal variable and a binary variable is that in the latter the two values are mutually exclusive. For instance, the variable *color* with values *black, white* would be nominal, whereas the variable *is-black?* with values *yes, no* would be binary. Sometimes the distinction is more conceptual than practical.

Since this meta-knowledge is usually not handy, we use in this work a frequency-based approach, as follows. Let x_{ik}, x_{jk} be two binary values and let P_{lk} be again the fraction of values of variable k that take on the value x_{lk}. We define: $s_{ijk} = h(1 - P_{ik}, 1 - P_{jk})$, where $h(x, y) = \frac{2xy}{x+y}$ is the *harmonic mean* between x and y. This measure compares the *agreement on the rarity* of each value: the similarity is higher the less frequent the values are.

6.0.10. Set variables

These can be regarded as generalized *nominal* variables whose values are not single elements of the space but subsets thereof. Let $\mathcal{S}$ be a *set* space and $x_{ik}, x_{jk} \in \mathcal{S}$. When $\mathcal{S}$ is finite (so that all its subsets are),

$$s_{ijk} = \frac{\#\{x_{ik} \cap x_{jk}\}}{\#\{x_{ik} \cup x_{jk}\}}, \qquad \text{if } x_{ik} \neq \emptyset \vee x_{jk} \neq \emptyset \qquad \text{and } s_{ijk} = 1 \qquad \text{if } x_{ik} = x_{jk} = \emptyset \tag{7}$$

is a similarity measure with $s_{max} = 1$, which reduces to the *overlap* (nominal variables) for singleton sets.

6.0.11. Fuzzy variables

For variables representing fuzzy sets, similarity relations from the point of view of fuzzy theory have been defined elsewhere [11], and different choices are possible. In possibility theory, the *possibility* expresses the likeliness of co-occurrence of two vague propositions, with a value of one standing for absolute certainty. For two fuzzy sets $\tilde{A}, \tilde{B}$ of a reference set X, it is defined as:

$$\Pi_{(\tilde{A})}(\tilde{B}) = \sup_{u \in X} (\mu_{\tilde{A} \cap \tilde{B}}(u)) = \sup_{u \in X} (\min (\mu_{\tilde{A}}(u), \, \mu_{\tilde{B}}(u)))$$

In our case, if $\mathcal{F}_k$ is an arbitrary family of fuzzy sets, and $x_{ik}, \, x_{jk} \in \mathcal{F}_k$, the following similarity relation can be used $s_{ijk} = \Pi_{(x_{ik})}(x_{jk})$. Notice that this measure indeed fulfills axioms **S1, S2** and **S3**.

7. Experimental Comparison

A number of experiments are carried out to illustrate the validity of the approach, using eight benchmarking problems, selected as representatives because of the richness in data heterogeneity, taken from [12] and [3]. Their main characteristics are displayed in Table 1. For every data set, the available documentation is analysed for an assessment on the more appropriate treatment. Missing information is also properly identified. There is hence an explicit *transfer of knowledge* from the domain to the heterogeneous neuron model. Specifically, some originally "continuous" variables are treated as ordinal since this makes much more sense. Examples would be *number of times pregnant* or *heart pulse*. Among these, there are variables that, besides being endowed with a total order relation, display a source of vagueness (coming from their *subjective* character) that has to be modeled. For instance, this is the case of the *temperature of extremities* (`cold, cool, normal, warm`) or the *abdominal distension* (`none, slight, moderate, severe`). These variables are treated as ordinal by respecting the number and order of the values.

The HNN is compared to a standard radial basis function network (RBF) and to a multilayer perceptron (MLP). The RBF neuron model has provision for a smoothing parameter, different for every hidden neuron, as follows:

$$\Gamma_i(\mathbf{x}) = exp\left(-\frac{||\mathbf{x}-\mu_i||^2}{2\sigma_i^2}\right), \sigma_i \in \mathbb{R}, \mu_i \in \mathbb{R}^n \tag{8}$$

while the MLP neuron model computes the classical function:

$$\Gamma_i(\mathbf{x}) = \lambda\left(\mathbf{x}\cdot\mu_i - \theta_i\right), \theta_i \in \mathbb{R}, \mu_i \in \mathbb{R}^n \tag{9}$$

where λ is the logistic function. The three networks are trained using exactly the same procedure to exclude this source of variation from the analysis. The RBF and MLP networks need a pre-processing of the data, following the recommendations in [12]. Specifically, ordinal variables are mapped to an equidistant linear scale, a 1-out-of-c coding is used for nominal ones and an extra input is added for those variables with missing values. The network architecture is fixed to one single layer of hidden neurons ranging from 1 to 30 plus as many output neurons as required by the task, logistic for classification problems and linear otherwise. The input variables for the RBF and MLP network are standardized to zero mean, unit standard deviation. This is not needed by the heterogeneous neurons because they compute a normalized measure, but is beneficial for the other networks. The weights (including biases and standard deviations) for the RBF network are let to vary in $[-10, 10]$, a sufficiently wide range given the normalization chosen; the same interval is used for the hidden-to-output weights in all the networks. The BGA main parameters are set to $\mu = 60, \tau = 5$. The aggregation constant in (2) is set to $q = 1$.

Table 1. Basic features of the data sets. ’Missing’ refers to the *percentage* of missing values, p to the number of examples and n to that of variables. Types are: C (continuous), N (nominal), I (integer), D (ordinal) and F (fuzzy). ’Output’ is R for regression or a number indicating the number of classes.

Name	p	n	Data Types	Output	Missing
Annealing	898	28	6C,19N,3I	6	none
Audiology	226	69	8D,61N	4	2.3%
Boston Housing	506	13	11C,1I,1B	R	none
Credit Screening	690	15	6C,6N,3B	2	0.6%
Heart Disease	920	13	3C,5N,3I,2B	2	16.2%
Horse Colic	364	20	6D,5N,2I,5C,2B	3	26.1%
South African Heart	462	9	7C,1I,1N	2	none
Servo Data	167	4	2I,2N	R	none
Solar Flares	1066	9	4N,3D,2B	2	none

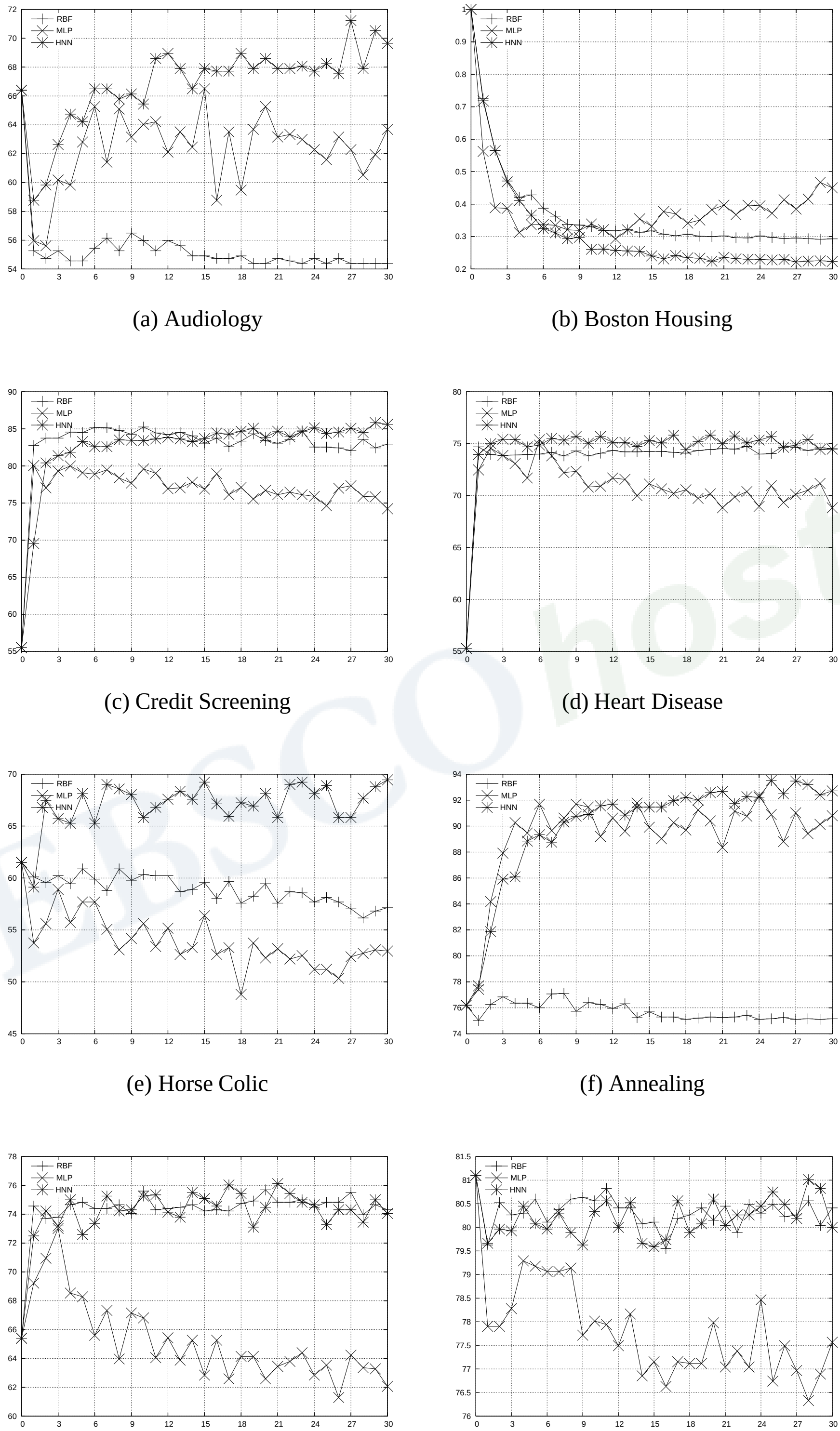
The output is presented for all networks in 1-out-of-c mode (that is, c outputs) for classification problems with c classes. The error measure in this case is *generalized cross-entropy* (GCE) [13]. For regression problems, the mean is subtracted to the continuous output and the *normalized root mean square* (NRMS) error is reported. Each data set is split in three parts, for training (TR), validation (VA) and test (TE), as 50%-25%-25%. The BGA task is to minimize error (either NRMS or GCE) on the TR part, until 500 generations or convergence. Then the network having the lowest error on the VA part is returned. The reported performance is the NRMS or GCE of that network on the TE part. This process is repeated ten times and the average is taken.

We present performance results in Fig. 3(a) to (h). In each figure, the abscissa represents the number of neurons in the hidden layer, and the ordinate the average NRMS or GCE of that network on the TE part, as explained above. For classification problems, the abscissa value 0 is used to indicate the percentage of examples in the majority class (that is, the minimum acceptable performance, as a reference). For regression problems, this value is set to 1.0. Note that for classification problems, higher values are better, whereas for regression problems, lower values are better. The HNN shows enhanced performance for most of the problems, specially for those displaying higher heterogeneity or missingness, and less so when this is not the case. This is reasonable since the HNN behaves as a RBF network when there is no heterogeneity or missingness.

Table 2. Weights for one of the neurons in the *Horse Colic* problem.

Name	Type	Value
age	BINARY	`Adult`
rectal temperature	REAL	38.48
pulse	INTEGER	62
respiratory rate	INTEGER	40
temperature of extremities	ORDINAL	`normal`
peripheral pulse	ORDINAL	`normal`
mucous membranes	NOMINAL	`normal pink`
capillary refill time	BINARY	`less than 3 secs.`
pain estimation	NOMINAL	`no pain/alert`
peristalsis	ORDINAL	`normal`
abdominal distension	ORDINAL	`none`
nasogastric tube gas emission	ORDINAL	`significant`
nasogastric reflux	ORDINAL	`none`
nasogastric reflux pH	REAL	1.12
rectal examination	NOMINAL	`decreased`
abdomen	NOMINAL	`firm feces`
packed cell volume	REAL	42.34
total protein	REAL	14.1
abdominocentesis appearance	NOMINAL	`clear`
abdominocentesis total protein	REAL	0.22
γ_i	REAL	0.7286

The interpretability of the learned weights is also enhanced, since the weights are of the same type as their matching input. To illustrate this point, one of the obtained neurons is



(a) Audiology

(b) Boston Housing

(c) Credit Screening

(d) Heart Disease

(e) Horse Colic

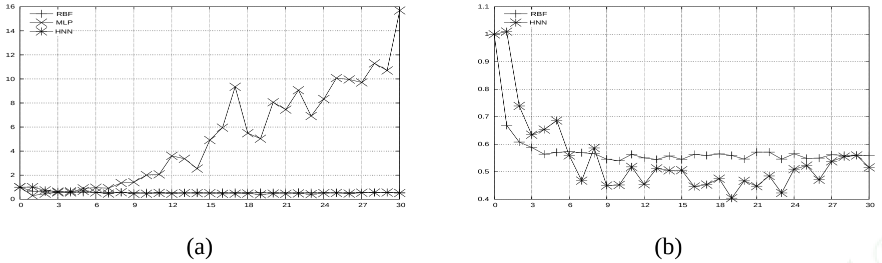
(f) Annealing

(g) South African Heart

(h) Solar Flares

Figure 3. Generalization results for the networks. See text for an explanation of axis.

shown in Table 2, corresponding to the *Horse Colic* problem, chosen because it displays a good amount of heterogeneity. This assigns a well-defined meaning to individual weights and to the neuron outcome (a similarity degree in $[0, 1]$), and at the same time permits to interpret the trained network in the original input domain (i.e., a collection of measurements about horses). In contrast, the weight vector for each RBF or MLP neuron consists (for this problem) of a vector of 55 real numbers.



(a) (b)

Figure 4. Generalization performance results for the Servo Data. Left: heavy overfit by the MLP network. Right: only RBF and HNN.

We additionally present performance results for the Servo Data (Fig. 4), which are specially interesting. This is a small data set with only 4 variables (none of them continuous) and less than 200 examples. For this data set the MLP overfits heavily (Fig. 4, left), due to the amplification in the number of variables introduced by the coding scheme of the nominal ones, which are many-valued, and the low number of examples. If we focus only on the RBF and the HNN, the compared results cannot be appreciated in Fig. 4 (left); a zoom of these two networks reveals that the HNN obtains lower errors for most network sizes, Fig. 4 (right).

8. Related Work

In classical artificial intelligence (AI) systems, the notion of *similarity* appears mainly in case-based reasoning (CBR), where learning by analogy and instance-based learning fuse. The popularity of CBR systems has boosted its use and signaled its key role in cognitive tasks [1]. As a matter of fact, some form of similarity is inherent in the majority of AI approaches. There is an agreement in that it is easier to respond intelligently to a stimulus if previous responses made under *similar* circumstances can be recalled. Similarity coefficients have had a long and successful history in the literature of cluster analysis and data clustering algorithms [4].

The origins of learning by similarity in artificial neural systems can be traced back to the pioneering works of Hebb and his now classic book [2]. He postulated that the functionality of ANNs had to be determined by the strengths of the neural connections. This functionality should be adjusted to *increase* the likeliness of getting a *similar* response to *similar* inputs in the future, provided the elicited response is the desired one. In the opposite situation, the weights should be adjusted to decrease this likeliness. However, this inspiring idea has not been fully exploited in the prevalent neuron models.

There have been few attempts to incorporate heterogeneous information into the workings of an artificial neuron in a principled way. The main contribution is perhaps encountered in heterogeneous distance proposals, where separate distance calculations are used for nominal and continuous variables [19]. The authors present an extension of the RBF model to nominal quantities and missing values. The RBF network is used in its original interpolative definition (i.e., there is one hidden node for each example in the training set). The extension consists in the use of the Value Difference Metric [20] for distance computations involving nominal variables. An Euclidean metric is used to account for the partial distances. All the similarities used have to be distance-based, which can be a limiting factor in many circumstances. Missing values are handled with a *pessimistic* semantics, defining the distance involving a missing value to be the maximum possible distance (actually a value of one). The results point to an increase in performance w.r.t. standard Euclidean distance when the amount of nominal information is significant, giving support to a deeper and more precise formulation of neuron models as similarity computing devices. The Support Vector Machine [21] could also in principle handle heterogeneous data types and missing values, by the use of a proper kernel. However, this "universal heterogeneous kernel" is presently lacking in the literature. Note that there are interesting links between the HNN and a SVM, in that a kernel is generally regarded as a form of similarity measure [23].

9. Conclusion

An heterogeneous neuron that computes a *similarity index* in $[0, 1]$, followed by a logistic function has been proposed. The conceptual and functional definition of a neuron as a similarity computing device permits to base the computation on principled grounds, and carries a number of advantages: first, the task performed by the hidden neurons is clearly delimited, allowing for a deeper influence on the workings of a neural network; second, it gives a precise *semantics* to the neuron computation, thereby facilitating the interpretation process.

Neural architectures making use of such neurons are termed *heterogeneous neural networks*. A body of experimental evidence has shown very promising results in several difficult learning tasks. These results also illustrate the importance of data preparation and the use of a more adequate similarity measure that captures relations in the data. As further work, we are currently working on an clustering-based alternative training scheme.

References

[1] Aamodt, A., Plaza, E. Case-based reasoning: Foundational issues, methodological variations and system approaches. *AI Communications*, 7(1):39-59, 1994.

[2] Hebb, D. *The organization of behaviour*, John Wiley, 1949.

[3] P.M. Murphy, D. Aha. UCI Repository of machine learning databases. UCI Dept. of Information and Computer Science, 1991.

[4] J. L. Chandon, S. Pinson. *Analyse Typologique*. Masson, 1981.

[5] Hille, E. *Methods in Classical and Functional Analysis* . Addison-Wesley, 1971.

[6] H. Mühlenbein, D. Schlierkamp-Voosen. Predictive Models for the Breeder Genetic Algorithm. *Evolutionary Computation*, 1(1), 1993.

[7] L. Belanche. Evolutionary Optimization of Heterogeneous Problems. In *Lecture Notes in Computer Science* 2439, pp. 475-484, 2002.

[8] T. Bäck. *Evolutionary Algorithms in Theory and Practice* . Oxford Press, 1996.

[9] D. Lin. An Information-Theoretic Definition of Similarity. Proceedings of International Conference on Machine Learning, 1998.

[10] Everitt, B. Cluster Analysis. Heinemann Books, 1977.

[11] L. Zadeh. Fuzzy Sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1: 3-28, 1978.

[12] L. Prechelt. Proben1: A set of Neural Network Benchmark Problems and Benchmarking Rules. Fac. für Informatik. Karlsruhe Univ. Tech. Rep. 21/94, 1994.

[13] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford, 1995.

[14] V. Kaburlassos, V. Petridis. Learning and decision-making in the framework of fuzzy lattices. *New learning paradigms in soft computing* . Physica-Verlag, 2002.

[15] B. Natarajan. *Machine learning: A theoretical approach*. Morgan Kaufmann, 1991.

[16] J.A. Anderson, E. Rosenfeld (eds.) *Neurocomputing: foundations of research*. MIT press, 1988.

[17] S.M. Omohundro. Geometric Learning Algorithms. Tech. Rep. 89-041. Univ. of Berkeley, CA, 1989.

[18] V. Tresp, S. Ahmad, R. Neuneier. Training Neural Networks with Deficient Data. In NIPS 6, 1994.

[19] D.R. Wilson, T.R. Martinez. Heterogeneous Radial Basis Function Networks. In Procs. of the 6th Intl. Conf. on Neural Networks (ICANN'96), vol. 2, pp. 1263-1267, 1996.

[20] C. Stanfill, D. Waltz. Toward memory-based reasoning. *Communications of the ACM* 29, 1986.

[21] C. Cortes, V. Vapnik. Support-Vector Networks. *Machine Learning*, 20, 1995.

[22] Voigt, H.M., Mühlenbein, H., Cvetkovic, D. Fuzzy recombination for the continuous Breeder Genetic Algorithm. In Procs. of ICGA'95.

[23] B. Schölkopf, A. Smola. *Learning with kernels*. MIT Press, Cambridge, MA, 2002.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 18

TUNING DIFFERENTIAL EVOLUTION FOR ARTIFICIAL NEURAL NETWORKS

Magnus Erik Hvass Pedersen, Andrew John Chipperfield
Hvass Laboratories Technical
Report no. HL0803, 2008

Abstract

The efficacy of an optimization method often depends on the choosing of a number of behavioural parameters. Research within this area has been focused on devising schemes for adapting the behavioural parameters during optimization, so as to alleviate the need for a practitioner to select the parameters manually. But these schemes usually introduce new behavioural parameters that must be tuned. This study takes a different approach in which finding behavioural parameters that yield good performance is considered an optimization problem in its own right and can therefore be attempted solved by an overlaid optimization method. In this work, variants of the general purpose optimization method known as Differential Evolution have their behavioural parameters tuned so as to work well in the optimization of an Artificial Neural Network. The results show that DE variants using so-called adaptive parameters do not have a general performance advantage as previously believed.

Keywords: Numerical optimization, stochastic, multi-agent, parameter tuning.

1. Introduction

The optimization method known as Differential Evolution (DE) was originally introduced by Storn and Price [1] and offers a way of optimizing a given problem without explicit knowledge of the gradient of that problem. This is particularly useful if the gradient is difficult or even impossible to derive. Gradient-based optimization for the dataflow model known as an Artificial Neural Network (ANN) is called BackPropagation (BP) and was developed independently by Werbos and Rumelhart et al. [2] [3] [4]. Deriving the gradient for an optimization problem is not always trivial, so optimizers such as DE which do not rely on the gradient can be used in rapid prototyping of new ANN variants and mathematical models in general.

Optimizing an ANN is an interesting problem because it may easily contain a large number of parameters that are highly interrelated and do not lend themselves well to direct human interpretation. DE has been used for optimizing ANNs by Ilonen et al. [5] who conclude that DE has performance comparable to the classic gradient-based approach, although results presented show that the gradient-based BP consistently out-performs DE. It has been recognized since the inception of DE that different choices of behavioural parameters cause it to perform worse or better on particular problems and the selection of good parameters is a challenging art, see for example Storn et al. [6] [7], Liu and Lampinen [8], and Zaharie [9]. There has been a trend in recent years to try and make the DE parameters automatically adapt to new problems during optimization, hence alleviating the need for the practitioner to select the parameters by hand, see for example Price et al. [7], Liu and Lampinen [10], Qin et al. [11] [12], and Brest et al. [13]. But these DE variants with so-called adaptive parameters just introduce new parameters that must then be tuned by the practitioner and has therefore merely deferred this difficult issue without actually eliminating it.

Finding the best choice of behavioural parameters for an optimization method is considered here to be an optimization problem in its own right and hence attempted solved by an overlaying optimization method. This is known as Meta-Optimization, Meta-Evolution, Super-Optimization, etc. See for example the work by Grefenstette [14], Bäck [15], Keane [16], Meissner et al. [17], and the more statistically oriented approaches by François and Lavergne [18], Czarn et al. [19], and Nannen and Eiben [20]. We have previously used a meta-optimization technique that is both simple and efficient for tuning DE parameters to perform well on benchmark problems [21], and that technique is also used here for tuning the DE parameters to perform well in ANN optimization, which poses other challenges.

The paper is organized as follows. Section 2. gives a brief overview of the class of ANNs to be optimized. Section 3. describes the DE variants considered in this work. Section 4. describes an optimization method especially suited for meta-optimization, and section 5. describes how to employ it as an overlaid meta-optimizer. The experimental settings and results are given in section 6., and overall conclusions are found in section 7..

2. Artificial Neural Networks

An ANN is a dataflow model inspired by the complex connections of cells in a biological brain. The type of ANN used in this study is a fully connected feedforward network with a single hidden layer having four nodes. This means the data flows from an input layer, through a single hidden layer, and finally reaches the output layer. This kind of ANN has its origin in research dating back several decades (see [22, Section 1.9] for a detailed historical account). More recent texts on this and other kinds of ANN can be found in [22] [23]. The ANN has parameters, usually called weights, that can be adjusted to provide different mappings from input to output. The optimization task is then to change these weights so a desired correspondence between inputs and outputs of the ANN is achieved. Under certain circumstances, and provided the input-output mapping is well-behaved, this approximation can be achieved arbitrarily well [24] [25] [26], simply by adding more hidden nodes to an ANN having a single hidden layer whose nodes use bounded, compressive functions such as the Sigmoid function described below. However, the datasets used in the experiments here are not necessarily well-behaved as they may contain contradictory entries or have missing features needed to properly distinguish them.

To briefly describe the computation of this kind of ANN, let the j'th node of the i'th layer be denoted by n_{ij}. The node has a bias-value b_{ij} for offsetting the influence of its inputs. The weight denoted by w_{ijk} connects the k'th node from the previous layer to node n_{ij}. For an ANN having just a single hidden layer, the computation of the j'th element of the ANN output $\vec{o}$ can be expressed as a single, flattened formula:

$$o_j = b_{3,j} + \sum_{k=1}^{N_2} w_{3,j,k} \cdot \sigma \left(b_{2,k} + \sum_{l=1}^{N_1} w_{2,k,l} \cdot n_{1,l} \right) \tag{1}$$

Where $\vec{n}_1 \in \mathbb{R}^{N_1}$ is the first layer of the network containing its input, and N_1 is the number of nodes in this first layer. The number of nodes in the hidden layer is N_2, and the number of nodes in the output layer is N_3. The parameters subject to optimization are all the bias-values and weights for the entire ANN, that is, b_{ij} and w_{ijk} for all valid combinations of i, j, and k. The $\sigma(\cdot)$ part of this formula represents the calculation of the k'th node in the hidden layer of the ANN, where the compressive function used here is called the Sigmoidal function $\sigma : \mathbb{R} \to (0,1)$, defined as: $\sigma(x) = 1/(1+e^{-x})$

A fitness measure must first be defined in order to optimize the weights and bias-values of an ANN. Let T be the set of input-output data the ANN must be trained to mimic. An element in the dataset T consists of input data $\vec{t_i}$ and its associated output data $\vec{t_o}$. Passing the input data $\vec{t_i}$ to the ANN and computing the actual ANN output using Eq.(1) yields $\vec{o}$. Accumulating the error between desired and actual ANN output for all data pairs in T gives a measure for how well the ANN mimics a desired input-output mapping. This is known as the Mean Squared Error (MSE), defined as:

$$\text{MSE} = \frac{1}{N_3 \cdot |T|} \sum_{(\vec{t_i}, \vec{t_o}) \in T} \sum_{j=1}^{N_3} (o_j - t_{oj})^2 \tag{2}$$

where normalization is done with both the number of input-output pairs $|T|$, and the number of ANN output-nodes N_3. This is not standard in the literature, but has the advantage of giving a uniform fitness measure that can be compared independently of the given dataset and ANN sizes. The MSE measure is also sometimes halved in the literature to facilitate a cleaner derivation of the gradient, but that is ignored here. It is customary to split the dataset T and use one part during optimization (or training) of the ANN, and the other part to test the generalization ability of the ANN, as a simple form of statistical cross-validation [27]. However, this is ignored here as this study is primarily concerned with the performance of DE rather than that of the ANN.

