# THREE-TERM BACKPROPAGATION ALGORITHM FOR CLASSIFICATION PROBLEM

FADHLINA IZZAH BINTI SAMAN

A project report submitted in partial fulfillment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Faculty of Computer Science and Information System
Universiti Teknologi Malaysia

APRIL, 2006

*“To my beloved family and friends, thanks for being there, throughout this journey”*

# ACKNOWLEDGEMENT

In the name of Allah, Most Merciful, Most Compassionate. It is by God's willing, I was able to complete this project within the time given. This project would not have been possible without the support of many people. Firstly, I would like to take this opportunity to thank my supervisor, Associate Professor Dr. Siti Mariyam bt. Shamsuddin for her guidance, patience and support for me throughout the period of time in doing this project. I also would like to thank Dr. Siti Zaiton Mohd. Hashim as my co-supervisor for her valuable comments and suggestions to further improve my analysis and writing. My deepest gratitude also goes out to my examiners, Dr. Ali Selamat and Dr. Norazah Yusof for their constructive comments and suggestions in evaluating my project.

A very special thanks to my family and all my friends for always being there for me and giving their love and support when I most needed it. Lastly I would like to thank all staff and lecturers of Faculty of Computer Science and Information System, Universiti Teknologi Malaysia for their help and support.

# ABSTRAK

*Standard Backpropagation Algorithm (BP)* merupakan algoritma yang digunakan secara secara meluas dalam melatih Rangkaian Neuron (*Neural Network*) dan ia telah berjaya digunakan dalam banyak aplikasi berbeza. Algoritma ini pada kebiasaannya menggunakan dua parameter pembelajaran iaitu Kadar Pembelajaran, $\alpha$ dan juga Faktor Momentum, $\beta$. Walaupun algoritma ini telah dianggap berjaya, namun terdapat beberapa kelemahan dan kekangan yang wujud. Antaranya ialah kewujudan *local minima*, kadar penumpuan yang perlahan dan juga pengiraan yang kompleks diperlukan pada pembaikan algoritma yang pernah dilakukan sebelum ini. Zweiri *et. al.* (2003) telah mencadangkan parameter pembelajaran ketiga iaitu Faktor Berkadaran, $\gamma$ untuk mengatasi masalah tersebut. Algoritma baru ini dinamakan *Three-Term BP*. Projek ini dijalankan untuk mengkaji keberkesanan algoritma *Three-Term BP* dan kemudiannya membuat perbandingan dengan algoritma *standard BP*. Untuk itu, eksperimen telah dijalankan dengan mengimplementasikan *Three-Term BP* keatas tiga set data iaitu Balloon, Iris dan Cancer. Data-data ini digunakan untuk mewakili data berskala kecil, sederhana dan besar. Berdasarkan hasil eksperimen yang diperolehi, *Three-Term BP* hanya menunjukkan prestasi yang lebih baik daripada *standard BP* jika ianya menggunakan data berskala kecil sahaja. Ia adalah berkemungkinan daripada ketidakstabilan algoritma tersebut jika menggunakan data berskala sederhana dan besar seperti yang telah ditunjukkan dalam bahagian analisa kajian ini.

# ABSTRACT

Standard Backpropagation Algorithm (BP) is a widely used algorithm in training Neural Network that is proven to be very successful in many diverse application. This algorithm utilizes two term parameters which are Learning Rate, $\alpha$ and Momentum Factor,$\beta$. Despite the general success of this algorithm, there are several drawbacks and limitations which some of them are the existence of local minima, slow rates of convergence and some of the modification of BP algorithm requires complex and costly calculations at each iteration, which offset their faster rates of convergence. To overcome this problem, a third learning parameter, Proportional Factor ($\gamma$) has been proposed by Zweiri *et. al.*, (2003). This new algorithm is called Three-Term BP. This study investigates the performance of Three-Term BP and compares its performance with standard BP. To achieve this objective, experiments were conducted by implementing Three-Term BP to three dataset which are Balloon, Iris and Cancer dataset. These datasets represents small, medium and large scale data respectively. The results obtained showed that Three-Term BP only outperforms standard BP while using small scale data but not in case of medium and large dataset. This might be caused by the instability of the network while using medium and large dataset as it has been proven in analysis part of the study.

## TABLE OF CONTENT

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS

| | | |
|---|---|---|
| $W_{ij}$ | - | Weight connected between node *i* and *j* |
| $\theta_i$ | - | Bias of node *i* |
| $O_j$ | - | Output of node *j* |
| $W_{ij}(t)$ | - | Weight from node *i* to node *j* at time t, |
| $\Delta W_{ij}$ | - | Weight adjustment |
| $\alpha$ | - | Learning rate |
| $\beta$ | - | Momentum term |
| $\gamma$ | - | Proportional factor |
| $\delta_j$ | - | Error signal at node *j* |
| $T_j$ | - | Target output value at node *j* |
| $O_j$ | - | Actual output of the network at node *j* |
| $t_{kj}$ | - | Target value from output node (*k*) to hidden node (*j*) |
| $o_{kj}$ | - | Network value from output node (*k*) to hidden node (*j*) |
| $E_k$ | - | Error at output unit *k* |
| $m$ | - | Number of input nodes |
| $n$ | - | Number output nodes |
| $e_s$ | - | Difference between output and target at each iteration |

# LIST OF ABBREVIATION

| | | |
|---|---|---|
| ANN | - | Artificial Neural Network |
| BP | - | Backpropagation |
| MLP | - | Multilayer Perceptron |
| MSE | - | Mean Squared Error |
| NN | - | Neural Network |

# LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Classification can be defined as deciding the category or grouping to which an input value belongs (Ishbushi *et al.*, 1999). Classification is defining a set of groups by their characteristics, then each case is analyzed and put it into the group it belongs. Classification can be used to understand the existing data and to predict how new instances will behave. The goal of data classification is to organize and categorize data in distinct classes. A model is first created based on the data distribution. The model is then used to classify new data. Given the model, a class can be predicted for new data.

The Neural Network (NN) must be trained to classify certain data patterns to certain outputs. The Backpropagation (BP) algorithm is a supervised learning method for multi-layered feedforward NN. It is commonly used for learning algorithm for training NN (Okine, 1999).

The BP algorithm involves backward error correction of the network weights. Training is usually done by weights updating iteratively using a mean-square error function. Traditionally, the standard BP algorithm utilizes two terms parameters called Learning Rate (α) and momentum factor (β) for weight adjustment.

## 1.2 Problem Background

BP algorithm is a widely used learning algorithm in NN. Despite the general success of the BP algorithm, there are several drawbacks and limitations that still exist (Zweiri *et al.*, 2003). Limitations of standard BP:

1 The existence of temporary, local minima resulting from the saturation behaviour of the activation function.

2 The slow rates of convergence. Convergence rate is relatively slow for network with more than one hidden layer.

3 Some of the modification of BP algorithm requires complex and costly calculations at each iteration, which offset their faster rates of convergence.

Several approaches had been proposed by researchers to overcome these problems. Wang *et al.* (2003) had proposed an improved BP to avoid local minima where 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 during the learning process. However, this approach did not produce good results on large problems and practical applications. Yu and Liu (2002), proposed an adaptive learning rate and momentum coefficient for faster convergence. The acceleration technique proposed was Backpropagation with Adaptive Learning rate and Momentum factor (BPALM). The learning rate and momentum are adjusted at each iteration.

Kim *et al*. (2004) has used Genetic Algorithm (GA) to determine learning rate and momentum parameter. Although GA is proven to give better performance than the standard BP which uses trial and error method to obtain the learning rate and momentum, but is changes the structure of the BP computation. Jia and Yang (1993) proposed an improved algorithm with stochastic attenuation momentum factor. Compared with the standard BP algorithm, the algorithm claims to effectively cancel the negative effect on momentum of network. However, the computation of this proposed approach is complex since it uses the correlation matrix in defining the momentum. Roy (1994) proposed a new method to compute dynamically the optimal learning rate. By using this method, a range of good static learning rates could be found. Although this method could improve the performance of standard BP, the algorithm is computationally complex and might take longer to train than standard BP. Yu and Chen (1996) proposed a BP algorithm using dynamically optimal learning rate and momentum factor. This approach claims to have remarkable savings in running time. However, this approach also its disadvantages of having complex computation and storage burden for estimating the optimal learning rate and momentum factor at most triple that of the standard BP.

Although many approaches had been taken by many researchers to improve the performance of standard BP, the alterations done in the BP algorithm sometimes require complex calculations at each iteration, which offset their faster rate of convergence. Another disadvantage of most acceleration techniques used is that they must be tuned to fit the particular application. The summary of comparisons between five approaches that had been implemented in improving BP performance is shown in Table 1.1.

**Table 1.1 :** List of few approaches in improving BP performance

| **Researcher** | **Approach** |
|---|---|
| Roy (1994) | Dynamically compute optimal learning rate |
| Ng *et al.* (1996) | Generalized BP. Change the derivative of the activation function to magnify backward propagated error signal |
| Yu and Chen (1996) | BP learning using dynamically optimal learning rate and also momentum factor |
| Yu and Liu (2002) | Acceleration technique, BPALM by employing adaptive learning rate and momentum factor. Both terms are adjusted at each iteration |
| Wang *et al.* (2003) | Each training pattern has its own activation functions of neurons in hidden layer |

The standard BP algorithm usually uses two parameters which are Learning Rate (α) and Momentum Factor, (β) for controlling the weight adjustment of the ANN. Zweiri *et al.* (2003) proposes an additional term, proportional factor (γ) for BP learning. In this study, a new term γ of Zweiri *et al*, (2003) will be implemented in addition of the existing terms ά and β. to calculate the change of weight for BP learning enhancement

This study attempts to evaluate the efficiency of implementing the proportional factor, γ as the third term for BP algorithm and to do a comparative study between the Three-Term BP and standard BP. The integration of the γ as a third term is done with the hope of improving and speeding the BP learning without having to change the BP structure into a more complex structure. It maintains the computation complexity of standard BP as it is, since $\gamma$ is added in the formulation of network's weight adaptation.

## 1.3 Problem Statement

Although the standard BP provides a solution to the problem of learning in multilayer networks, it also has its own weaknesses and limitations. The BP learning can be improved by selection of better activation function and optimal learning parameters which are learning rate and momentum values (Ng *et al.*, 1996). This study will focus on learning parameters by adding a third term which is the proportional factor, $\gamma$. The $\gamma$ factor will be implemented in the network weight adjustment to test its efficiency.

Therefore, the hypothesis of this study can be stated as:

> *How efficient is the Proportional Factor,$\gamma$ in Three-Term BP compared to Standard BP in terms of the convergence speed and classification accuracy?*

## 1.4 Project Aim

The aim of the project is to investigate the efficiency of Three-Term BP proposed by Zweiri *et al*. (2003) by introducing the $\gamma$ term. A comparison between

Three-Term and Standard BP is carried out. These algorithms will be used to solve classification problem using universal data which are Balloon, Iris and Cancer dataset. Various values of $\alpha$, $\beta$ and $\gamma$ will be experimented to search for better parameters tuning for Three-Term BP.

## 1.5 Project Objectives

The objectives of the study are defined as follows:

1. To investigate the efficiency of Proportional Factor, $\gamma$ of Three-Term BP in classification problem.
2. To compare the performance between standard BP and Three-Term BP.

## 1.6 Project Scopes

The project scopes are defined as follows:

1. Balloon, Iris and Cancer dataset will be used as the training and testing data set which represent small, medium and large scale data.
2. Implement the standard BP and Three-Term BP using Microsoft Visual C++ 6.0.

## 1.7 Significance of Project

This project will investigate the performance of Three-Term BP proposed by Zweiri *et al.* (2003), by comparing it with Standard BP. The evaluation will be carried out since there is no extensive comparison between Standard BP and Three-Term BP has been done before and to see whether it can give better convergence rate for BP learning or not. The results of this study can be used to verify the efficiency of Three-Term BP and will contribute in future works for BP improvement.

## 1.8 Project Plan

This project will be carried out in two semesters. The first part of the project is done in the first semester where the understandings of literature review and methodology that will be used are done. Gathering information about the study is a crucial part of this part since thorough understanding is needed in order to really implement the proposed approach. Most of the information is obtained from articles and journal that can be downloaded from the Institute of Electrical and Electronic Engineering (IEEE) website and ScienceDirect website. The second part of the project is to implement the Three-Term BP and analyze the results with standard BP.

## 1.9 Organization of the Report

This report consists of four chapters which are the introduction, literature review, methodology and preliminary result. The first chapter presents introduction to the study and why this study is being conducted. It also gives the objectives and scope of the study. Chapter 2 provides reviews on ANN, standard and Three-Term BP. Chapter 3 discusses on the methodology used to carry out the study systematically. It also contains the formulation of Three-Term BP. While Chapter 4 discusses the experimental results of training and testing data using both standard and Three-Term BP. Chapter 1 is the conclusion and suggestion for future works.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

Backpropagation (BP) algorithm is a supervised learning technique used for training Multi-Layer Perceptrons (MLPs). BP is used to calculate the gradient of the error of the network with respect to the network's modifiable weights. The BP algorithm attempts to minimize the difference (or error) between the desired and actual outputs in an iterative manner. For each iteration, the weights involved in the network are adjusted by the algorithm to make the error decrease along a descent direction (Yu and Chen, 1997).

