

STUDY OF COST FUNCTIONS IN THREE TERM BACKPROPAGATION FOR  
CLASSIFICATION PROBLEMS

PUSPADEVI A/P KUPPUSAMY

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

OCTOBER 2008

To my beloved grandmother, father, mother, brother and sister

## ACKNOWLEDGEMENTS

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my supervisor, Assoc. Prof. Dr. Siti Mariyam Shamsuddin for encouragement and guidance. Without her continued support and interest, this thesis would not have been the same as presented here.

My fellow postgraduate students should also be recognised for their support. My sincere appreciation also extends to my colleagues and friends who have provided assistance at various occasions. Their views and tips are useful indeed. I am grateful to all my family members too.

## ABSTRACT

Three Term Backpropagation was proposed in 2003 by Zweiri, and has outperformed standard Two Term Backpropagation. However, further studies on Three Term Backpropagation in 2007 indicated that the network only surpassed standard BP for small scale datasets (below 100 instances) but not for medium and large scale datasets (above 100 instances). It has also been observed that by using Mean Square Error (MSE) as a cost function in Three Term Backpropagation network, has some drawbacks such as incorrect saturation and tend to trap in local minima, resulting in slow convergence and poor performance. In this study, substantial experiments on implementing various cost functions on Three Term BP are executed to probe the effectiveness of this network. The performance is measured in terms of convergence time and accuracy. The costs functions involve in this study include Mean Square Error, Bernoulli function, Modified cost function and Improved cost function. These cost functions were introduced by previous researchers. The outcome indicates that MSE is not an ideal cost function to be used for Three Term BP. Besides that, the results have also illustrated that improve cost function's converges faster, while modified cost function produces high accuracy in classification

## ABSTRAK

Algoritma rambatan balik dengan tiga terma telah diperkenalkan oleh Zweiri pada 2003, dan telah berjaya mengatasi prestasi rangkaian rambatan balik tradisi iaitu rangkaian rambatan balik dua terma. Walaubagaimanapun, kajian yang telah dilaksanakan pada 2007 telah mendapati bahawa rangkaian rambatan balik tiga terma hanya dapat mengatasi prestasi rangkaian rambatan balik tradisi pada data yang bersaiz kecil (kurang daripada 100 data) dan bukan pada data yang bersaiz sederhana atau besar (besar dari 100 data). Oleh yang demikian, boleh dinyatakan bahawa fungsi ralat piawai iaitu Ralat Min Kuasa Dua mempunyai beberapa kelemahan seperti penumpuan yang amat perlahan, sering terperangkap pada minima setempat dan prestasi yang kurang baik. Kajian ini menjalankan eksperimen yang komprehensif terhadap beberapa fungsi ralat bagi rangkaian rambatan balik tiga terma bagi mencari keberkesanan fungsi kos tersebut. Prestasi rangkaian diukur dari aspek kepantasan kadar penumpuan dan ketepatan pengelasan. Fungsi kos yang terlibat adalah Ralat Min Kuasa Dua, fungsi ralat '*Bernoulli*', fungsi ralat yang telah 'diubahsuai', dan fungsi ralat pembaikan. Hasil kajian mempamerkan bahawa fungsi Ralat Min Kuasa Dua tidak begitu sesuai untuk algoritma rambatan balik tiga terma. Hasil kajian juga telah memperlihatkan bahawa fungsi ralat pembaikan memberi kadar penumpuan yang pantas manakala fungsi ralat yang 'diubahsuai' memberikan kadar pengelasan yang lebih tepat.

## TABLE OF CONTENT

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENTS	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENT	vii
	LIST OF TABLES	xii
	LIST OF FIGURES	xiv
	LIST OF SYMBOLS	xviii
	LIST OF ABBREVIATION	xix
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Introduction	1
	1.2 Problem Background	3
	1.3 Problem Statement	4
	1.4 Project Aim	5
	1.5 Objectives	6
	1.6 Project Scope	6
	1.7 Significance of The Project	7
	1.8 Organization of Report	8
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>9</b>

2.1	Introduction	9
2.1.1	The Neuron	10
2.1.2	Diagram of Neuron	11
2.1.3	Bias of a Neuron	12
2.1.4	Activation function	12
2.1.5	Network Architecture	13
2.2	Research Trends of Backpropagation (BP) Learning	14
2.3	Backpropagation (BP)	21
2.3.1	Two Term Backpropagation Algorithm	22
2.4	Two Term Backpropagation Parameters	24
2.4.1	Learning Rate	25
2.4.2	Momentum term	26
2.5	Three Term Backpropagation	26
2.5.1	Proportional term	28
2.6	Research Trends of Cost function in Backpropagation Network	28
2.7	Cost Function	39
2.7.1	Mean square error	39
2.7.2	Bernoulli Cost Function (BL)	41
2.7.3	Modified Cost Function	42
2.7.4	Improved Cost Function (IC)	46
2.8	Importance of Error Function	48
2.9	Comparison	49
2.10	Classification	50
<b>3</b>	<b>RESEARCH METHODOLOGY</b>	<b>53</b>
3.1	Introduction	53
3.2	Methodology	54
3.3	Defining Dataset Attributes	56
3.3.1	Balloons	56
3.3.2	Cancer	57
3.3.3	Diabetes	57
3.3.4	Pendigits	58

3.3.5	Summary of Datasets	58
3.4	Characterization of Network Architecture	59
3.4.1	Balloon Dataset	60
3.4.2	Cancer Dataset	60
3.4.3	Diabetes Dataset	61
3.4.4	Pendigits Dataset	62
3.5	Determine Network Parameters and Formulation of MSE Cost Function	64
3.6	Determine Network Parameters and Formulation of Bernoulli Cost Function	65
3.7	Determine Network Parameters and Formulation of Modified Cost Function	65
3.8	Determine Network Parameters and Formulation of Improved Cost Function	66
3.9	Training and Testing Three Term BP with Various Cost	67
3.10	Implementation of 'K+10' & K+100' Increment Rule	70
3.11	Summary	71
<b>4</b>	<b>EXPERIMENTAL RESULT</b>	72
4.1	Introduction	72
4.2	Experiments Setup	73
4.3	Implementation of various Cost Function	74
4.4	Implementation of T-Test	75
4.5	Analysis of Comparison Parameters	76
4.5.1	Epoch size	76
4.5.2	Network Error	77
4.5.3	Convergence Time	78
4.5.4	Accuracy	78
4.6	Experimental Result	79
4.6.1	Result of Three Term BP for Balloon Dataset	79
4.6.1.1	Result of Three Term BP with MSE Cost Function for Balloon Dataset	80
4.6.1.2	Result of Three Term BP with BL Cost	82

	Function for Balloon Dataset	
4.6.1.3	Result of Three Term BP with MM Cost	84
	Function for Balloon Dataset	
4.6.1.4	Result of Three Term BP with IC Cost	87
	Function for Balloon Dataset	
4.6.2	Result of Three Term BP for Cancer Dataset	89
4.6.2.1	Result of Three Term BP with MSE Cost	90
	Function for Cancer Dataset	
4.6.2.2	Result of Three Term BP with BL Cost	92
	Function for Cancer Dataset	
4.6.2.3	Result of Three Term BP with MM Cost	95
	Function for Cancer Dataset	
4.6.2.4	Result of Three Term BP with IC Cost	98
	Function for Cancer Dataset	
4.6.3	Result of Three Term BP for Diabetes Dataset	100
4.6.3.1	Result of Three Term BP with MSE Cost	101
	Function for Diabetes Dataset	
4.6.3.2	Result of Three Term BP with BL Cost	103
	Function for Diabetes Dataset	
4.6.3.3	Result of Three Term BP with MM Cost	105
	Function for Diabetes Dataset	
4.6.3.4	Result of Three Term BP with IC Cost	108
	Function for Diabetes Dataset	
4.6.4	Result of Three Term BP for Pendigits Dataset	110
4.6.4.1	Result of Three Term BP with MSE Cost	111
	Function for Pendigits Dataset	
4.6.4.2	Result of Three Term BP with BL Cost	113
	Function for Pendigits Dataset	
4.6.4.3	Result of Three Term BP with MM Cost	115
	Function for Pendigits Dataset	
4.6.4.4	Result of Three Term BP with IC Cost	118
	Function for Pendigits Dataset	
4.7	Performance Comparison of Three Term BP with	120

