THE EFFECT OF ADAPTIVE PARAMETERS ON THE PERFORMANCE OF BACK PROPAGATION

NORHAMREEZA BINTI ABDUL HAMID

A thesis submitted in
fulfillment of the requirement for the award of the
Degree of Master of Information Technology

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

APRIL, 2012

ABSTRACT

The Back Propagation algorithm or its variation on Multilayered Feedforward Networks is widely used in many applications. However, this algorithm is well-known to have difficulties with local minima problem particularly caused by neuron saturation in the hidden layer. Most existing approaches modify the learning model in order to add a random factor to the model, which overcomes the tendency to sink into local minima. However, the random perturbations of the search direction and various kinds of stochastic adjustment to the current set of weights are not effective in enabling a network to escape from local minima which cause the network fail to converge to a global minimum within a reasonable number of iterations. Thus, this research proposed a new method known as Back Propagation Gradient Descent with Adaptive Gain, Adaptive Momentum and Adaptive Learning Rate (BPGD-AGAMAL) which modifies the existing Back Propagation Gradient Descent algorithm by adaptively changing the gain, momentum coefficient and learning rate. In this method, each training pattern has its own activation functions of neurons in the hidden layer. The activation functions are adjusted by the adaptation of gain parameters together with adaptive momentum and learning rate value during the learning process. The efficiency of the proposed algorithm is compared with conventional Back Propagation Gradient Descent and Back Propagation Gradient Descent with Adaptive Gain by means of simulation on six benchmark problems namely breast cancer, card, glass, iris, soybean, and thyroid. The results show that the proposed algorithm extensively improves the learning process of conventional Back Propagation algorithm.

ABSTRAK

Algoritma *Back Propagation* atau variasinya pada *Multilayered Feedforward Networks* digunakan secara meluas dalam pelbagai aplikasi. Walau bagaimanapun, algoritma ini terkenal dengan masalah *local minima* yang disebabkan oleh *neuron saturation* dalam *hidden layer*. Kebanyakan pendekatan sedia ada, mengubahsuai model pembelajaran dengan menambah faktor rawak pada model tersebut untuk mengatasi masalah terperangkap pada *local minima*. Walau bagaimanapun, arah pencarian *random perturbations* dan pelbagai jenis *stochastic adjustment* bagi set pemberat semasa tidak efektif untuk menghindari masalah *local minima* yang menyebabkan model tersebut gagal dalam proses pembelajaran pada iterasi tertentu. Justeru itu, kajian ini mencadangkan satu kaedah baru dikenali sebagai *Back Propagation Gradient Descent with Adaptive Gain, Adaptive Momentum and Adaptive Learning Rate* (BPGD-AGAMAL) yang mengubahsuai algoritma *Back Propagation Gradient Descent* sedia ada dengan menukar *gain,* momentum dan *learning rate* secara adaptif. Dalam kaedah ini, setiap corak latihan mempunyai *activation function* tersendiri pada neuron dalam *hidden layer*. *Activation function* dilaraskan dengan penyesuaian parameter *gain* di samping mengubah nilai momentum dan *learning rate* semasa proses pembelajaran. Keberkesanan algoritma yang dicadangkan dibandingkan dengan *Back Propagation Gradient Descent* yang konvensional dan *Back Propagation Gradient Descent with Adaptive Gain* dan disahkan secara simulasi pada enam jenis masalah iaitu *breast cancer, card, glass, iris, soybean, and thyroid*. Hasil keputusan jelas menunjukkan bahawa algoritma yang dicadangkan berkeupayaan meningkatkan proses pembelajaran jika dibandingkan dengan algoritma *Back Propagation* yang konvensional.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF SYMBOLS AND ABBREVIATIONS

θ_j	-	Bias for the j^{th} unit.
η	-	Learning rate
α	-	Momentum coefficient
$a_{net,j}$	-	Net input activation function for the j^{th} unit.
a_1	-	BPGD
a_2	-	BPGD-AG
a_3	-	BPGD-AGAMAL
c	-	Gain of the activation function
e	-	Exponent
$f(x)$	-	Function of x
f	-	The squashing or activation function of the processing unit
o_i	-	Output of the i^{th} unit
o_j	-	Output of the j^{th} unit
o_k	-	Output of the k^{th} output unit
t_k	-	Desired output of the k^{th} output unit
w_{ij}	-	Weight of the link from unit i to unit j
w_{jk}	-	Weight on the link from node j to k
$x < 0$	-	x is less than 0
$x > 1$	-	x is greater than 1
$x \geq 0$	-	x is greater than or equal to 0
$-1 \leq x \leq 1$	-	x is greater than or equal to -1 and x is less than or equal to 1
A	-	All algorithms

E	-	Error function
H	-	Performance of the BPGD-AGAMAL against BPGD on measuring criteria
J	-	Performance of the BPGD-AGAMAL against BPGD-AG on measuring criteria
K	-	Improvement ratio of the BPGD-AGAMAL against BPGD on measuring criteria
LB	-	Lower bound
M	-	Improvement ratio of the BPGD-AGAMAL against BPGD-AG on measuring criteria
N	-	Performance of BPGD on measuring criteria
Q	-	Performance of BPGD-AG on measuring criteria
R	-	Performance of BPGD on measuring criteria
UB	-	Upper bound
ADALINE	-	Adaptive Linear Element
AI	-	Artificial Intelligence
ANN	-	Artificial Neural Network
BP	-	Back Propagation
BPGD	-	Back Propagation Gradient Descent
BPGD-AG	-	Back Propagation Gradient Descent with Adaptive Gain
BPGD-AGAMAL	-	Back Propagation Gradient Descent with Adaptive Gain, Adaptive Momentum and Adaptive Learning Rate
CPU	-	Central Processing Unit
GD	-	Gradient Descent
IWS	-	Initial Weight Selection
MLFNN	-	Multilayer Feedforward Neural Network
MLP	-	Multilayer Perceptron
MSE	-	Mean Squared Error
NN	-	Neural Network
OBP	-	Optical Back Propagation
RBF	-	Radial Basis Function

SD	-	Standard Deviation
SPLNN	-	Single Layer Perceptron Neural Network
UCIMLR	-	University California Irvine Machine Learning Repository

LIST OF APPENDICES

LIST OF PUBLICATIONS

Journals:

(i) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, Mohd Najib Mohd Salleh (2011) "Accelerating Learning Performance of Back Propagation Algorithm by Using Adaptive Gain Together with Adaptive Momentum and Adaptive Learning Rate on Classification Problems." International Journal of Software Engineering and Its Applications. Vol. 5, No. 4, pp. 31-44.

(ii) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, Mohd Najib Mohd Salleh (2011) "Improvements of Back Propagation Algorithm Performance by Adaptively Changing Gain, Momentum and Learning Rate." International Journal on New Computer Architectures and Their Applications. Vol. 1, No. 4, pp. 889-901.

(iii) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali (2011) "The Effect of Adaptive Gain and Momentum in Improving Training Time of Back Propagation algorithm on Classification problems", International Journal on Advanced Science, Engineering And Information Technology, Vol. 1, No. 2, pp. 178-184.

(iv) Nazri Mohd Nawi, Norhamreeza Abdul Hamid (2010) "BPGD-AG: A New Improvement of Back-Propagation Neural Network Learning Algorithms with Adaptive Gain", Journal of Science and Technology UTHM 2010, Vol 2, No. 2, pp. 83-102. ISSN-2229-8460.