The performance of BP algorithm usually depends on network parameters such as Learning Rate, Momentum Factor, network size that includes number of layers and number of node per layer and initial weights assigned to the network. Traditionally, standard BP algorithm or the Two-Term BP utilizes two terms parameters which are the Learning Rate and Momentum Factor for controlling the weight adjustment (Zweiri *et al.*, 2003). Although standard BP is a famous algorithm for training neural network, it has been observed that its convergence rate is extremely slow, especially for the networks with more than one hidden layer (Yu *et al*, 1995). Another drawback of standard BP is the existence of local minima resulting from the saturation behaviour of the activation function (Zweiri *et al.*, 2003). Many researches had been done in order to improve the performance of standard BP algorithm. However, the algorithm modification usually involves complex calculations at each iteration.

Zweiri *et al.* (2003) had proposed a third term, which is the Proportional Factor to overcome the problems of Standard BP. The algorithm, Three-Term BP has never been implemented before in terms of programming. Thus, this project is carried out mainly to investigate the performance of Three-Term BP and compare it with Standard BP.

## 2.2 Artificial Neural Network

Artificial Neural Network (ANN) is an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation.

Neural networks are different from conventional computing or statistical systems. The networks were inspired by the structure and operation of biological neurons. Knowledge is stored in the topology of the network itself rather than in explicitly coded data structures. Neural networks are composed of many simple processing units or artificial neurons joined through numerous interconnections. These neurons are usually organized into groups called layers. The input layer is connected to the output layer through junctions with a hidden layer. The network learns by a process involving the modification of the connection weights between neurons and layers.

ANN can be classified as either feedforward, recurrent, modular, stochastic and many others, depending on how data is processed through the network. The feedforward neural networks are the first and simplest type of neural networks. In this network, the information moves in only one direction which is forward from the input nodes, through the hidden nodes and to the output nodes. The connections are formed by connecting each of the nodes in a given layer to all of the neurons in the next layer. In this way every node in a given layer is connected to every other node in the next layer.

Usually there are at least three layers (Yam and Chow, 1999) to a feedforward network which are an input layer, a hidden layer, and an output layer. The input layer does no processing. It is where the data is fed into the network. The input layer then feeds into the hidden layer. The hidden layer, in turn, feeds into the output layer. The actual processing in the network occurs in the nodes of the hidden layer and the output layer.

When enough neurons are connected together in layers, the network can be trained to do useful things using a training algorithm. Feedforward networks usually can be trained to do classification or identification type tasks on unfamiliar data.

Another way of classifying ANN types is by their method of learning, as some ANN employs supervised learning while others are referred to as unsupervised or self-organizing. Supervised learning is a learning process where both the input and outputs are provided. Unsupervised learning is a learning process where the network is provided with inputs but not with desired outputs.

The most common neural network model for feedforward networks is the multilayer perceptional (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown. A graphical representation of an MLP is shown in Figure 2.1.

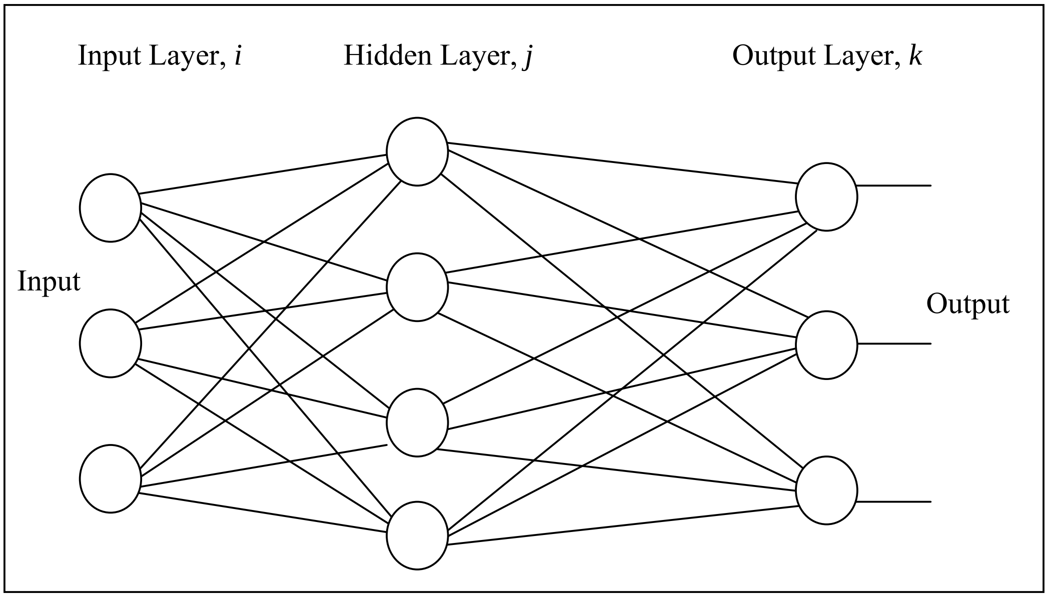


**Figure 2.1** MLP Architecture

ANN is used to train input data so that it can generate the appropriate output according to the desired target. Before the training process starts, all weights must be initialized to small random numbers. This is to make sure that the network is not saturated by large values of the weights. The process of training an ANN is based on five steps (Auda *et al.*, 1990). The steps are as follows:

1. Select the training pair from training set, applying the input vector to the network input
2. Calculate the output of the network.
3. Calculate the error between the network output and the desired output.
4. Adjust the weights of the network in a way that minimizes the error.
5. Repeat step 1 through 4 until the error is acceptably low.

## 2.3 Standard Backpropagation Algorithm

The BP algorithm is one of the most popular methods in training MLPs. There are two relatively standard definitions of backpropagation (Fogel *et al.*, 2003). The first defines backpropagation as a procedure for efficiently calculating the derivatives of some function of the outputs of any nonlinear differentiable system, with respect to all inputs and parameters of that system, through calculations proceeding backwards from outputs to inputs. The second standard definition of backpropagation is any technique for adapting the weights or parameters of a nonlinear system by using such derivatives or equivalent.

Network size usually refers to the number of hidden layers and of neurons in each layer. The network size is a compromise between generalization and convergence. Convergence is the capacity of the network to learn the patterns on the training set and generalization is the capacity to respond correctly to new patterns. The best way is to implement the smallest network possible, so it is able to learn all patterns and, at the same time, provide good generalization (Yu *et al.*, 1997).

### 2.3.1 BP Learning

The learning algorithm is performed in two stages (Roy, 1994) which are feed-forward and feed- backward. In the first phase the inputs are propagated through the layers of processing elements, generating an output pattern in response to the input pattern presented.

The output for *j* th layer is given by

$$Output = f(net_j) = \sum_j W_{ji} O_i + \theta_j$$

where,

*Wji* is the weight connected between node *i* and *j*,

$\theta_j$ is the bias of node *i*,

$O_i$ is the output of node *j*.

The most common activation function of a neuron *f(x)* is sigmoid function (Wang *et al*, 2003) as shown below:

$$f(net_j) = \frac{1}{(1 + e^{-net_j})}$$

In the second phase, the errors calculated in the output layer are then back propagated to the hidden layers where the synaptic weights are updated to reduce the error. This learning process is repeated until the output error value, for all patterns in the training set, are below a specified value.

The BP algorithm will change the current weights iteratively such that the system error function, *E* is minimized. This process is repeated iteratively until convergence is achieved. Typically, the error measure used in BP algorithm is the mean square error (MSE) (Yu and Liu, 2002). The aim of the learning is to minimize error of the output signal by modifying the weights, as shown above. The mean square error for *p* th output node is defined as follows:

$$E_p = \frac{1}{2}\sum_{j=1}^{N}(t_{pj} - o_{pj})^2$$

where,

$E_p$ = error for the $p$ th presentation vector

$t_{kj}$ is the desired value from output node ($k$) to hidden node ($j$)

$o_{kj}$ is the network value from output node ($k$) to hidden node ($j$)

The weight adaptation in standard BP is defined as:

$$\Delta W_{kj}(n) = \alpha\,(t_k - o_k)o_k(1 - o_k)\,O_j + \quad \beta\Delta W_{kj}(n-1)$$

From equation above, we could see that there are two terms added to the equation which are the α and β. These are the two terms that are usually utilized in standard BP or also known as two-term BP. The terms are added for reasons that will be stated in next section.

### 2.3.2 Two Term Parameters

The learning rate parameter (α) is used to determine how fast the BP method converges to the minimum solution. In the conventional BP learning rule, α is a decisive factor in regard to the size of weights adjustments made at each iteration and therefore it affects the convergence rate. In standard BP, α is constant throughout the training. The BP performance is very sensitive to the proper setting of the α term.

The weight adaptation of BP learning from the *k*th layer to the *j*th layer containing an α term is written as follows:

$$\Delta W_{kj}(n) = -\alpha\left(-\left(t_k - o_k\right)o_k\left(1 - o_k\right)O_j\right) \tag{2.1}$$

The best choice of α depends on problem and needs trial and error before a good choice is found. According to Yu and Liu (2002) the larger the learning rate, the bigger the step and the faster the convergence. However, if the α value is made too large the algorithm will become unstable. On the other hand, if the α value is set to too small, the algorithm will take a long time to converge. According to Sexton and Gupta (2000), if the α value is too small, it will lead to slow convergence. If the α value is too big, oscillation and overshooting of minimum will occur. The summary for behaviour of α are shown in Table 2.1.

**Table 2.1 :** Behaviour of α

| **Value** | **Effect** |
|---|---|
| Small | • Slow convergence |
| Big | • Bigger steps<br>• Faster convergence<br>• Oscillation and overshooting of minimum if α value too big |

Another possible way to improve the rate of convergence is by adding some momentum to the weight adjustment expression (Yang and Yang, 1993). This can be accomplished by adding a fraction of the previous weight change to the current weight change. β allows a network to respond not only to the local gradient, but also to recent trends in the error surface. β allows the network to ignore small features in the error surface. This term encourages movement in the same direction of successive steps. Without β a network may get stuck in a shallow local minimum. The use of β might smooth out the oscillations and produce a stable trajectory (Sexton and Gupta, 2000). As the β coefficient increases, the oscillation in the output is reduced. By using β in weight adaptation, a larger learning rate can be used while maintaining the stability of the algorithm. β also tends to accelerate convergence. The weight adaptation equation from *k*th layer to the *j*th layer with both α and β terms is written as follows:

$$\Delta W_{kj}(n) = \alpha\,(t_k - o_k)o_k(1 - o_k)\,O_j + \quad \beta \Delta W_{kj}(n-1) \tag{2.1}$$

where,

β is proportional to the previous value of the incremental change of the weights

The advantages of using β term can be summarized as follows:

1. Might smooth out oscillations occur in learning
2. Larger α value can be used if β is added in weight adaptation calculation
3. Encourages movement in the same direction of successive steps

## 2.4 Three-Term Backpropagation Algorithm

The BP algorithm is commonly used for training ANN. In standard BP, two terms, α and β are used for controlling the weight adjustment along the steepest descent direction and for dampening oscillations. The BP algorithm is popular and used for many applications. However, its convergence rate is relatively slow, especially for networks with more than one hidden layer (Zweiri *et al.*, 2003). The reason for this is the saturation behaviour of the activation function used for the hidden and the output layers (Yu and Chen, 1996). Since the output of a unit exists in the saturation area, the corresponding descent gradient takes a very small value, even if the output error is large, leading to very little progress in the weight adjustment.

This study will focus on the implementation of Three-Term BP. The standard BP weight adaptation equation given by (2.1) is modified by adding an extra term in order to increase the BP learning speed. The modification is done by adding a third term proposed by Zweiri *et al*. (2003). The third term, being the proportional term (γ)γ is proportional to:

$$e(W(k)) = \mathrm{e_s}$$

where,

$\mathrm{e_s}$ represents the difference between the output and target at each iteration

Hence the new weight adaptation for three-term BP is defined as follows:

$$\Delta W(k) = \alpha(-\nabla E(W(k))) + \beta\Delta W(k-1) + \gamma e(W(k)) \tag{2.2}$$

where,

α is proportional to the derivative of *E* (*W*(*k*)),

β is proportional to the previous value of the incremental change of the weights,

γ is proportional to $\mathrm{e_s}$

From equation (2.2), *E*(*W*(*k*)) can also be written as

$$\mathrm{E(W(k))} = \delta_{\mathrm{k}} O_j \tag{2.3}$$

Also from equation (2.2), for batch learning, e*(W(k))* could be written as

$$e(W(k)) = [e_s e_s \ldots e_s]^{\tau}$$

where,

vector *e* is of appropriate dimension of τ,

and,

$$\mathrm{e_s} = [T_j - O_j]$$

where,

$e_s$ represents the difference between the output and the target at each iteration.