The traditional way of optimizing ANN weights is to use a gradient-based optimization method to follow the path of steepest descent of the MSE measure. This is known as BackPropagation (BP) because it first makes a forward pass and computes the node values for the entire ANN, and then propagates the MSE errors backwards through the ANN to determine adjustments for the weights and bias-values. The BP method was developed independently by Werbos and Rumelhart et al. [2] [3] [4]. The stochastic gradient is used in this study, where a single input-output pair is picked randomly from the dataset T, and the gradient is computed for just that data pair. This requires much less computational time than accumulating the gradient for all data pairs in T, and taking small enough steps yields satisfactory results [28].

Once the ANN weights and bias-values have been optimized for the MSE measure, the ANN can be used for classification tasks as well, as proven by Richard and Lippmann [29]. This study uses 1-of-m classification [22] [28] [30], where each ANN output represents a class, and the output with the highest value is chosen as the classification. The Classification Error (CLS) is computed as the number of data-pairs in T, for which the ANN output classification does not match that of the desired output, divided by the total number of data pairs $|T|$. This yields a CLS value in the range $[0, 1]$, where a lower CLS value means a better classification rate.

3. Differential Evolution

An optimization method that can be used instead of BP for optimizing ANN weights and bias-values, is the population-based method known as Differential Evolution (DE) due to Storn and Price [1]. DE does not use the gradient of the problem it is optimizing, but maintains a number of candidate solutions, here called agents, creating new agents by combining randomly chosen agents from its population, and accepting the new agents in case of fitness improvement.

3.1. Basic Variants

DE employs evolutionary operators that are dubbed crossover and mutation in its attempt to minimize some fitness function $f : \mathbb{R}^n \to \mathbb{R}$. The operators are typically applied in turn but have been combined for a more concise description in the following. The basic DE/rand/1/bin variant is believed to be one of the best performing and hence one of the most popular of the basic DE variants [7] [31]. Formally, let $\vec{y} \in \mathbb{R}^n$ be the new candidate solution for an agent whose current position in the search-space is $\vec{x}$, and let $\vec{a}$, $\vec{b}$, and $\vec{c}$ be the positions of distinct and randomly chosen agents, which must also be distinct from the agent $\vec{x}$ that is currently being updated. The elements of $\vec{y} = [y_1, \cdots, y_n]$ are then computed as follows:

$$y_i = \begin{cases} a_i + F \cdot (b_i - c_i) & , i = R \vee r_i < CR \\ x_i & \text{, else} \end{cases} \tag{3}$$

where $F \in [0, 2]$ is a user-adjustable parameter called the differential weight, and the randomly chosen index $R \in \{1, \cdots, n\}$ ensures at least one element differs from that of the original agent: $y_R \neq x_R$. While the rest of the elements are either chosen from the original position $\vec{x}$ or computed from combining other agents' positions, according to the user-adjustable crossover probability $CR \in [0, 1]$, and with $r_i \sim U(0, 1)$ being a uniform random number drawn for each vector element i. Once the new potential position $\vec{y}$ has been computed, it will replace the agent's current position $\vec{x}$ in the case of improvement to the fitness. An additional user-adjustable parameter of DE is the population size NP, that is, the number of agents in the DE population.

The DE/best/1/bin variant has been long out of favour with researchers and practitioners because it is believed to have inferior performance with tendencies for premature convergence [7] [31]. The DE/best/1/bin variant replaces Eq.(3) with:

$$y_i = \begin{cases} g_i + F \cdot (a_i - b_i) & , i = R \vee r_i < CR \\ x_i & \text{, else} \end{cases} \tag{4}$$

where $\vec{g}$ is the population's best known position until now. In the original version of this, the agents $\vec{a}$ and $\vec{b}$ are chosen to be different not only from each other, but also from the agent $\vec{x}$ currently being processed, but it simplifies the implementation a good deal if the latter is not required. This variant is called DE/best/1/bin/simple, nicknamed The Joker, and was shown in [21] to have an overall good performance

if the behavioural parameters were properly tuned.

3.2. Perturbed & Adaptive Parameters

An attempt to remedy the need for a user to determine DE parameters that are good for a given optimization problem, is to perturb the behavioural parameters during optimization. Several schemes for perturbing the differential weight F have been proposed in the literature [7] [32], where the ones used by Storn himself are generally just the Dither and Jitter schemes [31] [33]. It was noted in [21] that these two schemes are actually identical, with the exception of Dither perturbing F once for every vector being processed in Eq.(3) or (4), where as Jitter perturbs it for each vector-element being processed. A common notation may therefore be used for both Dither and Jitter, having a midpoint F_{mid} and a range F_{range} for the pertubation:

$$F \sim U(F_{mid} - F_{range}, F_{mid} + F_{range})$$

The midpoint can be anywhere between $F_{mid} \in [0, 2]$, and the perturbation range can be anywhere between $F_{range} \in [0, 3]$, which were chosen to allow for unusual values when F_{mid} and F_{range} are tuned automatically later in this study.

The concept of perturbing behavioural parameters can be taken a step further, by also using the fitness improvement to guide selection of behavioural parameters that yielded good performance. Examples of such adaptive schemes for DE parameters are the Fuzzy Adaptive DE (FADE) by Liu and Lampinen [10], and the Self-adaptive DE (SaDE) due to Qin et al. [11] [12]. In this study the DE variant known as Janez DE (JDE) will be used, which is due to Brest et al. [13], and has been chosen because its performance has been found to compare well against other so-called state-of-the-art DE variants [13] [34]. It is presented by its authors as a Self-Adaptive DE as well, because it eliminates the need for a user to select the F and CR parameters - yet ironically introduces 8 new user-adjustable parameters to achieve this. This tendency is common for such DE variants, and indeed also for the comparatively simpler Dither and Jitter variants described above which merely perturb the behavioural parameters. The JDE variant works as follows. First assign start values to F and CR, F_{init} and CR_{init} respectively. Then before computing the new potential position of a DE agent using Eq.(3) or (4), decide what parameters F and CR to use in that formula. With probability $\tau_F \in [0, 1]$ draw a new random $F \sim U(F_l, F_l + F_u)$, otherwise reuse F from previously, where each DE agent retains its own F parameter. Similarly each agent retains its own CR parameter, for which a new random value $CR \sim U(CR_l, CR_l + CR_u)$ is picked with probability $\tau_{CR} \in [0, 1]$ otherwise the old CR value for that agent is reused. Whichever F and CR values are being used in the computation of Eq.(3) or (4), they will survive to the next iteration or be discarded along with the agent's new potential position $\vec{y}$, according to fitness improvement.

4. Local Unimodal Sampling

To tune the behavioural parameters of DE variants an overlaid optimization method is used which is called Local Unimodal Sampling (LUS) and is described in more detail in [21]. Briefly, LUS is somewhat related to the Matyas [35] and Rastrigin [36] families of optimization methods, which work by sampling a fraction of the search-space surrounding the current position, and moving to the new position in case of improvement to the fitness. LUS samples the new potential position $\vec{y} \in \mathbb{R}^n$ from the neighbourhood of the current position $\vec{x}$ by adding a random vector $\vec{a}$:

$$\vec{y} = \vec{x} + \vec{a},\ \vec{a} \sim U\left(-\vec{d}, \vec{d}\right)$$

with $\vec{d}$ being the current search-range, initially chosen as the full range of the search-space and decreased during an optimization run as described below. When LUS is used for meta-optimization these represent different choices of DE behavioural parameters, meaning e.g. that $\vec{x} = [NP,\ CR,\ F]$ for DE/rand/1/bin. The search-space therefore constitutes all valid choices of behavioural parameters, whose boundaries will be detailed later.

LUS decreases its sampling-range during optimization so as to focus the search, in a manner related to those described in earlier methods [37] [38] [39] [40] [41] [42], only more simply. When a sample fails to improve the fitness, the search-range $\vec{d}$ is decreased by multiplying with a factor q, defined as:

$$\vec{d} \leftarrow q \cdot \vec{d},\ q = 2^{-\beta/n} \tag{5}$$

where n is the dimensionality of the search-space, and $0 < \beta < 1$ causes slower decrease of the search-range, where as $\beta > 1$ causes more rapid decrease. For the experiments in this paper, a value of $\beta = 1/3$ is used as it has been found to yield good results on a broad range of problems.

5. Meta-Optimization

The parameters controlling the behaviour and efficacy of DE are usually found by manual experimentation, which is time-consuming and susceptible to human misconceptions of the inner-working of the optimization method. The problem of finding the best choice of behavioural parameters for a given optimization method, is considered here as an optimization problem in its own right and is termed Meta-Optimization. In other words, the idea is to have an optimization method act as an overlaying meta-optimizer, trying to find the best performing behavioural parameters for another optimization method, which in turn is used to optimize the ANN weights. The overall concept is depicted in figure 1. It is important to understand that parameter tuning by way of meta-optimization is done in an offline manner, and hence differs from adaptation of DE parameters which is done in an online manner during optimization. The DE parameters found using meta-optimization can therefore be used by other researchers and practitioners without the need for augmenting their DE implementation with adaptive schemes. This also means that meta-optimization can be readily employed on other optimization methods than just DE. The approach to meta-optimization from [21] is used here, albeit with a simplification because the tuning is done with regard to just one ANN problem, where as [21] would tune the DE parameters for multiple problems.

The crux of automatically finding good behavioural parameters for an optimization method, is to define an appropriate meta-fitness measure that can be made the subject of meta-optimization. The meta-fitness measure must reflect how the optimization method is ultimately to be used, but at the same time allow for efficient meta-optimization. A typical way of performing optimization with DE is to let it run for some predetermined and seemingly adequate number of iterations. This means the performance of DE and a given choice of behavioural parameters, can be rated in terms of the ANN fitness that can be obtained within this number of iterations. Since DE is stochastic by nature, it will be likely that it gives a different result for each optimization run, and a simple way of lessening this stochastic noise is to perform a number of optimization runs in succession, using the average of the fitness values obtained to guide the meta-optimization. But repeating optimization runs of DE also causes the computational time to grow undesirably. A simple way of saving computational time is to preemptively abort a meta-fitness evaluation, once the meta-fitness becomes worse than that needed for the overlaying meta-optimizer to accept the new parameters, and the meta-fitness is known not to improve for the rest of the evaluation. This technique is generally termed Preemptive Fitness Evaluation and has been used by researchers for decades, although its original author is difficult to establish as the technique is seldom mentioned in the literature.

To employ preemptive fitness evaluation when meta-optimizing the DE parameters, the meta-fitness measure must be ensured to be non-decreasing and hence only able to grow worse, so as the evaluation can be aborted safely without risk of it improving in later parts of its evaluation. Since the global minimum for the underlying ANN fitness measure is zero (see Eq.(2)), the best performance of DE when optimizing ANN weights is also a meta-fitness value of zero. The algorithm for computing the meta-fitness is shown in figure 2, where the preemptive fitness limit is denoted L, and is the limit beyond which the meta-fitness evaluation can be aborted. This limit is passed as an argument by the overlaying LUS meta-optimizer, so that L is the meta-fitness of the currently best known choice of DE parameters. Depending on the experimental settings, the time-saving resulting from the use of preemptive fitness evaluation in meta-optimization, ranges from approximately 50% to 85%.

6. Experimental Results

This section gives the experimental settings and results for optimizing ANN weights using DE variants with their behavioural parameters tuned accordingly.

6.1. Optimization Settings

Five ANN datasets are used in this study and they are all taken from the Proben1 library [30]. These particular datasets are chosen because they give acceptable MSE results within a reasonable number of optimization iterations. All five datasets are classification problems and their specifications are detailed in table 1. The Cancer dataset diagnoses a breast tumour as being either benign or malignant from various microscopic measurements. The Card dataset determines whether a credit-card can be granted an applicant by a financial institute according to various data of that applicant. The Gene dataset detects intron/exon boundaries in genetic sequences from a window of 60 DNA sequence elements. The Soybean dataset recognizes diseases in soybeans according to various measurements on the soybean and the plant itself. The Thyroid dataset diagnoses performance of the thyroid from patient query data.

Table 1. ANN dataset specifications, giving for each dataset the number of inputs, the number of possible classifications, the total number of ANN weights and bias-values when using an ANN with a single hidden layer having 4 nodes, and the number of input-output pairs in the dataset.

Problem	Inputs	Classes	Weights	Data Pairs
Cancer	9	2	50	699
Card	51	2	218	690
Gene	120	3	499	3175
Soybean	35	19	427	683
Thyroid	21	3	103	7200

To lessen the effect of stochastic variation 50 optimization runs are performed on each ANN problem and the results averaged. In each of these ANN optimization runs a number of iterations are executed which equals 20 times the total number of weights and bias-values for the problem in question. For instance, as the ANN Gene problem can be seen from table 1 to have 499 weights, it means 9980 optimization iterations are performed in each run. An iteration is here taken to mean a single fitness evaluation of the MSE measure. Using these settings the ANN Cancer problem is the fastest to optimize and takes approximately 25 seconds for all 50 runs, while the ANN Gene problem takes approximately 80 minutes for all 50 optimization runs, when executed on an Intel Pentium M 1.5 GHz laptop computer using an implementation in the ANSI C programming language.

All ANN weights and bias-values are initially picked randomly and uniformly from the range $(-0.05, 0.05)$, which works well for classic BP and DE alike:

$$\forall\, i,\, j,\, k:\; w_{ijk} \sim U(-0.05, 0.05),\; b_{ij} \sim U(-0.05, 0.05)$$

Boundaries for the weights and bias-values are used to limit the search for optima. These boundaries were chosen from experiments with several different optimization methods, suggesting they are applicable in general. Although classic BP does not require such boundaries, the values chosen do not impair the performance for the class of problems considered here. The boundaries are:

$$\forall\, i,\, j,\, k:\; w_{ijk} \in [-7, 7],\; b_{ij} \in [-7, 7]$$

When an optimizing agent steps outside the boundaries of the search-space, it will be moved back to the boundary value.

6.2. Meta-Optimization Settings & Results

To fairly compare the performance of DE variants against each other, their behavioural parameters will have to be tuned to perform their best. It is recommended in [21] that experiments be done to uncover the specialization and generalization ability of an optimizer. Here, however, these experiments are combined so the DE parameters will be meta-optimized to perform well on a single ANN problem and those parameters will be used on the remainder of the ANN problems as well.

The experimental settings for meta-optimization are as follows. Six meta-optimization runs are conducted with the LUS method as the overlaying meta-optimizer trying to find the best performing behavioural parameters of DE. Each meta-optimization run has a number of iterations equalling 20 times the number of parameters to be optimized. So for basic DE which has three behavioural parameters, namely NP, CR, and F, 60 iterations will be performed for each meta-optimization run of the overlaying LUS method. DE is in turn made to perform 10 optimization runs on the ANN Cancer problem in each of these iterations, as described above. The Cancer problem is used for meta-optimization because it is the one that is fastest to compute. These meta-optimization settings were found to be adequate in locating the behavioural parameters that cause the DE variants to perform well on that ANN problem. Meta-optimization using these settings requires approximately 30 minutes of computation time per DE variant.

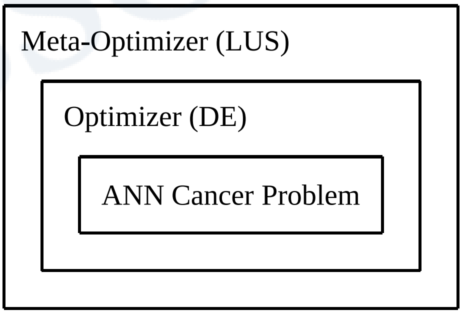


Figure 1. The concept of meta-optimization. The LUS optimization method is used as an overlaid meta-optimizer for finding good behavioural parameters of DE, which in turn is used to optimize the ANN weights for the Cancer problem.

The boundaries for the DE parameter search-spaces are shown in table 2, where it should be noted that we allow for smaller population sizes NP than usually [6] [7] [8], in part because so few optimization iterations are allowed here and each DE agent will therefore perform more iterations, but also because these have been found to be adequate, and the smaller a parameter search-space is the faster is the discovery of good parameter choices using meta-optimization.

Table 3 shows the best performing parameters for the DE/rand/1/bin variants found through meta-optimization. These parameters have abnormally small population sizes NP,

```
• Initialize the run-counter: $i \leftarrow 1$, and the fitness-sum: $s \leftarrow 0$.
• While $(i \leq M)$ and $(s < L)$, do:
    – Perform an optimization run on an ANN problem using DE with the given choice of behavioural parameters.
    – Add the best fitness obtained in the run (call it $\bar{f}$) to the fitness-sum: $s \leftarrow s + \bar{f}$.
    – Increment the run-counter: $i \leftarrow i + 1$.
• Return $s$ to the overlaying LUS meta-optimizer as the meta-fitness value of DE with the given choice of behavioural parameters.
```

Figure 2. Algorithm for performing a single meta-fitness evaluation, for rating the performance of DE using a given choice of behavioural parameters.

Table 2. Boundaries for the parameter search-spaces of DE variants as used in the meta-optimization experiments.

DE Variant	Behavioural Parameters		
Basic	$NP \in \{4, \cdots, 200\}$	$CR \in [0, 1]$	$F \in [0, 2]$
Dither	$NP \in \{4, \cdots, 200\}$	$CR \in [0, 1]$	$F_{mid} \in [0, 2]$ $F_{range} \in [0, 3]$
Jitter	$NP \in \{4, \cdots, 200\}$	$CR \in [0, 1]$	$F_{mid} \in [0, 2]$ $F_{range} \in [0, 3]$
JDE	$NP \in \{4, \cdots, 200\}$	$CR_{init} \in [0, 1]$ $CR_l \in [0, 1]$ $CR_u \in [0, 1]$ $\tau_{CR} \in [0, 1]$	$F_{init} \in [0, 2]$ $F_l \in [0, 2]$ $F_u \in [0, 2]$ $\tau_F \in [0, 1]$

compared to the advise typically given in the literature [1] [6] [8] [32], although small populations have been reported to work well on certain problems [7]. The differential weight F is also somewhat different from commonly advised, especially concerning the amount of perturbation in the Dither and Jitter variants, e.g. Jitter is recommend used with $F_{range} = 0.0005$ [32]. However, the crossover probability CR is similar to that often recommended. The JDE parameters seem to follow the same tendencies as the simpler DE variants, although their intrinsic meaning is more difficult to establish due to the increased complexity of that DE variant. The meta-optimized parameters for the DE/best/1/bin/simple variants are also shown in table 3. These parameters differ from the DE/rand/1/bin parameters in that they use larger population sizes NP, higher crossover probabilities CR, and smaller differential weights F. Deducing this from a conceptual or philosophical study of how these DE variants actually work would seem to be impossible, whereas meta-optimization allows us to automatically find good choices of behavioural parameters, without knowing the analytic cause of their good performance.

Table 3. DE parameters tuned for the ANN Cancer problem.

Variant		Behavioural Parameters		
DE/rand/1/bin	Basic	$NP = 9$	$CR = 0.804681$	$F = 0.736314$
	Dither	$NP = 9$	$CR = 0.824410$	$F_{mid} = 0.833241$ $F_{range} = 1.553993$
	Jitter	$NP = 7$	$CR = 0.916751$	$F_{mid} = 0.809399$ $F_{range} = 0.716685$
	JDE	$NP = 9$	$CR_{init} = 0.658907$ $CR_l = 0.835000$ $CR_u = 0.034128$ $\tau_{CR} = 0.774923$	$F_{init} = 0.946048$ $F_l = 0.400038$ $F_u = 0.807296$ $\tau_F = 0.242418$
DE/best/1/bin/simple	Basic	$NP = 44$	$CR = 0.967665$	$F = 0.536893$
	Dither	$NP = 47$	$CR = 0.954343$	$F_{mid} = 0.391178$ $F_{range} = 0.843602$
	Jitter	$NP = 13$	$CR = 0.933635$	$F_{mid} = 0.513652$ $F_{range} = 1.007801$
	JDE	$NP = 18$	$CR_{init} = 0.777215$ $CR_l = 0.889368$ $CR_u = 0.110632$ $\tau_{CR} = 0.846782$	$F_{init} = 1.393273$ $F_l = 0.319121$ $F_u = 0.933712$ $\tau_F = 0.619482$

6.3. Results of ANN Weight Optimization

Table 4 shows the results of using these DE variants to optimize the weights for all the ANN problems, not just the Cancer problem for which the DE parameters were specifically tuned. The results show that the DE/best/1/bin/simple variants generally have an advantage over the DE/rand/1/bin variants, but there does not appear to be a general advantage to using adaptive schemes over fixed DE parameters, as long as they have been tuned properly.

Figures 3-7 show the averaged fitness traces for the optimization runs that led to the results in table 4. What is important to note from these fitness traces, is that they are not only similar in terms of the end results, but also in terms of the rate of convergence. This is remarkable because it indicates the underlying dynamic behaviour of these DE variants are indeed similar.

Comparing the DE results to those of using classic BP in table 5 (using a hand-tuned step-size of 0.05 common in the literature, see e.g. [28]), it can be seen that DE is not quite capable of the same general performance as BP, but DE performs well enough to be a viable optimizer, for instance in the development of new kinds of ANN for which gradients have not yet been derived.

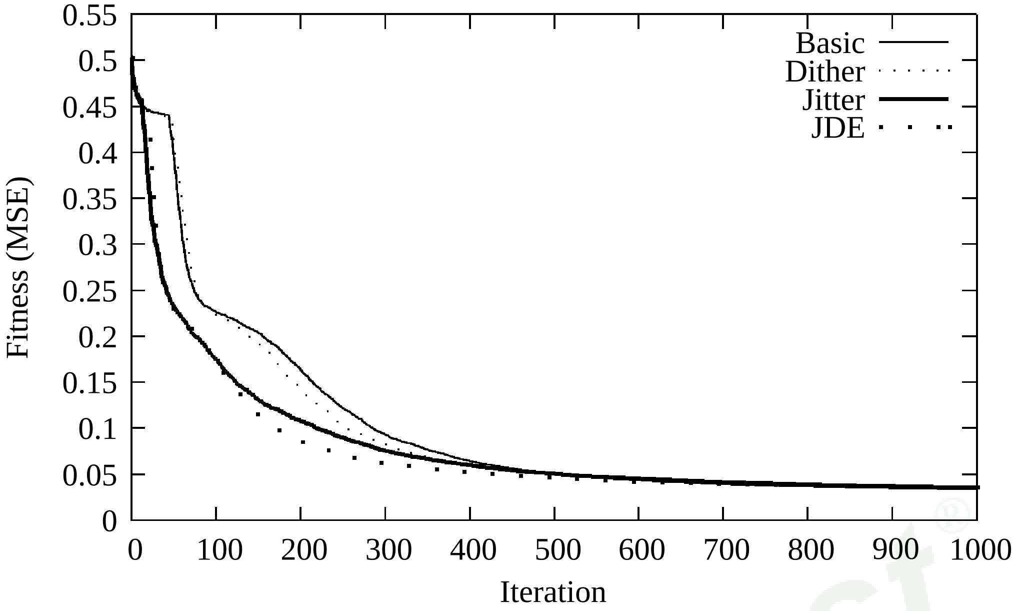


Figure 3. Fitness traces from using DE/best/1/bin/simple variants to optimize the ANN **Cancer** problem. Plot shows the average fitness obtained at each iteration of 50 optimization runs.

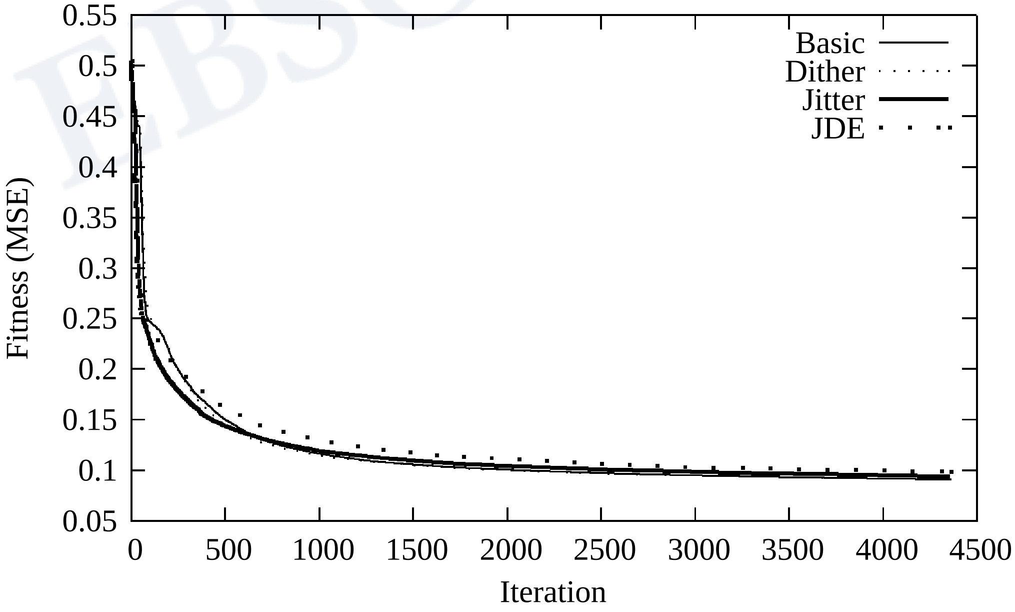


Figure 4. Fitness traces from using DE/best/1/bin/simple variants to optimize the ANN **Card** problem. Plot shows the average fitness obtained at each iteration of 50 optimization runs.