various Cost Function	
4.7.1 Balloon Datasets	121
4.7.1.1 Error	122
4.7.1.2 Convergence Time	123
4.7.1.3 Accuracy Percentage	124
4.7.2 Cancer Datasets	125
4.7.2.1 Error	126
4.7.2.2 Convergence Time	127
4.7.2.3 Accuracy Percentage	128
4.7.3 Diabetes Datasets	129
4.7.3.1 Error	130
4.7.3.2 Convergence Time	131
4.7.3.3 Accuracy Percentage	132
4.7.4 Pendigits Datasets	133
4.7.4.1 Error	134
4.7.4.2 Convergence Time	135
4.7.4.3 Accuracy Percentage	136
4.8 T-Test	137
4.8.1 T-test for Error Value	137
4.8.1.1 Balloon Data	137
4.8.1.2 Cancer Data	140
4.8.1.3 Diabetes Data	143
4.8.1.4 Pendigits Data	146
4.8.1.5 Overall T-test Result for error value	149
4.8.2 T-test for Convergence Time	150
4.8.2.1 Balloon Data	150
4.8.2.2 Cancer Data	153
4.8.2.3 Diabetes Data	155
4.8.2.4 Pendigits Data	158
4.8.2.5 Overall T-test Result for Convergence Time	160
4.8.3 T-test for Accuracy	161
4.8.3.1 Balloon Data	161

4.8.3.2	Cancer Data	161
4.8.3.3	Diabetes Data	164
4.8.3.4	Pendigits Data	167
4.8.3.5	Overall T-test Result for accuracy	169
4.9	Summary	170
<b>5</b>	<b>CONCLUSION AND FUTURE WORK</b>	173
5.1	Introduction	173
5.2	Contribution of the Study	174
5.3	Suggestion for future works	175
	<b>REFERENCE</b>	176

## LIST OF TABLES

<b>TABLE NO</b>	<b>TITLE</b>	<b>PAGE</b>
2.1	Previous Studies in Backpropagation Learning Enhancements	18
2.2	Effect of learning rate value	25
2.3	Effect of Momentum Term Value	26
2.4	Previous Error Function Improvements	33
3.1	Summary of Datasets	58
3.3	Summary of different Dataset's Network Architecture	64
4.1	Test result of Balloon dataset with MSE cost function	80
4.2	Test result of Balloon datasets with BL cost function	83
4.3	Test result of Balloon dataset with MM cost function	85
4.4	Test result of Balloon dataset with IC cost function	87
4.5	Test result of Cancer dataset with MSE cost function	91
4.6	Test result of Cancer datasets with BL cost function	93
4.7	Test result of Cancer dataset with MM cost function	96
4.8	Test result of Cancer dataset with IC cost function	98
4.9	Test result of Diabetes dataset with MSE cost function	101
4.10	Test result of Diabetes datasets with BL cost function	103
4.11	Test result of Diabetes dataset with MM cost function	105
4.12	Test result of Diabetes dataset with IC cost function	108
4.13	Test result of Pendigits dataset with MSE cost function	111
4.14	Test result of Pendigits datasets with BL cost function	114
4.15	Test result of Pendigits dataset with MM cost function	116
4.16	Test result of Pendigits dataset with IC cost function	118
4.17	Cost functions position in term of comparison parameters	121

4.18	Error produced by different cost function for Balloon Dataset	122
4.19	Convergence Time by different cost function for Balloon Dataset	123
4.20	Accuracy Percentage by different cost function for Balloon Dataset	124
4.21	Cost Functions Position in term of comparison parameters	125
4.22	Error produced by different cost function for Cancer Dataset	126
4.23	Convergence Time by different cost function for Cancer Dataset	127
4.24	Accuracy Percentage by different cost function for Cancer Dataset	128
4.25	Cost functions position in term of comparison parameters	129
4.26	Error produced by different cost function for Diabetes Dataset	130
4.27	Convergence Time by different cost function for Diabetes Dataset	131
4.28	Accuracy Percentage by different cost function for Diabetes Dataset	132
4.29	Cost functions position in term of comparison parameters	133
4.30	Error produced by different cost function for Pendigits Dataset	134
4.31	Convergence Time by different cost function for Pendigits Dataset	135
4.32	Accuracy Percentage by different cost function for Pendigits Dataset	136
4.33	Comparison mean and standard deviation Balloon data error	137
4.34	Comparison mean and standard deviation Cancer data error	140
4.35	Comparison mean and standard deviation Diabetes data error	143
4.36	Comparison mean and standard deviation Pendigits data error	146

4.37	Comparison mean and standard deviation Balloon data Convergence Time	150
4.38	Comparison mean and standard deviation Cancer data Convergence Time	153
4.39	Comparison mean and standard deviation Diabetes data Convergence Time	155
4.40	Comparison mean and standard deviation Pendigits data Convergence Time	158
4.41	Comparison mean and standard deviation Cancer accuracy.	161
4.42	Comparison mean and standard deviation Diabetes accuracy	164
4.43	Comparison mean and standard deviation Pendigits accuracy.	167

## LIST OF FIGURES

FIGURE NO	TITLE	PAGE
2.1	Diagram of Neuron	11
2.2	Different Activation Function	12
2.3	Neural Network Architecture	13
3.1	A general Framework of the proposed study	55
3.2	Balloon Datasets's Network Structure	60
3.3	Cancer Datasets's Network Structure	61
3.4	Diabetes Datasets's Network Structure	62
3.5	Pendigits Datasets's Network Structure	63
4.1	Error Convergence of Balloon Dataset with MSE Cost Function	81
4.2	Convergence Time of Balloon dataset with MSE cost function	81
4.3	Accuracy Percentage (%) of Balloon dataset with MSE cost function	82
4.4	Error Convergence of Balloon Dataset with BL Cost Function	83
4.5	Convergence Time of Balloon dataset with BL cost function	84
4.6	Accuracy Percentage (%) of Balloon dataset with BL cost function	84
4.7	Error Convergence of Balloon Dataset with MM cost function	85
4.8	Convergence Time of Balloon dataset with MM cost function	86

4.9	Accuracy Percentage (%) of Balloon dataset with MM cost function	86
4.10	Error Convergence of Balloon Dataset with IC cost function	88
4.11	Convergence Time of Balloon dataset with IC cost function	88
4.12	Accuracy Percentage (%) of Balloon dataset with IC cost function	89
4.13	Error Convergence of Cancer Dataset with MSE Cost Function	91
4.14	Convergence Time of Cancer dataset with MSE cost function	92
4.15	Accuracy Percentage (%) of Cancer dataset with MSE cost function	92
4.16	Error Convergence of Cancer Dataset with BL Cost Function	94
4.17	Convergence Time of Cancer dataset with BL cost function	94
4.18	Accuracy Percentage (%) of Cancer dataset with BL cost function	95
4.19	Error Convergence of Cancer Dataset with MM cost function	96
4.20	Convergence Time of Cancer dataset with MM cost unction	97
4.21	Accuracy Percentage (%) of Cancer dataset with MM cost function	97
4.22	Error Convergence of Cancer Dataset with IC cost function	99
4.23	Convergence Time of Cancer dataset with IC cost function	99
4.24	Accuracy Percentage (%) of Cancer dataset with IC cost function	100
4.25	Error Convergence of Diabetes Dataset with MSE Cost Function	102
4.26	Convergence Time of Diabetes dataset with MSE cost function	102
4.27	Accuracy Percentage (%) of Diabetes dataset with MSE cost function	103