Some of the modifications of BP algorithms require complex and costly calculations at each iteration, which will only offset the faster rates of convergence that is obtained using the modified BP algorithms. As we can see from equation (2.2), the Three-Term BP will maintain the simplicity of standard BP algorithm. In the paper by Zweiri *et al.* (2003), Three-Term BP was tested on XOR problem and it had significantly increased the convergence speeds while maintaining the simplicity and efficiency of standard BP. the characteristics of each learning parameters in Three-Term BP is summarized in Table 2.2.

**Table 2.2 :** Characteristics of BP Learning Parameters

| **Learning Rate (α)** | **Momentum Factor (β)** | **Proportional Factor (γ)** |
|---|---|---|
| Proportional to the derivative of $E(W(k))$ | Proportional to the previous value of the incremental change of the weights | Proportional to the difference between the output and the target at each iteration, $e_s$ |
| Added to increase convergence speed | Added to smooth out oscillation, increase convergence speed | Added to increase convergence speed, escape from local minima |
| Too large value will make network unstable | Enable the network to use larger α value | Does not effect the complexity of standard BP |

## 2.5 Summary

In this chapter, we have discussed about the concept of ANN and one of the widely used learning algorithm to train the ANN which is the BP algorithm. The BP learning is discussed thoroughly to give a better understanding of the complete formulation of BP network. The formulation of term $\gamma$ is also discussed in this chapter to give a clear description and comparison between the weight adaptation of standard BP and Three-Term BP.

# CHAPTER 3

# METHODOLOGY

This chapter discusses the methodology that will be used in this project and describes the techniques and parameters that are required in BP learning. The next section will discuss the experiments and analysis of results in order to investigate the efficiency of Three-Term BP. A comparison between standard BP and Three-Term BP in solving classification problems is also addressed.

## 3.1 Introduction

There are a few basic steps in order to implement the standard BP. These basic steps will also be used in implementing the Three-Term BP. The basic steps could be defined as follows:

1. Determine training patterns from dataset
2. Define neural network architecture
3. Start training standard BP
4. Implement the third term,$\gamma$ in addition for standard BP

5. Train Three-Term BP
6. Do comparison of standard BP and three term BP
7. Experiment and analysis

These basic steps can be followed in order to implement both standard and Three-Term BP. However, step 4 is not used in implementing standard BP since standard BP only utilizes two existing terms which are the α and β. The steps are divided into three phases. The first phase involves implementing standard BP, the second involves implementing Three-Term BP and the last phase involves carrying out the experiment and analysis of both algorithms. The training of Three-Term BP will only be carried out in the second phase of this project. A general framework of this study is shown in Figure 3.1.

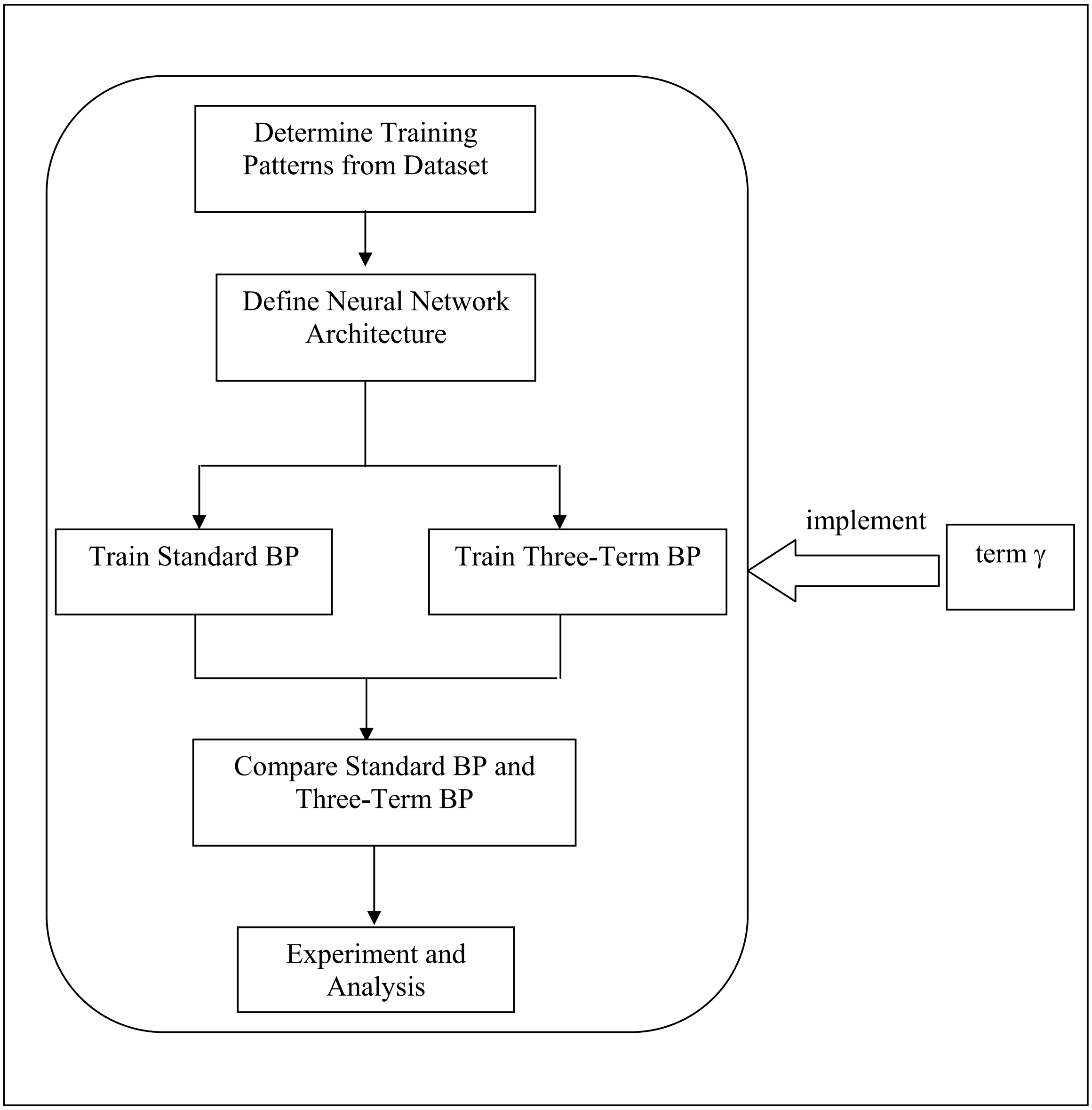

Determine Training Patterns from Dataset
Define Neural Network Architecture
Train Standard BP
Train Three-Term BP
implement
term γ
Compare Standard BP and Three-Term BP
Experiment and Analysis

**Figure 3.1** A General Framework of the Study

## 3.2 Dataset

The dataset used in solving classification problem using BP algorithm are Balloon dataset, Iris dataset and Cancer dataset. These data represents small, medium and large scale data based on their size. Balloon data is chosen to represent small scale data, Iris data represents medium scale data and Cancer data represents large scale data. These data are used to evaluate the performance of both standard and Three-Term BP algorithms in terms of classification accuracy and convergence speed.

### 3.2.1 Balloon Dataset

Balloon dataset is used to represent small scale data. It is used for classifying four balloon attributes which are colour (yellow, purple), size (large, small), act (stretch, dip) and age (adult, child) into a class, either inflated (T) or not (F). Balloons dataset contains 16 instances. The network will have 4 inputs to represent information for each attribute and 1 output to represent either it is inflated or not. The Balloon dataset were split with 12 for training data and 4 for testing data, totalling 16 instances for the whole Balloon dataset.

There are four sets of data in Balloons dataset that represents different conditions of an experiment which are:

1. Adult-stretch data: Inflated is true if age = adult or act = stretch
2. Adult + stretch data: Inflated is true if age = adult and act = stretch
3. Small - yellow data: Inflated is true if (colour = yellow and size = small) or
4. Small – yellow + adult – stretch data: Inflated is true if (colour = yellow and size = small) or (age = adult and act = stretch)

The summary for Balloon dataset for classification is shown in Table 3.1 below:

**Table 3.1 :** Summary of Balloon Attributes for Classification

| No. | Attribute | Value1 | Value2 |
|---|---|---|---|
| **1** | Colour | Yellow | Purple |
| **2** | Size | Large | Small |
| **3** | Act | Stretch | Dip |
| **4** | Age | Adult | Child |

### 3.2.2 Iris Dataset

The classifying of Iris dataset involves classifying the data of petal width, petal length, sepal width and sepal length into three classes of species which are iris Setosa, Versicolor and Verginica. This dataset contains data for the three classes with 25 instances for each class. The input consists of 4 numeric attributes related to the length and the width of the sepals and petals of iris plant. The network will have 4 inputs and 3 outputs to separate the pattern. Summary of Iris dataset attributes for classification is shown in Table 3.2.

**Table 3.2 :** Summary of Iris Attributes for Classification

| No. | Attribute | Value |
|---|---|---|
| **1** | Petal Width | Numeric Values |
| **2** | Petal Length | |
| **3** | Sepal Width | |
| **4** | Sepal Length | |

### 3.2.3 Cancer Dataset

The classifying of Cancer dataset involves classifying the data into two classes of breast lump's diagnosis which are either benign or malignant. These data were obtained from automated microscopic examination of cells collected by needle aspiration. All inputs are continuous variables and 65.5% of the examples are benign. The data set was originally generated at hospitals at the University of Wisconsin Madison, by Dr. William H. Wolberg. The data set includes 9 inputs and 1 output. The data are split into 500 for training, and 100 for testing, totaling of 600 instances. Summary of Cancer dataset attributes for classification is shown in Table 3.3.

**Table 3.3 :** Summary of Cancer Attributes for Classification

| | **No. of Instances** |
|---|---|
| **Training** | 500 |
| **Testing** | 100 |
| **Total of Instances** | 600 |

## 3.3 Defining Neural Network Architecture

To train the standard BP using the chosen universal datasets, we must first define the network architecture. In this project, the network architecture that will be used consists of three layers which are one input layer, one hidden layer and one output layer. Defining the network architecture usually involves the selecting of an appropriate number for input, hidden and output layers and also in selecting the number of nodes in each layer according to which application will be used. In this project, the network architecture is defined as to solve to classification problem for Balloon, Iris and Cancer data.

The number of nodes required in each layer differs from one dataset to another. Each input node defined represents the set of problem that will be classified and the output node represents the classes that the input data will belong to after classification has been done. The number of hidden nodes will be defined as (Masters, 1993):

$$\text{number of hidden nodes} = \sqrt{m * n}$$

where,

$m$ is the number of input nodes,

$n$ is number output nodes

The summary of defined neural network architecture for this project is shown according to dataset.

### 3.3.1 Balloon Dataset

The Balloon dataset has 4 attributes for each instances of Balloon which are colour, size, act and age. Therefore, the number of input nodes to represent this problem is 4. Meanwhile the number of output node for this dataset is only 1 since there is only one possible outcome for each combination of attributes which is either the balloon is inflated (T) or not inflated (F). The complete summary of the network architecture used in representing this problem can be viewed in Table 3.4. The network structure for this dataset is shown is Figure 3.2.

**Table 3.4 :** Network Architecture For Balloon Dataset

| | |
|---|---|
| Input Layer | 1 |
| Hidden Layer | 1 |
| Output Layer | 1 |
| Input Nodes | 4 |
| Hidden Nodes | 2 |
| Output Nodes | 1 |

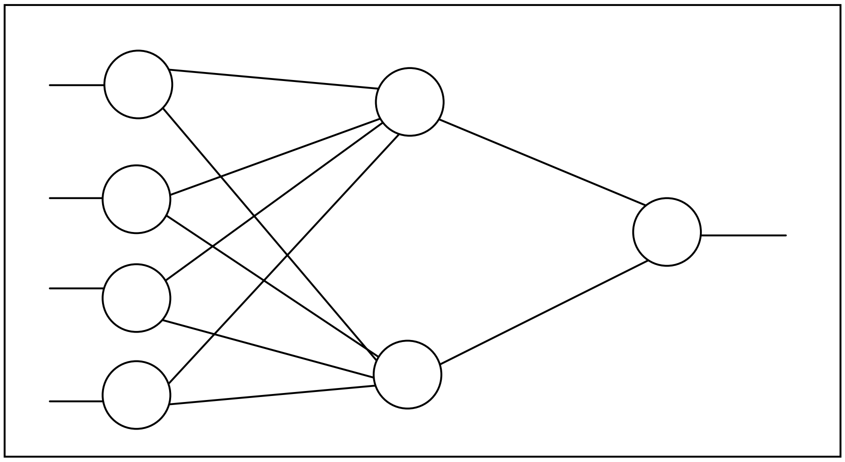


**Figure 3.2** Network Structure for Balloon Dataset

### 3.3.2 Iris Dataset

The network architecture for Iris dataset comprises of 4 input nodes, 3 output nodes and 3 hidden nodes. The Iris dataset has 4 attributes for each instances of Iris which are the petal width, petal length, sepal width and sepal length. This is why 4 input nodes are chosen to represent this problem. Meanwhile, for the selection of the output nodes are based on the 3 types of Iris that each instances will be classified into which are Setosa, Versicolor and Verginica. Therefore, the number of output nodes chosen for this problem is 3. The complete summary of the network architecture used in representing this problem can be viewed in Table 3.5. The network structure for this dataset is shown is Figure 3.3.