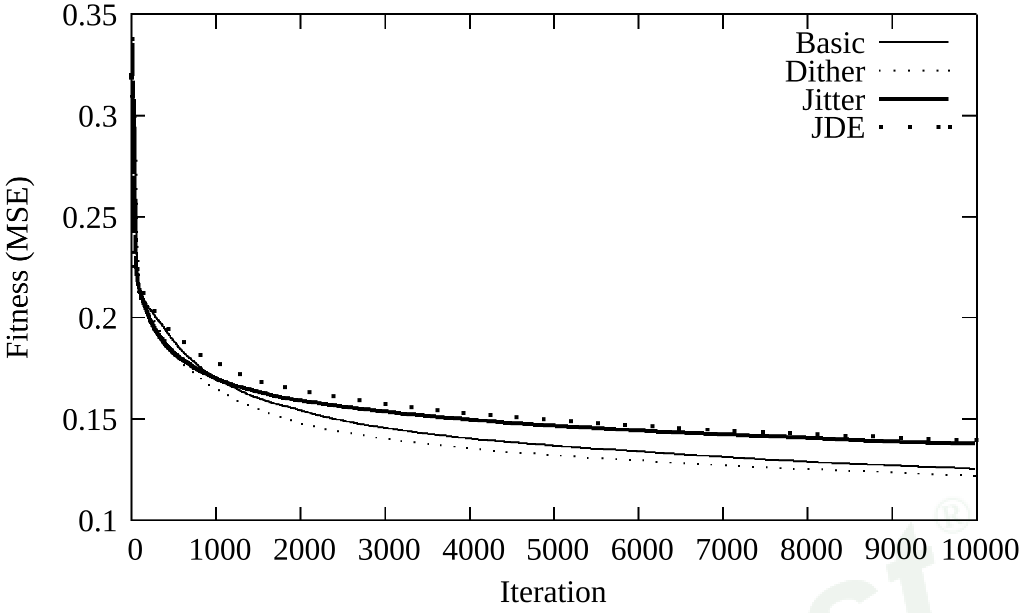


Figure 5. Fitness traces from using DE/best/1/bin/simple variants to optimize the ANN **Gene** problem. Plot shows the average fitness obtained at each iteration of 50 optimization runs.

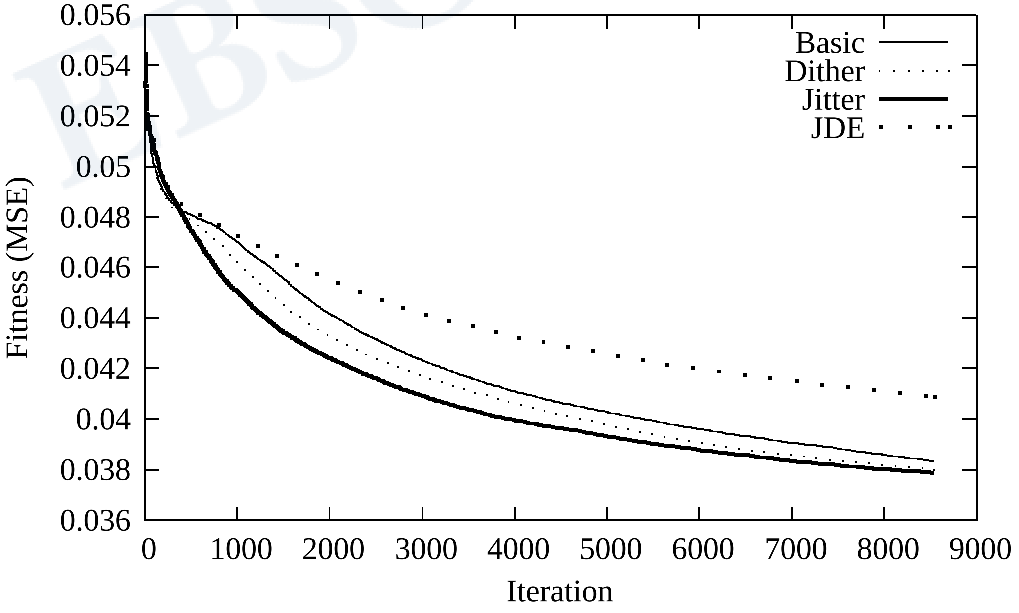


Figure 6. Fitness traces from using DE/best/1/bin/simple variants to optimize the ANN **Soybean** problem. Plot shows the average fitness obtained at each iteration of 50 optimization runs.

Table 4. Results of ANN weight optimization using eight DE variants with tuned parameters. Table shows the average MSE and CLS obtained over 50 runs. Numbers in parentheses are the standard deviations. For each ANN dataset the best results are printed in bold-face.

	Variant	DE/rand/1/bin		DE/best/1/bin/simple	
		MSE	CLS	MSE	CLS
Cancer	Basic	0.043 (0.011)	0.037 (0.007)	**0.034 (0.004)**	0.033 (0.004)
	Dither	0.046 (0.010)	0.039 (0.008)	0.035 (0.004)	0.033 (0.004)
	Jitter	0.046 (0.013)	0.040 (0.009)	0.036 (0.007)	**0.032 (0.005)**
	JDE	0.043 (0.009)	0.039 (0.007)	0.035 (0.004)	0.033 (0.004)
Card	Basic	0.097 (0.008)	0.123 (0.010)	**0.091 (0.004)**	0.118 (0.007)
	Dither	0.114 (0.017)	0.135 (0.023)	**0.091 (0.004)**	0.119 (0.007)
	Jitter	0.099 (0.011)	0.124 (0.017)	0.094 (0.011)	**0.117 (0.012)**
	JDE	0.105 (0.010)	0.127 (0.009)	0.098 (0.008)	0.122 (0.006)
Gene	Basic	0.143 (0.010)	0.288 (0.038)	0.125 (0.009)	0.236 (0.037)
	Dither	0.158 (0.010)	0.331 (0.039)	**0.122 (0.011)**	**0.222 (0.039)**
	Jitter	0.154 (0.009)	0.322 (0.037)	0.138 (0.011)	0.273 (0.051)
	JDE	0.155 (0.009)	0.334 (0.038)	0.139 (0.010)	0.281 (0.042)
Soybean	Basic	0.041 (0.001)	0.588 (0.054)	**0.038 (0.001)**	0.580 (0.052)
	Dither	0.042 (0.002)	0.651 (0.056)	0.038 (0.002)	0.578 (0.051)
	Jitter	0.040 (0.001)	0.614 (0.053)	0.038 (0.002)	**0.552 (0.054)**
	JDE	0.042 (0.001)	0.629 (0.049)	0.041 (0.001)	0.628 (0.055)
Thyroid	Basic	0.045 (0.001)	0.074 (2e-4)	**0.042 (0.001)**	**0.073 (0.002)**
	Dither	0.045 (0.001)	0.074 (2e-4)	**0.042 (0.001)**	**0.073 (0.002)**
	Jitter	0.045 (0.001)	0.074 (1e-4)	0.042 (0.002)	**0.073 (0.002)**
	JDE	0.045 (0.001)	0.074 (2e-17)	0.043 (0.001)	0.074 (0.001)

Table 5. Optimization of ANN weights using BP. Table shows the average MSE and corresponding CLS obtained over 50 runs. Numbers in parentheses are the standard deviations.

Problem	MSE	CLS
Cancer	0.034 (0.003)	0.035 (0.002)
Card	0.107 (0.002)	0.143 (0.002)
Gene	0.072 (0.004)	0.133 (0.010)
Soybean	0.030 (0.001)	0.488 (0.030)
Thyroid	0.047 (3e-5)	0.074 (3e-17)

7. Conclusion

Experiments were conducted with meta-optimizing the behavioural parameters of eight different DE variants when used for optimizing ANN weights. The DE variants ranged from

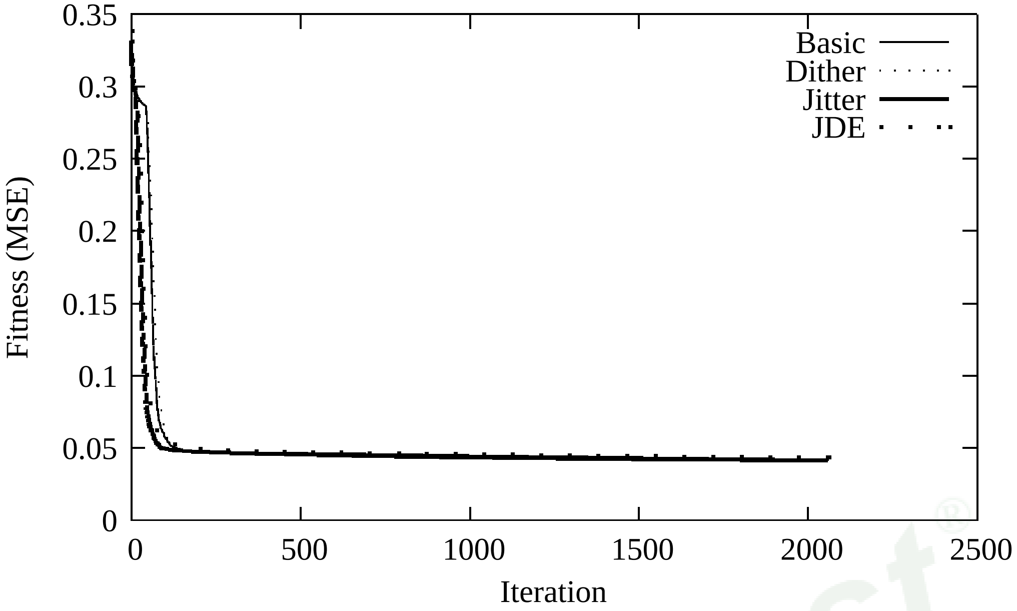


Figure 7. Fitness traces from using DE/best/1/bin/simple variants to optimize the ANN **Thyroid** problem. Plot shows the average fitness obtained at each iteration of 50 optimization runs.

the basic having fixed behavioural parameters during optimization, to DE variants employing some form of perturbation or adaptation of the parameters, designed with the intent of better

adapting and generalizing to new optimization problems. The results show that there is no general advantage to these so-called adaptive parameter schemes, when compared to the basic DE which has fixed behavioural parameters during optimization, if only those parameters are chosen properly. We suggest using an automated process to parameter tuning such as the meta-optimization technique presented, which is both simple and effective, and does not require for the practitioner to attain special knowledge about what causes good performance for the optimizer in question, as coarse parameter ranges will suffice for initializing the meta-optimization process.

8. Acknowledgments

Dr. Rainer Storn (researcher at Rohde & Schwarz, Germany) is thanked for suggestions on what DE variants to be used in this study. Professor Emeritus Rein Luus (University of Toronto, Canada) is thanked for his comments to the first draft of this paper. Associate Professor Janez Brest (University of Maribor, Slovenia) is thanked for supplying source-code for the JDE variant.

9. Source-Code

Source-code implemented in the ANSI C programming language and used in the experiments in this paper, can be found in the SwarmOps library on the internet address: `http://www.Hvass-Labs.org/`

References

[1] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341 – 359, 1997.

[2] P.J. Werbos. *Beyond Regression: new tools for prediction and analysis in the behavioural sciences*. PhD thesis, Harvard University, Cambridge, MA, 1974.

[3] P.J. Werbos. *The Roots of Backpropagation: from ordered derivatives to neural networks and political forecasting*. Wiley-Interscience, 1994.

[4] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: foundations*, pages 318–362. MIT Press, 1986.

[5] J. Ilonen, J-K. Kamarainen, and J. Lampinen. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 17(1):93–105, 2003.

[6] R. Storn. On the usage of differential evolution for function optimization. In *Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 519–523, Berkeley, CA, USA, 1996.

[7] K. Price, R. Storn, and J. Lampinen. *Differential Evolution – A Practical Approach to Global Optimization*. Springer, 2005.

[8] J. Liu and J. Lampinen. On setting the control parameter of the differential evolution method. In *Proceedings of the 8th International Conference on Soft Computing (MENDEL)*, pages 11–18, Brno, Czech Republic, 2002.

[9] D. Zaharie. Critical values for the control parameters of differential evolution algorithms. In *Proceedings of MENDEL 2002, 8th International Mendel Conference on Soft Computing*, pages 62–67, Bruno, 2002.

[10] J. Liu and J. Lampinen. A fuzzy adaptive differential evolution algorithm. *Soft Computing*, 9(6):448–462, 2005.

[11] A.K. Qin and P.N. Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *Proceedings of the IEEE congress on evolutionary computation (CEC)*, pages 1785–1791, 2005.

[12] A.K. Qin, V.L. Huang, and P.N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation, Accepted*, 2008.

[13] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark functions. *IEEE Transactions on Evolutionary Computation* , 10(6):646–657, 2006.

[14] J.J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions Systems, Man, and Cybernetics*, 16(1):122–128, 1986.

[15] T. Bäck. Parallel optimization of evolutionary algorithms. In *Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature (PPSN)*, pages 418–427, London, UK, 1994. Springer-Verlag.

[16] A.J. Keane. Genetic algorithm optimization in multi-peak problems: studies in convergence and robustness. *Artificial Intelligence in Engineering* , 9:75–83, 1995.

[17] M. Meissner, M. Schmuker, and G. Schneider. Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. *BMC Bioinformatics*, 7(125), 2006.

[18] O. François and C. Lavergne. Design of evolutionary algorithms – a statistical perspective. *IEEE Transactions on Evolutionary Computation* , 5(2):129–148, 2001.

[19] A. Czarn, C. MacNish, K. Vijayan, B. Turlach, and R. Gupta. Statistical exploratory analysis of genetic algorithms. *IEEE Transactions on Evolutionary Computation* , 8(4):405–421, 2004.

[20] V. Nannen and A.E. Eiben. A method for parameter calibration and relevance estimation in evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO)* , pages 183–190, Seattle, USA, 2006.

[21] M.E.H. Pedersen and A.J. Chipperfield. Parameter tuning versus adaptation: proof of principle study on differential evolution. Technical Report HL0802, Hvass Laboratories, 2008.

[22] S. Haykin. *Neural Networks: a comprehensive foundation* . Prentice Hall, 2nd edition, 1999.

[23] C.M. Bishop. *Neural Networks For Pattern Recognition* . Oxford University Press, 1995.

[24] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* , 2:303–314, 1989.

[25] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[26] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

[27] L.K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[28] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[29] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate bayesian a-posteriori probabilities. *Neural Computation*, 3(4):461–483, 1991.

[30] L. Prechelt. Proben1 – a set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Faculty of Informatics, University of Karlsruhe, Germany, 1994.

[31] R. Storn. Private correspondance, 2008.

[32] R. Storn. Differential evolution research – trends and open questions. In U. K. Chakraborty, editor, *Advances in Differential Evolution*, chapter 1. Springer, 2008.

[33] R. Storn. Optimization of wireless communications applications using differential evolution. In *SDR Technical Conference*, Denver, 2007.

[34] J. Brest, B. Bošković, S. Greiner, V. Žumer, and M.S. Maučec. Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft Computing*, 11:617–629, 2007.

[35] J. Matyas. Random optimization. *Automation and Remote Control*, 26(2):246–253, 1965.

[36] L.A. Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automation and Remote Control*, 24(10):1337–1342, 1963.

[37] V.A. Mutseniyeks and L.A. Rastrigin. Extremal control of continuous multi-parameter systems by the method of random search. *Engineering Cybernetics*, 1:82–90, 1964.

[38] M.A. Schumer and K. Steiglitz. Adaptive step size random search. *IEEE Transactions on Automatic Control*, 13(3):270–276, 1968.

[39] J.P. Lawrence III and K. Steiglitz. Randomized pattern search. *IEEE Transactions on Computers*, C-21(4):382–385, 1972.

[40] R. Luus and T.H.I. Jaakola. Optimization by direct search and systematic reduction of the size of search region. *American Institute of Chemical Engineers Journal (AIChE)*, 19(4):760–766, 1973.

[41] G. Schrack and M. Choit. Optimized relative step size random searches. *Mathematical Programming*, 10(1):230–244, 1976.

[42] F.J. Solis and R.J-B. Wets. Minimization by random search techniques. *Mathematics of Operation Research*, 6(1):19–30, 1981.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 19

ARTIFICIAL NEURAL NETWORKS: A CHEMIST'S PERSPECTIVE IN MODELING AND DESIGN OF HUMAN IMMUNODEFICIENCY VIRUS-1 (HIV-1) NON-NUCLEOSIDE REVERSE TRANSCRIPTASE INHIBITORS (NNRTIS)

Nitin S. Sapre*[1], Swagata Gupta[2], Nilanjana Pancholi[1] and Neelima Sapre[3]
[1]Shri G.S. Institute of Technology and Sciences, Indore, MP, India
[2]B.L.P. Govt. P.G. College, MHOW, MP, India
[3]S.V. College of Engineering, Indore, MP, India

ABSTRACT

The modern drug discovery has entered a realm wherein artificial intelligence plays a pivotal role in handling the huge overflow of the data and extracting vital information related to structure-activity/property relationships and rational drug design. In this communication, we present an overview of applications of Artificial Neural Networks (ANN) specifically in the drug discovery arena of Acquired Immunodeficiency Syndrome (AIDS) with special reference to modeling and design of Non-nucleoside Reverse Transcriptase Inhibitors (NNRTIs) of Human Immunodeficiency Virus Type-1 (HIV-1). Although both linear and non-linear techniques are appropriate in distinguishing the biologically active from the inactive compounds, ANN has consistently shown better predictive ability in case of training and test data sets both, thus can serve as an excellent tool for statistical modeling. Also, a comparison of ANN technique with linear and other non-linear techniques is portrayed in brief. Recent trends in application of ANN technique in drug development for AIDS are also discussed in the communication. The flexibility of the neural architecture serves in better understanding of the trained dataset. Since the last two decades, ANN has emerged as a tool to enhance a descriptive model,

[*] Department of Applied Chemistry, Shri G.S. Institute of Technology and Sciences, Indore, MP, India

though the higher computational cost encountered in deriving a neural network compared to linear methods might be of concern, but certainly not a limiting factor.

INTRODUCTION

"Necessity is the mother of invention." The vital force behind unravelling the mystery of nature is the human endeavour which has resulted in huge influx of information in almost every field of scientific domain. The data thus generated needs to be handled judiciously, so as to extract meaningful results out of it. Such meticulous job requires newer techniques with better reliability and accountability. Here again, the nature rules the roost. The answer lies in being adept with the adaptability to the flux of data and honing the skills for data handling in the newer context and improvising them for the future applications. One of the major queries is what are the important aspects that guide the data handling techniques or say data modeling? Is it the

- Volume of data? or
- Nature of data? or
- Applicability of the results relevant to a particular field? Or
- All of them.

"Thanks" to the advent of computing technology! The finesses of statistics, also known as statistical modeling, is being used since long back, but the silicon chip invention has revolutionised the modeling methods and techniques by reducing the computing time drastically. Though, the problem of the quantity has been tackled, hitherto, the quality of the results extracted is yet under the scanner. Here comes the role of "artificial intelligence", which is just burgeoning into a promising tool to solve the data handling job with reasonable analysis and reliability.

MATHEMATICAL METHODS IN CHEMISTRY

The modern drug discovery process has its foundation based on quantitative structure activity/property/toxicity relationship (QSAR/QSPR/QSTR) studies and its wide applicability to chemometrics, pharmacodynamics, pharmacokinetics and toxicology is well evident.[1-,4] Mathematical modelling is very crucial in these studies and often decides the fate of the model derived. Fortunately, newer and faster methods are being evolved using linear, nonlinear, statistical and machine learning methods. At the same time, the older methods are being revisited and improvised. Methods such as multiple linear regression (MLR), partial least squares (PLS), neural networks (NN), support vector machines (SVM) are some of the conventionally used methods, which are being constantly upgraded by improving the kernel algorithms or by combining them with other methods. Some of the newer methods proposed are gene expression programming (GEP), project pursuit regression (PPR), local lazy regression (LLR) etc. [5,6]

One of the most commonly used earlier methods for generating statistical models is multiple linear regression (MLR).[7] The simple and easily interpretability of mathematical expression of MLR speaks of its utility, but at times lead to ambiguity for the inter-correlated descriptors. Some of the modifications incorporated in MLR has lead to best multiple linear regression (BMLR), [8,9] genetic algorithm based multiple linear regression (GA-MLR),[10-14] stepwise MLR,[15-18] factor analysis MLR (FA-MLR),[15-18] heuristic methods (HM) etc.[19-22] In case of BMLR, computational time is enhanced with too many descriptors and is unable to build a mono-parametric model. [23] HM is a faster method and is an excellent tool for descriptor selection prior to building of a model. [24,25] GA-MLR, FA-MLR etc. are the results of fusion of different methods resulting in selection of appropriate variables, before a regression is worked out.[26,27] All the above mentioned methods have a major drawback as they are applicable to systems showing linear relationship only.

PLS has wide applicability in the case of 3D QSAR analysis such as molecular field analysis. [28,29] Like MLR, combination of PLS with other mathematical methods enhances the model's performance. Some of the methods used for such combinations are genetic partial least squares (G/PLS), factor analysis partial least squares (FA-PLS), orthogonal signal correction partial least squares (OSC-PLS) etc. G/PLS, one of the most widely used QSAR tool, uses genetic function algorithm (GFA) to derive the relevant basis functions to be used in the model and then fitted by PLS regression, thus facilitating derivation of larger QSAR equations and minimising the chances of over-fitting. [30,31,32] FA-PLS uses factor analysis to deduce relationship between the variables and thus selecting the desired variables for PLS regression.[33] In OSC-PLS, the orthogonal signal correction reduces the complexity of a model by filtering the undesired orthogonal components from the signals, thus help in deriving a more precise model.[34-36]

Support vector machine (SVM) is a newer type of machine learning method. [37] Earlier, developed for pattern recognition problems, its application to regressions in QSAR studies has been very encouraging. The various types of SVM are, least square support vector machine (LS-SVM), grid research support vector machine (GS-SVM), potential support vector machine (P-SVM) and genetic algorithm support vector machine (GA-SVM). [38-42] These are computationally simpler and are equally applicable for linear as well as non linear systems. It has been reported to be an alternative powerful tool for QSAR studies. It has the ability to perform better generalization of the dataset, compared to MLR and ANN, which implement the empirical risk minimization principle and may not converge to global solutions. [43,44]

Gene expression programming (GEP) is one of the latest chemometric tools and has evolved from genetic algorithms and genetic programming (GP). [45,46] It uses the expression of a genome in the form of expression trees and is simpler than the cellular gene progression. It has generated quite satisfactory results and can be a promising tool to express the nonlinear correlation ship of the variables with biological activity. [47-49] The major drawback in this method is the generation of complex equations and non-reproducibility of the predicted results and is of great concern.

As the name suggests, Project pursuit regression (PPR) is a beautiful blend of projection and pursuit. [50] It has evolved as a strong chemometric method for deriving reliable projections from high-dimensional data into lower dimensional space by means of linear projections based on estimation of trivariate settings. [51,52] Thus, it can tackle many shortcomings of the existing non-parametric regression analyses. With this tool, even highly complex models can be handled, as linear combinations of descriptors are modelled using general smooth functions. Also, significantly, in the case of PPR, all the interactions can be depicted graphically, thus allowing modification in the crucial parameters (smoother bandwidth and terminal threshold) of the procedure. Moreover such projections can be infinite and, thus, can reveal the intricate relationship within the model. Some of the QSAR studies have projected it as a highly reliable regression tool. [53-55]

Local lazy regression (LLR) is again a different, but an excellent approach, which retrieves a prediction by locally interpolating the neighbouring query based on distance measure, in place of considering the whole dataset. [56] Within a given dataset, there may be different subsets of molecules, which represent specific combination of similar features. Thus, instead of studying global SAR trends, as in the case of most of the conventional chemometric tools, the local SAR analysis is emphasised in LLR. Thus, this method is well-suited for large datasets, but the main focus is on the predictability and thus relevant interpretation of structure-activity cannot be achieved because of its non-global nature. The results are highly dependent on the efficiency of local neighbourhood selection and uncorrelated features may lead to wrong identification of neighbour. [57,58]

The present communication mainly deals with neural networks, therefore, in depth application of this method in computer-aided drug design (CADD) for HIV-1 will be described in detail. The intelligence-factor in Homo-Sapiens is associated with the central nervous system and more appropriately, the brain. The biological neural network comprises of a number of simple functional units called neurons, which are interconnected to form network and they process information from the receptor to the effector. On the similar guidelines, the concept of artificial neural network (ANN) has evolved wherein simple nodes or units, process the information emulating a brain. [59, 60] The assembly of the miniature units work collectively performing parallel computing. Here the more significant term is the network as the final outcome of the whole simulation depends on how these small units are linked with each other. The result also depends on their behavioural adjustments to the internal parameters, so as to adapt and modify their individual scores to some external control or nodal inter-competition. The concept of neural network was first proposed by Alan Turing in way back 1948. [61]

Neural networks (NN) process the input data and evolve hidden models of relationships and thus can deal with non-linear biological systems. For SAR studies, the most commonly used methods are, feed forward neural network (FFNN), back propagation neural network (BPNN), Kohonen neural network (KNN), counter propagation neural network (CPNN), radial basis function neural network (RBFNN) and general regression neural network (GRNN). FFNN is the most primitive type of neural network, where there is unidirectional flow of information and there are no loops in the neural architecture. The BPNN is the most widely used architecture with supervised learning method, where learning is by back propagation of errors and simultaneously the weights are corrected explicitly. The single layered KNN is based on self organised topological feature maps and unsupervised learning. Also known as self-organised mapping (SOM), KNN can be regarded as a clustering method and can be utilised to visualize the structure of a high dimensional data. CPNN is a two layered network, which is based on content-dependent data storage and retrieval method. The three layered RBFNN distribute the hidden vectors to the hidden layer, which uses radial basis function as a nonlinear function to operate on the input data. The radial basis function can be linear, cubic, spline, quadratic etc. Compared with BPNN, RBFNN requires a short training time and the global minimum of error is definitely obtained during the training and thus, the neural architecture can be conveniently implemented. It gives reliable results when the input data has relatively smaller dimensionality. [36,62-66] GRNN are the Bayesian networks, which perform regression using kernel based approximation. It is a four layered (input, hidden, summation and output) network with very fast computing speed, but drawback of this method is the requirement of a larger memory space. [67-70]

At the grass root level, the ANN methods handle the data with enormous flexibility, wherein the sole impetus lies on the architecture of the networking system. Its flexibility originates from considering varied number of neurons, the interconnectivity between them, altering the strength of the connecting nodes to have a desired signal outcome. [71] The beauty of the application of ANN methods lies in their adaptability, where, trained (supervised), untrained (unsupervised) as well as reinforced network can be applied to solve the problem. Thus, an ANN also known as simulated neural network (SNN) is an interconnected arrangement of nodes or neurons that uses a mathematical or a computational model for data processing based on connectionist approach to computation.