Proceedings:

(i) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, Mohd Najib Mohd Salleh. "Accelerating Learning Performance of Back Propagation Algorithm by Using Adaptive Gain Together with Adaptive Momentum and Adaptive Learning Rate on Classification Problems", In T.-h. Kim et al. 13‑15 April 2011, UCMA 2011, Part II, CCIS 151, pp. 573-584, 2011. © Springer-Verlag Berlin Heidelberg 2011.

(ii) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, Mohd Najib Mohd Salleh. "Learning Efficiency Improvement of Back Propagation Algorithm by Adaptively Changing Gain Parameter together with Momentum and Learning Rate". In J. M. Zain et al. ICSECS 2011, Part III, CCIS 181, pp. 812-824, 2011. © Springer-Verlag Berlin Heidelberg 2011.

(iii) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, Mohd Najib Mohd Salleh. (2011): "Solving Local Minima Problem In Back Propagation Algorithm Using Adaptive Gain, Adaptive Momentum and Adaptive Learning Rate On Classification Problems", Proceeding in International Conference On Mathematical and Computational Biology 2011 (ICMCB 2011), Renaissance Melaka Hotel, Malacca, 12-14 April 2011.

(iv) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, Mohd Najib Mohd Salleh. "A Review on Back Propagation Algorithm." Proceeding in The Second World Conference on Information Technology 2011 (WCIT 2011), Antalya, Turkey, 23-27 November 2011.

(v) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, Mohd Najib Mohd Salleh. "Accelerating Learning Performance of Back Propagation Algorithm by Using Adaptive Gain Together with Adaptive Momentum and Adaptive Learning Rate on Classification Problems." Proceeding in The International Conference on Advanced Science, Engineering and Information Technology 2011 (ICASEIT 2011), pp. 178-184, Kuala Lumpur, 14-15 January 2011.

(vi) Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali. “The Effect of Adaptive Gain and Adaptive Momentum in Improving Training Time of Gradient Descent Back-Propagation Algorithm on Approximation Problem”. Proceeding in 4th International Conference on Postgraduate Education (ICPE-4 2010), pp. 463-466, Kuala Lumpur, 26-28 November 2010.

(vii) Norhamreeza Abdul Hamid, Nazri Mohd Nawi “The Effect of Adaptive Gain and Momentum in Improving Training Time of Back Propagation Algorithm on Approximation Problem.” Proceeding in Conference on Postgraduates Incentive Research Grant 2010 (CoGIS 2010), 15 July 2010.

(viii) Norhamreeza Abdul Hamid, Nazri Mohd Nawi (2009): ” The Effect of Gain Variation of Activation Function in Improving Training Time of Back Propagation Neural Network on Classification Problems, Proceeding in Kolokium Kebangsaan Pasca Siswazah Sains dan Matematik 2009 (KOLUPSI '09), UPSI, 21 December 2009.

(ix) Nazri Mohd Nawi, R.S. Ransing, Mohd Najib Mohd Salleh, Rozaida Ghazali, Norhamreeza Abdul Hamid: “The Effect of Gain Variation in Improving Learning Speed of Back propagation Neural Network Algorithm on Classification Problems”. Proceeding in Symposium on Progress in Information and Communication Technologies (SPICT'09), Malaysia, 7-8 December 2009.

LIST OF AWARDS

(i) **Bronze Medal in Malaysia International Technology Expo [MiTE 2010]:** Nazri Mohd Nawi, Norhamreeza Abdul Hamid, Rozaida Ghazali, Mohd Najib Mohd Salleh. "BPGD-AG: The New Improved Back Propagation Algorithm."

CHAPTER 1

INTRODUCTION

1.1 An Overview

The Artificial Neural Network (ANN) is an Artificial Intelligence (AI) methodology using computational models with architecture and operations is inspired by human knowledge on biological nervous systems, particularly the brain, to process information. This distribution of knowledge provides a property of fault tolerance and potential for massive parallel implementation (Haykin, 2009).

Over the years, the acceptance level in the applications of ANN has been growing because it is proficient in capturing process information in a black box mode. Due to its ability to solve problems with relative ease of use, robustness to noisy input data and execution speed, and due its ability to analyse complicated systems without accurate modelling in advance, ANN has successfully been implemented across an extraordinary range of problem domains, in areas as diverse as pattern recognition and classification (Nazri *et al.*, 2010b), signal and image processing (Sabeti *et al.*, 2010), robot control (Subudhi & Morris, 2009), weather prediction (Mandal *et al.*, 2009), financial forecasting (Yu *et al.*, 2009), and medical diagnosis (Nazri *et al.*, 2010a).

The Multilayer Perceptron (MLP) is a well-known and the most frequently used type of ANN (Popescu *et al.*, 2009). It is suitable for a large variety of applications (Fung *et al.*, 2005). A standard MLP consists of an input layer, one or more hidden layer(s), and an output layer. Every node in a layer, it is connected to other node in the adjacent forward layer where each connection has a weight associated with it.

Learning is a basic and essential characteristic of MLP. Learning refers to the ability to learn from experience through network examples, to generalise the captured knowledge for expectation solutions, and to self-update in order to improve its performance. During the learning phase, the network learns by adjusting the weights so it is able to predict the correct class of the input samples (Han & Kamber, 2006).

The ANN uses Back Propagation (BP) algorithm to perform parallel training to improve the efficiency of MLP's network. The BP algorithm is the most popular, effective, and easiest algorithm to produce a model for MLP's complex network. This algorithm has produced a large class of network types with many diverse topologies and training methods. The BP algorithm is a supervised learning method that involves backward error correction of the network weights. This algorithm uses a gradient descent (GD) method that attempts to minimise the error of the network by moving down the gradient of the error curve (Alsmadi *et al.*, 2009). The weights of the network are adjusted by the algorithm. Consequently, the error is reduced along a descent direction.

Although BP algorithm has been successfully applied to a wide range of practical problems (Haofei *et al.*, 2007; Lee *et al.*, 2005), it has some limitations. Since BP algorithm uses GD method, the problems include slow learning convergence and easy to get trapped at local minima (Bi *et al.*, 2005; Otair & Salameh, 2005). Moreover, the convergence behaviour of the BP algorithm depends on the selection of network topology, initial weights and biases, learning rate, momentum coefficient, activation function, and value for the gain in the activation function.

In the last decade, a significant number of methods have been produced to improve the efficiency and convergence rate (Kathirvalavakumar & Thangavel, 2006; Naimin *et al.*, 2006; Nazri *et al.*, 2010b; Nazri *et al.*, 2008; Otair & Salameh, 2005). Those studies showed that the BP performance was affected by many factors, for instances learning structure, initial weight, learning rate, momentum coefficient, and activation function.

1.2 Problem Statements

The BP algorithm is well-known for its extraordinary ability to derive meaning from complicated or imprecise data that are too complex to be noticed by either humans or other computer techniques. In some practical applications of BP, fast response to external events within an extremely short time are highly insisted and expected. However, the extensively used GD method clearly cannot satisfy large scale applications and when higher learning performances are required. Furthermore, this type of algorithm has the uncertainty in finding the global minimum of the error criterion functions. To overcome those problems, a research has been done to improve the training efficiency of conventional BP algorithm by introducing adaptive gain variation of activation function known as Back Propagation Gradient Descent With Adaptive Gain (BPGD-AG) proposed by Nazri *et al.* (2008). It has been proven that the performances of the proposed method (BPGD-AG) are better than the conventional BP.

Although the analysis results shown by Nazri *et al.* (2008) demonstrated that the method significantly increased the learning speed and outperformed the standard algorithm with constant gain in learning the target function, however during the training, it was noticed that the method only updated weights, bias and gain update expressions adaptively whereas the learning rate and momentum term were keep constant until the end of the training. The challenge of this research was to prove by simulations, that the adaptive momentum and adaptive learning rate also have the significant effects in improving the current working BPGD-AG algorithm on some classification problems.