4.28	Error Convergence of Diabetes Dataset with BL Cost Function	104
4.29	Convergence Time of Diabetes dataset with BL cost function	104
4.30	Accuracy Percentage (%) of Diabetes dataset with BL cost function	105
4.31	Error Convergence of Diabetes Dataset with MM cost function	106
4.32	Convergence Time of Diabetes dataset with MM cost function	107
4.33	Accuracy Percentage (%) of Diabetes dataset with MM cost function	107
4.34	Error Convergence of Diabetes Dataset with IC cost function	109
4.35	Convergence Time of Diabetes dataset with IC cost function	109
4.36	Accuracy Percentage (%) of Diabetes dataset with IC cost function	110
4.37	Error Convergence of Pendigits Dataset with MSE Cost Function	112
4.38	Convergence Time of Pendigits dataset with MSE cost function	112
4.39	Accuracy Percentage (%) of Pendigits dataset with MSE cost function	113
4.40	Error Convergence of Pendigits Dataset with BL Cost Function	114
4.41	Convergence Time of Pendigits dataset with BL cost function	114
4.42	Accuracy Percentage (%) of Pendigits dataset with BL cost function	115
4.43	Error Convergence of Pendigits Dataset with MM cost function	116
4.44	Convergence Time of Pendigits dataset with MM cost	117

	function	
4.45	Accuracy Percentage (%) of Pendigits dataset with MM cost function	117
4.46	Error Convergence of Pendigits Dataset with IC cost function	119
4.47	Convergence Time of Pendigits dataset with IC cost function	119
4.48	Accuracy Percentage (%) of Pendigits dataset with IC cost function	120
4.49	Error produced by different cost functions for Balloon dataset	122
4.50	Convergence Time by different cost functions for Balloon Dataset	123
4.51	Accuracy Percentage by different cost functions for Balloon dataset	124
4.52	Error produced by different cost functions for Cancer dataset	126
4.53	Convergence Time by different cost functions for Cancer Dataset	127
4.54	Accuracy Percentage by different cost functions for Cancer dataset	128
4.55	Error produced by different cost functions for Diabetes dataset	130
4.56	Convergence Time by different cost functions for Diabetes Dataset	131
4.57	Accuracy Percentage by different cost functions for Diabetes dataset	132
4.58	Error produced by different cost functions for Pendigits dataset	134
4.59	Convergence Time by different cost functions for Pendigits Dataset	135
4.60	Accuracy Percentage by different cost functions for Pendigits dataset	136

## LIST OF SYMBOLS

$W_{ij}$	Weight connected between node $i$ and $j$
$\theta_i$	Bias of node $I$
$a_i$	Output 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
$W_j$	Weight of neuron $j$ ,
$X_j$	Input of neuron $j$ .
$\delta_j$	Error at node $j$
$T_j$	Target output value at node $j$
$O_j$	Actual output of the network at node $j$ .
$\Delta W(k-1)$	Previous weight change.
$e(W(k))$	Difference between the output and the target at each iteration.
$y_{pj}$	Output of the $j^{\text{th}}$ neuron in the hidden layer
H	Number of neurons in the hidden layer

**LIST OF ABBREVIATION**

ANN	- Artificial Neural Network
BP	- Backpropagation
NN	- Neural Network
MSE	- Mean Squared Error
BL	- Bernoulli Error function of Chow et al. (1994)
MM	- Modified Error function of Shamsuddin <i>et al.</i> (2001)
IC	- Improved Error function of Zhang et al. (2007)

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 Introduction**

Artificial Neural Network (ANN) is a model of reasoning based on the human brain. It consists of a number of simple highly interconnected processors known as neurons, which are analogous to the biological neural cells of the brain. These neurons are connected by a large number of weighted links (Ibrahim dan Al-shams, 1997). Learning is a fundamental and essential characteristic of ANN. It is capable of learning through the network experiences to improve their performance. When ANN is exposed to a sufficient number of samples, it can generalise well to other data that they have not yet encountered (Negnevitsky, 2004).

Generally, ANN can be trained using backpropagation (BP) developed by Rumelhart, Hinton and Williams in 1986. Studies have shown that BP has been proven to be very successful in many diverse applications (Hauger, 2003). ANN training usually updates the weights iteratively using the negative gradient of a Mean Squared Error (MSE) function, multiplied by the slope of a sigmoid activation function. MSE is

referred to the difference between desired and actual output values. The error signal is then backpropagated to the lower layers (Zweiri *et al.*, 2003).

Then an activation function will transform the input into its own value range accordingly. There are many activation functions available such as step, sign, linear and sigmoid. The most popular activation function is sigmoid function. The sigmoid function transforms the input, which can have any value between plus and minus infinity into reasonable value in the range between 0 and 1 (Hauger, 2003). BP network's neuron uses this function to produce a standard outputs.

The outputs will be compared with the targeted output and it will backpropagates to adjust the weights. There are two parameters used in controlling weight adjustment of standard backpropagation. These are learning rate (LR) and momentum factor (MF). Recently, a new term known as proportional factor is added to the formulation to speed-up the weight adjusting process by Zweiri *et al.* (2003). This formulation is known as three term BP.

The derivative of the cost function is one of the factors in the equation of weight adjustment. This is important to determine the success of the application, to train the network with an error function that resembles the objective of the problem at hand (Falas and Stafilopatis, 1999). In most practical applications, MSE is the most commonly used cost function in BP network.

## 1.2 Problem Background

Three Term Backpropagation was proposed by Zweiri *et al.* (2003). It involves Proportional Factor (PF) besides Learning Rate (LR) and Momentum Factor (MF) for error adjustment in the algorithm. According to Zweiri *et al.*, it has outperformed standard Two Term Backpropagation with less complexity, low computational cost and easy tuning to suit a particular application. It is noted that the new algorithm archives efficiency while maintaining a similar computational complexity to the conventional BP algorithm. This is in contrast to other alternative BP algorithms, which requires complex and costly calculations at each iteration to archive faster rates on convergence. Moreover in contrast to the proposed algorithm, most standard acceleration techniques must be tuned to fit particular application. This new term also can be viewed as being analogous to the common three term proportional integral derivative (PID) algorithm used in feedback control. PID controller is a generic control loop feedback mechanism widely used in industrial control systems. However, further studies on Three Term Backpropagation by Shamsuddin, Darus and Saman (2007) indicated that the network only outperformed standard BP for small scale datasets (less than 100 instances) but not for medium and large scale datasets (more than 100 instances).

Meanwhile, researches have identified proper cost function is being an important factor to improve the performance of Two Term BP in terms of convergence speed (Humpert, 1994; Neelakanta, 1996; Dhiantravan, 1996; Oh and Lee, 1999; Taji *et al.*, 1999; Shamsuddin *et al.*, 2001; Jiang *et al.*, 2003; Wang *et al.*, 2004; Lv and Yi, 2005; Choi *et al.*, 2005; Otair and Salameh, 2006; Zhang, 2007), in terms of higher accuracy (Telfer and Szu, 1994; Rimer and Martinez, 2006) and to overcome the problems of getting stuck into local minima (Telfer and Szu, 1994; Oh and Lee, 1999; Jiang *et al.*, 2003; Wang *et al.*, 2004; Bi *et al.*, 2004; Zhang *et al.*, 2007).

It has been observed that, Mean Square Error cost function employed has drawbacks such as incorrect saturation and tend to trap in local minima, resulting in slow convergence and poor performance (Rimer and Martinez, 2006). Besides that, it gives more emphasis on reducing the larger errors as compared to smaller errors due to the squaring that takes place. Also due to the summation of the errors for all input patterns, if a class is not well presented and happens to have small errors, it may be completely ignored by the learning algorithm (Falas and Stafylopatis, 1999).

The need to improve Three Term BP is foreseen, where if a better cost function is applied in the Three Term it could perform better. This is due to the successfulness of researches that claims Two Term BP performed better with their novel cost functions instead of MSE (Wang *et al.*, 2004; Lv and Yi, 2005; Choi *et al.*, 2005; Otair and Salameh, 2006; Zhang, 2007; Rimer and Martinez, 2006)

### **1.3 Problem Statement**

In Three Term Backpropagation, MSE is employed as its cost function. It has been observed that, MSE cost function employed has drawbacks resulting in slow convergence and poor performance. Falas and Stafylopatis (1999) studied on impact of cost function in neural network classifier. Their result showed that a cost function other than the usual mean square gives a better performance, both in terms of the number of epochs needed for training, as well as the obtained generalization ability of the trained network.