SIMPLE DESCRIPTION OF NEURAL NETWORK ARCHITECTURE

Simply speaking, a neuron is a regression equation having a non-linear output. Non-linear models can be fitted using more than one such neuron. The backward propagation network consists of input neurons (encoded molecular descriptors), which are multiplied by weight of each neuron. These products are summed for each neuron and a non-linear transfer function is applied. The bias is used for shifting the transfer function to the right or left side. The transformed sums are consequently multiplied to the output weights, where they are finally added, transformed and interpreted. Since, this is a supervised method, the required output (biological activity) should be known for each input vector so that the residual error (The difference between the experimental activity and network's predicted activity) can be evaluated. This calculated error propagates backward through the network, so that the weights are adjusted to observe the same input pattern and the residual error is minimized. The pattern is repeated many times till either the network cracks a relationship or no relationship is established. [72-75] Selecting the number of neurons in the hidden layer is very crucial in designing the network and the ratio rho (ρ) plays a significant role in determining the number of hidden neurons [76]
where,

$$\rho = \frac{\textit{number of datapoints in the network}}{\textit{number of connections in the network}}$$

The ratio is maintained in the range 1.0 - 3.0. If ρ < 1.0, the network simply memorizes the data, and if ρ > 3.0, the network is not able to generalize. [77]

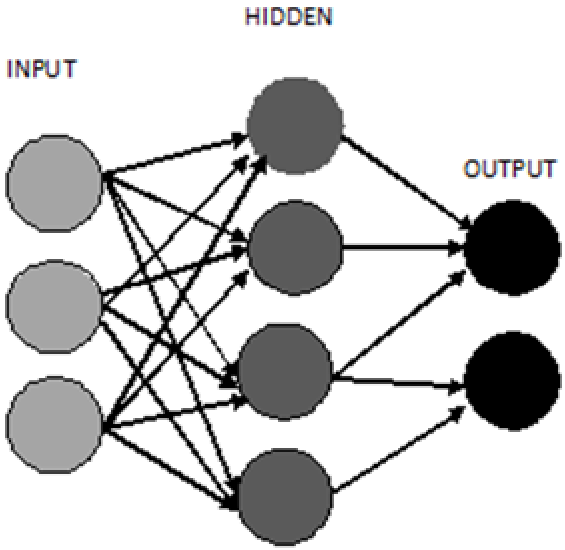


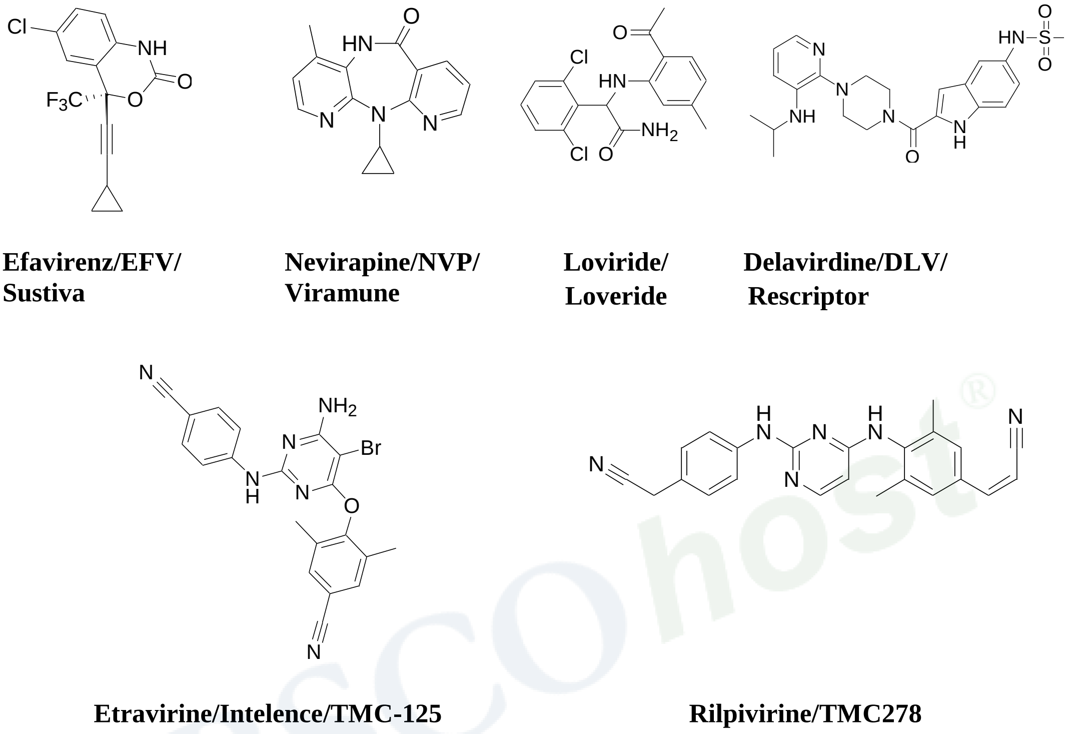
Figure 1. A simple depiction of neural network architecture showing input, hidden and output layer.

Relying injudiciously on the statistical modeling techniques can give us direction or can misdirect. Among the plethora of the modeling ways to tackle the computer generated data, the most important question is, can there be a 'golden rule' to judge the most suited model? In the next section, this is being discussed with special reference to NNRTIs and HIV-1 in general.

INTRODUCTION TO AIDS AND HIV-1

Acquired Immunodeficiency Syndrome (AIDS), is one of the most serious pandemic concern, playing havoc worldwide which has lead to an unprecedented biomedical research. [78-81] The mutagenetic agent, responsible for this disease is human immuno-deficiency virus-1 (HIV-1) belonging to a unique category of retroviruses. Since its discovery as early as in mid 1980's, efforts are being constantly made by scientists to tame the virus and crack its permanent cure. [82-85] The virus acts by progressively retarding the immune system of the body by depleting the CD4$^+$ T helper lymphocytes, leading to varied and opportunistic infections. [86,87] The major hurdle in the therapy is encountered because of the 'quasispecies' nature of the virus, thus leading to mutations and high level virologic cross-resistance. Current treatment seeks to suppress the symptoms by slowing down the viral load. [88,89] With the advent of highly active antiretroviral therapy (HAART), the "cocktail therapy" has been able to provide clinically relevant reduction of viral load, which act by targeting the multiple proteins which are essential for sustaining the various stages of life-cycle of the virus, namely, on binding to the host, transcription, integration, replication, assembly and release. [90,91]

Based on the application stage of the drug inhibitors, they are classified into (i) the reverse-transcriptase inhibitors (RTIs, which itself is classified into nucleoside reverse transcriptase (NRTIs), nucleotide reverse transcriptase inhibitors (NtRTIs) and non-nucleoside reverse transcriptase inhibitors (NNRTIs), (ii) integrase inhibitors, (iii) protease inhibitors and (iv) fusion inhibitors. [92,93]



Efavirenz/EFV/ Sustiva

Nevirapine/NVP/ Viramune

Loviride/ Loveride

Delavirdine/DLV/ Rescriptor

Etravirine/Intelence/TMC-125

Rilpivirine/TMC278

Figure 2. Some of the representative types of anti-retroviral inhibitors.

In the present communication, we limit our studies to NNRTIs. Their antiviral potency, comparatively lower toxicity and higher specificity have made them to occupy a crucial position in the HAART therapy. The binding mechanisms of NNRTIs have been dealt with elsewhere. [94,95] To qualify as NNRTIs, independent of their structure, they should specifically bind to the allosteric hydrophobic pocket, located in the p66 subunit, approximately 10Å from the polymerase binding site, at a concentration that is significantly lower than the concentration required to affect normal cell viability. [96, 97]

Application of chemometrics and molecular modeling have been a decisive factor in the journey of the NNRTIs into a new era, wherein computational chemists aided by crystallographers and medicinal chemists have lead to second-generation NNRTIs. Here comes the role of modeling techniques in generating a qualitatively more appropriate model. But, the question arises, how? Computational techniques based on quantitative structure/property-activity relationship (QSAR/QSPR) is one of the most widely approach for the prediction of biological activities in the case of drug design. [98-105] The basis of QSAR analysis lies on the assumption that the variation in the properties of the compounds can be correlated with respective change in their molecular structure that can be encoded as 'molecular descriptors'. Some of the computational techniques which have been used to derive a model from these descriptors are multiple linear regression (MLR), partial least square regression (PLS), artificial neural network (ANN), support vector machine (SVM) etc.

[71] Among these, some of the most conventionally used modeling method are MLR and PLS. The major problem encountered with these techniques is that the number of descriptors generated using computational techniques may be more than the number of the compounds in the dataset. In such a case, the correlation obtained might be a chance correlation. MLR, though, a widely used technique, but is inadequate to explain a complex biological system, as it assumes linear relationship between biological activity and the descriptors. It has been observed that the flexibility associated with ANN helps to discover the hidden more complex non linear behaviour in the experimental data, thus presenting a better model. Owing to the aforementioned reasons, there is a growing interest among the computational scientists in the application of ANN in the field of QSAR.

COMPARISON WITH OTHER LINEAR AND NON-LINEAR TECHNIQUES IN CASE OF NNRTIS

Very less work has been reported on application of ANN to the NNRTIs. Also, most of the reported work on NNRTIs has been limited to 4,5,6,7-Tetrahydro-5-methylimidazo[4,5,1-jk][1,4]benzodiazepin-2(1H)-ones (TIBO) and 1-[2-hydroxyethoxy) methyl]-6-(phenylthio)thymine (HEPT) derivatives.[106-112]

With conventional modelling techniques, though widely used, the major problem encountered is of multicollinearity. Though, PCA helps in tackling the problem of multi-collinearity (encoding of essentially the same information repetitively) while analysing molecular symmetry by creating completely orthogonal axes from the original attributes, but the main drawback remains with the principal components (PCs) constructed in such a way is that, the attributes thus selected may or may not be relevant to the property of interest. [113-116]

Tackling non-linearity of the dataset is a Herculean task. Usually, there seems to be substantial improvement in the modelling statistics while using neural network approach which provides the evidence for the nonlinear relationship between structure and biological activity. [71]

A QSAR analysis based on feed forward three layer neural network with four descriptors characterizing the electronic, steric and hydrophobic aspect of TIBO derivatives by Douali and co-workers demonstrated ANN to be better performer than MLR and as expected the robust model could well depict the role of hydrophobicity as in the case of NNRTIs. [106]

Freitas et al. has reported the application of multivariate image analysis applied to quantitative structure-activity relationship (MIA-QSAR) method coupled to principle component analysis-adaptive neuro-fuzzy inference systems (PCA-ANFIS) to predict the non-linearities of the system with reference to TIBOs. [117] The proposed neuro-fuzzy model has a multilayer neural network-based fuzzy system. The proposed model gave a highly predictive model which accounted for non-linear behaviour not considered by MLR and PLS. The results obtained were found to be better than that of ANN, [18] thus explaining the fact that the given dataset required non-linear local modeling than non-linear global modeling.

A comparative QSAR study carried out on TIBO derivatives using topological, structural, physicochemical, electronic and spatial descriptors was modelled with multiple regression (stepwise regression, FA-MLR, GFA-MLR, PLS, FA-PLS) as linear methods and GFA (spline) and ANN as nonlinear methods. [18] When all the descriptors were used in combination, the nonlinear methods were superior and among them GFA (spline) performed better than ANN. But when other different combinations of descriptors were used, the results of mathematical modeling varied, viz, with topological and structural descriptors, FA-PLS was the best in the case of external validation. Similarly, for the combination of structural, physicochemical, electronic and spatial descriptors, GFA (spline) showed best results.

A computational modelling approach on the TIBO derivatives using Kier-Hall electrotopological state indices by Sapre et al. [107] using MLR (LOO-CV and N-CV) gave the best result (r^2= 0.913, MSE= 0.190). When the same dataset was subjected to ANN using back propagation method, the results were improved (r^2= 0.938, MSE=0.141), thus reaffirming the non-linear phenomenon involved in relating descriptors with biological activity.

The utility of ANN is not just limited to generation of a robust and predictive mathematical model, but also has been extended as virtual screening tools aimed at exploration of bio-active molecules in chemical databases and virtual chemical libraries. [118,119] A 3D-QSAR study using group center overlap approach on TIBO derivatives by the same authors reported comparison of MLR and ANN and the applicability of the derived model for datamining novel hits from large chemical databases. [108] Manual selection method, LOO cross-validation, N-cross-validation, feature selection method using forward selection, backward elimination and simulated annealing was used for MLR as well as ANN. With the exception of LOO-CV, in all the cases, ANN outperformed MLR, thus suggesting the overall robustness of the ANN models. The cross-validation using test set also gave satisfactory results. Further, the robust model obtained has been used as a filter for screening novel and modified compounds with enhanced anti-viral activity from an external chemical database. 16 hits could be retrieved from the database that may be optimised for potential therapeutic utility. Use of ANN could be successfully implemented for assessing the ligand efficiencies using template based docking and tabu-clustering for the TIBO derivatives.[120] In this case too, ANN (r^2= 0.922) was better than MLR (r^2=0.851), suggesting the robustness of the template-based binding conformations of the inhibitors.

While performing QSAR studies on indolyl aryl sulphones, using physicochemical, topological, structural parameters and appropriate indicator variables and subsequently modelling them with linear (stepwise regression analysis, PLS, FA-MLR, GFA-MLR, GFA/PLS) as well as non linear (ANN) tools and externally validating them, Roy and Mandal have reported that the outcome of the ANN method (r^2=0.781) was superior to all the other linear methods. Among the linear methods, GFA-MLR (r^2=0.736) was the best. [16]

HEPT derivatives are one of the extensively studied NNRTIs. Heravi and Paraster modelled a dataset of eighty HEPTs using MLR and ANN and compared their results with earlier workers. [109,110] They used 6-6-1 neural network and could get lower MSE for ANN (0.073) compared to MLR (0.170) and Luco- Ferratti's model (0.084). They also tried to verify the descriptors used by Luco and Ferratti and developed an ANN model, but the model started overfitting after 7000 iterations (In contrast to 18000 iterations for the present ANN model), which suggests that all the descriptors were not able to show all the features of the inhibitory properties of the HEPT derivatives.

Mager modified ANN analysis by hybrid canonical-correlation neural-network approach applied to the HEPTs so as to minimise the danger of over fitting. [111] The results were more reliable as the dimensionality of the two subsets (biological and chemical variables) was reduced. Subsequently the two subsets were used as output and input respectively in an optimized back propagation neural network approach. This approach not only yielded a robust model, but also predicted the noncovalent interactions responsible for the antiviral and cytotoxic actions at the binding site in a consistent way.

The nonlinear aspects of structure-anti-HIV activity are clearly implicated in case of the HEPTs. The NNs outperforms the MLR. [117,121,122] Six NN architectures of 7-x-1 (x=3-8) were trained and all of them were found to better than the MLR. [122] The number of descriptors should be selected as such to encompass all the structural variations crucial for antiviral activity. In the case of HEPTs, the ANN model could successfully register the importance of the steric and hydrophobic contribution to the antiviral activity. [122] Variation of network output with variation in 1/steric, [76, 77] while keeping the outputs constant, could very well demonstrate the favourable steric interaction of moderately sized molecules within the NNIBP. Similarly, a parabolic dependency of logP was observed with antiviral activity, thus showing an optimum hydrophobicity of 4.32, which is quite in good agreement with the hydrophobicity range (2.31 – 5.72) as predicted by Garg et al. for the NNRTIs. [98] It is now well evident that hydrophobicity is not only a crucial factor involving protein-ligand interactions, but also has important function in drug transport and cell permeability.

In a study by us involving K.H. electrotopological indices for fifty five HEPT derivatives using MLR and ANN, ANN r^2= 0.901) outweighed the MLR (r^2=0.894). The results were complemented with LOO-CV as well as N-CV and test set. The ANN model was robust enough to data mine PubChem database and was able to retrieve 14 hit compounds with better predicted anti-viral activity [unpublished results].

We have also performed ANN studies based on back propagation method on NNRTIs such as 2-Amino-6-arylsulfonylbenzonitrile (AASBN) and 5-Alkyl-2-alkylamino-6-(2, 6-difluorophenylalkyl)-3, 4-dihydropyrimidin-4(3H)-one (DABO), which is being presented here [Unpublished studies]. In all the cases, over fitting parameter was taken care of so as to have minimum chance correlation possibility.

A 2D QSAR study using topological indices (5 descriptors), performed on forty AASBN analogs also critically showed the superiority of ANN (r^2= 0.911, MSE= 0.147) over MLR (r^2=0.801, MSE=0.269). The results were complimented by the test set as well (ANN: r^2= 0.759; MLR: r^2=0.657). A 2D QSAR study involving eleven K. H. E-state indices and fifty six AASBN also showed a similar behaviour, where the best MLR model (r^2= 0.820) trailed behind the ANN model (r^2= 0.903). The test set too reciprocated the same behaviour (ANN: r^2= 0.809; MLR: r^2=0.750). A 3D QSAR study on AASBN analogs using five 3D descriptors based on group center overlap templates of steric (S_{ALL}) , hydrogen donor (HD_{ALL}), hydrogen acceptor (HA_{ALL}), positive charge (POS_{ALL}) and ring (R_{ALL}) for the training set of fifty seven AASBN derivatives was also performed and comparison of ANN was made with MLR. The outcome of the analysis was as expected, wherein ANN (r^2= 0.880) was relatively better than the MLR (r^2= 0.872). Similar trait was observed in the case of test set as well (ANN: r^2= 0.761; MLR: r^2=0.749).

Similarly, DABO analogs were subjected to 2D QSAR using topological indices, using K. H. Electrotopological indices and 3D QSAR using four 3D descriptors namely, steric(S_{ALL}) , hydrogen donor(HD_{ALL}), hydrogen acceptor(HA_{ALL}), ring(R_{ALL}) and ClogP. In the case of 2D QSAR using topological indices, ANN (r^2= 0.885, MSE= 0.096) was found to be superior compared to MLR (r^2= 0.830, MSE= 0.141) for the training set. The test set too behaved in the same way (ANN: r^2= 0.680; MLR: r^2=0.670). In the case of 2D QSAR using seven E-state indices, in the comparative analyses, ANN (r^2= 0.920, MSE= 0.155) outperformed MLR (r^2= 0.871, MSE= 0.214) for the training as well as test set (ANN: r^2= 0.843; MLR: r^2=0.796). In the case of 3D QSAR, as expected, better performance was observed in the case of ANN as compared to MLR in the case of training (MLR: r^2= 0.805, MSE= 0.245; ANN: r^2= 0.939, MSE= 0.077) as well as test set (MLR: r^2= 0.745; ANN: r^2= 0.843). comparison of 3D QSAR studies of AASBN and DABO inhibitors reveal an interesting fact that in case of AASBN inhibitors, the difference between the cross-validated r^2 of ANN and MLR is very small in the case of training as well as the test set, whereas the difference is appreciable in case of the DABO compounds. The reason being obvious that incorporation of ClogP along with 3D variables in the case of DABOs has lead to non-linearity, which was well discriminated by the ANN model compared to MLR.

ANN has been also successfully implemented in the prediction of phenotypic susceptibility to the antiretroviral drugs using physicochemical properties of the primary enzymatic structure (amino-acid sequences. [123] For the said purpose, an ANN trained with 10-fold internal cross-validation was used. The correlation coefficients obtained for the NNRTIs (efavirenz and nevirapine) was satisfactory, thus suggesting the applicability of ANN for predicting the mutations.

COMPARISON WITH OTHER LINEAR AND NON-LINEAR TECHNIQUES IN CASE OF HIV

Here it will be worth mentioning, a brief comparision of some of the recent advances in molecular modeling techniques and their comparision in the arena of HIV research (other than NNRTIs) as a whole. In the QSAR studies performed on CCR1, three different methods, namely, linear (PCR) and non-linear (PC-ANN and PC-LS-SVM) were studied and compared, wherein non-linear methods showed better predictability and among the two, the SVM model was better. [124] The non-linearity was considered using RBF kernel function.

A recent QSAR study based on CCR5 binding affinity has suggested that the use of gene expression programming (GEP) for selection of variables and deriving simultaneous non-linear model can be more convenient than ANN and SVM. [125] Though ANN projected better result (r^2= 0.9714) than SVM and GEP (r^2= 0.9550 and 0.9557 respectively), but GEP implements a simple and convenient method of variable selection by inferring the K-expression from the number of arguments of the function in a gene, without building the expression tree. The “black box” nature of ANN does not give a complete expression or representation of the behavioural pattern in a simple interpretable manner. Comparatively, SVM serves as a popular approach in handling the high dimensionality of a particular data structure. In case of SVM, the major drawback is that its performance can be ruined by inefficient variable selection.

The advent of X-ray crystallography has enabled experimental determination of 3D structural information of the protein –ligand complexes and in turn has facilitated the use of more complicated and powerful machine learning algorithms using ligand-based molecular fingerprints and descriptors. For the above applications, accurate and versatile classification models could be generated by state-of-the-art non-linear methods like artificial neural network (ANN), Random forest (RF), support vector machine (SVM), naïve Bayesian classifier (NBC) and linear discriminate analysis (LDA). [126] These four machine learning algorithms were used to study the screening efficiencies of pharmacophore-based interaction fingerprint (Pharm-IF) scores, residue base fingerprint (PLIF) scores and GLIDE scores. SVM, ANN and RF methods were better at creating learning models even with few numbers of crystal structures available for the target protein. The learning effects of SVM were the best with any size of the training set. Comparatively, the performances of the ANN model improved slowly, as it strongly depended on the first choice of the learning samples. NBC did not behave as expected, as NBC treats all the variables independently and builds probabilistic models and cannot learn the complicated relationship and interactions between the variables. The diverse protein-ligand interaction pattern analysis by incorporating the docking poses of known active compounds along with the crystal structures showed significant learning effects in the case of RF models with substantial enhancement in the enrichment factor whereas SVM did not show any significant learning effect. As docking poses can have incorrect binding modes, it affects the sensitivity of the SVM method, whereas RF method is a robust statistical method against data containing noise. [126]

The neural networks have some draw back inherent in its architecture such as over training, over-fitting and network optimization. Owing to random initialization of the networks and variation of stopping criteria, reproducibility of the results is a major concern. [127] Interpretation of the obtained models on the basis of the descriptors is difficult as there is no such final equation similar to that of regression analysis. [124] There are no exact principles for working out an exact topology of the network, thus it is necessary to optimise the adjustable parameters f the network.

CONCLUSIONS

In the current scenario, the drug discovery has evolved into a phase, where the main onus lies on rigorous model validation based on robust and predictive analyses. Thus, researchers face a substantial challenge of choosing appropriate quantitative and qualitative methods for performing the analyses. QSAR based on "structure-property similarity principle" suggest that the user-defined or arbitrary molecular similarity methods perform reasonably well in narrow, well-defined situations, but the relationships between structural attributes and bioactive/toxicological properties are not always clearly defined. In a multivariate space, to handle objectively defined relationships, a robust statistical method is required. Though inferior to method such as SVM, ANN performs better than most of the currently used methods. The applicability of ANN is crucial when large numbers of observations accumulate and the problem is not well enough understood to derive a procedural programme. ANN has been able to provide solutions where the other conventional statistical modelling techniques reach limitation, especially, when non-linearity of a dataset is considered. This communication has been aimed at throwing light on various modelling methods currently being used and their comparison with special reference to the wide applicability of the neural networks. With the present limited knowledge of the ANN methods in the case of NNRTIs, more practical applications are to be undertaken so as to improvise the theoretical aspects of neural architecture and have a better application to extract maximum amount of information as possible, thus enhancing the "computer intelligence".

ACKNOWLEDGMENTS

The authors wish to acknowledge Dr. PK Sen, Director, SGSITS, Indore and Dr. RK Tugnawat, Principal, BLPPGC, MHOW for their encouragement in pursuing this work.

REFERENCES

[1] Du, H.; Wang, J.; Hu, Z.; Yao, X. Quantitative Structure-Retention relationship study of the constituents of saffron aroma in SPME-GC-MS based on the projection pursuit regression method. *Talanta* 2008, *77*, 360-365.