**Table 3.5 :** Network Architecture For Iris Dataset

| | |
|---|---|
| Input Layer | 1 |
| Hidden Layer | 1 |
| Output Layer | 1 |
| Input Nodes | 4 |
| Hidden Nodes | 3 |
| Output Nodes | 3 |

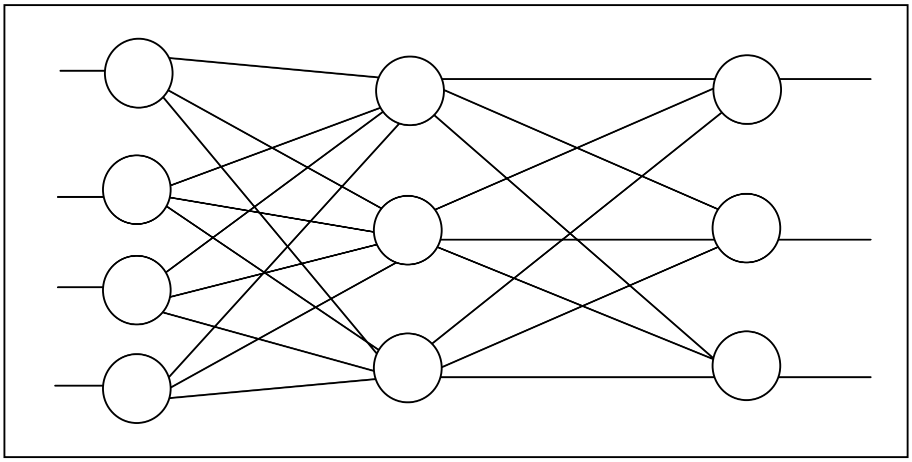


**Figure 3.3** Network Structure for Iris Dataset

### 3.3.3 Cancer Dataset

The network architecture for Cancer dataset comprises of 9 input nodes, 1 output nodes and 3 hidden nodes. The Cancer dataset has 9 attributes for each instances of Cancer. This is why 9 input nodes are chosen to represent this problem. Meanwhile, for the selection of the output node is based on the one output class which is type of Cancer. Therefore, the number of output node chosen for this problem is 1 node. The value for this class is either benign or malignant type of breast cancer diagnosis. The complete summary of the network architecture used in representing this problem can be viewed in Table 3.6. The network structure for this dataset is shown is Figure 3.4.

**Table 3.6 :** Network Architecture For Cancer Dataset

| | |
|---|---|
| Input Layer | 1 |
| Hidden Layer | 1 |
| Output Layer | 1 |
| Input Nodes | 9 |
| Hidden Nodes | 3 |
| Output Nodes | 1 |

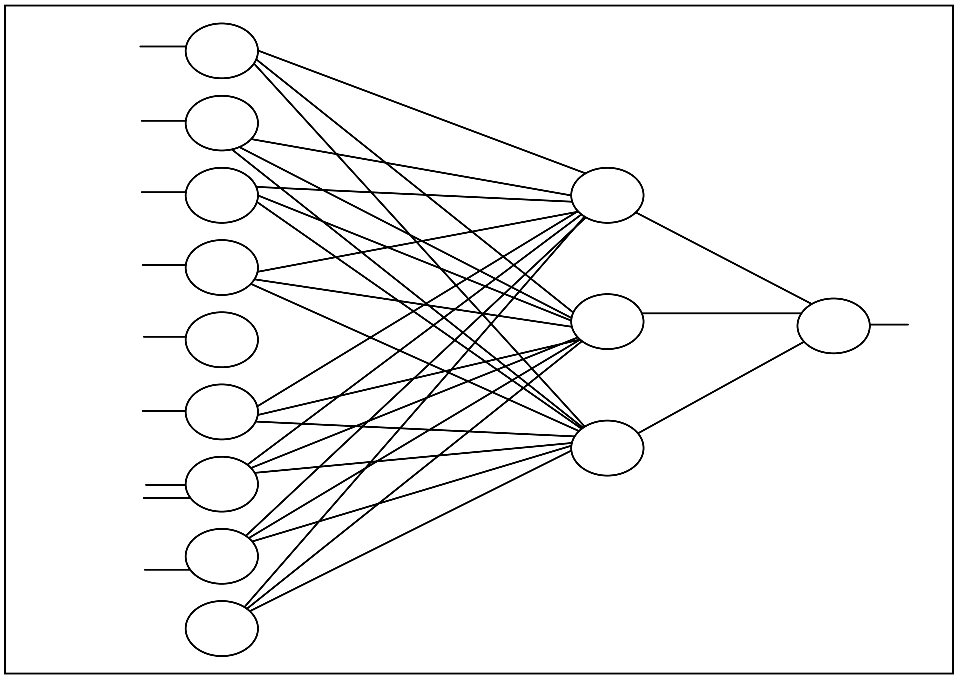


**Figure 3.4** Network Structure for Cancer Dataset

## 3.4 Training BP algorithm

The standard BP needs to be trained to minimize the error measure by adjusting the weights. The network will be trained after all the network structures have been defined. In order to train the network, the initial weights and bias must be defined. Another parameter that is needed to be defined is the activation function. This project will use sigmoid function as an activation function. The maximum error must also be defined as it would be used as comparison with the network error and the training will be repeated until the network error is less than maximum error.

The basic steps in training BP are as follows:

1. Apply input to the network.
2. Calculate the output.

3. Compare the resulting output with the desired output for the given input. This is called the error.
4. Modify the weights and threshold θ for all neurons using the error.
5. Repeat the process until error reaches an acceptable value which means that the NN was trained successfully, or if a maximum count of iterations is reached, then it means the NN training was not successful.

The same steps will also be used to train Three-Term BP. The difference between standard BP and Three-Term BP is in terms of the weight adjustments, and will be discussed in next section.

## 3.5 Implementing Proportional Factor, γ

The standard BP usually utilizes two term which are the learning rate, α and momentum factor, β.The proportional factor, γ will be implemented in Three-Term BP alongside the other two terms α and β. The value of γ will be determined using trial and error method.

This study focuses on batch learning using Balloon, Iris and Cancer dataset. Batch learning will only do the training phase and then the testing phase. It holds no responsibility for performance during learning, unlike online learning. The BP algorithm is modified by adding an extra term in order to increase the BP learning speed. This term is proportional to *e(W(k))* which represents the difference between the output and the target at each iteration.

The error measure, *E* used in this project is the Mean Square Error (MSE). *E* is defined as follows:

$$E_p = \frac{1}{2}\sum_{j=1}^{N}(t_{kj} - o_{kj})^2$$

where,

$E_p$ = error for the *p*th presentation vector

$t_{kj}$ is the desired value from output node (*k*) to hidden node (*j*)

$o_{kj}$ is the network value from output node (*k*) to hidden node (*j*)

The weight changes are proportional to the derivative of *E*. For example, the change in weights between output layer, *k* and hidden layer, *j* can be written as follows:

$$\Delta W_{kj} = -\alpha \frac{\partial E}{\partial W_{kj}}$$

where,

$\alpha$ is learning rate

By chain rule, equation above can be written as:

$$\frac{\partial E}{\partial W_{kj}} = \frac{\partial E}{\partial net_k} \times \frac{\partial net_k}{\partial W_{kj}} \qquad (3.1)$$

Let the error signal, $\delta_k$ be

$$\delta_k = \frac{\partial E}{\partial net_k} \qquad (3.2)$$

Since $net_k = \sum_k W_{kj} O_j + \theta_k$, by doing a partial derivation of it we will get

$$O_j = \frac{\partial net_k}{\partial W_{kj}} \qquad (3.3)$$

By substituting (3.3) and (3.2) into (3.1), we will get

$$\frac{\partial E}{\partial W_{kj}} = \delta_k \times O_j \tag{3.4}$$

From (3.2), we know that $\delta_k = \frac{\partial E}{\partial net_k}$.

This is obtained by chain rule

$$\delta_k = \frac{\partial E}{\partial o_k} \times \frac{\partial o_k}{\partial net_k} \tag{3.5}$$

The partial derivative of error function, $E = \frac{1}{2}\sum_k (t_{kj} - o_{kj})^2$ can be written as

$$\frac{\partial E}{\partial o_k} = -\left(t_k - o_k\right) \tag{3.6}$$

The output of *k* th layer is given by $o_k = \frac{1}{1+e^{-net_k}}$.

Therefore the partial derivative of $o_k$ is written as

$$\frac{\partial o_k}{\partial net_k} = o_k\left(1 - o_k\right) \tag{3.7}$$

By substituting (3.6) and (3.7) into (3.5), we will get

$$\delta_k = -\left(t_k - o_k\right) o_k \left(1 - o_k\right) \tag{3.8}$$

By substituting (3.8) into (3.4), we will get

$$\frac{\partial E}{\partial W_{kj}} = -\left(t_k - o_k\right) o_k \left(1 - o_k\right) \times O_j \tag{3.9}$$

The weight adaptation between output layer and hidden layer now can be written as

$$\Delta W_{kj} = -\alpha \frac{\partial E}{\partial W_{kj}} \tag{3.10}$$

By substituting (3.9) into (3.10), we will get

$$\Delta W_{kj}(n) = -\alpha\left(-\left(t_k - o_k\right)o_k\left(1 - o_k\right)O_j\right)$$

$$\Delta W_{kj}(n) = \alpha\left(t_k - o_k\right)o_k\left(1 - o_k\right)O_j \tag{3.11}$$

By adding momentum term $\beta$ to equation (3.11), the weight adaptation is now

$$\Delta W_{kj}(n) = \alpha\left(t_k - o_k\right)o_k\left(1 - o_k\right)O_j + \quad \beta\Delta W_{kj}(n-1) \tag{3.12}$$

$\gamma$ factor is added to equation (3.12) giving the weight adaptation as

$$\Delta W(k) = \alpha\left(-\nabla E\left(W(k)\right)\right) + \beta\Delta W(k-1) + \gamma e\left(W(k)\right) \tag{3.13}$$

where,

$\alpha$ is proportional to the derivative of $E(W(k))$,

$\beta$ is proportional to the previous value of the incremental change of the weights,

$\gamma$ is proportional to $e_s$

From equation (3.13), $E(W(k))$ can also be written as

$$\mathrm{E(W(k))} = \delta_{\mathrm{k}} O_j \tag{3.14}$$

where,

$\delta_{\mathrm{k}}$ can be obtained from equation (3.5)

Also from equation (3.13), for batch learning, e*(W(k))* could be written as

$$e(W(k)) = \left[ e_s e_s \ldots e_s \right]^\tau$$

where,

vector *e* is of appropriate dimension of τ,

and, $$\mathrm{e_s} = \lfloor T_j - O_j \rfloor$$

where,

$\mathrm{e_s}$ represents the difference between the output and the target at each iteration.

The weights adaptation between hidden layer , *j* and input layer, *i* of the standard BP algorithm is similar as updating weight between output layer, *k* and hidden layer, *j* (Ng *et al.*, 1996).

### 3.6 Comparing Standard and Three-Term BP

In order to evaluate the performance of the Three-Term BP, a comparison between the performance of standard BP and Three-Term BP will be done. To compare between the standard and Three-Term BP, both algorithms will be using the same network parameters which are the number of layers, number of nodes for each layer, same range of initial values, maximum error, learning rate, momentum value. The difference is the Three-Term BP will use the proportional factor to adjust the weights. This part will be carried out after the implementation of γ is done.

## 3.7 Experiment and Analysis

The experiment and analysis part must be carried out in this project to achieve one of its objectives which is to evaluate the performance of Three-Term BP and do a comparative study between the standard and Three-Term BP in terms of performance, after using the same value for the network parameters for both standard and Three-Term BP.

The analysis part will be done in testing the network of standard and Three-Term BP. The testing is done to evaluate the performance of the Three-Term BP algorithm in solving the classification problem.

## 3.8 Summary

This chapter discusses mainly about the methodology that will be used throughout this project. The basic steps in the methodology are discussed and a general framework of the study is also shown in this chapter.

The network architecture is first defined and then the standard BP will be trained. The implementation of the third term, $\gamma$ will be done and then the Three-Term BP will be trained to minimize the error measurement. The experiment and analysis is another crucial part of this project since this project focuses mainly on doing a comparative study between Three-Term BP and standard BP, and also evaluating the performance of Three-Term BP in solving classification problem.

# CHAPTER 4

# EXPERIMENTAL RESULTS AND ANALYSIS

This chapter discusses the experimental results of this project and its analysis. The experimental results are obtained from training both the standard and Three Term BP. The results are measured from its performance in terms of convergence rate and classification accuracy of the Balloon, Iris and Cancer dataset.