Thus, in this study Mean Square Error, Bernoulli Cost Function of Chow *et al.* (1994), Modified Cost Function of Shamsuddin *et al.* (2001) and Improved Cost Function of Zhang *et al.* (2007) are exploited in Three Term BP to probe the convergence time and accuracy. These cost function were selected because of the simplicity of the formulation that helps to incorporate easily into the Three Term BP. Besides that those cost functions has been tested on various classification problems and proven to be performed well in the Two Term BP. The classification domain was selected or this study since BP is successful in this domain.

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

Three Term BP would yield faster convergence speed and better classification accuracy with cost functions other then MSE.

#### **1.4 Project Aim**

The aim of this project is to study the effectiveness of exploiting novel cost functions introduced by researches in past years to improve the Two Term BP to be applied in Three Term BP to increase the convergence speed and to produce high accuracy.

## 1.5 Objectives

In order to accomplish the hypothesis of the study, few objectives have been identified.

1. To study the cost functions of previous researches especially Mean Square Error (MSE) cost function, Bernoulli (BL) cost function, Modified (MM) cost function and Improved (IC) cost function.
2. To conduct experimental comparisons of MSE cost function, BL cost function, MM cost function and IC cost function in Three Term BP for classification problems.

## 1.6 Project Scope

The scopes of this project are defined as follows:

- I. Datasets that will be employed are Balloon with 16 instances, Cancer with 500 instances, Diabetes with 768 instances and Pendigits with 1000 instances.
- II. Three Term BP with the following cost functions are used in this study:
  - a. Three Term BP with MSE cost function
  - b. Three Term BP with BL cost function of Chow *et al.* (1994)
  - c. Three Term BP with MM cost function of Shamsuddin *et al.* (2001)
  - d. Three Term BP with IC cost function of Zhang *et al.* (2007)

- III. Develop Three Term BP with MSE cost function, Three Term BP with BL cost function, Three Term BP with MM cost function and Three Term BP with IC cost function using Microsoft Visual C++ 6.0.
- IV. Experiments will be conducted for Three Term BP only. Two Term BP will not be tested.
- V. The network architecture is three layers consist of one input layer, one hidden layer and one output layer to standardize the comparison criteria.
- VI. Experimental setting with 'K+10 or K+100 Increment Rule' for the number of epochs.

### **1.7 Significance of the Project**

This project studied the performance of Three Term BP with MSE cost function, Three Term BP with BL cost function, Three Term BP with MM cost function and Three Term BP with IC cost function. The outcomes of this study will contribute to verify the performance of those cost functions for Three Term BP. Furthermore, this study will spark future research in Three Term BP algorithm.

## **1.8 Organization of Report**

This report consists of five chapters. The chapter 1 presents introduction to project, problem background, objective, scope and significant of this study. Chapter 2 reviews the ANN, Two Term BP, Three Term BP, Research trends of BP Learning, Research trends of cost function in BP Network, MSE cost function, BL cost function, MM cost function and IC cost function and also importance of cost functions. Chapter 3 discusses on the methodology used in this study. It also explains details of datasets being used and network architectures. Chapter 4 is the experimental result study. Chapter 5 is the conclusion and suggestion for future work.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

An Artificial Neural Network (ANN) can be defined as information processing paradigm. It is inspired by information processing system of human's biological nervous systems. It is composed of a large number of highly interconnected processing elements known as neurons to solve specific problems. An important criterion of neural network is the capability of learning. Synaptic connections that exist between the neurons of the nervous system are adjusted in order to learn where else, ANN also learns by adjusting the weights between neurons.

So, ANN can be defined as a machine learning approach inspired by the way in which the brain performs a particular learning task. ANN is consists of a number of artificial neurons which receives a number of inputs. A function called activation function is applied to these inputs that results in activation level of neuron (output value of the neuron). Knowledge about the learning task is given in the form of examples called training examples.

Neural network is defined by architecture, neuron model and the learning algorithm. Architecture refers to a set of neurons and links connecting neurons. Each link has a weight. Neuron model refers to information processing unit of the neural network. Besides that, a learning algorithm is used to train the NN by modifying the weights in order to model a particular learning task correctly on the training examples. The aim of neural network is to obtain a Neural Network that generalizes well that behaves correctly on new instances of the learning task.

### 2.1.1 The Neuron

Basic information processing unit of ANN is called neuron. It consists of a set of links, weights, adding function and activation function. Firstly, a set of links describe neuron inputs with weights.

$$u = \sum_{j=1}^m W_j X_j \quad (2.1)$$

where,

$W_j$  is Weight of neuron  $j$ ,

$X_j$  is Input of neuron  $j$ .

Then there is an adder function, which also known as linear combiner for computing the weighted sum of the inputs. In addition it has an activation function for limiting the amplitude of the neuron output.

$$y = \varphi(u + b) \quad (2.2)$$

where,

$\varphi$  is activation function,

$u$  is weighted sum,

$b$  is bias.

### 2.1.2 Diagram of Neuron

As can be seen from Figure 2.1, artificial neurons basically consist of inputs that are multiplied by weights (strength of the respective signals), and then computed by a mathematical function that determines the activation of the neuron. Another function (which may be the identity) computes the output of the artificial neuron. Sometimes it is in dependence of a certain threshold. ANN combines artificial neurons in order to process information.

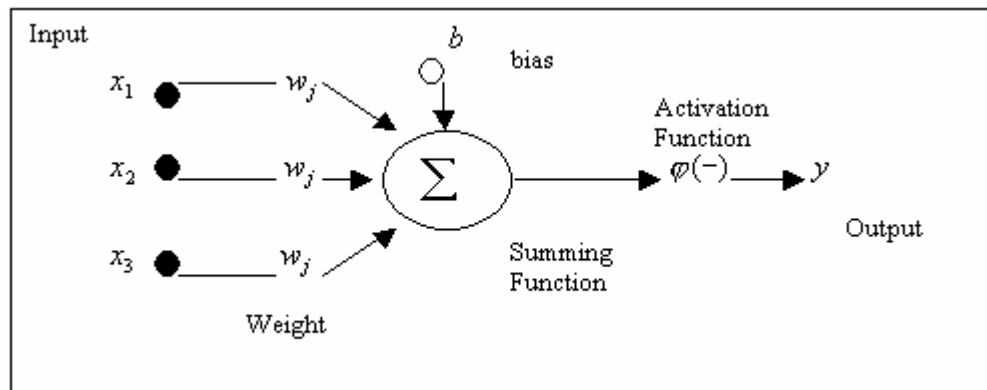


Figure 2.1 Diagram of Neuron

### 2.1.3 Bias of a Neuron

The bias  $b$  has effect to the weighted sum  $u$ , where  $v = u + b$ . It is an external parameter of the neuron and is modeled by adding an extra input.  $v$  is called induced field of the neuron.

### 2.1.4 Activation function

The choice of activation function determines the neuron model (Negnevitsky, 2004). There are several activation functions available. For example step, sign, sigmoid, linear and so on. Some examples are as follows:

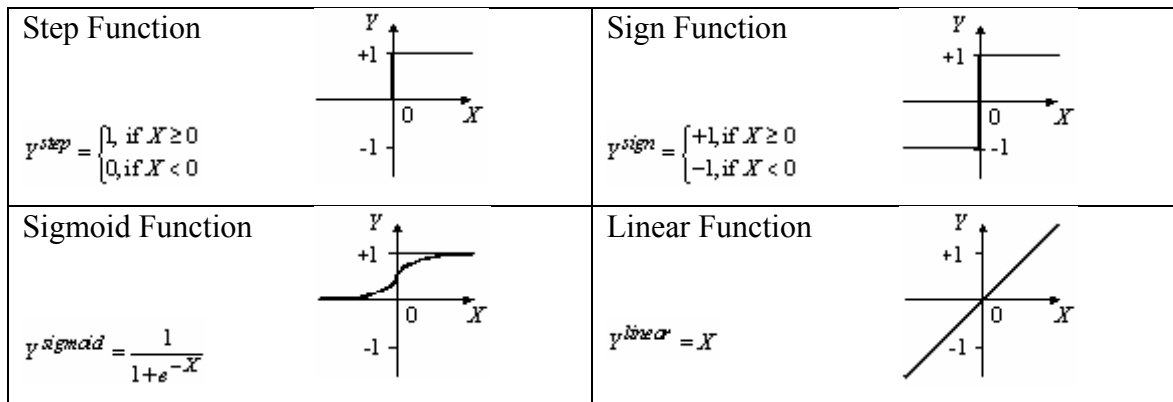


Figure 2.2 Different Activation Function

### 2.1.5 Network Architecture

Three different classes of network architectures are single-layer feed-forward, multi-layer feed-forward and recurrent. The architecture of a neural network is linked with the learning algorithm used to train the network (Kaushik, 2007).