[2] Xia, B.; Liu, K.; Gong, Z.; Zheng, B.; Zhang, X.; Fan, B. Rapid toxicity prediction of organic chemicals to Chlorella vulgaris using quantitative structure-activity relationships methods. *Ecotoxicol. Environ. Saf.* 2009, *72*, 787-794.

[3] Sapre, N. S.; Gupta, S.; Pancholi, N.; Sikarwar, A.; Sapre, N., 2-Dimensional Quantitative Structure-Activity Relationship Modeling Study of Glycine/N-methyl-D-aspartate Antagonist Inhibition: Genetic Function Approximation vis-a-vis Multiple Linear Regression Methods, *Acta. Chim. Slov*. 2007, *54*, 797-804.

[4] Srivani, P.; Srinivas, E.; Raghu, R.; Sastry, G.N. Molecular modeling studies of pyridopurinone derivatives--potential phosphodiesterase 5 inhibitors. *J. Mol. Graph. Model.* 2007, *26*, 378-390.

[5] Equbal, T.; Silakari, O.; Ravikumar, M. Exploring three-dimensional quantitative structural activity relationship (3D-QSAR) analysis of SCH 66336 (Sarasar) analogues of farnesyltransferase inhibitors. *Eur. J. Med. Chem.* 2008, *43*, 204-209.

[6] Bhonsle, J.B.; Bhattacharjee, A.K.; Gupta, R.K. Novel semi-automated methodology for developing highly predictive QSAR models: application for development of QSAR models for insect repellent amides. *J. Mol. Model.* 2007, *13*, 179-208.

[7] Li, J.; Li, S.; Lei, B.; Liu, H.; Yao, X.; Liu, M.; Gramatica, P.; A new strategy to improve the predictive ability of the local lazy regression and its application to the QSAR study of melanin-concentrating hormone receptor 1 antagonists. *J Comput Chem.* 2010, *15;31(5)*,973-985.

[8] Kahn, I.; Sild, S.; Maran, U. Modeling the toxicity of chemicals to Tetrahymena pyriformis using heuristic multilinear regression and heuristic back-propagation neural networks. *J. Chem. Inf. Model.* 2007, *47*, 2271-2279.

[9] Liu, P.; Long, W.; Current Mathematical Methods Used in QSAR/QSPR Studies. *Int J Mol Sci.* 2009 *10(5)*, 1978–1998.

[10] Gharagheizi, F.; Tirandazi, B.; Barzin, R. Estimation of aniline point temperature of pure hydrocarbons: A quantitative structure-property relationship approach. *Ind. Eng. Chem. Res.* 2009, *48*, 1678-1682.

[11] Gharagheizi, F.; Mehrpooya, M. Prediction of some important physical properties of sulfur compounds using quantitative structure-properties relationships. *Mol. Div.* 2008, *12*, 143-155.

[12] Om, A.S.; Ryu, J.C.; Kim, J.H. Quantitative structure-activity relationships for radical scavenging activities of flavonoid compounds by GA-MLR technique. *Mol. Cell. Toxicol.* 2008, *4*, 170-176.

[13] Riahi, S.; Ganjali, M. R. ; Norouzi, P.; Jafari, F. Application of GA-MLR, GA-PLS and the DFT quantum mechanical (QM) calculations for the prediction of the selectivity coefficients of a histamine-selective electrode. *Sensors and Actuators B: Chemical,* 2008, *148*, 13-19.

[14] Fisz, J.J. Combined genetic algorithm and multiple linear regression (GA-MLR) optimizer: Application to multi-exponential fluorescence decay surface. *J. Phys. Chem. A* 2006, *110*, 12977-12985.

[15] Roy K.; Roy P. P. Comparative chemometric modeling of cytochrome 3A4 inhibitory activity of structurally diverse compounds using stepwise MLR, FA-MLR, PLS, GFA, G/PLS and ANN techniques. Eur. *J. Med. Chem.* 2009, *44(7)*, 2913-22.

[16] Roy K.; Mandal A. S. Predictive QSAR modeling of CCR5 antagonist piperidine derivatives using chemometric tools. *J. Enzyme Inhib. Med. Chem.* 2008, *24*, 205–223.

[17] Roy K.; Mandal A. S. Development of linear and nonlinear predictive QSAR models and their external validation using molecular similarity principle for anti-HIV indolyl aryl sulfones. *Enzyme Inhib. Med. Chem.* 2008, *23(6)*, 980-95.

[18] Roy K.; Mandal A. S. Predictive QSAR modeling of HIV reverse transcriptase inhibitor TIBO derivatives. Eur. *J. Med. Chem.* 2009, *44(4)*, 1509-24.

[19] Xia, B.; Ma, W.; Zheng, B.; Zhang, X.; Fan, B. Quantitative structure-activity relationship studies of a series of non-benzodiazepine structural ligands binding to benzodiazepine receptor. *Eur. J. Med. Chem.* 2008, *43*, 1489-1498.

[20] Liu, H.; Yao, X.; Zhang, R.; Liu, M.; Hu' Z.; Fan, B. The accurate QSPR models to predict the bioconcentration factors of nonionic organic compounds based on the heuristic method and support vector machine. *Chemosphere* 2006, *63*, 722-733.

[21] Lu, W.J.; Chen, Y.L.; Ma, W.P.; Zhang, X.Y.; Luan, F.; Liu, M.C.; Chen, X.G.; Hu, Z.D. QSAR study of neuraminidase inhibitors based on heuristic method and radial basis function network. *Eur. J. Med. Chem.* 2008, *43*, 569-576.

[22] Li, X.; Luan, F.; Si, H.; Hu, Z.; Liu, M. Prediction of retention times for a large set of pesticides or toxicants based on support vector machine and the heuristic method. *Toxicol. Lett.* 2007, *175*, 136-144.

[23] Katritzky, A.R.; Pacureanu, L.; Dobchev, D.; Karelson, M. QSPR modeling of hyperpolarizabilities. *J. Mol. Model.* 2007, *13*, 951-963.

[24] Qin, S.; Liu, H.X.; Wang, J.; Yao, X.J.; Liu, M.C.; Hu, Z.D.; Fan, B.T. Quantitative Structure- Activity Relationship study on a series of novel ligands binding to central benzodiazepine receptor by using the combination of Heuristic Method and Support Vector Machines. *QSAR Comb. Sci.* 2007, *26*, 443-451.

[25] Si, H.Z.; Yao, X.J.; Liu, H.X.; Wang, J.; Li, J.Z.; Hu, Z.D.; Liu, M.C. Prediction of binding rate of drug to human plasma protein based on heuristic method and support vector machine. *Acta Chim. Sinica* 2006, *64*, 415-422.

[26] Gharagheizi, F. QSPR studies for solubility parameter by means of Genetic Algorithm-Based Multivariate Linear Regression and generalized regression neural network. *QSAR Comb. Sci.* 2008, *27*, 165-170.

[27] Elliott, G.N.; Worgan, H.; Broadhurst, D.; Draper, J.; Scullion, J. Soil differentiation using fingerprint Fourier transform infrared spectroscopy, chemometrics and genetic algorithm-based feature selection. *Soil Biol. Biochem.* 2007, *39*, 2888-2896.

[28] Rogers, D.; Hopfinger, A.J. Application of genetic function approximation to quantitative structure-activity-relationships and quantitative structure-property relationships. *J. Chem. Inf. Comput. Sci.* 1994, *34*, 854-866.

[29] Sammi, T.; Silakari, O.; Ravikumar, M. Three-dimensional quantitative structure-activity relationship (3D-QSAR) studies of various benzodiazepine analogues of gamma-secretase inhibitors. *J. Mol. Model.* 2009, *15*, 343-348.

[30] Li, Z.G.; Chen, K.X.; Xie, H.Y.; Gao, J.R. Quantitative structure - activity relationship analysis of some thiourea derivatives with activities against HIV-1 (IIIB). *QSAR Comb. Sci.* 2009, *28*, 89- 97.

[31] Samee, W.; Nunthanavanit, P.; Ungwitayatorn, J. 3D-QSAR investigation of synthetic antioxidant chromone derivatives by molecular field analysis. *Int. J. Mol. Sci.* 2008, *9*, 235-246.

[32] Roy, K.; Leonard, J.T. Topological QSAR modeling of cytotoxicity data of anti-HIV 5-phenyl-1- phenylamino-imidazole derivatives using GFA, G/PLS, FA and PCRA techniques. *Indian J. Chem. Sect. A-Inorg. Bio-Inorg. Phys. Theor. Anal. Chem.* 2006, *45*, 126-137.

[33] Leonard, J.T.; Roy, K. Comparative QSAR modeling of CCR5 receptor binding affinity of substituted 1-(3,3-diphenylpropyl)-piperidinyl amides and ureas. *Bioorg. Med. Chem. Lett.* 2006, *16*, 4467-4474.

[34] Wold, S.; Antti, H.; Lindgren, F.; Ohman, J. Orthogonal signal correction of near-infrared spectra. *Chemometr. Intel. Lab. Syst.* 1998, *44*, 175-185.

[35] Liang, G.Z.; Mei, H.; Zhou, Y.; Yang, S.B.; Wu, S.R.; Li, Z.L. Using SZOTT descriptors for the development of QSAMs of peptides. *Chem. J. Chinese Univ.* 2006, *27*, 1900-1902.

[36] Yang, S.B.; Xia, Z.N.; Shu, M.; Mei, H.; Lue, F.L.; Zhang, M.; Wu, Y.Q.; Li, Z.L. VHSEH Descriptors for the Development of QSAMs of Peptides. *Chem. J. Chinese Univ.* 2008, *29*, 2213- 2217.

[37] Du, H.Y.; Wang, J.; Hu, Z.D.; Yao, X.J.; Zhang, X.Y. Prediction of fungicidal activities of rice blast disease based on least-squares support vector machines and project pursuit regression. *J. Agric. Food Chem.* 2008, *56*, 10785-10792.

[38] Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* 1995, *20*, 273-297.

[39] Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* 1999, *9*, 293-300.

[40] Niazi, A.; Jameh-Bozorghi, S.; Nori-Shargh, D. Prediction of toxicity of nitrobenzenes using ab initio and least squares support vector machines. *J. Hazard. Mater.* 2008, *151*, 603-609.

[41] Liu, H.X.; Yao, X.J.; Zhang, R.S.; Liu, M.C.; Hu, Z.D.; Fan, B.T. Prediction of the tissue/blood partition coefficients of organic compounds based on the molecular structure using least-squares support vector machines. *J. Comput. Aided Mol. Des.* 2005, *19*, 499-508.

[42] Pourbasheer, E.; Riahi, S.; Ganjali, M. R.; Norouzi, P. Application of genetic algorithm-support vector machine (GA-SVM) for prediction of BK-channels activity. *Eur. J. Med. Chem.* 2009, *44*, 5023-5028.

[43] Yao X. J.; Panaye A.; Doucet J. P.; Zhang R. S.; Chen H. F.; Liu M. C.; Hu Z. D.; Fan B. T. Comparative study of QSAR/QSPR correlations using support vector machines,

radial basis function neural networks, and multiple linear regression. *J. Chem. Inf. Comput. Sci.* 2004, *44(4)*, 1257-66.

[44] Zhao C. Y.; Zhang H. X.; Zhang X. Y.; Liu M. C.; Hu Z. D. Fan B. T. Application of support vector machine (SVM) for prediction toxic activity of different data sets. *Toxicology*. 2006, *217(2-3)*, 105-19.

[45] Si, H.Z.; Zhang, K.J.; Hu, Z.D.; Fan, B.T. QSAR model for prediction capacity factor of molecular imprinting polymer based on gene expression programming. *QSAR Comb. Sci.* 2007, *26*, 41-50.

[46] Si, H.Z.; Yuan, S.P.; Zhang, K.J.; Fu, A.P.; Duan, Y.B.; Hue, Z.D. Quantitative structure activity relationship study on EC5.0 of anti-HIV drugs. *Chemometr. Intel. Lab. Syst.* 2008, *90*, 15-24.

[47] Wang, T.; Si, H.Z.; Chen, P.P.; Zhang, K.J.; Yao, X.J. QSAR models for the dermal penetration of polycyclic aromatic hydrocarbons based on Gene Expression Programming. *QSAR Comb. Sci.* 2008, *27*, 913-921.

[48] Luan, F.; Si, H.Z.; Liu, H.T.; Wen, Y.Y.; Zhang, X.Y. Prediction of atmospheric degradation data for POPs by gene expression programming. *SAR QSAR Environ. Res.* 2008, *19*, 465-479.

[49] Si, H.Z.; Wang, T.; Zhang, K.J.; Hu, Z.D.; Fan, B.T. QSAR study of 1,4-dihydropyridine calcium channel antagonists based on gene expression programming. *Bioorg. Med. Chem.* 2006, *14*, 4834-4841.

[50] Friedman, J.H.; Stuetzle, W. Projection Pursuit Regression. *J. Am. Stat. Assoc.* 1981, *76*, 817- 823.

[51] Yuan, Y.N.; Zhang, R.S.; Hu, R.J. Prediction of Photolysis of PCDD/Fs Adsorbed to Spruce [Picea abies (L.) Karst.] Needle Surfaces Under Sunlight Irradiation Based on Projection Pursuit Regression. *QSAR Comb. Sci.* 2009, *28*, 155-162.

[52] Du, H.Y.; Zhang, X.Y.; Wang, X.; Yao, X.J.; Hu, Z.D. Novel approaches to predict the retention of histidine-containing peptides in immobilized metal-affinity chromatography. *Proteomics* 2008, *8*, 2185-2195.

[53] Du, H.Y.; Wang, J.; Zhang, X.Y.; Hu, Z.D. A novel quantitative structure-activity relationship method to predict the affinities of MT3 melatonin binding site. *Eur. J. Med. Chem.* 2008, *43*, 2861-2869.

[54] Ren, Y.Y.; Liu, H.X.; Li, S.Y.; Yao, X.J.; Liu, M.C. Prediction of binding affinities to beta(1) isoform of human thyroid hormone receptor by genetic algorithm and projection pursuit regression. *Bioorg. Med. Chem. Lett.* 2007, *17*, 2474-2482.

[55] Yuan, Y.N.; Zhang, R.S.; Hu, R.J.; Ruan, X.F. Prediction of CCR5 receptor binding affinity of substituted 1-(3,3-diphenylpropyl)-piperidinyl amides and ureas based on the heuristic method, support vector machine and projection pursuit regression. *Eur. J. Med. Chem.* 2009, *44*, 25-34.

[56] Gunturi, S.B.; Archana, K.; Khandelwal, A.; Narayanan, R. Prediction of hERG Potassium Channel Blockade Using kNN-QSAR and Local Lazy Regression Methods. *QSAR Comb. Sci.* 2008, *27*, 1305-1317.

[57] Guha, R.; Dutta, D.; Jurs, P.C.; Chen, T. Local lazy regression: Making use of the neighborhood to improve QSAR predictions. *J. Chem. Inf. Model.* 2006, *46*, 1836-1847.

[58] Du, H.Y.; Watzl, J.; Wang, J.; Zhang, X.Y.; Yaol, X.J.; Hu, Z.D. Prediction of retention indices of drugs based on immobilized artificial membrane chromatography using

Projection Pursuit Regression and Local Lazy Regression. *J. Sep. Sci.* 2008, *31*, 2325-2333.

[59] Frank, M. J. Dynamic Dopamine Modulation in the Basal Ganglia: A Neurocomputational Account of Cognitive Deficits in Medicated and Non-medicated Parkinsonism. *Journal of Cognitive Neuroscience* 2005, *17*, 51–72.

[60] Egmont-Petersen, M.; de Ridder, D.; Handels, H. Image processing with neural networks - a review. *Pattern Recognition* 2002, *35(10)*, 2279–2301.

[61] Akman, V.; Blackburn, P. Alan Turing and Artificial Intelligence. *J. Logic, Lang. and Inform.* 2000, *9*,391-395.

[62] Gharagheizi, F. QSPR analysis for intrinsic viscosity of polymer solutions by means of GA-MLR and RBFNN. *Comput. Mater. Sci.* 2007, *40*, 159-167.

[63] Shi, J.; Luan, F.; Zhang, H.X.; Liu, M.C.; Guo, Q.X.; Hu, Z.D.; Fan, B.T. QSPR study of fluorescence wavelengths (lambda(ex)/lambda(em)) based on the heuristic method and radial basis function neural networks. *QSAR Comb. Sci.* 2006, *25*, 147-155.

[64] Tetko, I.V.; Solov'ev, V.P.; Antonov, A.V.; Yao, X.J.; Doucet, J.P.; Fan, B.T.; Hoonakker, F.; Fourches, D.; Jost, P.; Lachiche, N.; Varnek, A. Benchmarking of linear and nonlinear approaches for quantitative structure-property relationship studies of metal complexation with ionophores. *J. Chem. Inf. Model.* 2006, *46*, 808-819.

[65] Chen, H.F. Quantitative predictions of gas chromatography retention indexes with support vector machines, radial basis neural networks and multiple linear regression. *Anal. Chim. Acta* 2008, *609*, 24-36.

[66] Qi, J.; Niu, J.F.; Wang, L.L. Research on QSPR for n-octanol-water partition coefficients of organic compounds based on genetic algorithms-support vector machine and genetic algorithms radial basis function neural networks. *Huanjing Kexue* 2008, *29*, 212-218.

[67] Specht, D.F. A general regression neural network. *IEEE Trans. Neur. Netw.* 1991, *2*, 568-576.

[68] Szaleniec, M.; Tadeusiewicz, R.; Witko, M. How to select an optimal neural model of chemical reactivity? *Neurocomputing* 2008, *72*, 241-256.

[69] Agatonovic-Kustrin, S.; Turner, J.V. Artificial neural network modeling of phytoestrogen binding to estrogen receptors. *Lett. Drug Des. Disc.* 2006, *3*, 436-442.

[70] Ibric, S.; Jovanovic, M.; Djuric, Z.; Parojcic, J.; Solomun, L.; Lucic, B. Generalized regression neural networks in prediction of drug stability. *J. Pharm. Pharmacol.* 2007, *59*, 745-750.

[71] Gasteiger, J.; Zupan, J. Neural Networks in Chemistry. *Angew. Chem. Int. Ed. Engl.* 1993, *32*, 503-527.

[72] Prados, D. L. New learning algorithm for training multilayered neural networks that uses genetic-algorithm techniques. *Electron. Lett.* 1992, *28*, 1560–1561.

[73] Porto, V. W.; Fogel, D. B.; Fogel, L. J. Alternative neural network training methods. *IEEE Expert* 1995, *10*, 16–22.

[74] Maniezzo, V. Genetic evolution of the topology and weight distribution of neural networks. *IEEE Trans. Neural Networks* 1994, *5*, 39–53.

[75] Fang, J.; Xi, Y. Neural network design based on evolutionary programming. *Artificial Intell. Eng.* 1997, *11(2)*, 155–161.

[76] Andrea, T. A.; Kalayeh, H. Application of Neural Networks in Quantitative Structure-Activity Relationships of Dihydrofolate Reductase Inhibitors. *J. Med. Chem*. 1991, *34*, 2824-2836.

[77] So, S.; Richards, G. Application of Neural Networks: Quantitative Structure-Activity Relationships of the Derivatives of 2,4-Diamino- 5- (substituted-benzyl) pyrimidines as DHFR Inhibitors. *J. Med. Chem*. 1992, *35*, 3201-3207.

[78] Morris, K. The effect of HIV/AIDS on International Health. *Lancet Infect. Dis.* 2008, *8(8)*, 468-469.

[79] De Béthune, M. P. Non-nucleoside reverse transcriptase inhibitors (NNRTIs), their discovery, development, and use in the treatment of HIV-1 infection: A review of the last 20 years (1989–2009) *Antivir. Res.* 2010, *85*, 75–90.

[80] Silvestri, R.; Maga, G. Current State-of-the-art in Preclinical and Clinical Development of Novel Non-nucleoside HIV-1 Reverse Transcriptase Inhibitors. *Expert Opin. Therap. Patents* 2006, *16*, 939-962.

[81] Cohen, J. Up in smoke: epidemic changes course. *Science* 2006, 313, 487–488.

[82] McReynolds, K. D.; Gervay-Hague, J. Chemotherapeutic Interventions Targeting HIV Interactions with Host-Associated Carbohydrates, *Chem. Rev.* 2007, *107*, 1533-1552.

[83] Clark, R. D. A ligand's-eye view of protein binding. *J. Comput. Aided Mol. Des.* 2008, *22*, 507–521.

[84] Mestres, J.; Rohrer, D. C.; Maggiora, G. M. A molecular-field-based similarity study of non-nucleoside HIV-1 reverse transcriptase inhibitors. 2. The relationship between alignment solutions obtained from conformationally rigid and flexible matching. *J. Comput. Aided Mol. Des.* 2000, *14*, 39–51.

[85] Hannongbua, S.; Prasithichokekul, S.; Pungpo, P. Conformational analysis of nevirapine, a non-nucleoside HIV-1 reverse transcriptase inhibitor, based on quantum mechanical calculations. *J. Comput. Aided Mol. Des.* 2001, *15*, 997–1004.

[86] Dickinson, L.; Khoo, S.; Back, D. Pharmacokinetics and drug–drug interactions of Antiretroviral: An update *Antivir. Res.* 2010, *85*, 176–189.

[87] Almerico, A. M.; Tutone, M.; Lauria, A. Docking and multivariate methods to explore HIV-1 drug-resistance: a comparative analysis. *J. Comput. Aided Mol. Des.* 2008, 22, 287–297.

[88] Medina-Franco, J.L.; Rodr´ıguez-Morales, S.; Ju´arez-Gordiano, C.; Hern´andez-Campos, A.; Castillo, R. Docking-based CoMFA and CoMSIA studies of non-nucleoside reverse transcriptase inhibitors of the pyridinone derivative type. *J. Comput. Aided Mol. Des.* 2004, *18*, 345–360.

[89] Martins, S.; Ramos, M. J.; Fernandes, P.A. The Current Status of the NNRTI Family of Antiretrovirals Used in the HAART Regime against HIV Infection. *Curr. Med. Chem.* 2008, *15*, 1083-1095.

[90] Tomicha, C.H.; da Silvaa, P.; Carvalhoa, I.; Taft, C.A. Homology modeling and molecular interaction field studies of a-glucosidases as a guide to structure-based design of novel proposed anti-HIV inhibitors. *J. Comput. Aided Mol. Des.* 2005, *19*, 83–92.

[91] Medina-Franco, J.L.; Golbraikh, A.; Oloff, S.; Castillo, R.; Tropsha, A. Quantitative structure–activity relationship analysis of pyridinone HIV-1 reverse transcriptase inhibitors using the *k* nearest neighbor method and QSAR-based database mining. *J. Comput. Aided Mol. Des.* 2005, *19*, 229–242.

[92] Bustanji, Y.; Al-Masri, I.M.; Qasem, A.; Al-Bakri, A.G.; Taha, M.O. In Silico Screening for Non-nucleoside HIV-1 Reverse Transcriptase Inhibitors Using Physicochemical Filters and High-Throughput Docking Followed by In Vitro Evaluation. *Chem. Biol. Drug. Des.* 2009, *74*, 258–265.

[93] Krüger, D. M.; Evers, A. Comparison of Structure- and Ligand-Based Virtual Screening Protocols Considering Hit List Complementarity and Enrichment Factors, *ChemMedChem* 2010, *5*, 148–158.

[94] Catherine, S.; Adamson, E. O. Freed Novel approaches to inhibiting HIV-1 replication. *Antivir. Res.* 2010, *85*. 119–141.

[95] De Clercq E. The discovery of antiviral agents: Ten different compounds, ten different stories. *Med. Res. Rev.* 2008, *28(6)*, 929-953.

[96] Greenberg, M.L.; Cammack, N. Resistance to enfuvirtide, the first HIV fusion inhibitor. *J. Antimicrob. Chemother.* 2004, *54*, 333-340.

[97] Cihlar, T.; Ray, A. Nucleoside and nucleotide HIV reverse transcriptase inhibitors: 25 years after zidovudine. *Antivir. Res.* 2010, *85*, 25–33.

[98] Garg, R.; Gupta, S. P.; Gao, H.; Babu, M. S. Debnath, A. K. Comparative Quantitative Structure-Activity Relationship Studies on Anti-HIV Drugs. *Chem. Rev.* 1999, *99*, 3525-3601.

[99] Sattwa, M.A.; Kunal, R. Predictive QSAR modeling of HIV reverse transcriptase inhibitor TIBO derivatives. *Eur. J. Med. Chem.* 2009, *44 (4)*, 1509–1524.

[100] Klein, C.; Lawtrakul, L.; Hannongbua, S.; Wolschann, P. Accessible charges in structure-activity relationships: a study of HEPT-based HIV-1 RT inhibitors. *Sci. Pharm.* 2000, *68*, 25–40.

[101] Kiralj, R.; Ferreira, M. A priori molecular descriptors in QSAR: a case of HIV-1 protease inhibitors. I. The chemometric approach. *J. Mol. Graph. Model.* 2003, *21 (5)*, 435–448.

[102] Hemmateenejad, B.; Tabaei, S.M.; Namvaran, F. Computer-aided design of potential anti-HIV-1 non-nucleoside reverse transcriptase inhibitors by contraction of b-ring in TIBO derivatives. *J. Mol. Struct. Theochem.* 2005, *732 (1–3),* 39–45.

[103] Huuskonen, J. QSAR modeling with the electrotopological state: TIBO derivatives. *J. Chem. Inf. Comput. Sci.* 2001, *41*, 425–429.

[104] Maw, H.H.; Hall, L.H. E-state modeling of HIV-1 protease inhibitor binding independent of 3D information. *J. Chem. Inf. Comput. Sci.* 2002, *42*, 290–298.

[105] Gancia, E.; Bravi, G.; Mascagni, P.; Zaliani, A. Global 3D-QSAR methods: MSWHIM and autocorrelation. *J. Comput. Aided Mol. Des.* 2000, *14*, 293–306.