1.3 Objectives of the Study

This study embarks on the following objectives:

(i) To investigate the effects of some adaptive parameters such as learning rate, momentum, and gain variation in improving learning efficiency of data mining classification techniques.

(ii) To enhance the current working BPGD-AG algorithm introduced by Nazri *et al.* (2008) by choosing the optimal values for momentum, learning rate, and gain on some classification problems.

(iii) To assess the performances of the enhanced algorithm with the current working BPGD-AG in terms of processing time while preserving the accuracy.

1.4 Scope of the Study

This research focused only on enhancing the current working BPGD-AG algorithm (Nazri *et al.*, 2008). The performances of the proposed algorithm and the existing algorithm were compared and analysed in terms of processing time while preserving the accuracy. The six datasets from University California Irvine Machine Learning Repository (UCIMLR) (Frank & Asuncion, 2010) were employed in order to verify the efficiency of the proposed algorithm which includes of breast cancer (Mangasarian & Wolberg, 1990), card (Quinlan, 1993), glass (Evett & Spiehler, 1988), Iris (Fisher, 1936), soybean (Michalski & Chilausky, 1980), and thyroid (Coomans *et al.*, 1983) datasets. The simulations were carried out by using Matlab 7.10.0 (R2010a) on Pentium IV with 2 GHz HP Workstation, 3.25 GB RAM.

1.5 Aim of the Study

This study focused on enhancing the current working BPGD-AG by optimally choosing gain value together with momentum coefficient and learning rate that would change adaptively.

1.6 Significance of the Study

This study investigated the performances of BP algorithm particularly the current working BPGD-AG (Nazri et al., 2008) by changing momentum coefficient and learning rate adaptively on some classification problems in terms of processing time while preserving the accuracy. It was discovered in this study that adaptive gain together with adaptive learning rate and adaptive momentum improved further the performances of BP algorithm instead of the gain value as claimed by previous researchers.

1.7 Project Schedule

This project has been carried out in two years. The summary of the activity during the research process has been stated in **APPENDIX A**.

1.8 Outline of the Thesis

This thesis comprises of five chapters including the Introduction and Conclusion chapters. The followings are the synopsis of each chapter.

(i) ***Chapter 1*****:** *Introduction*. Apart from providing an outline of the thesis, this chapter contains an overview of the research work background, problem to be solved, objectives to achieve, scope, aim, and significance of the study.

(ii) ***Chapter 2*****:** *Literature Review*. This chapter consists of some efficient learning methods for BP algorithms. This chapter reviews some of the fundamental theory about ANN such as network architecture, learning algorithm and applications. This is followed by reviews on the research contributions made by many researchers in improving the training efficiency of ANN. At the end of this chapter, some of the advantages of using gain value together with adaptive learning rate and momentum are outlined. This chapter lays a foundation for introducing a new method in improving the learning efficiency of the proposed algorithm as described in Chapter 3.

(iii) ***Chapter 3*****:** *Research Methodology*. This chapter discusses the research methodology used to carry out the study systematically.

(iv) ***Chapter 4***: *Simulation Results and Analysis*. The new algorithm developed in Chapter 3 is further validated for its efficiency and accuracy on a variety of benchmark problems. The performances of the proposed algorithm were tested for comparison against the conventional BP algorithm and BPGD-AG algorithm. The performance evaluation was carried out based on its convergence rate and computational training time of classification problems (benchmark data). Hence, only the best values were given.

(v) ***Chapter 5***: *Conclusions and Future Works*. The contributions of the proposed algorithm are summarised and the recommendations are described for further continuation of work.

1.9 Summary of Chapter

The ANN is modelled in human brain and it consists of processing units known as artificial neurons that can be trained to perform complex calculations like the human brain. ANN uses the BP algorithm to perform parallel training for improving the efficiency of MLP’s network. The BP algorithm is a supervised learning method, which is the most popular method with its remarkable ability to derive meaning from complicated or imprecise data that are too complex to be noticed by either humans or other computer techniques. Despite many successful applications, the BP algorithm has several important limitations such as slow convergence rate and it can easily get trapped into local minima because it uses GD method. This study proposes a further improvement on the current working BPGD-AG algorithm by changing the momentum and learning rate value adaptively, which in turn would reduce the learning time and preserving the accuracy of the conventional BP algorithm.

CHAPTER 2

LITERATURE REVIEW

Apart from other competitive techniques in Artificial Intelligence (AI) (i.e. decision support system, expert system, computer vision, and so forth) such as fuzzy logic, genetic algorithm as well as statistical methods and analytic tools for instance, Artificial Neural Networks (ANN) are very powerful in solving complicated and non-linear problems. The reason for ANN being commonly used is because it can present some properties such as learning from examples and exhibiting some capability for generalisation beyond training data. The detailed of ANN is reviewed in this chapter.

This chapter is organised in the following manner: Section 2.1 provides the historical perspectives of ANN. Section 2.2 presents fundamental of ANN field includes the basic of node which is defined in Subsection 2.2.1, the activation function has been reviewed in Subsection 2.2.2, and Multilayer Feedforward Neural Network (MLFNN) has been illustrated in Subsection 2.2.3. The Back Propagation (BP) algorithm has been chosen in order to learn the MLFNN which has been discussed in Section 2.3. In some practical ANN applications, fast response to external events within tremendously short time are highly demanded and expected. However, the comprehensively used of BP algorithm based on gradient descent (GD) method obviously not satisfy in many applications especially large scale application and when higher learning accuracy as well as generalisation performances are obligatory. The reasons for this dissatisfaction have been explained in the Section 2.4. Over the years, many improvements and modifications of the BP learning algorithm have been reported and Section 2.5 outlines the previous researches on improving the BP training efficiency. Then, a detailed description of the method proposed by Nazri *et al.* (2008) is given in Section 2.6. This lays the foundation for the next chapter to improves further the learning efficiency of the method proposed by Nazri *et al.* (2008). Section 2.7 summarised this chapter.

2.1 The Historical Perspective

The concept of ANN approach began in 1943 when McCulloch and Pits introduced the first mathematical model of a biological neuron (McCulloch & Pitts, 1943). At that time, the significance of this model was its ability to compute any logical expression. Then, in 1949, Hebb proposed one of the first learning rules for the McCulloch and Pitts Neural Network known as Hebbian Learning Rule by dealing with ways in which synapses can change their efficiencies (Hebb, 1949). Afterward, as computer emerged in 1950s, several researchers attempted to utilise the new technology to create better performance of ANN. Later, in 1958 the first type of Perceptron was established by Rossenblatt who was the one of the early pioneers in ANN. In their experiments, the pattern recognition ability of Perceptron model was demonstrated by recognising different simple characters (Rosenblatt, 1958). Two years later, Widrow and Hoff developed models which was an adaptive linear element called ADALINE based on the least mean square algorithm. ADALINE became the first ANN to be applied in a commercial application. In 1967, Amari (1967) used the stochastic GD method for adaptive pattern classification.

Conversely in 1969, Minsky and Papert mathematically proved that there are certain serious limitations in Roseblatt’s NN model. Particularly, they justified that the perceptron model could not handle the XOR function (Minsky & Papert, 1969). Influenced by Minsky and Papert’s evidenced, only a few pioneering works on ANN during the 1970’s were undertaken. In 1972, Kohonen (1972) and Anderson (1972) independently proposed the mathematical model for associative memory trained by the Hebbian Learning Rule.