## 4.1 Introduction

The performance evaluation of standard and Three Term BP are carried out based on its convergence rate and accurate classification of presented problem. The standard BP is trained using Balloon, Iris and Cancer data as mentioned in chapter 3. In this study, two programs have been developed which are standard BP and Three Term BP. As described in chapter 3, the $\gamma$ term proposed by Zweiri *et. al.* (2003) is added in the weight adaptation of standard BP. The experiments are implemented with different types of classification problems where Balloon dataset represents small scale data, Iris dataset represents medium and Cancer dataset represents a large scale data. The analysis of both standard and Three-Term BP's performance will be discussed in next section.

## 4.2 Experimental Result

Balloon, Iris and Cancer data were used for training and testing the standard and Three-Term BP throughout this project. For standard BP, the experiments are divided into two set of tests which contains 9 trials each. For the first test (Test I), the same value of α and β are used in the range of [0.1, 0.9]. While in the second test (test II), the values of α and β are increased and decreased, respectively with 0.1 as the initial value for α and 0.9 for β.

For Three-Term BP, the experiments are divided into three set of tests which also contains 9 trials each. For the comparisons, Three-Term BP uses the same value range of α, β and γ which is also [0.1, 0.9]. The first test (Test I) uses the same values of α, β and γ for all 9 trials. The second test (Test II) uses the same increasing value for α and β and decreasing value for γ. α and β values start from 0.1 and γ value starts from 0.9. The third test (Test III) uses increasing value for α and decreasing values for both β and γ, where α value starts from 0.1 and β and γ values both start from 0.9.

### 4.2.1 Balloon Dataset Analysis

Balloon dataset are used to represent small scale data where the data consists of 16 instances. In Balloon dataset experiment, the network architecture consists of 4 input nodes, 2 hidden nodes and 1 output node. 12 instances were represented to the network as training data set and 4 instances as testing data set. The results of Standard BP are shown in Table 4.1(a) and (b), while for Three-Term BP, the results are shown in Table 4.2(a), (b) and (c).

**Table 4.1(a) :** Results of Standard BP in Balloon Dataset (Test I)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Error Generated** | 0.04993 | 0.0499 | 0.0497 | 0.0497 | 0.0492 | 0.0495 | 0.0481 | 0.0497 | 0.0493 |
| **Learning Iteration** | 1041 | 455 | 261 | 164 | 107 | 69 | 44 | 28 | 25 |
| **Process Time** | 1.000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| **Correct Classification** | 75% | 75% | 75% | 75% | 75% | 75% | 75% | 75% | 75% |

**Table 4.1(b) :** Results of Standard BP in Balloon Dataset (Test II)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 0.0499 | 0.0499 | 0.0494 | 0.0494 | 0.0495 | 0.0495 | 0.0499 | 0.0497 | 0.0497 |
| **Learning Iteration** | 115 | 112 | 110 | 108 | 107 | 106 | 105 | 105 | 105 |
| **Process Time** | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| **Correct Classification** | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% |

**Table 4.2(a) :** Results of Three-Term BP in Balloon Dataset (Test I)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Gamma Term** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Error Generated** | 0.0499 | 0.0499 | 0.0497 | 0.0493 | 0.0489 | 0.0493 | 0.0495 | 0.0364 | 0.0373 |
| **Learning Iteration** | 960 | 409 | 228 | 140 | 90 | 91 | 77 | 207 | 130 |
| **Process Time** | 1.000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| **Correct Classification** | 75.00 | 75.00 | 75.00 | 75.00 | 75.00 | 75.00 | 75.00 | 75.00 | 75.00 |

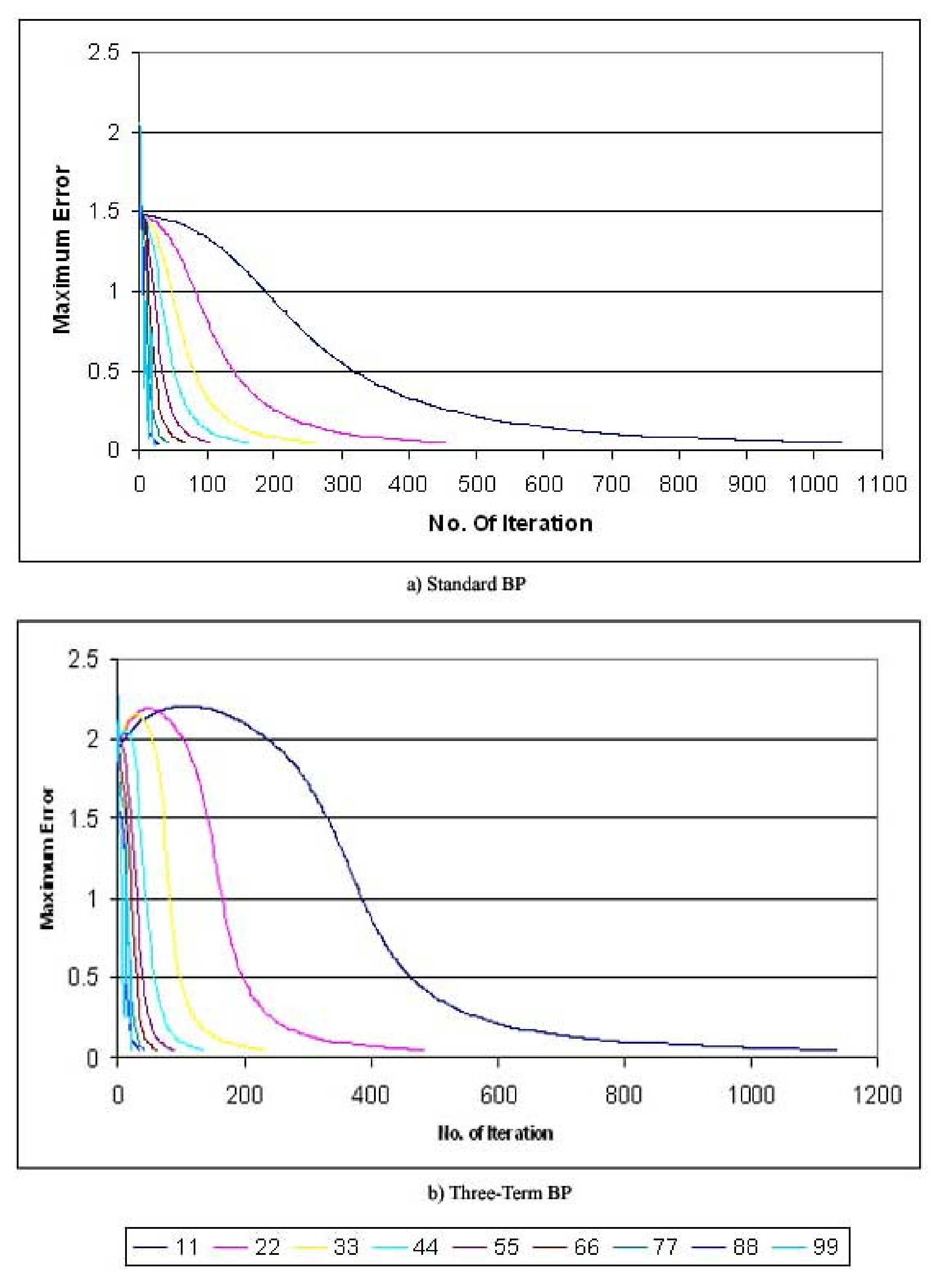
**Table 4.2(b) :** Results of Three-Term BP in Balloon Dataset (Test II)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Gamma Term** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 0.0499 | 0.0499 | 0.0498 | 0.0498 | 0.0489 | 0.0493 | 0.0473 | 0.0492 | 0.0426 |
| **Learning Iteration** | 1136 | 485 | 233 | 136 | 90 | 62 | 44 | 35 | 23 |
| **Process Time** | 1.000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| **Correct Classification** | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% |

**Table 4.2(c) :** Results of Three-Term BP in Balloon Dataset (Test III)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Gamma Term** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 0.0492 | 0.0488 | 0.0490 | 0.0498 | 0.0489 | 0.0494 | 0.0493 | 0.0495 | 0.0492 |
| **Learning Iteration** | 168 | 147 | 115 | 97 | 90 | 91 | 92 | 100 | 107 |
| **Process Time** | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| **Correct Classification** | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% | 75.00% |

From the results shown above, the results for Balloon dataset using Standard BP for all trials in both tests gave the same correct classification percentage which is 75%. However we could see that the experiments differ in terms of error generated and number of learning iterations.  The least number of learning iteration is given by TE9 of Test I with 25 iterations.  The same trial also gave the lowest error generated by the BP network which is 0.0493.  Three-Term BP also gave the same correct classification percentage as Standard BP which is 75%.  The best performance in terms of learning iteration is given by TE9 in Test II which is 23 iterations and it is also the least number of iterations compared to all experiments in Standard BP. Learning pattern for Balloon dataset in both algorithms is shown in Figure 4.1. These learning patterns are taken from the tests with the least number of iterations for both algorithms which is Test I for Standard BP and Test II for Three-Term BP.



a) Standard BP

b) Three-Term BP

**Figure 4.1** Characteristic Convergence of Balloon Dataset

Figure 4.1 a) shows that the errors for all trials in Standard BP converged in a quite similar pattern. The errors generated converged closely match the maximum error function specified earlier. The error converged closely to each other in TE6 to

TE9 for Standard BP where the learning iterations ranged from 65-70 iterations with TE9 as trial with the lowest number of iterations. For Three-Term BP as shown in Figure 4.1 b), the errors converged in quite different patterns from one to another. However, the error signal converged much faster to solution within only 23 iterations compared to solution obtained from Standard BP. The detailed analysis and comparison based on these results will be discussed in section 4.3.

In analysis and comparison part, the results that will be analyzed are the best results in terms of number of iterations for both Standard and Three-Term BP, since the percentage for correct classification is the same for both algorithms. Therefore TE9 of Test I is chosen for experiment using Standard BP and TE9 of Test II is chosen for Three-Term BP.

### 4.2.2 Iris Dataset

Iris dataset is used to represent medium scale data where the data consists of 150 instances. In Iris dataset experiment, the network architecture consists of 4 input nodes, 3 hidden nodes and 3 output node. 100 instances were represented to the network as training data set and 50 instances as testing data set. The results of Standard BP are shown in Table 4.3(a) and (b), while for Three-Term BP, the results are shown in Table 4.4(a), (b) and (c).

**Table 4.3(a) :** Results of Standard BP in Iris Dataset (Test I)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Error Generated** | 0.05 | 0.0499 | 0.0499 | 0.0495 | 0.0482 | 0.0495 | 0.0373 | 0.0377 | 19.994 |
| **Learning Iteration** | 16534 | 6472 | 3663 | 2814 | 3052 | 2452 | 1665 | 1064 | 50000 |
| **Process Time** | 21 | 9 | 5 | 4 | 4 | 4 | 2 | 2 | 54 |
| **Correct Classification** | 96% | 96% | 96% | 96% | 96% | 96% | 100% | 100% | 50% |

**Table 4.3(b) :** Results of Standard BP in Iris Dataset (Test II)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 0.04983 | 0.0482 | 0.0492 | 0.0499 | 0.0482 | 0.0448 | 0.0498 | 0.0492 | 0.0417 |
| **Learning Iteration** | 1473 | 2328 | 2773 | 3215 | 3052 | 3064 | 3174 | 3130 | 3152 |
| **Process Time** | 2 | 3 | 4 | 5 | 4 | 4 | 5 | 4 | 4 |
| **Correct Classification** | 96% | 96% | 96% | 96% | 96% | 96% | 96% | 96% | 96% |

**Table 4.4(a) :** Results of Three-Term BP in Iris Dataset (Test I)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Gamma Term** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Error Generated** | 0.04999 | 0.0472 | 0.0491 | 0.0497 | 0.0495 | 0.0047 | 39.622 | 40.00 | 80.00 |
| **Learning Iteration** | 10787 | 4437 | 3448 | 2408 | 2303 | 6622 | 50000 | 50000 | 50000 |
| **Process Time** | 16 | 7 | 6 | 4 | 3 | 11 | 74 | 67 | 62 |
| **Correct Classification** | 96% | 96% | 96% | 96% | 96% | 96% | 80% | 80% | 60% |