[106] Douali L.; Villemin D.; Cherqaoui D. Exploring QSAR of Non-Nucleoside Reverse Transcriptase Inhibitors by Neural Networks: TIBO Derivatives. *Int. J. Mol. Sci.* 2004, *5,* 48-55.

[107] Sapre, N. S.; Gupta, S.; Pancholi, N.; Sapre, N. Computational Modeling of Tetrahydroimidazo- [4,5,1-jk][1,4]-benzodiazepinone Derivatives: An Atomistic Drug Design Approach Using Kier-Hall Electrotopological State (E-State) Indices. *J. Comput. Chem.* 2008, *29*, 1699–1706.

[108] Sapre, N. S.; Gupta, S.; Pancholi, N.; Sapre, N. A Group Center Overlap Based Approach for ‘‘3D QSAR’’ Studies on TIBO Derivatives. *J. Comput. Chem.*2009, *30*, 922–933.

[109] Jalali-Heravi M.; Parastar F. Use of Artificial Neural Networks in a QSAR Study of Anti-HIV Activity for a Large Group of HEPT Derivatives. *J. Chem. Inf. Comput. Sci.* 2000, *40,* 147-154.

[110] Luco, J. M.; Ferretti, F. H. QSAR Based on Multiple Linear Regression and PLS Methods for the Anti-HIV Activity of a Large Group of HEPT Derivatives. *J Chem. Inf. Comput. Sci.* 1997, *37*, 392-401.

[111] Mager P. P. Hybrid canonical-correlation neural-network approach applied to nonnucleoside HIV-1 reverse transcriptase inhibitors (HEPT derivatives). *Curr. Med. Chem.* 2003, *10(17)*, 1643-59.

[112] Douali L.; Villemin D.; Cherqaoui D. Comparative QSAR based on neural networks for the anti-HIV activity of HEPT derivatives. *Curr. Pharm. Des.* 2003, *9(22)*, 1817-26.

[113] *Basak, S. C.; Bertelsen, S.; Grunwald, G. D.* Use of Graph Theoretic Parameters in Risk Assessment of Chemicals. *Toxicol. Lett.1995*, *79*, *239*−250.

[114] Gute, B. D.; Grunwald, G. D.; Mills, D.; Basak, S. C. Molecular similarity based estimation of properties: A comparison of structure spaces and property spaces.*SAR QSAR Environ. Res.* 2001, *11*, 363-384.

[115] Basak, S. C.; Gute, B. D.; Mills, D. Quantitative molecular similarity analysis (QMSA) methods for property estimation: A comparison of property-based, arbitrary, and tailored similarity spaces. *SAR QSAR Environ. Res*. 2002, *13*, 727-742.

[116] Basak, S. C.; Gute, B. D.; Mills, D.; Hawkins, D. M. Quantitative molecular similarity methods in the property/ toxicity estimation of chemicals: A comparison of arbitrary versus tailored similarity spaces. *J. Mol. Struct. (THEOCHEM)* 2003, *622*, 127-145.

[117] Goodarzi, M.; Freitas, M. P. MIA–QSAR coupled to principal component analysis-adaptive neuro-fuzzy inference systems (PCA–ANFIS) for the modeling of the anti-HIV reverse transcriptase activities of TIBO derivatives, *Eur. J. Med. Chem.* 2010, 45(4), 1352-1358.

[118] Tropsha, A. In Cheminformatics in Drug Discovery; Oprea, T., Ed.; Wiley-VCH: New York, 2005, pp. 437–455.

[119] Tropsha, A.; Cho, S. J.; Zheng, W. In Rational Drug Design: Novel Methodology and Practical Applications; Parrill A. L.; Reddy, M. R., Eds.; ACS symposium series, American Chemical Society Publication, 1999, pp 198–211.

[120] Sapre, N. S.; Gupta, S.; Sapre, N. Assessing Ligand Efficiencies using Template-based Molecular docking and Tabu-clustering on Tetrahydroimidazo-[4,5,1-jk][1,4]-benzodiazepinone (TIBO) Derivatives as HIV-1RT inhibitors. *J. Chem. Sci.* 2008, *120(4)*, 395-404.

[121] Douali L.; Villemin D.; Zyad A.; Cherqaoui D. Artificial neural networks: non-linear QSAR studies of HEPT derivatives as HIV-1 reverse transcriptase inhibitors. *Mol. Divers.* 2004, *8(1),* 1-8.

[122] Douali L.; Villemin D.; Cherqaoui D. Neural networks: Accurate nonlinear QSAR model for HEPT derivatives. *J. Chem. Inf. Comput. Sci.* 2003, *43(4)*, 1200-7.

[123] Kjaer J.; Høj L.; Fox Z.; Lundgren J. D. Prediction of phenotypic susceptibility to antiretroviral drugs using physiochemical properties of the primary enzymatic structure combined with artificial neural networks. *HIV Med.* 2008, *9(8)*, 642-52.

[124] Shahlaei, M.; Fassihi, A.; Saghaie, L. Application of PC-ANN and PC-LS-SVM in QSAR of CCR1 antagonist compounds: A comparative study, *Eur. J. Med. Chem* 2010 45(4):1572-82.

[125] Shi, W.; Zhang, X.; Shen, Q. Quantitative structure-activity relationships studies of CCR5 inhibitors and toxicity of aromatic compounds using gene expression programming, *Eur. J. Med. Chem.* 2010, *45*, 49–54.

[126] Sato, T.; Honma, T.; Yokoyama, S. Combining Machine Learning and Pharmacophore-Based Interaction Fingerprint for in Silico Screening. *J. Chem. Inf. Model.* 2010, *50,* 170–185.

[127] Darnag, R.; Mostapha Mazouz, E.L.; Schmitzer, A.; Villemin, D.; Jarid, A.; Cherqaoui, D. Support vector machines: Development of QSAR models for predicting anti-HIV-1 activity of TIBO derivatives, *Eur. J. Med. Chem.* 2010, 45(4), 1590–1597.

In: New Developments in Artificial Neural Networks Research ISBN: 978-1-61324-286-5
Editor: Robert W. Nelson © 2011 Nova Science Publishers, Inc.

Chapter 20

PRICING IN BANKING AND FINANCE BY UTILIZING ARTIFICIAL NEURAL NETWORKS

Vincenzo Pacelli[*]
University of Foggia – Italy

JEL Classification: C3, C5, G14, G20.
Keywords: Pricing; Banking and Finance; Artificial Neural Networks.

1. INTRODUCTION

The objective of this paper is to analyze the theme of the application of the intelligent learning systems (such as artificial neural networks, expert systems, fuzzy models and genetic algorithms) in the pricing in banking and finance.

All firms must settle a price for the services or products which they offer. The price is an important element of the marketing mix, being also an important source of income for the firm. As the competition in the financial systems has intensified, nowadays the settlement of correct prices has become an essential element for the marketing strategy of a bank.

The price must not be considered as a purely financial problem, calculated by estimating only the costs to which a margin for profit will be added. The settlement of the price must consider also the stakeholders point of view.

A peculiarity of the pricing in banking is also a partial lack of transparency which make difficult to understand the variables to analyze in order to forecast the phenomenon.

In banking and finance there are many factors that influence the pricing as shown in the figure below.

[*] Ph. D. - University of Rome “La Sapienza”, Tenured Assistant Professor of Economics of Financial Intermediaries, University of Foggia, Faculty of Economics. Via Caggese, 1 - 71100 - Foggia – Italy, E-mail: v.pacelli@unifg.it

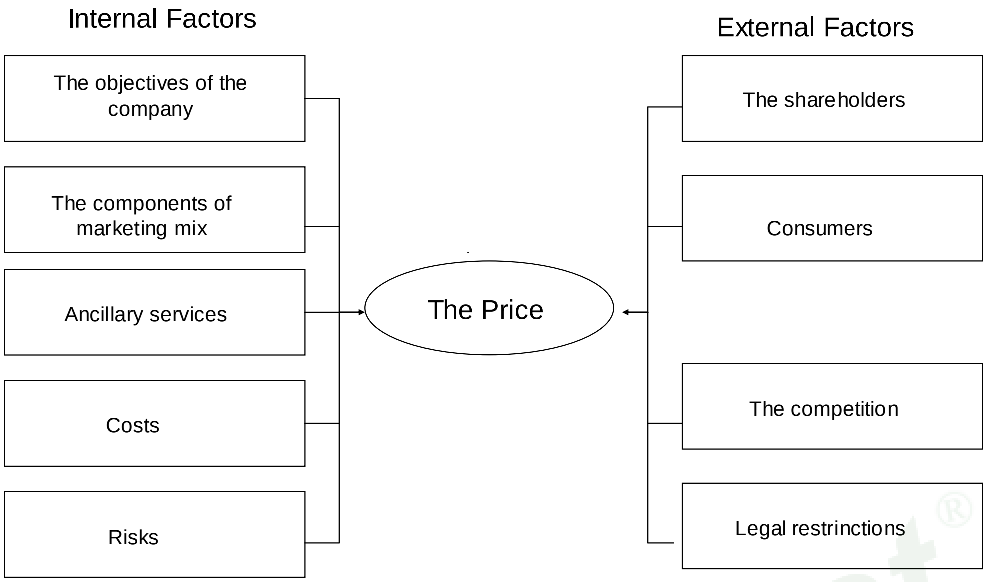


Figure 1. The factors that influence the pricing in banking and finance.

As we can see in figure 1, the factors are classified in internal and external ones. The internal factors are the ones inside the company and which are under its control as:

- the objectives of the company;
- the other variables of the marketing mix as the policies of product, communication and distribution;
- the structure of the costs;
- the ancillary services;
- the evaluation of the risk.

The external factors are the ones that influence the pricing outside the company and it has a reduced or inexistent control of them.

On their turn, these factors can be shared in internal factors of the activity sector as:

- competition;
- shareholders;

and in external ones as:

- consumers;
- the legislation.

2. Pricing in Banking and Finance By Utilizing Intelligent Learning Systems: A Literature Review

The literature on the application of artificial intelligence systems in the pricing in banking and finance has explored various aspect of the phenomenon as we can see by analyzing the studies of White (1988), Grudnitski and Osburn (1993), Refenes, Zapranis and Francis (1994), Trippi and Turban (1996), Anders, Korn and Schimitt (1998), Corazza and Gobbo (2002), Oh and Kim (2002), Farina and Amato (2004), Cao, Leggio and Schniederjans (2005), Chen (2007), Hyun-jung Kim and Kyung-shik Shin (2007), Thiagarajah, Appadoo and Thavaneswaran (2007), Andreou, Charalambous and Martzoukos (2008), Amjady and Keynia (2009), Po-Chang Ko (2009), Wang (2009), Liang, Zhang, Xiao and Chen (2009), Pacelli (2009), Zapranis and Alexandridis (2009), Esfahanipour and Aghamiri (2010), Fethi and Pasiouras (2010).

In his work White (1988)[1] reports the results of a project finalized to search for and decode nonlinear regularities in asset price movements using neural network modelling and learning techniques. He uses a linear autoregressive model of “order p”, that corresponds to a very simple two layer feed-forward network. He focuses on the case of IBM common stock daily returns. This exercise was carried out for a post sample period of 500 days, and a pre-sample period of 500 days. For the post-sample period it has been observed a correlation of – 0,0699; for the pre-sample period, it was 0,0751 (for comparison, the linear model gives post-sample correlation of – 0,207 and pre-sample correlation of 0,0996). Such results do not constitute convincing statistical evidence against the efficient markets hypothesis. The in-sample (training period) results are seen to be over-optimistic, being either the result of over-fitting (random fluctuations recognized incorrectly as nonlinearities) or of learning evanescent features (features which are indeed present during the training period, but which subsequently disappear). In either case the implication is the same: the neural network model is not a money machine.

The research of Grudnitski and Osburn (1993)[2] suggests the standard random walk assumption of futures price may actually be only a veil of randomness that shrouds a noisy nonlinear process. What is called to remove this veil and thereby make futures price forecasting possible is the application of nonlinear models by the author’s point of view. The research examines the feasibility of the neural networks to forecast price changes of S&P 500 stock index and gold futures based on past changes and historical open interest patterns that are held to represent the beliefs of a majority of the traders in these markets.

Refenes, Zapranis and Francis (1994)[3] examine the use of neural networks as an alternative to classical statistical techniques for forecasting within the framework of the APT (arbitrage pricing theory) model for stock ranking. They show that neural networks outperform these statistical techniques in forecasting accuracy terms, and give better model fitness in-sample by one order of magnitude. They identify intervals for the network parameter values for which these performance figures are statistically stable. Neural networks

[1] H. White, “Economic Prediction Using Neural Networks: The Case of IBM Stock Prices”, in *Proceedings of the Second Annual IEEE Conference in Neural Networks*, pp. 451-458, 1988.

[2] G. Grudnitski, L. Osburn, “Forecasting S&P and Gold Futures Prices: An Application of Neural Networks”, in *Journal of Futures Markets*, Volume 13, n. 6, pp. 631-643, 1993.

[3] A. P. Refenes, A. Zapranis, G. Francis, “Stock performance modelling using neural network; a comparative study with regression model”, in *Neural Networks*, Volume 7, pp. 375-388, 1994.

are usually criticised for not being able to provide an explanation of how they interact with their environment and how they reach an outcome. The authors show that by using sensitivity analysis, neural networks can provide a reasonable explanation of their predictive behaviour and can model their environment more convincingly than regression models.

Trippi and Turban (1996)[4] assemble a stellar collection of articles by recognized experts from industry and academia. They discuss neural network successes and failures, as well as identify the vast unrealized potential of neural networks in numerous specialized areas of financial decision making.

In the paper of Anders, Korn and Schimitt (1998)[5], the authors utilize statistical inference techniques to build neural network models which are able to explain the prices of call options written on the German stock index DAX. By testing for the explanatory power of several input variables serving as network inputs, some insight into the pricing process of the option market is obtained. The results indicate that statistical specification strategies lead to parsimonious networks which have a superior out-of-sample performance when compared to the Black/Scholes model. They further validate their results by providing plausible hedge parameters.

The paper of Corazza and Gobbo (2002)[6] considers neural networks applied to the evaluation of financial options. Initially, they proceed to the presentation of a particular type of neural network, the perceptron multylayer to supervised learning, highlighting key features such as the architectural structure, the estimation procedure, setting and development. Next, they apply the instrument presented to the evaluation of financial option call European-style written English stock index FTSE-100. They conclude by stating that the neuro-computational approach for the evaluation of financial options is a viable alternative to more traditional valuation techniques.

The study of Oh and Kim (2002)[7] proposes stock trading model based on chaotic analysis and piecewise nonlinear model. The core component of the model is composed of four phases. The first phase determines time-lag size in input variables using chaotic analysis. The second phase detects successive change-points in the stock market data and the third phase forecasts the change-point group with back propagation neural networks (BPNs). The final phase forecasts the output with BPN. The experimental results are encouraging and show the usefulness of the proposed model with respect to profitability.

The study of Cao, Leggio and Schniederjans (2005)[8] uses artificial neural networks to predict stock price movement (i.e., price returns) for firms traded on the Shanghai stock exchange. They compare the predictive power using linear models from financial forecasting literature to the predictive power of the univariate and multivariate neural network models. Their results show that neural networks outperform the linear models compared. These results

[4] R. R. Trippi, E. Turban, Neural Networks in Finance and Investing, Irwin Professional Pub., New York, 1996.

[5] U. Anders, T. H. Hann, G. Nakaheizadeh, "Testing for Nonlinearity with Neural Networks", in Weigend A. S., Abu Mustafa Y., Refens A. P. N. (a cura di), "Decision Technologies for Financial Engineering", in World Scientific, Singapore, 1997.

[6] M. Corazza, M. Gobbo, "Reti Neurali Artificiali per la valutazione di opzioni", in Working Paper n. 02, 06/Maggio, 2002.

[7] K. J. Oh, K. J. Kim, "Analyzing stock market tick data using piecewise nonlinear model", in Expert Systems with Applications, Volume 22, Issue 3, April, pp. 249-255, 2002.

[8] Q. Cao, K. B. Leggio, M. J. Schniederjans, "A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market", in Computers & Operations Research, Volume 32, Issue 10, October, pp. 2499-2512, 2005.

are statistically significant across our sample firms and indicate that the neural networks are a useful tool for stock price prediction in emerging markets, like China.

The study of Hyun-jung Kim and Kyung-shik Shin (2007)[9] investigates the effectiveness of a hybrid approach based on the artificial neural networks (ANNs) for time series properties, such as the adaptive time delay neural networks (ATNNs) and the time delay neural networks (TDNNs), with the genetic algorithms (GAs) in detecting temporal patterns for stock market prediction tasks. Since ATNN and TDNN use time-delayed links of the network into a multi-layer feed-forward network, the topology of which grows by on layer at every time step, it has one more estimate of the number of time delays in addition to several control variables of the ANN design. To estimate these many aspects of the ATNN and TDNN design, a general method based on trial and error along with various heuristics or statistical techniques is proposed. However, for the reason that determining the number of time delays or network architectural factors in a stand-alone mode does not guarantee the illuminating improvement of the performance for building the ATNN and TDNN model, they apply GAs to support optimization of the number of time delays and network architectural factors simultaneously for the ATNN and TDNN model. The results show that the accuracy of the integrated approach proposed for this study is higher than that of the standard ATNN, TDNN and the recurrent neural network (RNN).

In the paper of Thiagarajah, Appadoo and Thavaneswaran (2007)[10], the authors consider moment properties for a class of quadratic adaptive fuzzy numbers defined in Dubois and Prade. The corresponding moments of Trapezoidal Fuzzy Numbers (Tr.F.N's) and Triangular Fuzzy Numbers (T.F.N's) turn out to be special cases of the adaptive fuzzy number. A numerical example is presented based on the Black-Scholes option pricing formula with quadratic adaptive fuzzy numbers for the characteristics such as volatility parameter, interest rate and stock price. Their approach hinges on a characterization of imprecision by means of fuzzy set theory.

Andreou, Charalambous and Martzoukos (2008)[11] compare the ability of the parametric Black and Scholes, Corrado and Su models, and Artificial Neural Networks to price European call options on the S&P 500 using daily data for the period January 1998 to August 2001. They use several historical and implied parameter measures. Beyond the standard of neural networks, in their analysis they include hybrid networks that incorporate information from the parametric models. Their results are significant and differ from previous literature. They show that the Black and Scholes based hybrid artificial neural network models outperform the standard neural networks and the parametric ones. They also investigate the economic significance of the best models using trading strategies (extended with the Chen and Johnson modified hedging approach). They find that there exist profitable opportunities even in the presence of transaction costs.

[9] Hyun-jung Kim, Kyung-shik Shin , “A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets”, in Applied Soft Computing, Volume 7, Issue 2, March, pp. 569-576, 2007.

[10] K. Thiagarajah, S. S. Appadoo, A. Thavaneswaran, “Option valuation model with adaptive fuzzy numbers”, in Computers & Mathematics with Applications, Volume 53, Issue 5, March, pp. 831-841, 2007.

[11] P. C. Andreou, C. Charalambous, S. H. Martzoukos, “Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters”, in European Journal of Operational Research, Volume 185, Issue 3, 16 March, pp. 1415-1433, 2008.

In the paper of Amjady and Keynia (2009)[12], a new forecast strategy is proposed for day-ahead price forecasting of electricity markets. Their forecast strategy is composed of a new two stage feature selection technique and cascaded neural networks. The proposed feature selection technique comprises modified Relief algorithm for the first stage and correlation analysis for the second stage. The modified Relief algorithm selects candidate inputs with maximum relevancy with the target variable. Then among the selected candidates, the correlation analysis eliminates redundant inputs. Selected features by the two stage feature selection technique are used for the forecast engine, which is composed of 24 consecutive forecasters. Each of these 24 forecasters is a neural network allocated to predict the price of 1 h of the next day. The whole proposed forecast strategy is examined on the Spanish and Australia's National Electricity Markets Management Company (NEMMCO) and compared with some of the most recent price forecast methods.

In the paper of Liang, Zhang, Xiao and Chen (2009)[13], a simple and effective nonparametric method of forecasting option prices based on neural networks (NNs) and support vector regressions (SVRs) is presented. They first modified the improved conventional option pricing methods, allowing them to forecast the option prices. Then they employed the NNs and SVRs to further decrease the forecasting errors of the parametric methods. Since the conventional methods mimic the trends of movement of the real option prices, using these methods in a first stage allows the NNs and SVRs to concentrate their power in nonlinear curve approximation to further reduce the forecasting errors in a second stage. Finally, extensive experimental studies with data from the Hong Kong option market demonstrated the ability of NNs and SVRs to improve forecast accuracy.

The objective of the paper of Pacelli (2009)[14] is to propose an intelligent computing algorithm, represented by an artificial neural network model, to analyze the dynamics of stock prices of banks. Through the empirical application of the model developed, the author obtains indications about the ability of the artificial neural network model developed to generalize the phenomenon analyzed. So the research provides empirical results about the use of non-linear methods of analysis for the study of the dynamics of banks' stock prices, enriching the prospects for research in terms of methodological tools.

In the article of Po-Chang Ko (2009)[15], a neural regression (NR) model, which produces nonlinear coefficients of multiple regression model based on neural networks, is introduced to capture the option valuation's nonlinear characteristics effectively. The traditional linear regression uses the least-squares estimator to estimate the coefficient of a linear regression and thus may only produce suboptimal solutions. However, applying neural networks to forecast volatility in option pricing has increased in popularity in recent years since many studies have indicated that the conventional option pricing models are not accurate enough. The proposed neural regression model devotes to evaluate option values to improve on the

[12] N. Amjady, F.Keynia, "Day-ahead price forecasting of electricity markets by a new feature selection algorithm and cascaded neural network technique", in Energy Conversion and Management, Volume 50, Issue 12, December, pp. 2976-2982, 2009.

[13] X. Liang, H. Zhang, J. Xiao, J. Chen, "Improving option price forecasts with neural networks and support vector regressions", in Neurocomputing, Volume 72, Issues 13-15, August, 2009.

[14] V. Pacelli, "An Intelligent Computing Algorithm to Analyze Bank Stock Returns", in Aa. Vv. (eds) (2009), Emerging Intelligent Computing Technology and Applications, Lectures Notes on Computer Sciences, n. 5754, Springer Verlag, New York, 2009.

[15] Po-Chang Ko (2009), "Option valuation based on the neural regression model", in Expert Systems with Applications, Volume 36, Issue 1, January , pp. 464-471.

tracking error in the measurement of hedging capability. The NR model uses the variables introduced by the Black-Scholes Model and applies the multiple regressions (MR) model to re-price option values. It is worth noting that each corresponding weight coefficient in MR is constructed by a complete neural network rather than by a scalar value. By capturing the nonlinear behaviours of option pricing, the proposed NR model has lower tracking error and better hedging capability than the BS model and other studies.

The study of Wang (2009)[16] integrates new hybrid asymmetric volatility approach into artificial neural networks option-pricing model to improve forecasting ability of derivative securities price. Owing to combines the new hybrid asymmetric volatility method can be reduced the stochastic and nonlinearity of the error term sequence and captured the asymmetric volatility simultaneously. Hence, in the ANNS option-pricing model, the results demonstrate that Grey-GJR-GARCH volatility provides higher predictability than other volatility approaches.

In the paper of Zapranis and Alexandridis (2009)[17], the authors use neural networks in order to model the seasonal component of the residual variance of a mean-reverting Ornstein-Uhlenbeck temperature process, with seasonality in the level and volatility. This approach can be easily used for pricing weather derivatives by performing Monte Carlo simulations. Moreover, in synergy with neural networks they use wavelet analysis to identify the seasonality component in the temperature process as well as in the volatility of the temperature anomalies. This model is validated on more than 100 years of data collected from Paris, one of the European cities traded at Chicago Mercantile Exchange. Their results show a significant improvement over more traditional alternatives, regarding the statistical properties of the temperature process. This is important since small misspecifications in the temperature process can lead to large pricing errors.

In the paper of Esfahanipour and Aghamiri (2010)[18], Neuro-Fuzzy Inference System adopted on a Takagi-Sugeno-Kang (TSK) type Fuzzy Rule Based System is developed for stock price prediction. The TSK fuzzy model applies the technical index as the input variables and the consequent part is a linear combination of the input variables. Fuzzy C-Mean clustering implemented for identifying number of rules. Initial membership function of the premise part approximately defined as Gaussian function. TSK parameters tuned by Adaptive Nero-Fuzzy Inference System (ANFIS). Proposed model is tested on the Tehran Stock Exchange Indexes (TEPIX). This index with high accuracy near by 97,8% has successfully forecasted with several experimental tests from different sectors.

The paper of Fethi and Pasiouras (2010)[19] presents a comprehensive review of 196 studies which employ operational research (O.R.) and artificial intelligence (A.I.) techniques in the assessment of bank performance. Several key issues in the literature are highlighted. The paper also points to a number of directions for future research. They first discuss

[16] H. Y. Wang, “Nonlinear neural network forecasting model for stock index option price: Hybrid GJR–GARCH approach”, in Expert Systems with Applications, Volume 36, Issue 1, January, pp. 564-570, 2009.

[17] A. Zapranis, A. Alexandridis, “Weather derivatives pricing: Modeling the seasonal residual variance of an Ornstein–Uhlenbeck temperature process with neural networks”, in Neurocomputing, Volume 73, Issues 1-3, December, pp. 37-48, 2009.

[18] A. Esfahanipour, W. Aghamiri, “Adapted Neuro-Fuzzy Inference System on indirect approach TSK fuzzy rule base for stock market analysis”, in Expert Systems with Applications, Volume 37, Issue 7, July, pp. 4742-4748, 2010.

[19] M. D. Fethi, F. Pasiouras, “Assessing bank efficiency and performance with operational research and artificial intelligence techniques: A survey”, in European Journal of Operational Research, Volume 204, Issue 2, 16 July, pp. 189-198, 2010.

numerous applications of data envelopment analysis which is the most widely applied O.R. technique in the field. Then they discuss applications of other techniques such as neural networks, support vector machines, and multicriteria decision aid that have also been used in recent years, in bank failure prediction studies and the assessment of bank creditworthiness and under performance.