The limitations of the earlier Perceptron model was solved by the BP algorithm which originally introduced by Werbos (1974). Meanwhile, in 1976 Grossberg investigated self-organising networks derived from the human visual systems (Grossberg, 1976). In 1982, Hopfield introduced the first model of recurrent ANN which could be effectively used for solving computational problems (Hopfield, 1982). Another important development in 1982 was the self-organising maps which was proposed by Kohonen (1982). Parker (1985) and LeCun (1985) simultaneously rediscovered independently the BP algorithm for training feedforward neural network. Later on, the BP algorithm was reinvented and made popular by Rumelhart & McClelland (1986).

Once Rumelhart and McClelland answered the criticism of Minsky and Papert, a dramatic increase of interest in ANN occurred. The Boltzman Machine has been developed by Hinton and Sejnowski (1986) which was the first successful realisation of MLFNN. Kosko (1987) developed an adaptive Bi-directional Associative Memory using Hebbian Learning Rule. Also, in 1988, Broomhead and Lowe first introduced Radial Basis Function (RBF) network which provide an alternative to Multilayer Perceptron (MLP) (Broomhead & Lowe, 1988). While Cybenko (1989) proved that the ANN has the ability of universal function approximation. Meanwhile, Funahashi (1989) and Hornik *et al.* (1989) also proposed their findings on proving MLP network as universal approximator.

Subsequently, ANN has been widely implemented on many different areas. Nowadays, ANN has already extended from its simple pattern recognition problems to the very complicated problems. The significant improvements in computer technology as well as the rapid reduction in the cost of high powered computers have resulted in making the development of ANN applications a universally attractive and affordable option.

2.2 The Artificial Neural Networks

The ANN is one of the most popular approaches used extensively in machine learning, which is involved in the development of algorithms that enable computers to learn (Negnevitsky, 2005).

The ANN is a powerful set of adaptive learning technique in order to extract patterns and detect trends that are too complex to be identified otherwise (Kaya, 2009). Supplementary, ANN can exhibit a surprising number of characteristic of human brain (Elhag & Wang, 2007) which has the capability to learn from experience through examples fed to it, generalising the captured knowledge for future solutions and self-adapting (Negnevitsky, 2005). More specifically, ANN is a class of flexible nonlinear regression, discriminates and data reduction model. Indeed, various computational vision systems are developed based on ANN, essentially due to its main characteristics, which are robustness to noisily input data or outliers, execution speed, and possibly to be parallel implemented.

The ANN consists of very simple and highly interconnected nodes also called neurons which are analogous of the biological neurons in the brain that will explained further on the next subsection.

2.2.1 The Basic of Node

The very basic information processing unit of ANN is called node, neuron or unit. It is inspired by the biological neuron which resembles the function of the biological neuron.

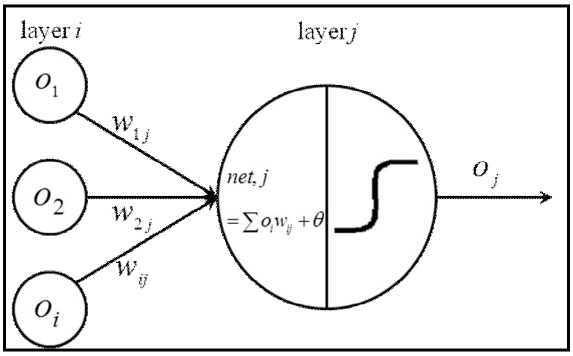

Figure 2.1: The simple node

The Figure 2.1 shows the network structure with inputs $(o_1, o_2, \ldots o_i)$ where o indicates the source of the input signal being connected to node j with weights $(w_{1j}, w_{2j}, \ldots w_{ij})$. Each input o is weighted before being sent to the node j by the connections strength or the weights factor w. This is followed by performing summation of the signals it receives, with each signal being multiplied by its associated weights on the connection. Moreover, it has internal bias, θ in order to enhance the performance of the network. The output net, j is then passed through a non-linear activation function in order to obtain the output o_j:

$$o_j = f\left[\sum_{i=1} w_{ij} o_i + \theta_j\right] \tag{2.1}$$

where,

o_j : output of the j^{th} unit.

o_i : output of the i^{th} unit.

w_{ij} : weight of the link from unit i to unit j.

f : function of activation function

θ_j : bias for the j^{th} unit.

2.2.2 An Activation Function

An activation function also known as transfer function is a non-linear function that determines the output from a summation function of the weighted inputs of the neuron (Engelbrecht, 2007). This function is used for limiting the amplitude of the output neuron. It can be linear or non-linear function. In the literature, the activation function also referred as a squashing function which squashes the permissible amplitude range of the output signal to some finite value. It generates an output value for a node in a predefined range as the closed unit interval $[0,1]$ or alternatively $[-1,1]$. There are various choices for the activation functions which are:

(i) Linear Function

Linear function (refer to Figure 2.2) provides an output proportional to the total weighted output, viz

$$y = f(x) = x \tag{2.2}$$

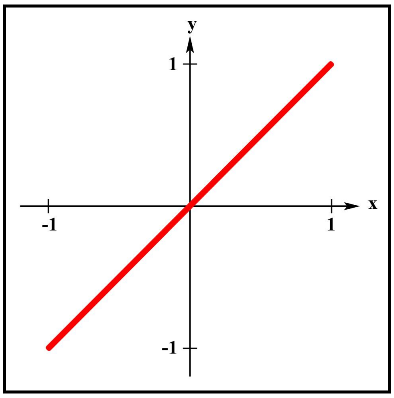

Figure 2.2: The linear function

(ii) Threshold Function

The threshold function maps the weighted input to a binary value $[0,1]$ as shown in Figure 2.3 which is given by

$$y = f(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & if \quad x < 0 \end{cases} \tag{2.3}$$

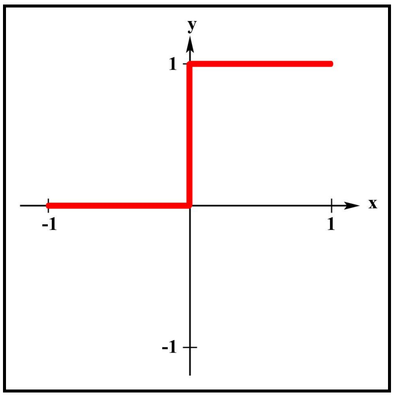

Figure 2.3: The threshold function

(iii) Piecewise Linear Function

The piecewise linear function (Figure 2.4) can have either a binary or bipolar range for the saturation limits of the output. The output for this function can be written as:

$$y = f(x) = \begin{cases} -0.5 & if \quad x < -0.5 \\ x & if \quad -0.5 \leq x \leq 0.5 \\ 0.5 & if \quad x > 0.5 \end{cases} \tag{2.4}$$

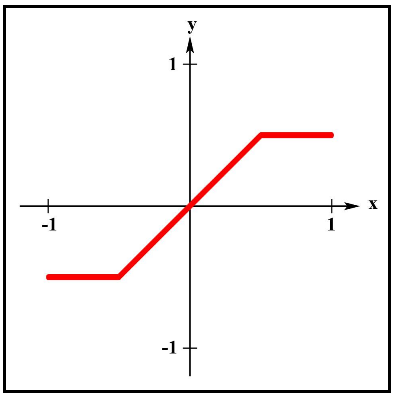

Figure 2.4: The piecewise linear function

(iv) Sigmoid Function

This type of activation function has an S-shaped graph and logistic form of the sigmoid transform the input which can have any value interval $[-\infty,\infty]$ into reasonable value asymptotically in the range between $[0,1]$ as seen in Figure 2.5.

$$y = f(x) = \frac{1}{1+e^{-cx}} \tag{2.5}$$

Where the parameter c controls the steepness of the function.