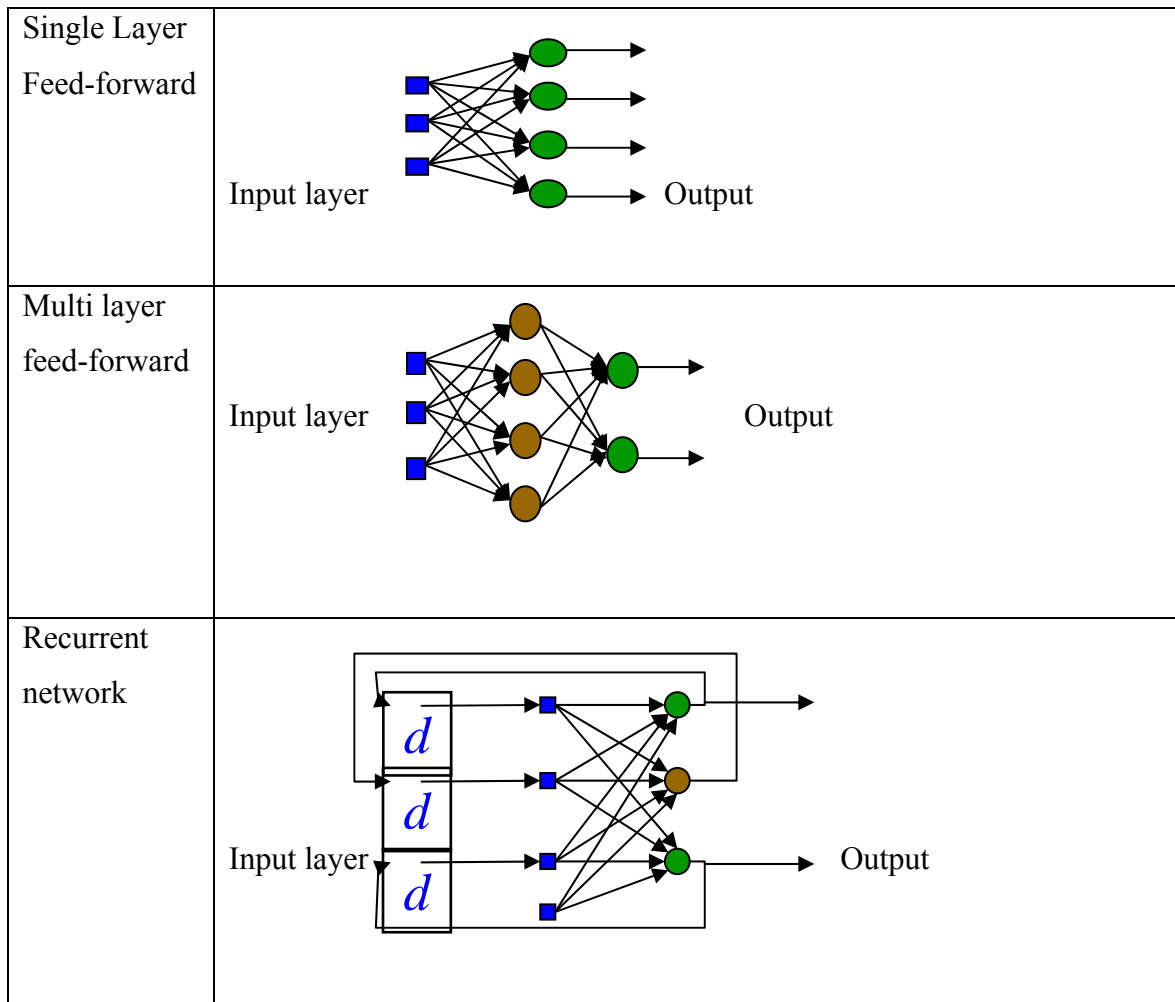


Figure 2.3 Neural Network Architecture

## 2.2 Research Trends of BP Learning

The BP algorithm is popular and used for many applications. BP is a method for calculating the first derivatives, or gradient, of the cost function required by some optimization methods. It is certainly not the only method for estimating the gradient. However, it is the most efficient (Edward, 2004). The major limitations of this algorithm are the existence of temporary, local minima resulting from the saturation behavior of the activation function, and the slow rates of convergence (Zweiri *et al.*, 2003). From 1988 to date, a number of researchers have done many modifications to standard BP algorithm which also known as Two Term BP to overcome these problems. A number of approaches have been implemented to improve the convergence speed. There are basically on selection of dynamic variation of learning rate and momentum, selection of better activation function and better cost function.

In 1993, Yam and Chow suggested a new learning algorithm for feed forward neural networks. This algorithm is based on modifications to the BP algorithm. The learning rate and momentum coefficient are adapted according to the coefficient of correlation between the downhill gradient and the previous weight update. It is said to be faster than the standard BP algorithm and has the additional advantage of being less affected by poor initial weight setting.

Modification on solving the weight matrix for the output layer using equations theory and least squares techniques was done by Verma and Mulawka (1994). This algorithm is presented for a certain and fast training process. Meanwhile Drago (1995) investigated the effectiveness of proposing an adaptive momentum for BP. In his finding, Adaptive Momentum Back Propagation (AMBP) capable of fast minimum search and the solutions are promising. Such effectiveness is achieved by making adaptive, in a

very simple and satisfactory way, both at the learning rate and the momentum term. The execution was generated by controlling and correcting the possible cost function to increase and to move in opposite direction of negative gradient.

Bossan *et al.* (1996) introduced a method to reduce the tendency of the classifiers to spend most of the time to achieve unnecessarily low mean square errors in very populated regions of the pattern space. While almost ignoring patterns in sparse regions of it until a large number of training steps occurs.

Chen *et al.* (1997) suggested a novel method to improve the performance of BP by randomizing the cost function. A randomized BP algorithm is thus obtained by choosing a sequence of weighting vectors over the learning phase. Since a given member cost function corresponds to a distinct error landscape, this randomization in effect shakes the error of delta-bar delta algorithm on a non-convex surface, thus increasing the probability of escaping from local minima.

Fukuoka *et al.* (1998) has proposed a modified BP method by keeping the sigmoid derivative relatively large while some of the error signals are large. For this purpose, each connecting weight in a network is multiplied by a factor in the range of (0,1), at a constant interval during a learning process.

A new generalized BP algorithm, which can effectively speed up the convergence rate and reduce the chance of being trapped in local minima was introduced by Ng *et al.* (1999). The new BP algorithm is to change the derivative of the activation function so as to magnify the backward propagated error signal, thus the convergence rate can be accelerated and the local minimum can be escaped. Ng *et al.* also investigated the convergence of the generalized BP algorithm with constant learning

rate. The weight sequences in generalized BP algorithm can be approximated by a certain ordinary differential equation.

Wen *et al.* (2000) investigated a new method in BP algorithm to avoid local minimum and proposed a mean of adding gradually training data and hidden units. It is actually an adaptive BP algorithm that can update learning rate and inertia factor automatically based on dynamical training error rate of change. He stated that the method can assure the convergence rate and there is no strict demand on selection of initial value.