3. An Artificial Neural Network Model for Pricing in Banking

Artificial Neural Networks are non-linear mathematical models that can be trained to map past and future values of time series data and thereby extract hidden relationship that govern the data. Artificial Neural Networks consist of multiple neurons which are extensively interconnected and organized in layers similar to those of a decision tree. These neurons work in unison to solve specific problems and, when trained through time series of data, become potent tools for analysis and forecasting.

A single artificial neuron, which is the basic element of the neural network, comprises several inputs (i_1, i_2, …, i_m) and one output (y) that can be written as follows:

$$y = f(i_n, w_n) \tag{1}$$

where, w_n are the function parameter weights of the function f.

Equation (1) is called an activation function. Each artificial neuron determines its output by applying an activation function and weights to inputs. An activation function of the Artificial Neural Network Model – ANN_m useful for pricing in banking is the symmetrical sigmoid function that can be written as follows[20]:

$$f(y) = 1 / [1 + \exp(-k\Psi y)] \tag{2}$$

where, $\Psi > 0$ is the slope of sigmoid in its point of inflection y = 0, and k is a constant. The output (y) of the symmetrical sigmoid function is included in the range [-1, 1].

With regard to architecture of the ANN_m, an architecture which can be used to analyze the phenomenon of pricing is the multilayer fully connected and characterized by *n* hidden layers an *n* neurons in each layer.

The response of an artificial neural network is determined by the synaptic values (weights) of the connections between the nodes of the neurons. In the same way biological nervous systems are able of learning by experience, artificial neural networks learn the reality analyzed by changing gradually their synaptic values (weights) ΔW_{ij} when trained through time series of data (pattern of input and output) and through the use of a learning algorithm.

The learning algorithm which can be used in the training of the ANN_m is the back propagation algorithm with learning for cycles (on line). In this way it is so developed an ANN_m feed-forward. In the learning for cycles, the changing of the synaptic values (weights) ΔW_{ij} of the ANN_m is calculated after each presentation of a single pattern. The new

[20] See Pacelli (2009).

configuration of the synaptic values (weights) after a cycle of training is calculated by adding the change obtained $\varDelta W_{ij}(t)$ to the previous configuration of the synaptic values $W_{ij}(t\text{-}1)$. The speed of learning is regulated by a constant η – the learning rate – which defines the portion of change that is applied to the synaptic values. The equation of learning can be write as follows:

$$W_{ij}(t) = W_{ij}(t\text{-}1) + \eta\ \Delta W_{ij}(t), \tag{3}$$

where, $0< \eta<1$.

The training of the ANN_m will be interrupted when the mean square error has become less of a defined value.

Once the learning phase was completed, the synaptic values are frozen and it is possible to study the response of the ANN_m on other patterns of data in the phase of test. The phase of test consists in the presentation of new patterns and in the calculation of activation of the nodes of the ANN_m with synaptic weights frozen.

A typical problem is represented by the overfitting of the neural networks that would decrease the probability that neural networks discover the general function that describes the phenomenon analyzed. This problem can be overcome by increasing the number of the pattern of input and output used for the training and the testing of the neural networks.

In order to be used to train the ANN_m, the patterns of input and output must be appropriately amended (normalised) through their transformation into the range [-1, 1] as follows:

$$I = I_{min} + [(I_{max}\text{-}I_{min})*(D\text{-}D_{min})/(D_{max}\text{-}D_{min})] \tag{4}$$

where:

- D_{min} and D_{max} are the extreme values of the ranges – namely the maximum and minimum values – of the variables of input and output;
- D is the value of each input and output;
- I_{max} and I_{min} are the new extreme values of the range or “-1” and “1”.

4. Concluding Remarks

The analysis of the studies mentioned above showed that, among the systems of the artificial intelligence applied to finance, neural networks are the most widely used models, followed by fuzzy systems.

In particular neural networks are mainly used for stock market prediction, but a part of literature has given attention to the application of these systems for pricing of derivates and in particular for options pricing. Less attention has been paid instead to the application of fuzzy systems. In this regard, in fact, we can note few contributes, as those of Thiagarajah, Appadoo and Thavaneswaran (2007) and Esfahanipour and Aghamiri (2010), concerning to the use of fuzzy systems respectively for stock price prediction and for options pricing.

Most of the authors previously cited, as White, Grudnitski and Osburn, Refenes, Zapranis and Francis, Oh and Kim, Hyun-jung Kim e Kyung-shik Shin, Cao, Leggio and Schniederjans and Pacelli have proposed a neural network model applied to stock market prediction.

The ability of neural networks to discover nonlinear relationships in input data makes them ideal for modelling nonlinear dynamic systems such as the stock market.

Various neural network configurations have been also developed to determine the validity of the efficient market hypothesis or to compare the neural models with statistical methods such as regression.

About the learning algorithm which can be used in the training of the ANN_m, back-propagation networks are the most commonly used networks because they offer good generalization abilities and are relatively straightforward to implement. Although it may be difficult to determine the optimal network configuration and network parameters, these networks offer very good performance when trained appropriately.

Other authors, as Anders, Korn and Schimitt, Corazza and Gobbo, Andreou, Charalambous and Martzoukos, Po-Chang Ko, Wang, and Liang, Zhang, Xiao and Chen, have considered neural networks applied to evaluating financial options.

The neuro-computational approach for the evaluation of financial options is a viable alternative to more traditional valuation techniques, such as those based on closed-form expressions and numerical techniques proposed in the literature[21].

Nowadays neural networks appear to be the best modelling methods currently available in lots of the sector of finance to analyze the phenomenon of pricing as they are able to capture nonlinearities in the system without human intervention. However, these computational methods have several limitations:

- the demand for a considerable amount of data to obtain satisfactory results;
- the overfitting problem, because of which the model tends to transpose the part "noisy" data;
- being a black box, it is not always interpretable the financial relation between the value of each explanatory variable and, therefore, its significance for the assessment.

Then it seems to be particularly promising a hybrid approach to the pricing in banking and finance in which neural networks and non linear models can work together with linear models.

REFERENCES

Amjady N., Keynia F. (2009), "Day-ahead price forecasting of electricity markets by a new feature selection algorithm and cascaded neural network technique", in *Energy Conversion and Management*, Volume 50, Issue 12, December, pp. 2976-2982.

Anders U., Hann T. H., Nakaheizadeh G. (1997), "Testing for Nonlinearity with Neural Networks", in Weigend A. S., Abu Mustafa Y., Refens A. P. N. (a cura di), "Decision Technologies for Financial Engineering", in *World Scientific*, Singapore.

[21] See Boyle (1977).

Andreou P. C., Charalambous C., Martzoukos S. H. (2008), "Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters", in *European Journal of Operational Research*, Volume 185, Issue 3, 16 March, pp. 1415-1433.

Banca d'Italia (2008), *Fondi comuni italiani: situazione attuale e possibili linee di intervento,* Rapporto del gruppo di lavoro sui fondi comuni italiani, luglio.

Black F., Scholes M. (1973), "The pricing of options and Corporate Liabilities", in *Journal of Political Economy.*

Black F., Cox J. C. (1976), "Valuing corporate securities: some effects of bond indenture provisions", in *Journal of Finance*, n. 31, pp. 351-367.

Boyle P. P. (1977), "Options: A Monte Carlo Approach", in *Journal of Financial Economics* , n. 4, pp. 323-338.

Broadie M., Glasserman P. (1996), "Estimating Security Price Derivatives Using Simulation", in *Management Science*, n. 42, pp. 269-285.

Cao Q., Leggio K. B., Schniederjans M. J. (2005), "A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market", in *Computers & Operations Research*, Volume 32, Issue 10, October, pp. 2499-2512.

Cetinã I., Mihail N. (2007), "Price Strategies in Banking Marketing", in *Economie teoreticã si aplicatã.*

Cetinã I. (2005), *Marketing financiar bancar,* Editura Economica, Bucaresti.

Chen H. S. (2007), "Computationally intelligent agents in economics and finance", in *Information Sciences*, Volume 177, Issue 5, 1 March, pp. 1153-1168.

Collin-Dufresne P., Goldstein R. (2001), "Do credit spreads reflect stationary leverage ratios?", in *Journal of Finance*, n. 56, pp. 1929-1957.

Corazza M., Gobbo M. (2002), "Reti Neurali Artificiali per la valutazione di opzioni", in *Working Paper* n. 02, 06/Maggio.

Cox J. C., Ross S. A., Rubistein M. E. (1979), "Option Pricing: A Simplified Approach", in *Journal of Financial Economics.*

De Rossi F. M., Gariboldi D., Leggieri G., Russo A. (2008), "Il marketing dei fondi comuni italiani. Modelli organizzativi, costi, andamento e nuove prospettive conseguenti all'introduzione della Mifid nell'ottica della vigilanza", *Consob, Quaderni di finanza*, n. 61, gennaio.

Draghi M. (2007), *Considerazioni finali,* Assemblea ordinaria dei partecipanti.

Duffie D., Singleton K. J. (1999), "Modeling term structures of defaultable bonds", in *Review of Financial Studies*, n. 12, pp. 687-720.

Eom Y. H., Helwege J., Huang J. Z. (2004), "Structural models of corporate

bond pricing: an empirical analysis", in *Review of Financial Studies*, n.17, pp. 499-544.

Esfahanipour A., Aghamiri W. (2010), "Adapted Neuro-Fuzzy Inference System on indirect approach TSK fuzzy rule base for stock market analysis", in *Expert Systems with Applications*, Volume 37, Issue 7, July, pp. 4742-4748.

Fethi M. D., Pasiouras F. (2010), "Assessing bank efficiency and performance with operational research and artificial intelligence techniques: A survey", in *European Journal of Operational Research*, Volume 204, Issue 2, 16 July, pp. 189-198.

Fondo interbancario di tutela dei depositi (2001), *Misurare e gestire il rischio di credito nelle banche,* Alpha test, capitolo 3.1.

Geske R. (1977), "The valuation of corporate liabilities as compound options", in *Journal of Financial and Quantitative Analysis*, n. 12, pp. 541-552.

Grudnitski G., Osburn L. (1993), "Forecasting S&P and Gold Futures Prices: An Application of Neural Networks", in *Journal of Futures Markets*, Volume 13, n. 6, pp. 631-643.

Gualandri E. (2008), "Il pricing nel settore finanziario: la prospettiva dei costi della regolamentazione e della vigilanza", in *Bancaria,* n.3.

Hyun-jung Kim, Kyung-shik Shin (2007), "A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets", in *Applied Soft Computing*, Volume 7, Issue 2, March, pp. 569-576.

Hull J. C. (1997), *Option, Futures and other derivates,* Prentice Hall.

Jarrow R. A., Protter P. (2004), "Structural versus reduced form models: a new information based perspective", in *Working paper*.

Lando D. (1994), *Three Essays on Contingent Claim Pricing*. Ph.D. dissertation, Cornell University.

Lawrence R. (1997), "Using neural networks to forecast stock market prices", *Department of Computer Science University of Manitoba*.

Leland H., Toft K. (1996), "Optimal capital structure, endogenous bankruptcy, and the term structure of credit spreads", in *Journal of Finance*, n. 51, pp. 987-1019.

Liang X, Zhang H., Xiao J., Chen J. (2009), "Improving option price forecasts with neural networks and support vector regressions", in *Neurocomputing*, Volume 72, Issues 13-15, August.

Linciano N., Marrocco. E. (2002), "Fondi di fondi e accordi di retrocessione sulle scelte di investimento e sui costi a carico dei patrimoni gestiti", in *Quaderni di finanza*, n. 49, gennaio.

Longstaff F, Schwartz E. (1995), "A simple approach to valuing risky fixed and floating rate debt", *Journal of Finance*, 50, 789-819.

Longstaff F. A., Schwartz E. S. (2001), "Valuing American options by simulation: a simple least squares approach", in *Review of Financial Studies*, n. 14, pp. 113-148.

Merton R. C. (1974), "On the pricing of corporate debt: the risk structure of interest rates", in *Journal of Finance*.

Oh K. J., Kim K. J. (2002), "Analyzing stock market tick data using piecewise nonlinear model", in *Expert Systems with Applications*, Volume 22, Issue 3, April, pp. 249-255.

Pacelli V. (2009), "An Intelligent Computing Algorithm to Analyze Bank Stock Returns", in Aa. Vv. (eds) (2009), *Emerging Intelligent Computing Technology and Applications*, Lectures Notes on Computer Sciences, n. 5754, Springer Verlag, New York.

Po-Chang Ko (2009), "Option valuation based on the neural regression model", in*Expert Systems with Applications*, Volume 36, Issue 1, January , pp. 464-471.

Refenes A. P., Zapranis A., Francis G. (1994), "Stock performance modelling using neural network; a comparative study with regression model", in *Neural Networks*, Volume 7, pp. 375-388.

Resti A., Sironi A. (2008), *Rischio e valore nelle banche,* Egea, Milano.

Romelli D. (2007), "Pricing e Basilea 2. Quali effetti sui tassi di interesse?", in *Quaderni di Ludovicianum,* Università Cattolica del Sacro Cuore, Milano.

Saa-Requejo J., Santa-Clara P. (1999), "Bond pricing with default risk", in *Working paper*.

Sabbatini R. (2008), "Risprmio: inizia l'era dei fondi quotati", in *Il Sole 24 Ore*, agosto.

Saita F., Resti A. (2009), “Prestiti bancari, rating interni e modelli Var: quale autonomia di pricing per le unità operative?”, in *Bancaria*, febbraio n. 2.

Saunders A. (1996), *Financial Institution Management*, II. Ed., Irwin, Boston.

Sironi A., Maccario A., Zazzara C. (2002), “Is bank’s cost of equity capital different across countries? Evidence from the 10 countries’ major banks”, in *WP* n.77, SDA Bocconi Research Division.

Thiagarajah K., Appadoo S. S., Thavaneswaran A. (2007), “Option valuation model with adaptive fuzzy numbers”, in *Computers & Mathematics with Applications*, Volume 53, Issue 5, March, pp. 831-841.

Trippi R. R., Turban E. (1996), *Neural Networks in Finance and Investing*, Irwin Professional Pub., New York.

Ursino G. (2006), *Falsa partenza per il turnover di portafoglio,* Plus24, Il Sole 24 Ore, marzo.

Ursino G. (2009), *Il turnover di portafoglio imbocca il viale del tramonto,* Plus24, Il Sole 24 Ore, aprile.

Wang H. Y. (2009), “Nonlinear neural network forecasting model for stock index option price: Hybrid GJR–GARCH approach”, in *Expert Systems with Applications*, Volume 36, Issue 1, January, pp. 564-570.

White H. (1988), “Economic Prediction Using Neural Networks: The Case of IBM Stock Prices”, in *Proceedings of the Second Annual IEEE Conference in Neural Networks*, pp. 451-458.

Zapranis A., Alexandridis A. (2009), “Weather derivatives pricing: Modeling the seasonal residual variance of an Ornstein–Uhlenbeck temperature process with neural networks”, in *Neurocomputing*, Volume 73, Issues 1-3, December, pp. 37-48**.**

INDEX

A

acid, 305
acidic, 170
active compound, 306
activity level, 84, 88, 114, 156
actual output, 142, 231
actuation, 26
actuators, 208
adaptability, 296, 299
adaptation, 48, 49, 234, 283, 291, 293
additives, 171
adhesion, 24
adjustment, ix, 77, 85, 89, 95, 99, 157
adsorption, 78, 79
age, 42, 184, 272
aggregation, 191, 262, 271
AIDS, xiv, 295, 300
air quality, 80
air temperature, xi, 151, 152, 153, 154, 161, 162, 163
Algeria, 109
allele, 264, 268
amino, 129, 131, 136, 305
amino acid, 129, 131, 136
amino acids, 129, 131, 136
amyotrophic lateral sclerosis, 191
anaerobic digestion, 205
aniline, 307
annealing, 303
antioxidant, 309
antiviral agents, 313
arbitrage, 319
aromatic compounds, 315
artificial intelligence, xiv, 186, 188, 192, 219, 274, 295, 296, 319, 323, 325, 327
Artificial Neural Networks, i, iii, v, vi, xii, xiv, 12, 23, 38, 41, 56, 57, 59, 60, 62, 75, 77, 91, 100, 120, 123, 139, 141, 167, 188, 189, 190, 207, 208, 222, 229, 257, 277, 278, 279, 281, 283, 285, 287, 289, 291, 293, 295, 314, 317, 321, 324
aspartate, 307
aspiration, 143, 144
assessment, viii, 41, 153, 185, 270, 323, 326
atmosphere, 78
auditory stimuli, 42
Austria, xi, 56, 139, 140, 147, 148, 182
automation, 208, 219
autonomy, 258

B

background noise, 95
bacteria, 201
Bahrain, 1
bandwidth, 298
bank failure, 324
banking, xiv, 317, 318, 319, 324, 326
bankruptcy, 328
banks, 322, 329
base, 36, 118, 171, 189, 205, 275, 302, 306, 323, 327
bauxite, 79
beer, xi, 139, 140, 147, 148
behaviors, 194
Beijing, 38
benchmarking, 258, 270
bending, 211
benefits, ix, 3, 109, 171
benign, 284
benzodiazepine, 308, 309
bias, 26, 113, 196, 279, 280, 284, 299
biodegradability, 197
biogas, 205
bioinformatics, 131
biological activities, 301

biological activity, 297, 299, 302, 303
biological systems, 2, 13, 298
biomass, 152, 194, 201, 205
biopolymer, 38
blends, 29, 37
blood, 309
BMI, 184
bonds, 327
bounds, 246, 261, 294
brain, 13, 43, 46, 56, 62, 149, 228, 278, 298
brain functions, 43, 46
brain growth, 56
branching, 173
Brazil, 153, 166
breeding, 26, 264
Brno, 292
building blocks, 42
butadiene, 27

C

Cairo, 57
calcium, 79, 80, 310
calibration, 3, 8, 93, 194, 293
cancer, 183, 184
candidates, 146, 264, 322
capillary, 272
capillary refill, 272
carbon, 27, 61, 75
cardiovascular disease, 184, 192
cardiovascular risk, 185
case study, 5, 255
casting, 171
catchments, 182
cation, 246
causal relationship, 113
CBS, 166
CDC, 21
CEC, 292
cell body, 45, 48, 124
cellulose, 81
central nervous system, 13, 298
challenges, 278
chaos, 191
chaotic behavior, 189
chemical, x, 13, 37, 78, 79, 80, 123, 124, 130, 131, 171, 172, 173, 194, 195, 303, 311
chemical industry, 124
chemical properties, 130
chemical reactivity, 311
chemicals, 78, 307, 314
chemometrics, 296, 301, 308
Chicago, 323
Chicago Mercantile Exchange, 323
China, 23, 29, 38, 166, 321
chromatograms, 131
chromatographic technique, 130, 136
chromatography, vii, x, 123, 124, 129, 130, 131, 136, 310, 311
chromosome, 264
classes, 2, 118, 236, 237, 271
classical conditioning, viii, 42, 43, 45, 46, 48, 54
classification, 13, 43, 46, 75, 84, 113, 150, 155, 222, 231, 236, 245, 246, 247, 248, 249, 250, 252, 254, 258, 271, 272, 280, 284, 305
climate, 88, 153
cluster analysis, 274
clustering, 100, 188, 274, 275, 298, 303, 314, 323
coding, 259, 264, 271, 274
cognition, 42
cognitive impairment, 191
cognitive science, viii, 41, 42
cognitive style, 55
combustion, 78
combustion processes, 78
commercial, 36, 78, 184
communication, xiv, 44, 113, 295, 298, 301, 306, 318
communities, 2
community, 2, 259
comparative analysis, 312
compatibility, 31
competing interests, 191
competition, xiv, 24, 298, 317, 318
complement, 88
complex interactions, 264
complexity, vii, 1, 110, 112, 186, 189, 190, 191, 194, 201, 286, 297
composites, vii, viii, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35, 36, 37, 59, 74, 75, 76, 179
composition, xiii, 27, 31, 129, 131, 132, 194, 201, 257, 258
compounds, xiv, 131, 205, 295, 301, 303, 304, 305, 307, 308, 313, 314
compression, 2, 3, 78, 94, 175
computation, 3, 26, 75, 118, 174, 189, 258, 259, 264, 275, 279, 282, 285, 299
computational performance, 118
computer, vii, viii, xi, 2, 25, 42, 52, 92, 125, 131, 169, 171, 172, 177, 189, 211, 255, 284, 298, 300, 306
computer technology, 92
computing, 2, 3, 12, 26, 84, 90, 112, 113, 122, 167, 172, 186, 275, 276, 279, 281, 283, 296, 298, 299, 322
condensation, ix, 77, 83, 84, 86, 87, 88, 89

conditioned response, 47, 50, 51
conditioned stimulus, 47
conditioning, 78, 88, 90
conduction, x, 109
conductivity, 195, 201
conference, 21, 57, 179, 219, 256
configuration, 3, 15, 81, 97, 115, 210, 325, 326
congress, 219, 292
connectivity, 49
constituents, 307
constructed wetlands, 205
construction, xi, 140, 150, 169, 170, 171
consumers, 318
consumption, xi, 78, 88, 139, 140, 147, 148
contact time, 81
containers, 173
contamination, 78
continuous data, 93
contour, viii, 23, 33
convergence, xiii, 46, 49, 52, 54, 55, 86, 116, 157, 257, 258, 287, 294
cooling, ix, 77, 78, 79, 80, 84, 86, 89, 90
correlation, x, 47, 50, 51, 52, 61, 72, 74, 123, 128, 129, 135, 152, 153, 172, 176, 196, 254, 297, 302, 303, 304, 305, 314, 319, 322
correlation analysis, 322
correlation coefficient, x, 72, 74, 123, 152, 196, 305
correlations, 166, 196, 309
corrosion, x, 80, 123, 124, 125, 127, 128, 129, 136
cost, x, xiii, xiv, 24, 79, 80, 81, 85, 109, 110, 157, 208, 257, 296, 329
covering, 81
CPU, 117
cracks, 75, 299
creativity, 55
creditworthiness, 324
Croatia, 123
crops, 78
cross-validation, 303, 305
crystal structure, 306
crystallization, 80
cutting force, 61, 75
cycles, 54, 55, 324
Cyprus, 56
cytochrome, 308
cytology, 190
cytotoxicity, 309
Czech Republic, 292

D

danger, 303
Darwinian evolution, 264
data analysis, 172, 176, 269
data collection, 153, 189
data gathering, 16
data mining, 190
data processing, 62, 299
data set, ix, xi, xiii, 64, 66, 86, 91, 93, 100, 112, 151, 175, 176, 188, 189, 190, 198, 216, 224, 227, 229, 230, 231, 236, 237, 246, 247, 248, 249, 250, 251, 252, 253, 254, 258, 259, 270, 271, 272, 274, 310
data structure, 305
database, 56, 112, 120, 153, 165, 178, 188, 303, 304, 312
decay, 46
defects, viii, 59
deficiency, 300
degradation, 310
dementia, 191
dendrites, 124
dependent variable, 239, 240, 241, 243
depression, 184
depth, 3, 93, 298
derivatives, 122, 179, 292, 302, 303, 304, 307, 308, 309, 313, 314, 315, 323, 329
destiny, 184
detection, 60, 75, 95, 190
developing countries, 152
deviation, 53
dew, 78, 79
DFT, 308
diet, 185
diffusion, 110, 117
diffusivity, 170
diluent, 32
dimensionality, 49, 52, 260, 282, 299, 303, 305
disability, 46
discharges, 197, 200
discrete variable, 257
discretization, 95
diseases, 184, 185, 189, 284
disorder, 188
displacement, 214
disposition, 188
dissolved oxygen, 195
distortions, 12, 94
distribution, 53, 54, 55, 111, 117, 184, 185, 238, 260, 311, 318
diversification, 143, 144, 147
diversity, 258
DNA, 284
doctors, 190
DOI, 21
draft, 291
drainage, xii, 181, 182, 183, 186, 187, 188

drawing, 79, 124, 186
drug design, xiv, 295, 298, 301
drug discovery, xiv, 295, 296, 306
drug interaction, 312
drugs, 305, 310, 314
drying, 78
duality, 212
durability, xi, 169, 170, 171
dynamic control, 17

E

economics, 124, 131, 327
educational process, 43
educational psychology, viii, 41
effluent, xii, 193, 194, 195, 196, 197, 199, 200, 201, 203, 204
Egypt, 41, 57, 59, 153, 166
elastomer compositions, 37
elastomers, 24, 39
e-learning, 55
electricity, xi, 139, 140, 147, 148, 152, 153, 322, 326
electron, 171, 173, 178, 179
elementary school, 42
elongation, 24, 29, 31, 33, 34, 35, 36, 37
emerging markets, 321
emission, x, 60, 61, 123, 125, 126, 136, 150, 272
employment, 21
encoding, 302
encouragement, 307
energy, ix, 79, 80, 109, 111, 152, 153, 166
engineering, x, xi, 26, 29, 60, 116, 131, 139, 169, 171, 172, 194, 222
entropy, 140, 150, 271
environment, xii, 2, 24, 54, 61, 92, 93, 172, 181, 184, 185, 186, 320
environmental conditions, 43, 46
environmental factors, 187
environmental stimuli, 43
epidemic, 312
epipolar geometry, 3, 9
epipolar line, 10
equality, 261, 262, 269
equilibrium, 110, 112, 190
equipment, ix, 77, 80, 208
equity, 329
estrogen, 311
ethylene, vii, 23, 26, 27
Europe, 153, 165, 182
evidence, 184, 191, 275, 302, 319
evolution, vii, 26, 118, 131, 187, 188, 292, 293, 294, 311
evolutionary computation, 229, 256
examinations, 188
exchange rate, 221, 222, 224, 227
excitation, 13, 213, 216
execution, 28, 173
exercise, 319
experimental design, 24, 36, 132
expert systems, xiv, 255, 317
exposure, 185, 230
external validation, 303, 308
extraction, 130, 188, 255
extrusion, 31