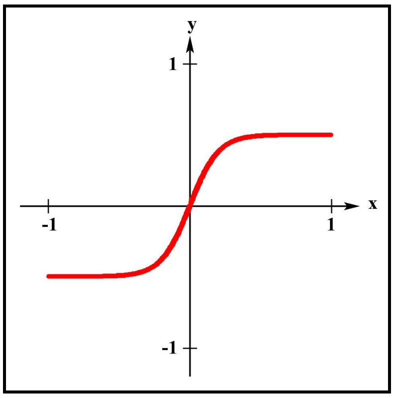

Figure 2.5: The sigmoid function

An activation function is one of the important parameter in the ANN. This function not only determining the decision borders, beside the value of the activation function also demonstrates the total signal strength of the node (Engelbrecht, 2007). Therefore, the selection of activation function cannot arbitrarily selected because it has a huge impact on the ANN performance.

The next subsection will explain the architecture of ANN. The ANN architecture refers to the way nodes are arranged in the network. There are various architecture of ANN which can be classified into three groups by the arrangement of neurons and the connection patterns of the layers. Those are feedforward network (such as Multilayer Feedforward, Radial Basis), recurrent network (such as Elman and Hopfiled), and self-organising network (such as Kohonen) (Haykin, 2009). This thesis only covered for MLP.

2.2.3 The Multilayer Perceptron

The MLP also equivalently known as Multilayer Feedforwad Neural Network (MLFNN) is one of the most popular and most frequently used type of ANN models due to its clear architecture and comparably simple algorithm (Popescu *et al.*, 2009). It can be used as a comprehensive function generator (Haykin, 2009). Moreover, it is suitable for a large variety of applications (Fung *et al.*, 2005).

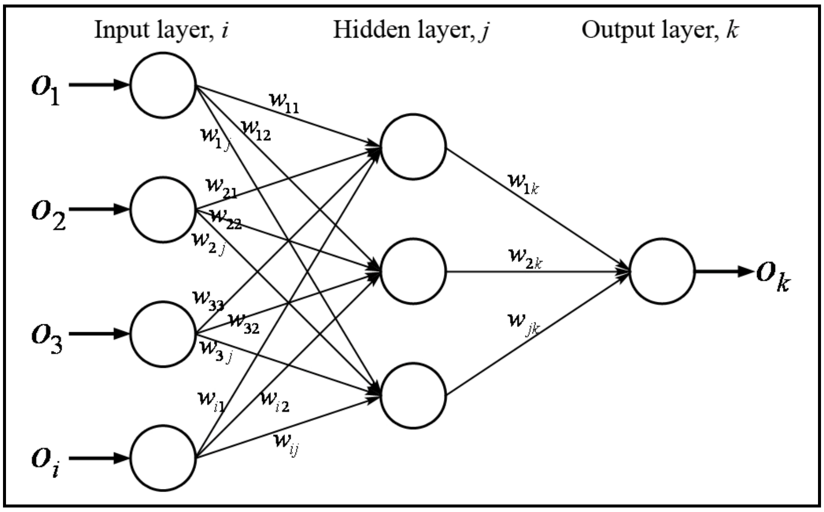

Figure 2.6: The Multilayer Perceptron

The MLP is composed by a set of sensorial nodes organised in three hierarchical of layers comprise of the input layer of nodes, one or more intermediary or hidden(s) layer of computational nodes, and the output layer of nodes that calculates the output of the network as shown in Figure 2.6. The consecutive layers are fully connected. The connections between the nodes of adjacent layers relay the output signals from one layer to the next. For example, in Figure 2.6 the input layer has 4 dimensional vectors, follow by the hidden layer which contains 3 hidden nodes, and finally the output layer which consists 1 output node. This ANN would be known as 4-3-1 network. The input signals propagate through the synaptic links between the layers. The synaptic link consists of the interconnections between the perceptron that carry a signal weight value. This weight value is modified while training the network using training algorithm. Typically, MLP networks are trained with the BP algorithm, which shall be discussed later in Section 2.3.

2.3 The Back Propagation (Supervised Learning)

The BP algorithm has been introduced by Werbos (1974) in order to overcome the drawback of previous ANN algorithm where single layer perceptron fail to solve a simple XOR problem. This type of ANN algorithm is a supervised learning algorithm since it requires a desired output in order to learn the network. The goal of BP algorithm is to create a model that currently maps the input to the output using historical data, thus the ANN model can be used to produce the output when the desired output is unknown (Engelbrecht, 2007). Currently, this synergistically developed BP architecture is the most popular, effective, and easy to learn model for complex, multilayered networks.

There are two types of BP algorithm in order to learn the ANN which are the batch mode learning algorithm and the incremental mode learning algorithm. In the batch mode, the weights values are modified after all patterns are presented, while in the incremental mode, the weights values are updated at every iteration after input pattern is presented. The batch mode learning is more robust, since the training step averages over all the training patterns. On the other hand, the incremental mode approaches appeals to some on-line adaptation applications.

BP is based on the GD method that endeavours to minimise the error of the network by moving down the gradient of error curve (Haykin, 2009). This type of algorithm is used more than all other combined and applied in many different types of applications (Alsmadi *et al.*, 2009).

BP mainly consists of two passes, a forward pass and backward pass. During the forward pass, this algorithm mapping the input values to the desired output through the network. The generated output pattern is obtained from a summation of the weighted input of node and maps to the network activation function. The output is calculated as follows:

$$o_j = \frac{1}{1+e^{-a_{net,j}}} \tag{2.6}$$

where,

$$a_{net,j} = \left(\sum_{i=1} w_{ij} o_i\right) + \theta_j \tag{2.7}$$

where,

o_j : output of the j^{th} unit.

w_{ij} : weight of the link from unit i to unit j.

$a_{net,j}$: net output for the j^{th} unit.

θ_j : bias for the j^{th} unit.

In the backward pass, this output pattern (actual output) is then compared to the desired output and the error signal is computed for each output unit. The signals are then transmit backward from the output layer to each unit in the transitional layer that contributes directly to the output and the weights are adjusted iteratively during the learning process, thus the error is reduced along a descent direction. The error function at the output neuron is defined as:

$$E = \frac{1}{2}\sum_{k=1}^{n}(t_k - o_k)^2 \tag{2.8}$$

where,

n : number of output nodes in the output layer

t_k : desired output of the k^{th} output unit

o_k : network output of the k^{th} output unit

The error function in a one dimensional weight space can be visualised as shown in Figure 2.7.

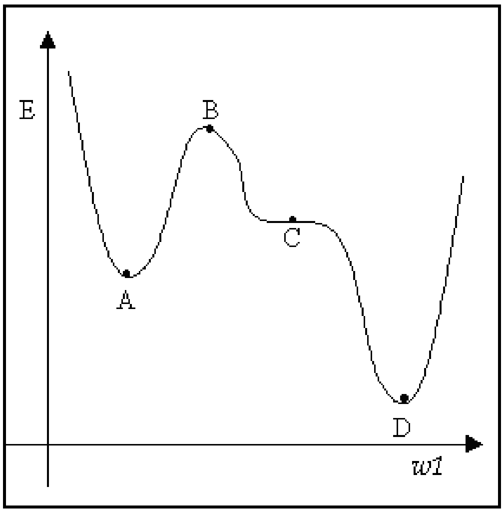

Figure 2.7: The schematic error functions for a single parameter w, showing for stationary points, at which $\nabla E(w) = 0$. Point A is a local minimum, point B is a local maximum, point C is a saddle point, and D is the global minimum

For networks with more than one layer of adaptive weights, the error function is a non-linear function of weights and may have many minima, which satisfy the following equation:

$$\nabla E(w) = 0 \tag{2.9}$$

Where $\nabla E(w)$ denotes the gradient of E with respect to weights. The point at which the value of the error function is smallest (point D in Figure 2.7) is called the global minima while all other minima are called local minima. There may also be other points, which satisfy conditions (Equation (2.9)) for instance local maxima (point B, Figure 2.7) or saddle point (point C, Figure 2.7).