Work by Abid S. *et al.* (2001) have proposed a new approach that minimizes a modified form of the criterion used in the standard BP algorithm. The criterion is based on the sum of the linear and the nonlinear quadratic errors of the output neuron. The quadratic linear error signal is appropriately weighted and the convergence of the new algorithm requires less iteration than the Standard BP.

Meanwhile, Yu C.C. and Liu B.D. (2003) proposed an acceleration technique called BPALM (Backpropagation with adaptive learning rate and momentum factor). They employed an adaptive learning rate and momentum factor based on conventional BP algorithm where the learning rate and momentum factor are adjusted accordingly to reduce the training time.

Zweiri *et al.* (2003) proposed proportional factor (PF) term in addition to the learning rate and momentum factor terms in BP network. The results indicate that this new algorithm offers higher convergence speeds compared to standard BP algorithm. Three term BP is able to escape from local minima, more robust, applicable to any

network with different activation functions and does not require complex and costly calculations at each iteration.

Wang *et al.* (2004) proposed an improved BP where each training pattern has its own activation functions of neurons in hidden layer to avoid local minima. The activation functions are adjusted by the adaptation of gain parameters during the learning process.

Pernía-Espinoza (2005) has introduced a TAO-robust BP learning algorithm. They combine the benefits of the non-linear regression model  $\tau$ -estimates (Introduced by Tabatabai and Argyros, 1993) with the BP algorithm to produce the TAO-robust learning algorithm in dealing with outliers. TAO is referred to the name given for the new algorithm developed.

Kathirvalavakumar and Thangavel (2006) studied new learning procedure for training single hidden layer feed forward network. This procedure trains the output layer and the hidden layer separately by introducing a new optimization criterion for the hidden layer. In their study, the existing methods are executed to find fictitious teacher signal for the output of each hidden neuron, modified standard BP algorithm and the new optimization criterion are combined to train the feed forward neural networks.

Recently in 2007, Wang *et al.* has proposed the individual inference adjusting learning rate technique (IIALR) to enhance the learning performance of the BP network. The mechanism of the weight adjustment in the IIALR is an individual learning rate for each weight. Furthermore, Sammy Siu (2007) also worked on improving the BP algorithm using evolutionary strategy. This is based on the theory of introducing the

factors of the chromosome and gene mutation rates that can enhance the flexibility of the mutation.

Another study by Guijarro and Fontenla (2007) revealed the importance of upgrading the learning speed of several BP algorithms, while preserving good optimization accuracy.

From previous literatures, we can conclude that the importance and the significant of enhancing BP learning are still stipulated, relevant and vital despite its existence for almost four decades. Table 2.1 summarizes the substantial studies that have been done on improving the BP learning.

Table 2.1: Previous Studies in Backpropagation Learning Enhancements

<b>Year</b>	<b>Researcher</b>	<b>Method</b>
1993	Yam and Chow	The learning rate and momentum coefficient are adapted according to the coefficient of correlation between the downhill gradient and the previous weight update.
1994	Verma and Mulawka	The modification is based on the solving of weight matrix for the output layer using theory of equations and least squares techniques.
1995	Drago <i>et al.</i>	An Adaptive Momentum Back Propagation (AMBP) for fast minimum search is proposed, which is said to achieves very satisfying performance.
1996	Bossan <i>et al.</i>	This method aims to reduce the tendency of classifiers to spend most of their time trying to

		achieve unnecessarily low mean square errors in very populated regions of the pattern space, while almost ignoring patterns in sparse regions of it until a large number of training steps occurs.
1997	Chen <i>et al.</i>	A randomized BP algorithm is proposed. It is obtained by choosing a sequence of weighting vectors over the learning phase.
1998	Fukuoka <i>et al.</i>	Each connecting weight in a network is multiplied by a factor in the range of (0,1) at a constant interval during a learning process. It is to keep the sigmoid derivative relatively large while some of the error signals are large.
1999	Ng and Leung	A new generalized back-propagation algorithm to change the derivative of the activation function so as to magnify the backward propagated error signal, thus the convergence rate can be accelerated and the local minimum can be escaped.
2000	Wen <i>et al.</i>	An adaptive backpropagation algorithm which can update learning rate and inertia actor automatically based on dynamical training error rate of change.
2001	Abid <i>et al.</i>	This approach minimizes a modified form of the criterion used in the standard backpropagation algorithm. This criterion is based on the sum of the linear and the nonlinear quadratic errors of the output neuron.

2002	Yu and Liu	BPALM (Backpropagation with adaptive learning rate and Momentum term) - adaptive learning rate and momentum term where the learning rate and momentum factor are adjusted at each iteration to reduce the training time.
2003	Zweiri <i>et al.</i>	An addition of an extra term, a proportional factor, besides learning rate and momentum factor of backpropagation is proposed in order to speed-up the weight adjusting process.
2004	Wang <i>et al.</i>	Improved BP where each training pattern has its own activation function of neurons in hidden layer to avoid local minima.
2005	Pernía-Espinoza <i>et al.</i>	The benefit of the non-linear regression model $\tau$ -estimates (introduced by Tabatabai and Argyros, 1993) is combined with the backpropagation algorithm to produce the TAO-robust learning algorithm.
2006	Kathirvalavakumar and Thangavel	New learning procedure for training single hidden layer feedforward network is proposed where this procedure trains the output layer and the hidden layer separately. A new optimization criterion for the hidden layer is proposed.
2007	Wang <i>et al.</i>	An Individual Inference Adjusting Learning Rate technique (IIALR) to enhance the learning performance of the Backpropagation Neural Network

2007	Sammy Siu <i>et al.</i>	Improving the Back-Propagation Algorithm Using Evolutionary Strategy
2007	Guijarro and Fontenla	A learning algorithm that applies linear-least-squares is presented.

### 2.3 Backpropagation

Backpropagation is a supervised learning algorithm used by multilayered neural network for learning purposes. It has been introduced by Rumelhart, Hinton and Williams in 1986. Now, the BP algorithm is the most widely used supervised learning technique to train feedforward ANN (Boquera *et al.*, 2007). The Backpropagation algorithm searches for weight values that minimize the total error of the network over the set of training examples.

BP consists of the repeated phase of two passes. There are forward pass and backward pass. During forward pass, the network is activated on one example and the error of the output layer is computed. Then during backward pass, the network error is used for updating the weights. From the output layer, the error is propagated backwards through the network, layer by layer. This is done by recursively computing the local gradient of each neuron. BP adjusts the weights of the NN in order to minimize the network total mean squared error (Kaushik, 2007).

### 2.3.1 Two Term Backpropagation Algorithm

Backpropagation refers to a simple method for calculating the gradient of the network, that is the first derivatives of the weights in the network. The primary objective of network training is to estimate an appropriate set of network weights based upon a training dataset. Simply stated, backpropagation is a method for calculating the first derivative of the cost function with respect to each network weight. It is certainly not the only method for estimating the gradient. However, it is the most efficient (Jones, 2004). This calculation is shown below (Narayan, 1997).

As mention above, in a backpropagation neural network, the learning algorithm has two phases. First, a training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output pattern is generated by the output layer.

$$\begin{aligned} output &= f(net_i) \\ net_i &= \sum_i W_{ij} O_j + \theta_i \end{aligned} \quad (2.3)$$

where,

$W_{ij}$  is the weight connected between node  $i$  and  $j$ ,

$\theta_i$  is the bias of node  $i$ ,

$O_j$  is the output of node  $j$ .

Shown below is the activation function. In this in this study sigmoid function is used. It is widely used in backpropagation:

$$f(net_i) = \frac{1}{(1 + e^{-net_i})} \quad (2.4)$$

If this pattern of output is different from the desired output, an error is calculated. It is then propagated backwards through the network from the output layer to the input layer. The weights will be modified as the error is propagated. The representation is as below:

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij} \quad (2.5)$$

where,

$W_{ij}(t)$  is the weight from node  $i$  to node  $j$  at time  $t$ ,

$\Delta W_{ij}$  is the weight adjustment.