F

fabrication, 38
factor analysis, 297
families, 262, 282
feature selection, 303, 308, 322, 326
features extraction, 17
feces, 272
fetal growth, 191
fiber, viii, 59, 60, 61, 63, 75
fibers, 60
fidelity, 116
film thickness, 114, 116
films, 110
filters, 94
financial, xiv, 284, 317, 320, 326
financial system, xiv, 317
fish, 78
fitness, 26, 29, 131, 229, 264, 279, 280, 281, 282, 283, 284, 286, 287, 288, 289, 291, 319
flexibility, x, xiii, xiv, 29, 109, 118, 207, 208, 211, 295, 299, 302
flooding, 80
flour, 78
fluctuations, 319
fluidized bed, 194
fluorescence, 308, 311
fluorescence decay, 308
football, 188
force, viii, 59, 60, 61, 62, 63, 64, 65, 66, 67, 70, 71, 72, 73, 74, 75, 265, 296
forecasting, x, xiii, 26, 139, 140, 141, 142, 147, 149, 172, 179, 196, 205, 221, 222, 223, 226, 227, 228, 254, 292, 319, 320, 322, 323, 324, 326, 329
formation, 30
formula, 224, 226, 267, 279, 281, 321
foundations, 276, 292
freedom, 92, 188, 208, 218
freezing, 98
friction, 208, 209, 210
fruits, 78

fuel consumption, 78
function values, 143
fungus, 38
fusion, 60, 61, 75, 297, 301, 313
fuzzy set theory, 115, 321
fuzzy sets, 262, 270

G

gastritis, 192
GCE, 271, 272
gel, 79
gene expression, 296, 305, 310, 315
genes, 264, 265
genetic background, xii, 181, 183, 184, 186, 191
genetic predisposition, xii, 181, 184, 185
genetic programming, 297
genetic testing, xii, 181, 184
genome, 184, 297
genomics, 184, 188
geometry, 3, 5, 9, 60, 61, 75, 76, 189, 259
Germany, 56, 165, 182, 183, 291, 294
gland, 45, 47
glucosidases, 312
glycol, 80, 90
graph, 224, 225, 226, 227
graphite, 60, 75
grass, 299
gravity, 208, 209
grids, 81
grounding, 230
groundwater, 194, 205
grouping, 95
growth, x, xii, 109, 191, 193, 194, 245, 259
growth factor, 191
guidelines, 298

H

HAART, 300, 301, 312
health, xii, 125, 181, 183, 184, 186, 187, 188, 191
health status, 187
heart disease, 190
hedging, 321, 323
height, 61, 70, 260
hepatitis, 150
heterogeneity, 60, 257, 258, 262, 270, 272, 274
heteroskedasticity, 224, 225, 227
high strength, 170
histamine, 308
histidine, 310
histogram, 173
HIV, xiv, 295, 298, 300, 304, 305, 308, 309, 310, 312, 313, 314, 315
HIV inhibitors, 312
HIV/AIDS, 312
HIV-1, vi, xiv, 295, 298, 300, 309, 312, 313, 314, 315
Hong Kong, 322
hormone, 307, 310
horses, 274
host, 300
human, 12, 62, 92, 125, 141, 172, 191, 278, 282, 296, 300, 308, 310, 326
human brain, 13, 141
human health, 125
humanism, 191
humidity, ix, xi, 77, 78, 79, 82, 83, 84, 86, 87, 88, 89, 151, 152, 153, 154, 155, 159, 160, 163, 164, 165, 166
Hungary, 182
hybrid, vii, xiii, 23, 26, 29, 36, 75, 90, 116, 221, 222, 223, 224, 225, 226, 227, 228, 263, 303, 321, 323, 326, 328
hydrocarbons, 307
hydrogen, 304
hydrophobicity, 302, 304
hypercube, 246
hypothesis, 264, 319, 326

I

identification, 90, 94, 129, 130, 167, 172, 184, 191, 209, 298
identity, 189, 259
ill-defined problems, ix, 77, 81
image, ix, xi, 2, 3, 5, 7, 8, 9, 10, 11, 12, 15, 17, 19, 22, 49, 91, 92, 93, 94, 95, 101, 169, 171, 173, 175, 177, 178, 302
image analysis, 171, 173, 177, 178, 302
image interpretation, 94
images, 10, 92, 93, 94, 173, 178
immersion, 126, 127
immune system, 300
imprinting, 310
improvements, ix, 109
income, xiv, 317
India, 56, 295
individuals, 131, 185, 188, 264, 265
induction, 186
industries, viii, 59, 78, 197
industry, x, 24, 60, 78, 92, 109, 320
inertia, 209
infection, 312
inflation, 228

information processing, 13, 259
infrared spectroscopy, 308
ingredients, 171
inhibition, 13
inhibitor, 308, 312, 313
initial state, 142
insertion, 255
institutions, xiii, 221
insulin, 191
integration, vii, x, 1, 13, 109, 259, 300
intelligence, 84, 110, 155, 219, 298, 306
intelligent systems, 256
interest rates, 328
interface, 24, 93, 101, 102
internal environment, 12
interpretability, xiii, 257, 258, 263, 272, 297
interrelations, 49
intervention, 92, 191, 326
intrinsic viscosity, 311
inversion, 209, 212, 213, 217, 219
ions, 126
Iraq, 207
IRC, 38
iris, 246, 247, 248, 249, 250, 252
irradiation, 153, 165, 166
issues, vii, 1, 55, 171, 275, 323
Italy, 21, 22, 181, 317
iteration, 86, 131, 140, 143, 144, 146, 147, 149, 157, 282, 284, 288, 289, 291

J

Japan, 21, 218
joints, xiii, 15, 17, 95, 207, 208, 209
justification, 55

K

Korea, 23, 27

L

labeling, 95
landscape, xii, 181, 183, 184, 186
latency, 46
lattices, 276
LEA, 109
leaching, 125, 128
lead, 30, 74, 170, 171, 173, 175, 177, 222, 223, 267, 297, 298, 300, 301, 305, 320, 323
Leahy, 140, 150
leakage, 48, 114
learning environment, 54
learning process, xii, 13, 44, 45, 48, 52, 55, 62, 114, 175, 207, 212, 259, 260
learning styles, 52, 55
learning task, 259, 275
Least squares, 309
legislation, 318
ligand, 303, 304, 305, 312
light, 306
lignin, 173
linear function, 142, 175, 208, 223
linear model, 222, 299, 305, 319, 320, 326
linear systems, 208, 297
linguistics, 42, 46
liquid chromatography, x, 123, 130
lithium, 79, 80, 90
low temperatures, 111
Luo, 38
lymphocytes, 300

M

machine learning, 275, 296, 297, 305
magnitude, 118, 196, 319
majority, 128, 129, 247, 272, 274, 319
Malaysia, 207
mammography, 190
management, xii, 39, 172, 191, 193, 194, 201, 202, 204
manipulation, 3, 264, 268
manpower, 160
manufacturing, 26, 131, 208
mapping, 17, 21, 84, 115, 116, 117, 122, 130, 155, 223, 234, 258, 260, 261, 262, 278, 279, 298
marketing, xiv, 317, 318, 327
marketing mix, xiv, 317, 318
marketing strategy, xiv, 317
mass, ix, 24, 77, 81, 82, 89
material surface, 81
materials, vii, viii, x, xi, 23, 26, 36, 42, 43, 54, 55, 59, 60, 61, 63, 75, 78, 79, 80, 123, 125, 127, 169, 170, 171, 172, 173, 177, 178
mathematical methods, 131, 297
mathematics, 131, 188, 189
matrix, vii, 1, 3, 6, 7, 8, 10, 11, 12, 15, 47, 50, 51, 52, 60, 61, 93, 94, 95, 209, 259
matter, 184, 185, 234, 240, 245, 260, 274
measurements, 27, 55, 112, 117, 125, 152, 165, 166, 178, 195, 254, 274, 284, 323
mechanical properties, vii, 23, 26, 27, 30, 31, 36
mechanical stress, 31
media, xi, 81, 169, 170

median, 260, 266, 267
medical, xii, 4, 181, 183, 188, 189, 190, 191
medicine, xii, 181, 183, 185, 186, 188, 189, 190
Mediterranean, 56, 182, 183, 184, 185
melanin, 307
melatonin, 310
memory, 42, 43, 46, 47, 50, 51, 52, 55, 119, 142, 143, 144, 276, 299
memory function, 43, 46
Metabolic, 13
metal ion, 125, 126, 127, 128
metallurgy, 75
metals, 128, 129
metaphor, xii, 181, 183
methodology, vii, viii, xi, 1, 4, 21, 26, 28, 29, 36, 41, 42, 47, 84, 141, 151, 228, 307
MFI, 27
microorganisms, 194
microscopy, 179
microstructure, xii, 170, 171, 172, 173, 177, 178
microstructures, xi, 169, 179
migration, 17
military, viii, 59
miniature, 298
misconceptions, 282
mission, 188
mixing, 263, 265
modelling, 43, 149, 179, 189, 190, 205, 219, 228, 296, 302, 303, 306, 319, 326
models, ix, x, xiii, xiv, 17, 29, 49, 55, 60, 77, 81, 101, 103, 106, 110, 111, 112, 116, 117, 118, 120, 130, 139, 141, 152, 153, 164, 165, 166, 189, 194, 221, 222, 224, 226, 227, 230, 257, 258, 262, 263, 275, 277, 297, 298, 303, 305, 306, 307, 308, 310, 315, 317, 319, 320, 321, 322, 324, 325, 326, 327, 328
modern science, 136
modifications, 171, 297
modules, 153
moisture, 60, 78, 79, 81
moisture content, 78, 81
molecular fingerprints, 305
molecular structure, 301, 309
molecules, 79, 298, 303, 304
momentum, 62, 86, 157
morbidity, 189
morphology, 127
mortality, 189
motion control, 2, 93
motivation, 255
motor neurons, 47
mucous membrane, 272
mucous membranes, 272
multimedia, viii, 42, 55
multiple factors, 190
multiple regression, 302, 322
municipal solid waste, 194, 205
mutation, 229, 265, 266, 268, 280
mutations, 265, 300, 305
myocardial infarction, 184, 190

N

nanodevices, 110
nanometer, 112
nanotechnology, 112
naphthalene, 173
nasogastric tube, 272
National Institutes of Health, 42
natural evolution, 26
natural selection, 26, 264
nerve, 13
nervous system, 12, 13, 60, 141, 172, 324
networking, 299
neural connection, 274
Neural Network Model, 56, 84, 324
neural systems, 150, 228, 274
neurobiology, 45
neurons, x, 12, 13, 14, 26, 45, 47, 48, 62, 84, 85, 86, 97, 98, 113, 115, 124, 126, 128, 129, 132, 135, 139, 140, 141, 142, 143, 144, 147, 150, 155, 156, 157, 158, 160, 172, 174, 175, 196, 216, 223, 225, 226, 229, 230, 231, 232, 234, 235, 236, 237, 238, 239, 245,맴246, 247, 248, 249, 250, 251, 252, 253, 254, 259, 263, 271, 272, 275, 298, 299, 324
neuroscience, viii, 41
neurotransmitter, 13
neurotransmitters, 13
nitrogen, 195
nodes, 26, 49, 99, 141, 172, 278, 279, 284, 298, 299, 324, 325
nonlinear dynamic systems, 326
nonlinear dynamics, 191
nonlinear systems, 26, 115
non-nucleoside reverse transcriptase inhibitors, 301, 312, 313
North America, 182, 292
nucleoside reverse transcriptase inhibitors, 301, 312

O

operations, 119, 175, 176
opportunities, 92, 171, 321
optical microscopy, 171

optimization, x, xiv, 26, 29, 36, 61, 75, 84, 112, 115, 116, 117, 120, 123, 124, 130, 131, 132, 136, 140, 142, 149, 150, 155, 196, 222, 228, 264, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 306, 321
optimization method, xiv, 277, 278, 280, 282, 283, 284, 285
organic chemicals, 307
organic compounds, 308, 309, 311
organism, 188, 194
originality, 46
oscillation, 86, 157
overlap, 114, 269, 270, 303, 304
overlay, 131
oxide thickness, 114, 115, 116
oxygen, 195

P

Pacific, 182
pain, 272
paradigm shift, 190
parallel, 7, 10, 62, 97, 118, 172, 190, 298
parents, 264, 265, 266, 267, 268
Pareto, 61
participants, 54
partition, 236, 309, 311
pathways, 46
pattern recognition, 13, 26, 90, 92, 167, 297
Pavlovian conditioning, 47, 50
Pavlovian learning, 47, 49, 50
PCA, 302, 314
PCDD/Fs, 310
PCR, 305
peptides, 309, 310
performance measurement, 55
performance modelling, 319, 328
peristalsis, 272
permeability, 170, 304
peroxide, 37
Perth, 256
Petroleum, 77, 88, 151, 164
pH, 195, 199, 200, 204, 272
pharmaceutical, 78
pharmacokinetics, 296
Philadelphia, 90
phosphate, 173
phosphorous, 195, 205
physical characteristics, 185
physical phenomena, 110
physical properties, 307
physicochemical properties, 305
physics, x, 109, 110, 131
plants, 194, 204, 205
plasticizer, 24, 32
plastics, 75, 78
PLS, 296, 297, 301, 302, 303, 308, 309, 314
Poisson equation, 110, 112
pollution, 152
polycyclic aromatic hydrocarbon, 310
polymer, 27, 60, 74, 173, 310, 311
polymer composites, 60, 74
polymer solutions, 311
polymerase, 301
polymers, vii, 23
polynomial functions, 134, 135
polyolefins, 37
polypropylene, vii, 23, 24, 26, 27, 31, 37
population, 26, 131, 185, 264, 265, 280, 281, 285, 286
population size, 264, 280, 285, 286
porosity, 170, 171
Portugal, 182
potato, 78
precipitation, 152
predictability, 298, 305, 323
prediction models, 205
premature death, xii, 181, 183
preparation, iv, 130, 178, 179, 275
price changes, 319
primary school, 46
principal component analysis, 175, 314
principles, 26, 110, 229, 306
prior knowledge, 194, 259
probability, xi, 26, 169, 173, 184, 185, 189, 191, 237, 238, 239, 242, 245, 251, 253, 265, 266, 267, 268, 280, 281, 282, 286, 325
probability distribution, 265, 266
probability theory, 189, 191
processing stages, 93
profit, xiv, 317
profitability, 320
programming, 214, 284, 292, 296, 297, 305, 310, 311, 315
project, 24, 296, 309, 319
promoter, 184
pronunciation, 47
propagation, viii, 59, 61, 62, 74, 92, 114, 142, 174, 175, 196, 197, 199, 201, 223, 231, 232, 235, 268, 292, 298, 299, 303, 304, 307, 320, 324, 326
proportionality, 49
protease inhibitors, 301, 313
protection, 24
proteins, 129, 300
prototypes, xiii, 257
pruning, 140, 150

Q

quantification, 179
quantization, 93
quantum confinement, 110
query, 284, 298

R

radiation, xi, 151, 152, 153, 154, 155, 158, 160, 161, 162, 163, 164, 165, 166
radius, 63, 233, 234
rainfall, 205
ramp, 48
rancid, 78
random walk, 319
raw materials, 78
real numbers, 264, 265, 267, 274
real time, 2, 92, 112, 147, 149
receptors, 47, 311
recognition, ix, 28, 43, 47, 49, 54, 91, 92, 93, 113, 191, 255, 256
recombination, 265, 267, 268, 276
recommendations, iv, 271
reconstruction, 188
rectal temperature, 272
rectification, 11
recycling, 24
reductionism, 191
redundancy, 43, 46
regenerate, 79
regeneration, 79, 80, 90
regression, 24, 61, 62, 98, 153, 165, 178, 196, 198, 202, 203, 204, 271, 272, 296, 297, 298, 299, 301, 302, 303, 306, 307, 308, 309, 310, 311, 319, 320, 322, 326, 328
regression analysis, 61, 62, 303, 306
regression equation, 24, 299
regression method, 307
regression model, 153, 165, 319, 320, 322, 328
reinforcement, 46, 219
reinforcement learning, 46, 219
relevance, 120, 259, 293
reliability, 296
remote sensing, 150
renewable energy, 219
repellent, 307
repetitions, 147
replication, 300, 313
reproduction, 229
requirements, 79, 118
researchers, xi, xiii, 12, 42, 46, 140, 169, 170, 175, 177, 196, 208, 209, 221, 230, 281, 283, 306
residual error, 115, 299
resistance, xiii, 170, 229, 300, 312
resolution, x, 123, 131, 134, 185
resources, 113
respiratory rate, 272
response, 2, 24, 28, 44, 45, 46, 47, 48, 49, 50, 51, 52, 55, 61, 98, 100, 131, 211, 213, 263, 274, 324, 325
restoration, 94
retardation, 173
retroviruses, 300
reverse transcriptase, 300, 308, 312, 313, 314
rings, 81
risk, 184, 185, 188, 190, 192, 283, 297, 318, 328
risk factors, 185
robotics, vii, 1, 2, 3, 5, 11, 12, 15, 17, 21, 26, 92, 209, 219
root, 140, 198, 226, 261, 272, 299
roughness, viii, 59, 61, 62, 64, 70, 72, 74, 75
roughness measurements, 64
rubber, vii, 23, 24, 26, 27, 31, 37, 39
rubbers, 24
rules, 115, 116, 141, 187, 188, 294, 296, 323

S

saturation, 48
Saudi Arabia, xi, 41, 77, 88, 151, 153, 160, 164, 165, 166
scaling, ix, 109
scanning electron microscopy, 171, 179
scattering, 112
Schrödinger equation, 112
science, x, 131, 139, 189
scientific investigations, 124
scope, xi, 81, 169, 172, 173, 265
sea level, 153
seasonal component, 323
seasonality, 323
securities, 323, 327
sedimentation, 195
selectivity, 308
self-study, 45
semantics, 188, 262, 275
semiconductor, x, 109
sensing, 3, 219
sensitivity, 11, 60, 231, 232, 234, 238, 239, 244, 245, 246, 247, 248, 249, 251, 252, 306, 320
sensors, xiii, 60, 61, 92, 207, 211
settlements, 194
shape, 55, 60, 95, 111, 236, 258, 262
shareholders, 318

shear, 60
showing, xiii, 110, 213, 257, 297, 300, 304
signals, 3, 13, 43, 45, 47, 48, 50, 61, 75, 119, 172, 190, 297
silica, 79
silicon, 111, 114, 115, 116, 164, 296
simulation, viii, x, xi, 5, 12, 15, 17, 33, 42, 47, 52, 54, 55, 109, 110, 111, 112, 117, 118, 120, 125, 131, 169, 176, 188, 211, 298, 328
simulations, xi, 2, 15, 110, 112, 116, 119, 120, 121, 169, 323
Singapore, 21, 320, 326
skin, 125
sludge, xii, 193, 194, 195, 196, 201, 202, 204, 205
smoking, 185
smoothing, 262, 271
social roles, 188
sodium, 79
software, 29, 36, 52, 62, 75, 112, 118, 126, 171, 211
solar cells, 164
solubility, 308
solution, ix, x, 8, 12, 26, 29, 48, 49, 79, 80, 81, 82, 83, 86, 89, 90, 91, 92, 93, 96, 100, 101, 105, 110, 123, 124, 125, 127, 140, 142, 143, 144, 145, 146, 147, 149, 185, 209, 210, 219, 258, 263, 264, 265, 280
solution space, 29, 143, 147
solvents, 131, 132, 136
South Africa, 271, 273
soybeans, 284
Spain, 121, 182, 183, 257
specialization, 285
specific heat, 84, 88
specifications, 284
spectroscopy, vii, x, 123, 124, 126, 131, 136
speech, 113, 255
spindle, viii, 59, 60, 61, 62, 63, 67, 70
Sri Lanka, 164
stability, ix, 2, 77, 208, 311
stakeholders, xiv, 317
standard deviation, 175, 176, 195, 271, 290
state, 2, 3, 48, 52, 120, 142, 150, 190, 228, 230, 231, 233, 235, 281, 303, 304, 305, 313
states, 112, 142
statistical inference, 320
statistics, 188, 190, 296, 302
steel, x, 61, 123, 124, 125, 126, 128, 129
stimulus, 12, 44, 47, 49, 50, 51, 52, 274
stock exchange, 320
stock markets, 321, 328
stock price, 228, 320, 321, 322, 323, 325
storage, 78, 298
stress, 61
stroke, 190
structural variation, 304
structure formation, 173
style, 52, 56, 320
styrene, vii, 23, 26, 27
substitution, 140, 150
subsurface flow, 205
succession, 111, 283
sulfur, 31, 37, 307
suppression, 62
surface area, ix, 77, 80, 81
surface modification, 37
surrogates, 117
susceptibility, 305, 314
sustainable energy, 219
sweat, x, 123, 124, 125, 126, 127
symmetry, 302
symptoms, 300
synapse, 13
synergistic effect, 265
synthesis, 208

T

tanks, 195
target, xii, 11, 15, 20, 24, 66, 98, 106, 112, 114, 131, 181, 184, 196, 216, 306, 322
target response, 131
teaching quality, 53
techniques, ix, x, xi, xii, xiv, 2, 24, 26, 91, 92, 93, 94, 101, 109, 110, 113, 114, 116, 123, 136, 140, 150, 152, 169, 170, 171, 172, 177, 178, 179, 196, 204, 208, 246, 294, 295, 296, 300, 301, 302, 305, 306, 308, 309, 311, 319, 320, 321, 323, 326, 327
technological developments, 92
technologies, 24, 25, 171, 177
technology, ix, 24, 109, 118, 152, 296
temperature, ix, xi, 27, 77, 78, 79, 80, 81, 82, 83, 84, 86, 87, 89, 90, 114, 151, 152, 153, 154, 155, 158, 159, 160, 164, 165, 270, 272, 307, 323, 329
tensile strength, 29, 31, 33, 34, 35, 36, 37
tenure, 147
test data, xiv, 62, 64, 66, 71, 72, 74, 143, 176, 227, 246, 295
testing, viii, xi, xii, 23, 27, 42, 55, 62, 64, 86, 127, 129, 132, 151, 158, 159, 160, 170, 178, 185, 225, 320, 325
texture, 61
therapy, 191, 300, 301
thermal energy, 80
thermoplastics, 24
thoughts, 229
thymine, 302

thyroid, 192, 284, 310
time series, x, xiii, 139, 140, 141, 147, 148, 149, 150, 221, 222, 223, 224, 225, 226, 227, 228, 321, 324
tissue, 309
TMC, 301
topology, 115, 175, 306, 311, 321
toxicity, 296, 301, 307, 309, 314, 315
toxicology, 296
trajectory, xii, 181, 185, 211, 216, 217
transaction costs, 321
transcription, 300
transformation, 2, 3, 6, 7, 8, 12, 96, 172, 175, 176, 224, 325
transformation matrix, 96
transformations, 6, 259
transistor, ix, 109, 119
translation, 5
transmission, 13, 172, 268
transparency, xiv, 317
transport, x, 109, 110, 112, 170, 178, 304
transportation, 171
trapezoidal membership, 267
treatment, xii, 185, 188, 193, 194, 195, 196, 197, 201, 202, 204, 205, 268, 270, 300, 312
trial, 28, 31, 116, 131, 140, 147, 149, 197, 199, 200, 204, 216, 223, 321
tunneling, 114
Turkey, xi, 91, 106, 107, 139, 140, 147, 148, 152, 153, 165, 166, 169, 193, 205, 221, 224, 228

U

UK, 122, 137, 217, 293
unconditioned, 47, 49, 50
unconditioned response, 47, 50
United States (USA), 21, 26, 37, 38, 46, 56, 57, 75, 106, 107, 137, 138, 205, 292, 293
universe, 183
updating, 85, 157
US Department of Commerce, 178
USA, 21, 26, 37, 38, 46, 56, 57, 75, 106, 107, 137, 138, 205, 292, 293
USSR, 255
UV, 37

V

validation, 62, 64, 66, 71, 72, 74, 132, 175, 178, 189, 196, 272, 279, 303, 306
valuation, 320, 321, 322, 326, 328, 329
vapor, 78, 79, 80, 81, 89, 90
variables, xiv, 26, 28, 81, 84, 86, 124, 126, 133, 172, 186, 188, 190, 194, 222, 223, 255, 258, 259, 260, 261, 262, 264, 266, 267, 268, 269, 270, 271, 274, 275, 297, 303, 305, 306, 317, 318, 320, 321, 323, 325
variations, 12, 196, 268
vector, 8, 10, 11, 12, 44, 45, 49, 51, 52, 85, 88, 89, 98, 113, 115, 142, 157, 209, 212, 214, 222, 228, 231, 232, 233, 243, 246, 260, 262, 269, 274, 280, 281, 282, 296, 297, 299, 301, 305, 308, 309, 310, 311, 315, 322, 324, 328
velocity, 15, 80
vibration, 61, 75
virus infection, 150
vision, ix, 2, 3, 91, 92, 188, 191, 219
visualization, 3, 130

W

Washington, 22, 205
waste, vii, 23, 24, 26, 31, 37, 39
wastewater, xii, 193, 194, 195, 196, 197, 199, 200, 201, 202, 204, 205
water, ix, 77, 78, 79, 80, 81, 84, 86, 87, 88, 89, 164, 170, 178, 182, 183, 185, 186, 187, 201, 205, 228, 311
water quality, 228
water vapor, 78, 79, 81, 89, 164
watershed, 182
wavelengths, 311
wavelet analysis, 323
wear, viii, 59, 62, 63, 64, 65, 67, 70, 75
web, 153, 165
weight changes, 86, 157
wetlands, 194
wood, 38, 81

Y

yield, xiv, 153, 241, 277, 282