Error is calculated by comparing the network output with the desired output by using Equation (2.8). The error signal (E) is propagated backwards through the network and is used to adjust the weights. The weights in the link connecting to output nodes (w_{jk}) are then modified based on the GD method as follows:

$$\Delta w_{jk}(n+1) = \eta\left(-\frac{\partial E}{\partial w_{jk}}\right) + \alpha \Delta w_{jk}(n) \tag{2.10}$$

$$= \eta \delta_k o_j + \alpha \Delta w_{jk}(n) \tag{2.11}$$

where:

o_j : output of the j^{th} hidden node.

The error is propagated backwards to compute the error specifically, at the hidden nodes:

$$\Delta w_{ij}(n+1) = \eta\left(-\frac{\partial E}{\partial w_{ij}}\right) + \alpha \Delta w_{ij}(n) \qquad (2.12)$$

$$= \eta \delta_j o_i + \alpha \Delta w_{ij}(n) \qquad (2.13)$$

where:

o_i : output of the i^{th} input node (which the same as the output value)

α : momentum coefficient

η : learning rate (step length)

i, j, k : subscripts i, j and k correspond to input, hidden, and output nodes respectively.

w_{jk} : weight on the link from node j to k.

w_{ij} : weight on the link from unit i to j.

δ_k : $o_k(1-o_k)(t_k - o_k)$ for output nodes.

δ_j : $o_j(1-o_j)\sum_k \delta_k w_{jk}$ for hidden nodes.

In this way, the error is propagated backwards to modify weights in order to minimise the error.

From the Equation (2.12), there are two parameters that have been added which are η (learning rate) and α (momentum coefficient). Those are the two parameters that generally employed in BP algorithm. The two parameters also known as two terms parameter. The two terms parameter are added for some reason which stated in the next subsection.

2.3.1 The Two Terms Parameter

The BP utilises two parameters which are learning rate and momentum coefficient. Those parameters are used for controlling the weight adjustment along the steepest descent direction and for dampening oscillations (Zweiri *et al.*, 2003).

(i) The learning rate

The learning rate is one of the most effective means to accelerate the convergence of BP learning which values lies between $[0,1]$. It is a crucial factor to control the variable of the neuron weight adjustments for each iteration during the training process and therefore it affects the convergence rate. In fact, the convergence speed is highly depending on the choice of learning rate value. The learning rate values need to be set appropriately since it dominate the performance of the BP algorithm. The algorithm will take longer time to converge with a large number of iterations or may never converge if the learning rate is too small. Conversely, the network will accelerate the convergence rate significantly although still possibly will cause the instability whereas the algorithm may oscillate on the ideal path and thus not reach a minimum if the learning rate value is too high. The best choice of learning rate is application-dependant and typically chosen by trial and error method. The adaptive learning rate hence can speed up the learning process which will be discussed later.

(ii) The momentum coefficient

Another effective approach regarding to hasten up the convergence and stabilise the training procedure is by adding some momentum coefficient to the network. Moreover, with momentum coefficient, the network can slide through shallow local minima. The value for the momentum coefficient usually in the interval $[0,1]$. The momentum coefficient adds a fraction of the previous weight change to the current weight update to the current weight adjustment which leads to faster convergence.

Although the BP algorithm is used extensively to estimate weights combination for ANN, it still has some limitations which will be pointed out in the next section.

2.4 Limitation of the Back Propagation Training Algorithm

The conventional BP has proved satisfactory when successfully applied in some real problems including prediction, pattern recognition, and classification. Unfortunately, despite the common success of BP in learning ANN, several major drawbacks are still required to be solved. Since BP algorithm uses GD method to update weights, the limitations comprise a slow learning convergence and easily get trapped at local minima (Bi *et al.*, 2005; Wang *et al.*, 2004).

Although the GD can be an efficient method to find the weight values that minimise an error measure, error surfaces frequently possess properties that make this procedure too slow to converge. There are several reasons for this slow rate of convergence which involve the magnitude and the direction components of the gradient vector. When the error surface is fairly flat along a weight dimension, the derivative of the weight is small in magnitude. Thus, the value of the weight is adjusted by a small amount and many procedures are obligatory to achieve a significant reduction in error. Alternatively, where the error surface is highly curved along a weight dimension, the derivative of the weight is large in magnitude. Thus, the value of the weight is adjusted by a large value which may exceed the minimum of the error surface along that weight dimension. Another reason for the slow rate of convergence for the GD method is that the direction of the negative gradient vector may not point directly towards the minimum of the error surface (Nazri, 2007).

It is noted that many local minima complications are closely associated to the neuron saturation in the hidden layer. When such saturation exists, neuron in the hidden layer will lose their sensitivity to the input signals and propagation chain is blocked severely. In some situation, the network can no longer learn. Furthermore, the convergence behaviour of the BP algorithm also depends on the selection of network architecture, initial weights and biases, learning rate, momentum coefficient, activation function, and value of the gain in the activation function.

Nevertheless, for these limitations of BP algorithm, several researches have been done to overcome these drawbacks.

2.5 Previous Research on Improving the Back Propagation Training Efficiency

In the recent years with the progress of researches and applications, the ANN technology has been enhanced and sophisticated. Many researches have been done to modify the conventional BP algorithm in order to improve the efficiency and performance of ANN training. Much works have been devoted to improve the generalisation ability of the networks. These implicated the development of heuristic techniques, based on properties studies of the conventional BP algorithm. These techniques include such idea as varying the learning rate, using momentum, and gain tuning of activation function. Various acceleration techniques have been proposed in heuristic technique.

2.5.1 Improving the Error Function

Since the sigmoid derivative which appears in the error function of the conventional BP algorithm has a bell shape, it sometimes causes slow learning convergence when output of a unit is near 0 or 1. In order to overcome the problems, the Optical Back Propagation (OBP) (Otair & Salameh, 2005) algorithm is designed to adjust the error. This algorithm applied on the output units. This kind of algorithm used for training process that depends on a MLP with a very small learning rate, especially when using a large training set size. Conversely, it does not guarantee to converge at global minima because if the error closes to maximum, the OBP error grows increasingly.

Meanwhile, Ng *et al.* (2006) localised generalisation error model for single layer perceptron neural network (SPLNN). This is an extensibility of the localised generalisation model for supervised learning with mean squared error minimisation. Though, this approach serves as the first step of considering localised generalisation error models of ANN.

2.5.2 Starting with Appropriate Weight

It has been shown that the BP method is sensitive to initial weights (Kolen & Pollack, 1991). Generally, weights initialised with small random values. However, starting with incorrect weight values will cause the network to be trapped in local minima or may lead to slow learning progress. For example, initial weight values which are too large can cause 'Premature Saturation'. Köppen *et al.* (2009) demonstrate that a complete analysis of the MLP weight space is possible. This approach based on clustering of the weight vectors after having trained an MLP with the BP algorithm. While Hyder *et al.* (2009) presents a new algorithm known as initial weight selection (IWS) to determine initial weights for ANN. The initial weights are carefully selected so that it will hasten up learning process.