The weight adjustment is computed by using the delta rule as shown below:

$$\Delta W_{ij} = \eta \delta_j x_i \quad (2.6)$$

where,

$\eta$  is learning rate ( $0 < \eta < 1$ ),

$\delta_j$  is error at node  $j$

The error  $\delta_j$  is as shown below:

$$\delta_j = T_j - O_j \quad (2.7)$$

where,

$T_j$  is the target output value at node  $j$ ,

$O_j$  is the actual output of the network at node  $j$ .

A generalized Delta rule with momentum term in the delta rule, which takes into account previous updates, is shown below:

$$\Delta W(k) = \alpha(-\nabla E(W(k))) + \beta\Delta W(k-1) \quad (2.8)$$

where,

$\alpha$  is Learning Rate,

$\nabla E(W(k))$  is gradient of  $E$  at  $W = W(k)$ ,

with  $k = 1; 2; 3; \dots; N$ , being the iteration number,

$\beta$  Momentum term,

$\Delta W(k-1)$  Previous weight change.

This learning process is repeated iteratively in other word continues until stopping condition is satisfied or convergence is achieved.

## 2.4 Two Term BP Parameters

Learning parameter that involved in standard Two Term BP 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 (Nii, 1999). These parameters will affect the convergence of the neural network.

### 2.4.1 Learning Rate

The learning rate in the BP learning rule is an important factor to determine the size of the weight adjustment made at each iteration which later will affect the convergence rate. From studies, it is discovered that small values for the learning rate will lead to small weight changes and large values will lead to large changes. The best choice of learning rate depends on the problem itself. If the chosen value is very small the descent will progress in small steps significantly increasing the total time to converge. On the other hand if the chosen value is too large for the error surface the search path will oscillate about the ideal path (Yu and Liu 2002). From the survey, we can conclude that, if learning rate is small then the algorithm learns the weights very slowly, while if it is large then the large changes of the weights may cause an unstable behavior with oscillations of the weight values. If the learning rate is 0, the network will not learn (Nii, 1999). Based on reports by other researchers, a reasonable range for learning rate is between 0.25 and 0.35 (Chow *et al.*, 1995).

Table 2.2 Effect of learning rate value

<b>Value</b>	<b>Effect</b>
Too Large	Oscillation and overshoot the solution vector.
Large	Faster Convergence
0	Will not converge
Small	Slow Convergence

### 2.4.2. Momentum term

According to Yu and Liu (2002), another possible way to improve the rate of convergence is by adding some momentum to the adjustment expression. This can be accomplished by adding a fraction of the previous weight change to the current weight change. This term encourage movement in the same direction on successive steps. The addition to such a step can help smooth out the descent path by preventing extreme changes in the gradient due to anomalies. Therefore it is likely to suppress any oscillations that result from changes in the slope of the error surface (Yu and Liu, 2002). Without momentum term, network may get stuck in the shallow local minimum (Fadhlina, 2002). In past research, momentum factors typically have been set between 0 and 1 (Chow *et al.*, 1995).

Table 2.3 Effect of Momentum Term Value

Value	Effect
Too large	Too much skipping
Large	Oscillation is reduced Faster Convergence
0	Get stuck in the shallow local minimum

## 2.5 Three Term Backpropagation

Eventough Standard Two Term backpropagation is well known algorithm but its convergence is relatively slow. The reason for this is the saturation behavior of the activation function used for the network layers. 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 (Zweiri *et al.*, 2003). A number of researches proposed modification to standard BP to improve the efficiency and convergence rate of the BP algorithm. Some requires complex calculation while others is designed to fit for particular domain or application.

A new approach to calculate the change of weight for the link joining the output layer unit to the hidden layer unit is presented by Zweiri *et al.*, (2003). In his approach, a third term is proposed in addition to the usual Learning Rate and Momentum Factor terms. It is called a proportional factor (PF). The results show that the proposed approach outperforms the conventional BP algorithm in terms of convergence speed and the ability to escape from learning stalls (Zweiri *et al.*, 2003).

The BP algorithm given by Equation 2.8 is modified by adding an extra term called proportional term. It is as shown below:

$$\Delta W(k) = \alpha(-\nabla E(W(k))) + \beta\Delta W(k-1) + \gamma e(W(k)) \quad (2.9)$$

where,

$\alpha$  is Learning Rate,

$\nabla E(W(k))$  is gradient of  $E$  at  $W = W(k)$ ,

with  $k = 1; 2; 3; N$ , being the iteration number,

$\beta$  is Momentum term,

$\Delta W(k-1)$  Previous Weight Change,

$\gamma$  is Proportional term,

$e(W(k))$  Difference between the output and the target at each iteration.

Note that BP algorithm given by Eq. (2.9) above has three terms.  $\alpha$  is proportional to the derivative of  $E(W(k))$ , while  $\beta$  is proportional to the previous value of the incremental change of the weights and  $\gamma$  is the Proportional Factor that is proportional to  $e(W(k))$ .

### 2.5.1 Proportional term

This proportional term is proportional to  $e(W(k)) \cdot e(W(k))$ .  $e(W(k))$  represents the difference between the output and the target at each iteration. This term is added to speed up the convergence. According to Zweiri (2003), the comparative test results indicate that the new algorithm convergence faster than the standard BP algorithm and percentage of successful trials for was higher than the standard BP algorithm besides it able to escape from local minima.

## 2.6 Research Trends of Cost function in Backpropagation Network

In 1993, Widder and Fiddy had presented a modification of the cost function based on the current performance of backpropagation error reduction rule. This cost function is able to accelerate BP learning. The advantage of this function was due to its good performance on using no priori knowledge of character recognition.

A new self-organizing net was proposed based on the principle of Least Mean Square Error Reconstruction (LMSER) of an input pattern by Xu (1993). A local learning rule called LMSER is naturally obtained for training nets. The benefit of this method is stated as converged points are stable and corresponding to the global minimum in the Mean Square Error (MSE).

In 1994, Chow *et al.* had introduced Bernoulli Error Measure Approach to train Feed forward ANN for classification problems. In his paper, the Bernoulli Error Measure is very suitable for learning feed forward ANN in classification problem.

In a study done by Telfer and Szu (1994), two minimum misclassification error (MME) energy functions are advanced to achieve minimum network complexity. The authors have stated that by minimizing the network complexity, the improvement on generalization and simplification can be done. The advantage of MME is due to its capability in achieving minimum network complexity.

Humpert (1994) proposed a linear combination of cross-entropy cost function (reminicent of the sum over the quadratic errors) and a new cost function (reminicent of the sum over the absolute value of the errors). This function is help to improve the learning speed.

In 1995, Oh and Lee introduced modified cost function where the error signal can be represented by a square function of the difference between the desired and actual. It resolves the slow learning and specialization problem in pattern recognition applications. Despite to his claimed that the function outperformed other approaches in complex problems, however the method is not tested on simple problems.

A set of minimum cross-entropy error measures, known as Csiszar's measures, are realized in terms of probabilistic attributes of the output and teacher value parameters (Neelakanta, 1996). This offers an alternative set of cost functions which train a neural network optimally towards convergence.

In 1996, Dhiantravan has investigated latent errors in Backpropagation (BP) algorithm. It is showed that the mean square latent error varies according to the correlation of the input vector sequence. A set of weights that gives the least mean error squared for each learning iteration are also obtained, resulting in faster convergence than a conventional BP. However the complexity of the computation was not discussed in the paper.

In 1997, Oh *et. al* proposed modified cost function to improve the error backpropagation (EBP) algorithm of multilayer perceptrons (MLP's) which suffers from slow learning speed. According to the author, the advantage of the method is due to its capability in accelerating the learning speed by reducing the probability of incorrect saturation. However the proposed method is demonstrated on handwritten digit recognition task only.

A new local error-backpropagation (LBP) algorithm based on the definition of a new local mean-squared cost function is introduced by Liu and Tseng (1999). The benefit of this method is the conjugate gradient (CG) of quadratic optimization methods in finding the global optimal solution of quadratic problems within finite steps.