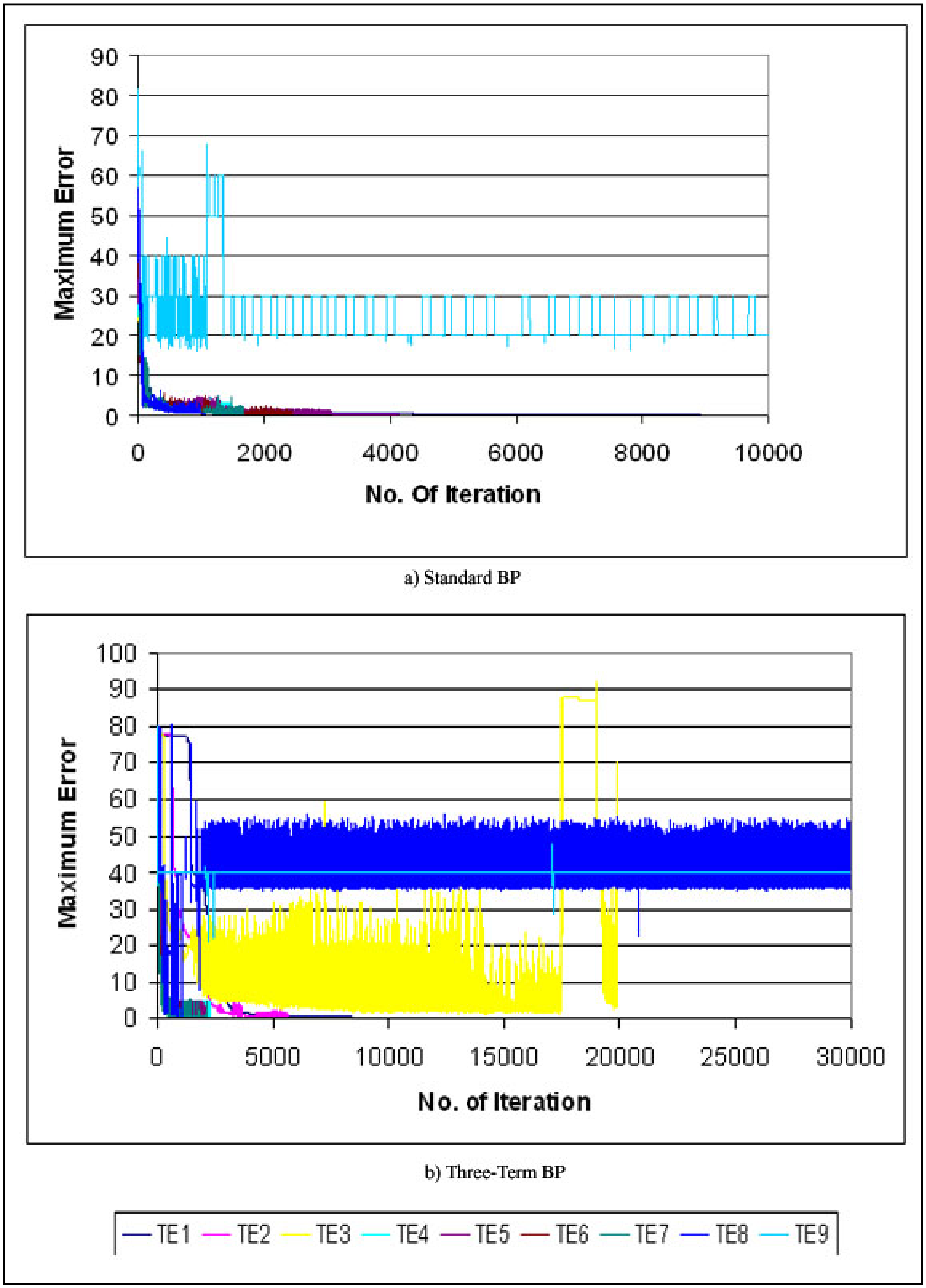
**Table 4.4(b) :** Results of Three-Term BP in Iris Dataset (Test II)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Gamma Term** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 0.05 | 0.0461 | 50.00 | 0.0499 | 0.0495 | 0.0498 | 0.0396 | 38.514 | 39.952 |
| **Learning Iteration** | 16218 | 5616 | 50000 | 2515 | 2303 | 2150 | 2149 | 50000 | 50000 |
| **Process Time** | 27 | 9 | 88 | 4 | 3 | 3 | 4 | 102 | 87 |
| **Correct Classification** | 96% | 96% | 96% | 96% | 96% | 96% | 96% | 96% | 96% |

**Table 4.4(c) :** Results of Three-Term BP in Iris Dataset (Test III)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Gamma Term** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 80 | 26.646 | 79.999 | 79.961 | 0.0495 | 0.0497 | 22.521 | 1.6718 | 0.0496 |
| **Learning Iteration** | 50000 | 50000 | 50000 | 50000 | 2303 | 2289 | 50000 | 50000 | 3286 |
| **Process Time** | 68 | 73 | 58 | 60 | 3 | 3 | 57 | 55 | 5 |
| **Correct Classification** | 60% | 76% | 56% | 56% | 96% | 96% | 76% | 76% | 96% |

From the results shown above, the results for Iris dataset using Standard BP gave a better correct classification percentage with almost all trials gave between 96% to 100% correct classification compared to Three-Term BP which had a percentage ranging from 56% to 96%. For standard BP, the best performance in terms of correct classification percentage and the lowest error generated is obtained from TE8 of Test I with 96% correct classification, an error of 0.0377 and the fastest processing time which is 2 seconds. Meanwhile for Three-Term BP, trial TE7 of Test II gave the best correct classification compared to other Three-Term experiments with 96% correct classification, error of 0.0396 and 4 seconds of processing time. From here we can see that the best result from Three-Term BP is not as satisfactory as result's from standard BP. Learning pattern for Iris dataset in both algorithms is shown in Figure 4.2. These learning patterns are taken from the tests with the best percentage of correct classification which is Test I for standard BP and Test II for Three-Term BP.



a) Standard BP

b) Three-Term BP

**Figure 4.2** Characteristic Convergence of Iris Dataset

Figure 4.2 shows that the errors for all trials in standard BP converged in quite a similar pattern except for TE9 where the error did not converge within 50,000 iterations. For Three-Term BP, the error converged in quite a different pattern from one to another. The errors generated did not converge into solution for TE3, TE8 and TE9. From here we can see that Standard BP gave a better range of error convergence compared to Three-Term BP. Error signals generated by standard BP ranged from 0.0377 to 19.994 while for Three-Term BP the error signals ranged from 0.0396 to 50.00.

In the analysis, only one single result is taken from both standard and Three-Term BP that is considerably good for all trials. Therefore the results from TE8 of Test I for standard BP and TE7 of Test II from Three-Term BP will be used in the analysis part. The detailed analysis and comparison based on these results will be discussed in section 4.3.

### 4.2.3 Cancer Dataset

Cancer dataset is used to represent large scale data where the data consists of 600 instances. In Cancer dataset experiment, the network architecture consists of 9 input nodes, 3 hidden nodes and 1 output node. 500 instances were represented to the network as training data set and 100 instances as testing data set. In this experiment, both Standard and Three-Term BP training could not converge within 10,000 iterations (Shafie, 2005). Therefore, a stopping criterion for training is set so the training will terminate after 10,000 iterations. The results of Standard BP are shown in Table 4.5(a) and (b), while for Three-Term BP, the results are shown in Table 4.6(a), (b) and (c).

**Table 4.5(a) :** Results of Standard BP in Cancer Dataset (Test I)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Error Generated** | 4.1273 | 3.6932 | 3.6610 | 3.6408 | 4.1562 | 4.3165 | 4.3587 | 3.7734 | 4.2359 |
| **Learning Iteration** | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |
| **Process Time** | 44 | 40 | 43 | 42 | 46 | 42 | 43 | 45 | 46 |
| **Correct Classification** | 50.00% | 50.00% | 50.00% | 49.00% | 49.00% | 50.00% | 50.00% | 49.00% | 49.00% |

**Table 4.5(b) :** Results of Standard BP in Cancer Dataset (Test II)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 4.2854 | 4.2848 | 3.0015 | 3.0016 | 4.1562 | 4.0346 | 4.0316 | 4.0282 | 4.0238 |
| **Learning Iteration** | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |
| **Process Time** | 42 | 42 | 42 | 40 | 40 | 41 | 40 | 45 | 42 |
| **Correct Classification** | 49.00% | 49.00% | 49.00% | 50.00% | 49.00% | 49.00% | 49.00% | 49.00% | 50.00% |

**Table 4.6(a) :** Results of Three-Term BP in Cancer Dataset (Test I)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Gamma Term** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Error Generated** | 4.0019 | 5.0003 | 5.5000 | 8.0000 | 11.500 | 5.0300 | 6.3706 | 5.3707 | 8.5000 |
| **Learning Iteration** | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |
| **Process Time** | 51 | 48 | 57 | 60 | 60 | 70 | 183 | 46 | 63 |
| **Correct Classification** | 48.00% | 48.00% | 48.00% | 47.00% | 49.00% | 48.00% | 48.00% | 48.00% | 49.00% |

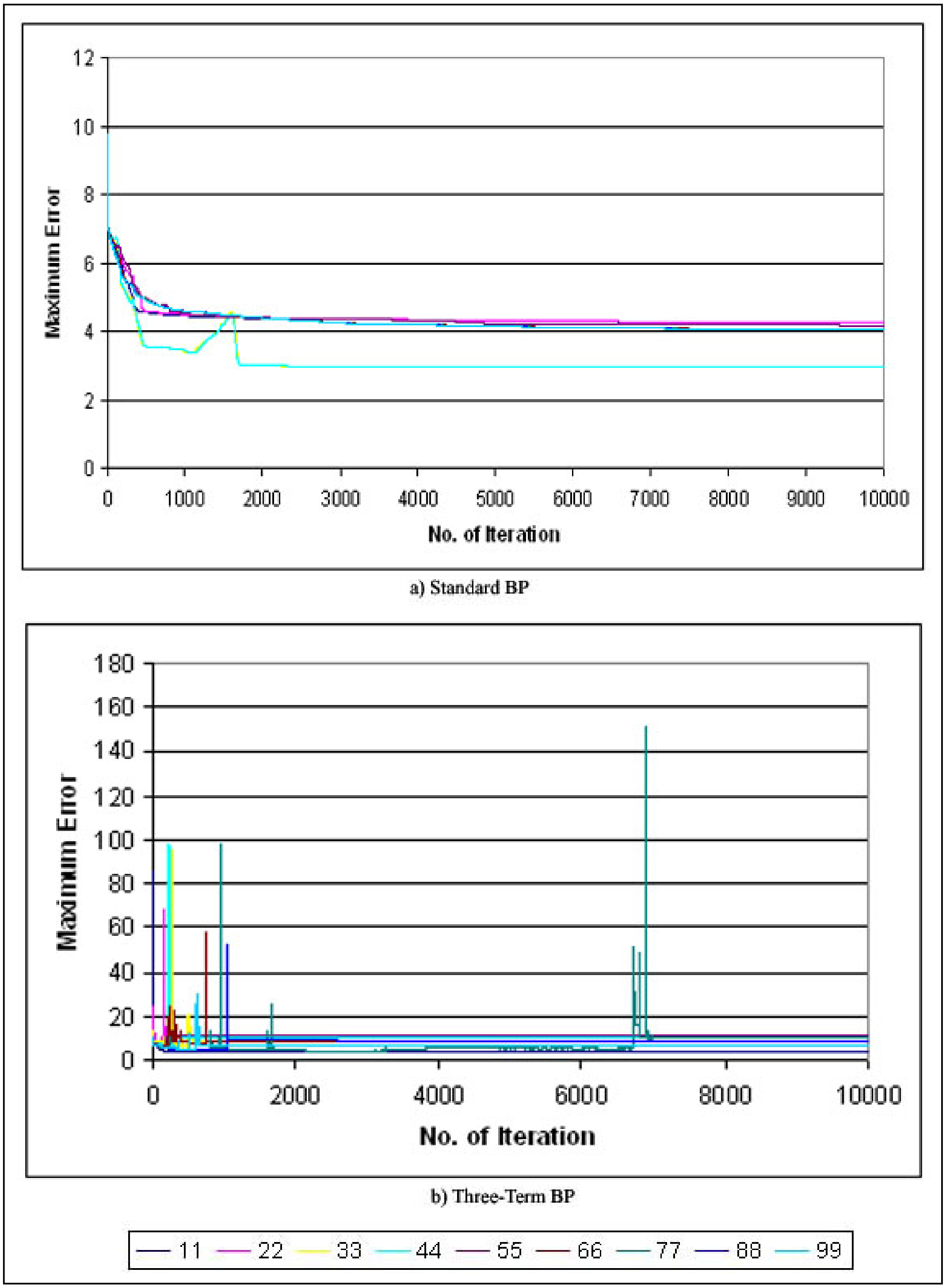
**Table 4.6(b) :** Results of Three-Term BP in Cancer Dataset (Test II)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Gamma Term** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 7.0000 | 7.0000 | 7.5000 | 7.5000 | 11.500 | 83.370 | 4.3239 | 3.3889 | 6.7473 |
| **Learning Iteration** | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |
| **Process Time** | 61 | 69 | 65 | 63 | 60 | 43 | 131 | 55 | 193 |
| **Correct Classification** | 48.00% | 49.00% | 49.00% | 49.00% | 49.00% | 47.00% | 49.00% | 47.00% | 47.00% |