2.5.3 Improving Activation Function

One of the main reasons for the slow convergence of conventional BP algorithm is the derivative of the activation function that leads to the occurrence of premature saturation of the neurons. Wang *et al.* (2004) proposed an improved BP algorithm caused by neuron saturation in the hidden layer. Each training pattern has its own activation function of hidden nodes in order to prevent neuron saturation when the network output has not acquired the desired signals. The activation functions are adjusted by the adaptation of gain parameters during the learning process. It has been shown that BP algorithm using gain variation term in an activation function converges faster than BP algorithm as will be discussed further in the next section.

2.5.4 Improving Two Terms BP Parameters

Learning parameter that involved in conventional Two Terms BP parameters are learning rate and momentum factor. The correct selections of these parameters separate the signal from the noise and avoid over-fitting of the signal. Those parameters will affect the convergence of the ANN.

(i) The Learning Rate

The value of learning rate usually set to be constant which means that the selected value is employed for all weights in the whole learning process. Later, Ye (2001) stated that the constant learning rate of the BP algorithm fails to optimise the search for the optimal weight combination. Hence, a search methodology has been classified as a "blind-search". While Li & Lin (2005) proposed the value of learning rate is calculated by the fuzzy reasoning. However, the algorithm needs to define a membership function for the fuzzy reasoning by try and error method. Meanwhile Yuemei & Hong (2009) improved the restraining through by auto-adapted learning rate, although the adjustment of the network weights is related with error gradient during the training. When the training has fallen into smooth area, error gradient is closed to zero. Then, the learning rate is large and the adjustment of weights will still be slow, which could cause slow convergence to the target error.

(ii) The Momentum Coefficient

Formerly, the momentum coefficient is typically preferred to be constant in the interval $[0,1]$. In spite of that, it is discovered from simulations that the fixed momentum coefficient value seems to hasten up learning only when the recent downhill gradient of the error function and the last change in weight have a parallel direction. When the recent negative gradient is in a crossing direction to the previous update, the momentum coefficient may cause the weight to be altered up the slope of the error surface as opposed to down the slope as preferred. This leads to the emergence of diverse schemes for adjusting the momentum coefficient value adaptively instead of being kept constant throughout the training process. The BP with adaptive momentum has been proposed by Xiaoyuan *et al.* (2009). This method can escape at local minima and hasten up the network learning. However, when the training enters smooth area, error gradient is closed to zero. Thus, the network will be converging slowly.

REFERENCES

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1986). A learning algorithm for boltzmann machines. *Cognitive Science, 9 (1),* pp. 147-169.

Alsmadi, M. K. S., Omar, K., & Noah, S. A. (2009). Back Propagation Algorithm: The Best Algorithm Among the Multi-layer Perceptron Algorithm. *International Journal of Computer Science and Network Security, 9 (4),* pp. 378-383.

Amari, S. (1967). A Theory of Adaptive Pattern Classifiers. *Electronic Computers, IEEE Transactions on, EC-16 (3),* pp. 299-307.

Anderson, J. A. (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences, 14 (3-4),* pp. 197-220.

Bi, W., Wang, X., Tang, Z., & Tamura, H. (2005). Avoiding the Local Minima Problem in Backpropagation Algorithm with Modified Error Function. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci., E88-A (12),* pp. 3645-3653.

Broomhead, D. S., & Lowe, D. (1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems 2,* pp. 321-355.

Castellani, M., & Rowlands, H. (2009). Evolutionary Artificial Neural Network Design and Training for Wood Veneer Classification. *Engineering Applications of Artificial Intelligence, 22 (4-5),* pp. 732-741.

Coomans, D., Broeckaert, I., Jonckheer, M., & Massart, D. L. (1983). Comparison of Multivariate Discrimination Techniques for Clinical Data - Application to The Thyroid Functional State. *Methods of Information Medicine, 22 (2),* pp. 93 - 101.

Cybenko, G. (1989). Approximation by Superposition of a Sigmoidal Function. *Mathematics of Control, Signals and Systems, 2,* pp. 303-314.

Elhag, T. M. S., & Wang, Y.-M. (2007). Risk Assessment for Bridge Maintenance Projects: Neural Networks versus Regression Techniques. *Journal of Computing in Civil Engineering, 21 (6)*, pp. 769-775.

Engelbrecht, A. P. (2007). *Computational Intelligence: An Introduction*. 2nd ed. England: John Wiley & Sons.

Eom, K., Jung, K., & Sirisena, H. (2003). Performance improvement of backpropagation algorithm by automatic activation function gain tuning using fuzzy logic. *Neurocomputing, 50*, pp. 439-460.

Evett, I. W., & Spiehler, E. J. (1988). Rule induction in forensic science. in (Ed.). *Knowledge Based Systems*. Halsted Press. pp. 152-160.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7 (2)*, pp. 179-188.

Frank, A., & Asuncion, A. (2010). *UCI Machine Learning Repository*. Retrieved on January 10, 2010 from http://archive.ics.uci.edu/ml.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks, 2 (3)*, pp. 183-192.

Fung, C. C., Iyer, V., Brown, W., & Wong, K. W. (2005). Comparing the Performance of Different Neural Networks Architectures for the Prediction of Mineral Prospectivity. *Proceedings of the Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 2005*. pp. 394-398.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics, 23 (3)*, pp. 121-134.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Haofei, Z., Guoping, X., Fangting, Y., & Han, Y. (2007). A neural network model based on the multi-stage optimization approach for short-term food price forecasting in China. *Expert Syst. Appl., 33 (2)*, pp. 347-356.

Haykin, S. S. (2009). *Neural Networks and Learning Machines*. New Jersey: Prentice Hall.

Hebb, D. (1949). *The Organization of Behavior: A Neuropsychological Theory*. Wiley.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, 79 (8)*, pp. 2554-2558.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Netw., 2 (5)*, pp. 359-366.

Hyder, M. M., Shahid, M. I., Kashem, M. A., & Islam, M. S. (2009). Initial Weight Determination of a MLP for Faster Convergence. *Journal of Electronics and Computer Science, 10*.

Jayalakshmi, T., & Santhakumaran, A. (2010). Improved Gradient Descent Back Propagation Neural Networks for Diagnoses of Type II Diabetes Mellitus. *Global Journal of Computer Science and Technology, 9 (5)*, pp. 94-97.

Kathirvalavakumar, T., & Thangavel, P. (2006). A Modified Backpropagation Training Algorithm for Feedforward Neural Networks*. *Neural Processing Letters, 23 (2)*, pp. 111-119.

Kaya, A. (2009). Slope Residual Strength Evaluation using Artificial Neural Networks. *Geotech Geol. Eng., 27 (2)*, pp. 281-289.

Kohonen, T. (1972). Correlation Matrix Memories. *Computers, IEEE Transactions on, C-21 (4)*, pp. 353-359.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43 (1)*, pp. 59-69.

Kolen, J. F., & Pollack, J. B. (1991). Back Propagation Is Sensitive To Initial Conditions. in R. P. Lippmann, J. E. Moody & D. S. Tpuretzky (Ed.). *Advances in Neural Information Processing Systems*. Denver: pp. 860-867.

Köppen, M., Kasabov, N., Coghill, G., Adam, S., Karras, D., & Vrahatis, M. (2009). Revisiting the Problem of Weight Initialization for Multi-Layer Perceptrons Trained with Back Propagation. in (Ed.). *Advances in Neuro-Information Processing*. Springer Berlin / Heidelberg. pp. 308-315.

Kosko, B. (1987). Adaptive bidirectional associative memories. *Appl. Opt., 26 (23)*, pp. 4947-4960.

LeCun, Y. (1985). A Learning Scheme for Asymmetric Threshold Networks. *Proceedings of the Cognitiva*. Paris, France: pp. 599-604.