**Table 4.6(c) :** Results of Three-Term BP in Cancer Dataset (Test III)

| | TE1 | TE2 | TE3 | TE4 | TE5 | TE6 | TE7 | TE8 | TE9 |
|---|---|---|---|---|---|---|---|---|---|
| **Learning Rate** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| **Momentum** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Gamma Term** | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| **Error Generated** | 3.8692 | 9.5000 | 8.5000 | 9.5000 | 11.500 | 8.0000 | 11.000 | 8.5000 | 7.0000 |
| **Learning Iteration** | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |
| **Process Time** | 59 | 59 | 57 | 61 | 60 | 55 | 59 | 57 | 59 |
| **Correct Classification** | 48.00% | 48.00% | 49.00% | 46.00% | 49.00% | 48.00% | 46.00% | 47.00% | 47.00% |

From the results shown above, the results for Cancer dataset using Standard and Three-Term BP gave almost similar correct classification percentage which fell within the range of 47% to 56%. For standard BP, the best performance in terms of correct classification percentage and the lowest error generated is obtained from TE4 of Test II with 50% correct classification and an error of 3.0016. It also gave smallest processing time with 40 seconds. For Three-Term BP, trial TE3 of Test III gave the best correct classification compared to other Three-Term experiments with 49% and processing time of 57 seconds. However, we can see that the error generated is not very satisfactory with increased error value of 8.5. In general, although these values are the best values for Three-Term BP, they are still not very satisfactory compared to the best result from standard BP. All three criteria which are the correct classification percentage, error generated and processing time fell behind those values generated from standard BP. The learning patterns for Cancer dataset in both algorithms are shown in Figure 4.3. These learning patterns are taken from the tests with the best results which is Test II for standard BP and Test III for Three-Term BP.



a) Standard BP

b) Three-Term BP

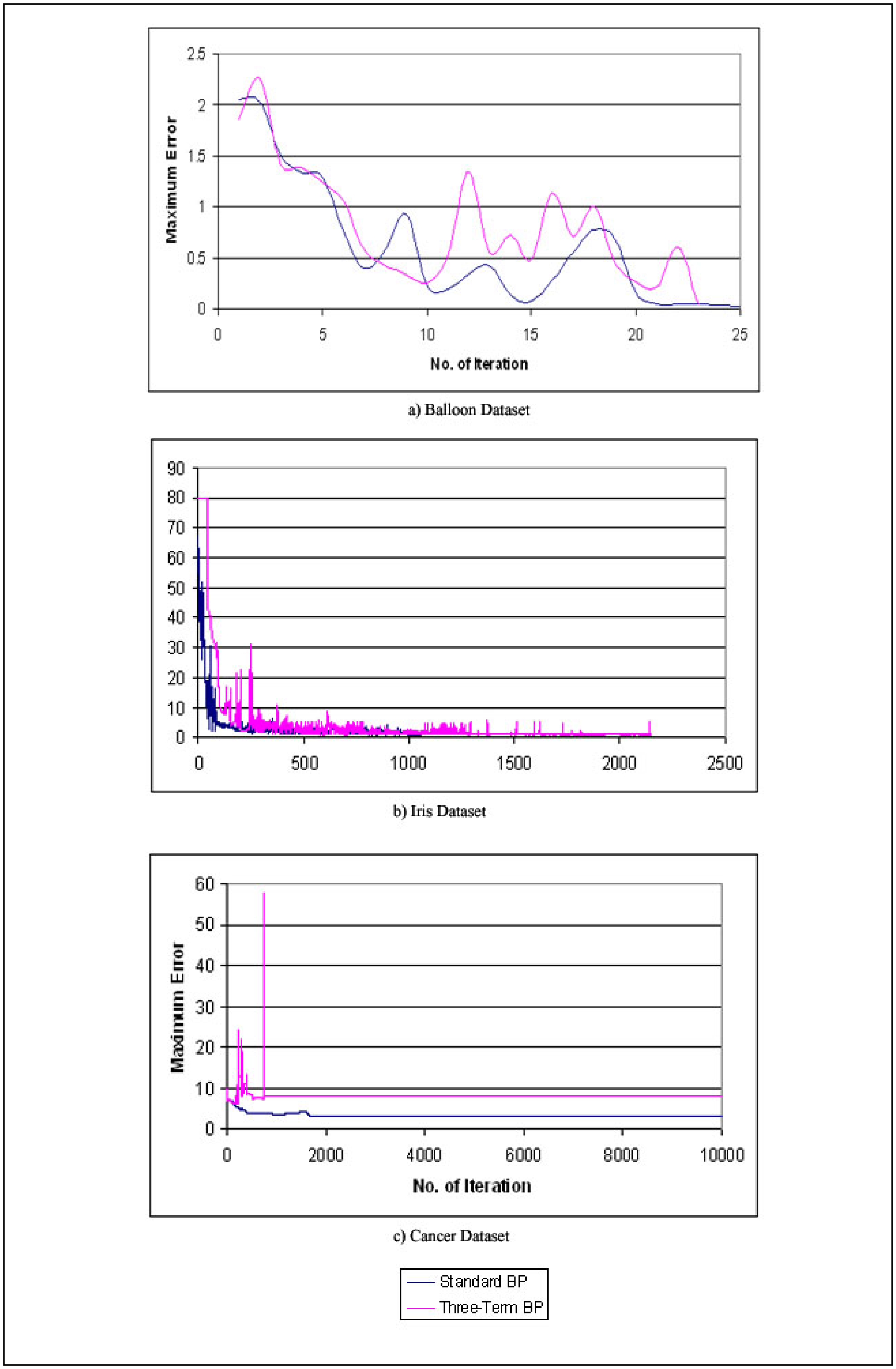
**Figure 4.3** Characteristic Convergence of Cancer Dataset

Figure 4.3 a) shows that the errors for all trials in Standard BP converged in a quite similar pattern except for TE4 which gave a lower range of convergence error. The errors generated converged closely match the maximum error function specified earlier. For Three-Term BP as shown in Figure 4.2 b), the errors were also generated in quite similar pattern except for TE6, where the error signals were generated steadily but at a much higher value of 152.30 at iteration 7000. From Figure 4.3 we can see that Standard BP gave a better range of error compared to Three-Term BP where the lowest error for standard BP was 3.0015 compared to 3.8692 generated by Three-Term BP. The detailed analysis and comparison based on these results will be discussed in section 4.3.

In analysis and comparison part, the results that will be analyzed are the best results in terms of correct classification percentage and processing time for both Standard and Three-Term BP. Therefore TE4 of Test II is chosen for experiment using Standard BP and TE3 of Test III is chosen for Three-Term BP.

## 4.3 Comparison of Standard BP and Three-Term BP Algorithm

This analysis is done to investigate the efficiency of Three-Term BP in solving classification problems. The comparisons are made between the results obtained from considerably good trials from both algorithms for each dataset. The comparisons between learning convergence of each dataset are illustrated in Figure 4.4. Table 4.7 shows the tabulated percentage of correct classification for each dataset while Figure 4.5 shows the corresponding graphs of the dataset.



a) Balloon Dataset

b) Iris Dataset

c) Cancer Dataset

**Figure 4.4** Convergence comparison between standard and Three-Term BP

**Table 4.7 :** Percentage of correct classification for standard and Three-Term BP

| Dataset | Standard BP | Three-Term BP |
|---|---|---|
| Balloon | 75% | 75% |
| Iris | 100% | 96% |
| Cancer | 50% | 48% |

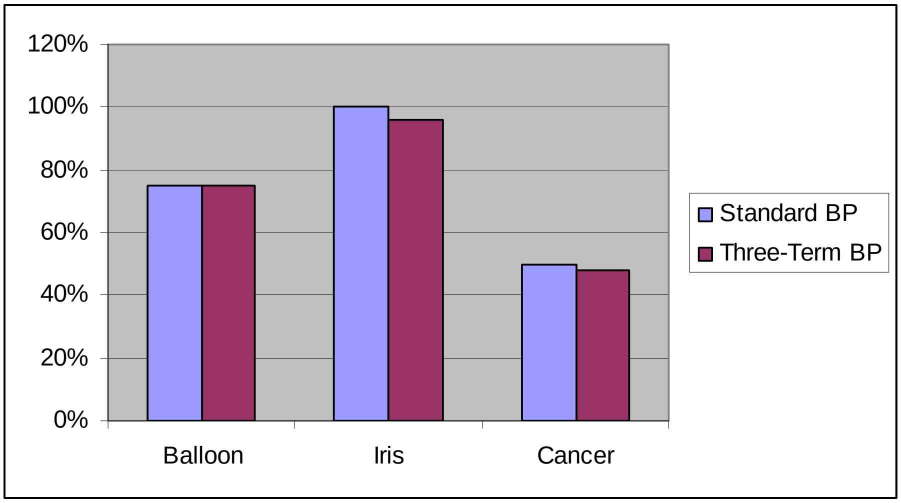


**Figure 4.5:** Comparison of correct classification between standard and Three-Term BP

For Balloon dataset, the learning pattern indicates that Three-Term BP converges faster than standard BP. This can be seen in Figure 4.4 where Three-Term BP converged within 23 iterations and standard BP converged within 25 iterations. For these best results, the value for $\alpha$, $\beta$ and $\gamma$ in Three-Term BP were all 0.9 and standard BP also had the same value for $\alpha$ and $\beta$ of 0.9. The results indicate that Three-Term BP generates less iteration, thus enhancing the learning speed. Both algorithms produced the same classification accuracy which is 75% but in terms of convergence accuracy, Three-Term BP once again produced a slightly better result with an error of 0.0426 instead of 0.0493 generated by standard BP.

For Iris dataset, the learning pattern indicates that Standard BP converges faster than standard BP. This can be seen in Figure 4.4 where Standard BP converged within 1064 iterations and Three-Term BP converged within 2149 iterations. The results indicate that error convergence produced by standard BP is lower than Standard BP. In terms of classification performance, standard BP also scored in giving a higher percentage of correct classification with a difference of 4% compared to Three-Term BP. Thus, standard BP also performs better than Three-Term BP for Iris dataset which is a medium scale data, in terms of correct classification percentage, processing time and error generated.

For Cancer dataset, the error produced by standard and Three-Term BP started off with almost the same value but for Three-Term BP, the error value increased tremendously during iteration 752 making the characteristic convergence of Three-Term BP ended with a high value or error of 8.5. Standard BP produced a higher percentage of correct classification compared to Three-Term BP with a difference of 2%. Thus, in classifying Cancer dataset which is a large scale data, standard BP performs better than Three-Term BP in terms of correct classification percentage, processing time and error generated.

## 4.4 Stability Analysis

The stability analysis is carried out to see whether the system, in this case the Three-term BP is stable under various scale of dataset presented to the network. The main task in doing this analysis is to find the eigen values, $\lambda_1$ and $\lambda_2$ for Cancer and Iris dataset respectively. To prove this statement, these derivations are made:

1. The conventional BP algorithm with two terms utilization has the following weight updating equation as shown in equation (4.1) below.

$$\Delta W(k) = -\alpha\nabla E(W(k)) + \beta\Delta W(k-1) \qquad (4.1)$$

where

E is the average of mean square function (MSE),

$\nabla E(W(k)) = \frac{1}{2k}\sum_{i=0}^{i=k}\sum_{s=0}^{s=i}(T(s) - O(s))^2$ , and

W is the network weight.

2. This study implements the third term, γ to increase the performance of standard BP. The term γ is proportional to $e(W(k)) = \frac{1}{k}\sum_{i=0}^{i=k}\sum_{s=0}^{s=i}(T(s) - O(s))$.

   The new weight adaptation of Three-Term BP is shown in equation (4.2) below.

$$\Delta W(k) = -\alpha\nabla E(W(k)) + \beta\Delta W(k-1) + \gamma e(W(k)) \qquad (4.2)$$

3. We want to analyze all local minima of the mean square error function that are only locally asymptotically stable points, equation (4.2) can be written as

$$W(k+1) = W(k) - \alpha\nabla E(W(k)) + \beta\Delta W(k-1) + \gamma e(W(k)) \qquad (4.3)$$

4. Local stability properties around an equilibrium point $(g_1, g_2)$ can be examined by using small signal analysis. (Zweiri *et al.*, 2003). Let $g_1 = W(k)$ and $\mathrm{g}_2 = W(k) - \mathrm{W(k-1)}$, then a state variable representation for equation (4.3) can be written as

$$g_1(k+1) = g_1(k) - \alpha\nabla E(g_1(k)) + \beta g_2(k) + \gamma e(g_1(k)) \qquad (4.4)$$

Note that, $\mathrm{g}_2 = \Delta \mathrm{W(k-1)}$, then (4.4) can be rewritten as

$$\begin{aligned} g_1(k+1) - g_1(k) &= -\alpha \nabla E(g_1(k)) + \beta g_2(k) + \gamma e(g_1(k)) \\ \Delta g_1(k+1) &= -\alpha \nabla E(g_1(k)) + \beta g_2(k) + \gamma e(g_1(k)) \\ \Delta W(k+1) &= -\alpha \nabla E(g_1(k)) + \beta g_2(k) + \gamma e(g_1(k)) \end{aligned}$$

5. From here we obtained another function

$$g_2(k+1) = -\alpha \nabla E(g_1(k)) + \beta g_2(k) + \gamma e(g_1(k)) \qquad (4.5)$$

Let $A = \nabla E(g_1(k))$ and $\mathrm{D} = g_2(k)$. Equation (4.4) and (4.5) can be represented into a linear equation, such as

$$\begin{bmatrix} g_1(k+1) \\ g_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 - \alpha A + \gamma D & \beta \\ -\alpha A + \gamma D & \beta \end{bmatrix} \begin{bmatrix} g_1(k) \\ g_2(k) \end{bmatrix} \qquad (4.6)$$

Equation (4.6) can be written in more compact form as

$$\phi(k+1) = \Theta \phi(k) \qquad (4.7)$$

6. It is well known that the discrete-time system in equation (4.7) is asymptotically stable if $\Theta$ has distinct eigen values, $\lambda_i$ that satisfy this condition (Zweiri *et al.*, 2003)

$$|\lambda_i| < 1$$

Let $\Theta$ be the matrix 2x2 that correspond to $\Theta = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ then the eigen values can be obtained from

$$\lambda_i = \frac{1}{2}(a + d \pm \sqrt{4bc + (a-d)^2}) \qquad (4.8)$$

By applying equation (4.6) into (4.8), we will get

$$\lambda_i = \frac{1}{2}(T \pm \sqrt{U+V}) \tag{4.9}$$

Where,

$$T = 1 - \alpha A + \gamma D + \beta$$
$$U = 4\beta(\gamma D - \alpha A)$$
$$V = 1 - \alpha A + \gamma D - \beta$$

The stability analysis is done on Iris dataset which covers small and medium scale data, and Cancer dataset which covers small and large scale data. The eigen values are calculated for system's generated errors and are shown in Table 4.8.

**Table 4.8 :** Eigen Values for Iris and Cancer Dataset

| | Iris Dataset | | Cancer Dataset | |
|---|---|---|---|---|
| | Small Scale | Medium Scale | Small Scale | Large Scale |
| $\lambda_1$ | 0.9998 | 55.8946 | 0.9705 | 4.3650 |
| $\lambda_2$ | 0.1015 | 25.6979 | 0.8243 | 0.1604 |

From the results in Table 4.8, we can see that both eigen values,$\lambda_1$ and $\lambda_2$ for small scale Iris and Cancer dataset had met the stability condition $|\lambda_i|<1$ whereas large scale Iris and Cancer dataset did not satisfy the condition. This indicates that the system is stable when represented with small scale data but it is not in a stable state when represented with medium or large data. This proves why Three-Term BP only outperforms standard BP in experiments involving small scale data but not when it involves medium and large scale data.

## 4.5 Discussion

In this section, two issues are needed to be addressed. First to investigate the efficiency of Three-Term BP with the additional γ factor and to compare the performance between standard and Three-Term BP. Table 4.9 is the summary of comparison between standard and Three-Term BP which has taken three analysis criteria into account which are the classification performance, the processing time and the error generated.

**Table 4.9 :** Summary of results analysis

| Analysis Criteria | Balloon | Iris | Cancer |
|---|---|---|---|
| **Classification Performance** | Three-Term BP better | Standard BP better | Standard BP better |
| **Processing Time** | Three-Term BP faster | Standard BP faster | Standard BP faster |
| **Error Convergence** | Three-Term BP lower | Standard BP lower | Standard BP lower |

Based on the analysis, Three-Term BP only gave a better performance in classifying Balloon dataset. Balloon dataset is used to represent small scale data. In Balloon dataset, Three-Term BP gave a higher correct classification performance, faster processing time and lower error convergence compared to standard BP. From Figure 4.4a) it is clearly shown that Three-Term BP excelled in generating less number of iteration compared to standard BP.

Meanwhile, in Iris and Cancer dataset, standard BP performed better in all three criteria compared to Three-Term BP. Iris dataset is used to represent medium scale data while Cancer dataset is used to represent large scale data. From Figure 4.4b) we could see that error convergence for Iris dataset in Three-Term BP is not stable after the first 100 iterations compared to standard BP which has a more stable error convergence pattern. Even though local minima is not obviously visible in this case, the instability of the error convergence for Three-Term BP has caused its poor performance compared to a more steady error convergence for standard BP.

The same results can also be seen from Cancer dataset. Standard BP scored better than Three-Term BP in classifying this dataset. Eventhough local minima is not visible in this experiment, error convergence for Three-Term BP is not stable within the first 500 iterations compared to standard BP which has a more stable error convergence pattern. Eventhough both algorithms did not converge to solutions for this dataset, the early iterations of standard BP had generated amore stable and lower value of error compared to Three-Term BP. Thus, standard BP had generated a faster processing time and correct classification percentage compared to Three-Term BP.

In this study, the implementation of $\gamma$ factor in Three-Term BP as the third term only enhances the performance in small scale data. This result is contradictory to the expected result in experiments using medium and large scale data which are Iris and Cancer dataset respectively. This conclusion is further proven by rescaling Iris and Cancer dataset into small scale data and the results of Three-Term BP using this newly scaled data are better than using standard BP (refer appendix A). This situation might had been caused by some reasons which are the instability of error convergence for Three-Term BP while using medium and large scale data and also higher error values for early iterations for these kind of dataset. Instability in error convergence might cause the convergence process to face difficulties in escaping local minima situation.

## 4.6 Summary

This chapter discusses about experimental results obtained from both standard and Three-Term BP. The implementation of $\gamma$ as the third term in Three-Term BP was tested on three types of dataset which are Balloon, Iris and Cancer dataset. The dataset used in this study is chosen based on their data size where Balloon dataset represents small scale data, Iris dataset represents medium and Cancer dataset represents large scale data. These data are represented is such ways to investigate the performance of Three-Term BP in various situations. The analysis part had analyzed results from both standard and Three-Term BP and comparisons are made based on these results.

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

This chapter discusses the work that has been done to complete this study, suggestions for future work and overall conclusion for this study. The main objective of this study is to investigate the efficiency of Three-Term BP and made a comparison of performance between standard and Three-Term BP.

## 5.1 Introduction

The performance evaluation of standard and Three Term BP are carried out based on its convergence rate and accurate classification of presented problem. The standard BP is trained using Balloon, Iris and Cancer dataset as mentioned in chapter 3. In this study, two programs have been developed which are standard BP and Three- Term BP.

Standard Backpropagation (BP) usually utilizes two terms parameters which are the Learning Rate and Momentum Factor for controlling the weight adjustment.

Although this algorithm has been proven to be very successful in training Neural Network to be used in many diverse applications, it has been observed that its convergence rate is extremely slow, especially for the networks with more than one hidden layer and trapped in local minima resulting from the saturation behaviour of the activation function. Eventhough many researches had been done to improve its performance, the modification usually involves complex calculations at each iteration. Zweiri et. al (2003) had proposed a $\gamma$ term of Three-Term BP with minimal addition to the computation complexity.

In this study, the implementations of both standard and Three-Term BP had been done and comparisons of results are made. The results and analysis are discussed in Chapter 4. The results obtained from using Three-Term BP in solving classification problems only outperform results from standard BP when it is represented with small scale data. Whereas by using medium and large scale data, the performance of Three-Term BP is poorer in both scale of data. This might have been caused by the instability of the network while processing medium and large data as discussed in section 4.4 of Chapter 4.

## 5.2 Summary of Work

In doing this project, the work had been done according to steps outlined in the project's methodology. The following are the summary of work in doing this project:

1. Determining training patterns from dataset
2. Implement standard BP
3. Derive the new weight adaptation by including the $\gamma$ term in the standard BP weight adaptation derivation
4. Implement $\gamma$ term in Three-Term BP

5. Conduct experiments using Balloon, Iris and Cancer dataset where each dataset represents small, medium and large scale data respectively
6. Analyze and compare the results obtained from both algorithms

## 5.3 Conclusion

Based on the results and analysis done in this study, it can be concluded that :

1. In this study, it is shown that Three-Term BP only outperforms standard BP when using small scale data, but not medium and large scale data that might be due to the instability in the algorithm when represented with medium and large scale data.

2. BP Three-Term BP is proven to be computationally less complex compared to some other algorithm modification techniques or methods that had been done before.

3. Eventhough in small scale data (Balloon dataset), both standard and Three-Term BP gave the same correct classification percentage, the performance is then evaluated in terms of less number of iterations and error generated.

4. Three-Term BP converges faster in small scale data compared to standard BP. This indicates that the proposed term $\gamma$ can be used to speed up the convergence and avoid local minima (only for small scale data).

## 5.4 Suggestions for Future Work

There are several suggestions and future works that can be done to further improve this project. The suggestions are:

1. Additional number of experiments be carried out to further validate the findings of this project.

2. Use optimization method to tune various network parameters such as initial weight, maximum error, number of hidden layer and hidden nodes etc. to the optimal value i.e using Genetic Algorithm (GA).

3. Use other activation function such as Arctangent and Logarithmic activation function.

4. More input data that represents all data scale can be considered and explored to find more effective results in training and testing the algorithm and to validate the findings of this project.

5. Other factors should be considered in studying the behaviour of Three-Term BP such as local minima problem and error function used.

# REFERENCES

Auda, G., Kamel, M. and Raafat, H. (1994). A New Neural Network Structure With Cooperative Modules. *IEEE International Conference on Neural Network Network, 1994.* 27 June-2 July, 1994. IEEE, 1301-1306.

Bilski, J. (2000). The Backpropagation Learning with Logarithmic Transfer Function. *Proceeding of The fifth Conference on Neural Network and Soft Computing*. 6-10 June, 2000. Poland : IEEE, 71-76.

Cao, F. and Zhang, Q. (2004). Neural Network Modeling and Parameters Optimization of Increased Explosive Electrical Discharge Grinding (IEEDG) Process For Large Area Polycrystalline Diamond. *Journal of Materials Processing Technology.* 149: 106-111.

Ishibuchi, H., Nakashima, T. and Murata, T, (1999). Performance Evaluation of Fuzzy Classifier Systems for Multidimensional Pattern Classification Problems. *IEEE Transactions on Systems, Man, and Cybernetics* 29 (5): 601-618.

Jia, Y. and Dali,Y. (1993). Analysis of The Misadjustment of BP Network and an Improved Algorithm. IEEE *International Symposium on Circuits and Systems 1993*. IEEE, 2592-2595.

Kim, G-H., Yoon, J-E., An, S-H., Cho, H-H. and Kang, K-I. (2004). Neural Network Model Incorporating A Genetic Algorithm in Estimating Construction Cost. *Building and Environment.* 39(11): 1333-1340.

Masters, T. *Practical neural network recipes in C++*. Boston:Academic Press. 1993.

Ng, S.C., Leung, S.H. and Luk, A. (1996). A Generalized Back-Propagation Algorithm For Faster Convergence. *IEEE International Conference on Neural Network Network, 1996.* IEEE, 409-413.

Okine, N.O.A. (1998). Analysis of Learning Rate and Momentum Term in Backpropagation Neural Network Algorithm Trained to Predict Pavement Performance. *Advances in Engineering Software* 30:291-302.

Roy, S. (1994). Factors Influencing The Choice of Learning Rate for a Backpropagation Neural Network. *1994 IEEE International Conference on Neural Networks*. 27 June-2 July, 1994. IEEE,503-507.

Rumelhart, D.E. and McClelland, J.L. (1986). Parallel Distributed Processing: Explorations in The Microstructure of Cognition. Vol 1. MIT press, Cambridge, MA.

Shafie, A.S. (2005). *Improved Two-Term Backpropagation Error Function With GA-Based Parameter Tuning For Classification Problem*. Universiti Teknologi Malaysia: Master Thesis.

Sexton, R.S. and Gupta, J.N.D. (2000). Comparative Evaluation of Genetic Algorithm and Backpropagation for Training Neural Networks. *Information Sciences* 129:45-59.

Yam, J.Y.F. and Chow, W.S. (2002). A weight initialization method for improving training speed in feedforward neural network. *Neurocomputing* 30:219-232.

Yu, C. C. and Liu B.D. (2002). A Backpropagation Algorithm With Adaptive Learning Rate and Momentum Coefficient. 2002. IEEE, 1218-1223.

Yu, X.H., Chen, G.A. and Cheng, S.X. (1995). Dynamic Learning Rate Optimization of the Backpropagation Algorithm. *IEEE Transactions on Neural Networks* May, 3. IEEE, 669-677.

Yu, X.H. and Chen, G.A. (1997). Efficient Backpropagation Learning Using Optimal Learning Rate and Momentum. *Neural Networks* 10(3):517-527.

Wang, X.G., Tang, Z., Tamura, H., Ishii, M. and Sun, W.D. (2003). An Improved Backpropagation Algorithm To Avoid The Local Minima Problem. *Neurocomputing* 56:455-460.

Zweiri, Y. H., Whidborne, J. F., Althoefer, K and Seneviratne, L.D. (2002). A new Three-Term backpropagation Algorithm With Convergence Analysis. *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*. May 2002. Washington, DC : IEEE, 3882-3887.

Zweiri, Y. H., Whidborne, J. F., Althoefer, K and Seneviratne, L.D. (2003). A Three-term Backpropagation Algorithm. *Neurocomputing* 50:305-318.