Lee, C.-H., & Lin, Y.-C. (2005). An adaptive neuro-fuzzy filter design via periodic fuzzy neural network. *Signal Process., 85 (2)*, pp. 401-411.

Lee, K., Booth, D., & Alam, P. (2005). A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms. *Expert Systems with Applications, 29 (1),* pp. 1-16.

Li, S., Okada, T., Chen, X., & Tang, Z. (2006). An Individual Adaptive Gain Parameter Backpropagation Algorithm for Complex-Valued Neural Networks. in J. Wang, Z. Yi, J. Zurada, B.-L. Lu & H. Yin (Ed.). *Advances in Neural Networks* - *ISNN 2006*. Springer Berlin / Heidelberg. pp. 551-557.

Maier, H. R., & Dandy, G. C. (1998). The effect of internal parameters and geometry on the performance of back-propagation neural networks: an empirical study. *Environmental Modelling and Software, 13 (2),* pp. 193-209.

Mandal, S., Sivaprasad, P. V., Venugopal, S., & Murthy, K. P. N. (2009). Artificial neural network modeling to evaluate and predict the deformation behavior of stainless steel type AISI 304L during hot torsion. *Applied Soft Computing, 9 (1),* pp. 237-244.

Mangasarian, O. L., & Wolberg, W. H. (1990). Cancer diagnosis via linear programming. *SIAM News, 23 (5),* pp. 1-18.

McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology, 5 (4),* pp. 115-133.

Michalski, R. S., & Chilausky, R. L. (1980). Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis. *International Journal of Policy Analysis and Information Systems, 4:2.*

Minsky, M. L., & Papert, S. (1969). *Perceptrons; An Introduction to Computational Geometry*. MIT Press.

Naimin, Z., Wei, W., & Gaofeng, Z. (2006). Convergence of gradient method with momentum for two-Layer feedforward neural networks. *Neural Networks, IEEE Transactions on, 17 (2),* pp. 522-525.

Nazri, M. N. (2007). *Computational Issues in Process Optimsation using Historical Data*. S. University: PhD.

Nazri, M. N., Najib, M. S. M., & Rozaida, G. (2010a). The Development of Improved Back-Propagation Neural Networks Algorithm for Predicting Patients with Heart Disease. in R. Zhu, Y. Zhang, B. Liu & C. Liu (Ed.). *Information Computing and Applications*. Springer Berlin / Heidelberg. pp. 317-324.

Nazri, M. N., Ransing, R. S., Najib, M. S. M., Rozaida, G., & Norhamreeza, A. H. (2010b). An Improved Back Propagation Neural Network Algorithm on Classification Problems. in Y. Zhang, A. Cuzzocrea, J. Ma, K.-i. Chung, T. Arslan & X. Song (Ed.). *Database Theory and Application, Bio-Science and Bio-Technology*. Springer Berlin Heidelberg. pp. 177-188.

Nazri, M. N., Ransing, R. S., & Ransing, M. S. (2008). An Improved Conjugate Gradient Based Learning Algorithm for Back Propagation Neural Networks. *International Journal of Information and Mathematical Sciences, 4 (1),* pp. 46-55.

Negnevitsky, M. (2005). *Artificial Intelligence a Guide to Intelligent Systems*. 2. Harlow, England: Addison Wesley.

Ng, W. W. Y., Yeung, D. S., & Tsang, E. C. C. (2006). Pilot Study on the Localized Generalization Error Model for Single Layer Perceptron Neural Network. *Proceedings of the Machine Learning and Cybernetics, 2006 International Conference on*. pp. 3078-3082.

Norhamreeza, A. H., & Nazri, M. N. (2009). The Effect of Gain Variation of Activation Function in Improving Training Time of Back Propagation Neural Network on Classification Problems. *Proceedings of the Kolokium Kebangsaan Pasca Siswazah Sains dan Matematik 2009*. UPSI: pp.

Norhamreeza, A. H., Nazri, M. N., & Ghazali, R. (2011). The Effect of Adaptive Gain and Adaptive Momentum in Improving Training Time of Gradient Descent Back Propagation Algorithm on Classification problems. *Proceedings of the Proceeding of the International Conference on Advanced Science, Engineering and Information Technology 2011*. Hotel Equatorial Bangi-Putrajaya, Malaysia: pp. 178-184.

Otair, M. A., & Salameh, W. A. (2005). Speeding Up Back-Propagation Neural Networks. *Proceedings of the Proceeding of the 2005 Informing Science and IT Education Joint Conference*. Flagstaff, Arizona, USA: pp. 167-173.

Ozel, T., Correia, A. E., & Davim, J. P. (2009). Neural Network Process Modelling for Turning of Steel Parts using Conventional and Wiper Inserts *International Journal of Materials and Product Technology 2009, 35 (1/2),* pp. 246-258

Parker, D. (1985). *Learning logic*. MIT: Technical Report TR-87.

Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer perceptron and neural networks. *WSEAS Trans. Cir. and Sys., 8 (7),* pp. 579-588.

Quinlan, J. R. (1987). Simplifying decision trees. *Int. J. Man-Mach. Stud., 27 (3),* pp. 221-234.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann.

Rosenblatt, F. (1958). The perception: a probabilistic model for information storage and organization in the brain. *Psychological Review, 65 (6),* pp. 386-408.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press.

Sabeti, V., Samavi, S., Mahdavi, M., & Shirani, S. (2010). Steganalysis and payload estimation of embedding in pixel differences using neural networks. *Pattern Recogn., 43 (1),* pp. 405-415.

Shepherd, A. J. (1997). *Second-Order Methods for Neural Networks: Fast and Reliable Training Methods for Multi-Layer Perceptrons*. London: Springer.

Subudhi, B., & Morris, A. S. (2009). Soft computing methods applied to the control of a flexible robot manipulator. *Applied Soft Computing, 9 (1),* pp. 149-158.

Sun, Y.-j., Zhang, S., Miao, C.-x., & Li, J.-m. (2007). Improved BP Neural Network for Transformer Fault Diagnosis. *Journal of China University of Mining and Technology, 17 (1),* pp. 138-142.

Thimm, G., Moerland, P., & Fiesler, E. (1996). The interchangeability of learning rate and gain in backpropagation neural networks. *Neural Comput., 8 (2),* pp. 451-460.

Wang, X. G., Tang, Z., Tamura, H., Ishii, M., & Sun, W. D. (2004). An improved backpropagation algorithm to avoid the local minima problem. *Neurocomputing, 56* pp. 455-460.

Watkins, D. (1997). *Clementine's Neural Networks Technical Overview*. Technical Report.

Werbos, P. (1974). *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. Harvard University.

Xiaoyuan, L., Bin, Q., & Lu, W. (2009). A New Improved BP Neural Network Algorithm. *Proceedings of the Proceedings of the 2009 Second International Conference on Intelligent Computation Technology and Automation - Volume 01*. IEEE Computer Society.

Ye, Y. C. (2001). *Application and Practice of the Neural Networks*. Taiwan: Scholars Publication.

Yu, L., Wang, S., & Lai, K. K. (2009). A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Appl. Soft Comput., 9 (2)*, pp. 563-574.

Yuemei, X., & Hong, Z. (2009). Study on the Improved BP Algorithm and Application. *Proceedings of the Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*. pp. 7-10.

Zweiri, Y. H. (2007). Optimization of a Three-Term Back Propagation Algorithm Used for Neural Network Learning. *International Journal of Computer Intelligence, 3 (4)*, pp. 322-327.

Zweiri, Y. H., Whidborne, J. F., & Seneviratne, L. D. (2003). A three-term backpropagation algorithm. *Neurocomputing, 50* pp. 305-318.