Oh and Lee (1999) proposed a new cost function at hidden layers to speed up MLP training. This cost function overcomes the stalling problem of the LBL algorithm without heuristics or non-optimal learning rates. However, the disadvantage of this

approach is if all hidden nodes and their targets are saturated before the training is success, then the proposed cost function does not work due to the slope term.

Rydvan and Milan (1999) studied and proposed a function of the type  $E = \sum \text{BIQ}(d,y)$ , where BIQ is a biquadratic function. According to his paper, the features and convergence of the networks, trained using this type of cost functions, are examined and a possible method of the choice of its actual parameters is also proposed.

Taji *et al.* (1999) has taken differentiable approximate functions for the absolute value function as an objective function. This approach is more robust and learns faster than standard BP when teacher signals include some incorrect data. However, to clarify that the proposed method is practically efficient, many and more detailed computational experiments are needed, particularly for investigating the performance of the method, which is influenced by the change of an approximate parameter and the best approximate function to be used.

In 2000, a Modified Cost function for On-Line Trained Neuro-Controller is introduced by Salem *et al.* The modified error function ensures high performance during on-line training due to its very simple structure. The authors have stated that the stability analysis need to be performed in future study.

Another modified cost function and adaptive parameter is also proposed by Shamsuddin *et al.* (2001). This modified cost function has been proved to improve convergence speed of standard BP with convincing recognition rate that is 100% recognition for XOR classification problem, data profitability and Kuala Lumpur Composite Index (KLCI) from Kuala Lumpur Stock Exchange (KLSE) and

handwritten/handprinted digit problem. However, for some datasets, the network does not converge.

Jiang *et al.* (2003) introduced different approaches based on Fourier kernel function for the generalized fast training criterion of neural networks. It is stated that the function overcomes the problems of getting stuck into local minima and slow convergence.

Furthermore, Wang *et al.* (2004) investigated a modified cost function for BP which can harmonize the update of weights, connected to the hidden layer and those connected to the output layer by adding one term to the conventional cost function. This cost function avoids the local minima problem yet more analysis on large problems and some detailed discussion on parameter settings is still required. The same function has been adapted in different problems by Bi *et al.* (2004) and Zhang *et al.* (2007).

Lv and Yi (2005) presented an absolute cost function using the Lyapunov method in 2005. This algorithm is more robust and faster for learning than standard BP when target signals include some incorrect data. It also avoids local minima. However, the function is tested on XOR problem only, and further justifications on its efficiency can be debated.

In the same year, an adaptive error-constrained least mean square (AECLMS) algorithm is derived and proposed using adaptive error-constrained optimization techniques by Choi *et al.* (2005). The cost function of the LMS algorithm is modified using augmented Lagrangian multipliers. The paper also showed improved performance in terms of convergence speed and misadjustment. Ng *et al.* (2006) considered localized generalization error model for single layer perceptron neural network (SLPNN). This is

an extensibility of the localized generalization error model for supervised learning with minimization of Mean Square Error. However, this work serves as the first step of investigating localized generalization error models of Multilayer perceptron Neural Network (MLPNN).

Otaïr and Salameh (2006) studied three algorithms and different versions of backpropagation training is proposed for stable learning and robustness to oscillations. The new modification consists of a simple change in the error signal function. This algorithms tested on OR problem, encoding problem and character recognition. Rimer and Martinez (2006) proposed Classification-Based (CB) cost functions that attempt to guide the network directly to correct pattern classification. It reduces average test error. However, this function only applicable for Classification-based (CB) problems. Finally, Zhang *et.al* (2007) has applied an cost function of Wang *et.al* (2004) for element neural network. This method can avoid the local minima problem and accelerate the speed of the convergence. Table 2.4 summarizes previous studies on the enhancement of BP cost function.

Table 2.4 Previous Cost function Improvements

<b>Year</b>	<b>Author</b>	<b>Description</b>
1993	Widder and Fiddy	The modification of the cost function is based on current performance of backpropagation error reduction rule. The advantage of this function is due to its good performance using no priori knowledge of character recognition.
1993	Xu	A new self-organizing net was proposed based on the principle of Least Mean Square Error Reconstruction (LMSER) of an input pattern. A local learning rule (LMSER) is naturally obtained for training nets. The

		advantage of this method is converged points are stable and corresponding to the global minimum in the Mean Square Error (MSE).
1994	Chow <i>et al.</i>	Bernoulli Error Measure Approach to train Feed forward ANN for classification problems.
1994	Telfer and Szu	Two minimum misclassification error (MME) energy functions are advanced to achieve minimum network complexity which is important for improving generalization and simplifying implementation. Advantage of MME is it is proven to achieve minimum network complexity.
1994	Humpert	A linear combination of cross-entropy cost function (reminicent of the sum over the quadratic errors) and a new cost function (reminicent of the sum over the absolute value of the errors) s was shown to improve the learning speed.
1995	Oh and Lee	A modified cost function where the error signal can be represented by a square function of the difference between the desired and actual. It resolves the slow learning and specialization problem in pattern recognition applications. However, the author has claimed that the function outperforms other approaches in complex problems and the method is not tested on simple problems.
1996	Neelakanta	A set of minimum cross-entropy error measures, known as Csiszar's measures, are realized in terms of

		probabilistic attributes of the output and teacher value parameters. This offers an alternative set of cost functions which train a neural network optimally towards convergence.
1996	Dhiantravan	Latent errors in the Backpropagation (BP) algorithm are investigated. A set of weights that gives the least mean error squared for each learning iteration are also obtained, resulting to faster convergence than a conventional BP.
1997	Oh	A modified cost function is proposed to improve the error backpropagation (EBP ) algorithm of multilayer perceptrons (MLP's). It accelerates the learning speed by reducing the probability of incorrect saturation. However the proposed method is demonstrated on handwritten digit recognition task only.
1999	Liu and Tseng	A new local error-backpropagation (LBP) algorithm based on the definition of a new local mean-squared cost function is introduced. Conjugate gradient (CG) of quadratic optimization methods is used to find global optimal solution of quadratic problems within finite steps. However, the disadvantage is if all hidden nodes and their targets are saturated before successful training, the proposed cost function does not work due to the slope term.
1999	Oh and Lee	A new cost function at hidden layers to speed up the training of multilayer perceptrons (MLP's). It

		overcome the stalling problem of the LBL algorithm without heuristics or nonoptimal learning rates.
1999	Rydvan, and Milan	This article studied and proposed a function of the type $E = \sum \text{BIQ}(d,y)$ , where BIQ is a biquadratic function. A possible method of the choice of its actual parameters is also proposed.
1999	Taji <i>et al.</i>	Approximate functions for the absolute value function were proposed as an objective function. This approach is more robust and learns faster than standard backpropagation when teacher signals include some incorrect data.
2000	Salem <i>et al.</i>	A Modified Cost function for On-Line Trained Neuro-Controller is introduced. This modified cost function ensures high performance during on-line training due to its very simple structure. Author has stated that stability analysis is needed to be performed in future study.
2001	Samsuddin <i>et al.</i>	New Modified cost function (MM) is proposed. It has been proved to improve convergence speed of standard backpropagation. However, for some datasets, the network does not converge.
2003	Jiang <i>et al.</i>	Different approaches are based on Fourier kernel function for the generalized fast training criterion of neural networks. It is stated that the function overcome the problems of getting stuck into local minima and slow convergence.

2004	Wang <i>et al.</i>	A modified cost function for backpropagation, which can harmonize the update of weights, connected to the hidden layer and those connected to the output layer by adding one term to the conventional cost function. It avoids the local minima problem yet more analysis on large problems and some detailed discussion on parameter settings is still required.
2004	Bi <i>et al.</i>	Proposed a modified cost function with added term that is similar to (Wang <i>et al.</i> , 2004) algorithm. Simulations on the modified XOR problem have been performed to show that the algorithm avoids local minima. It also proposes a method of using Lyapunov stability theory to directly minimize the absolute cost function.
2005	Lv and Yi	An absolute cost function using the Lyapunov method. This algorithm is more robust and faster for learning than standard backpropagation when target signals include some incorrect data. It also avoids local minima. However, the function is tested on XOR problem only.
2005	Choi <i>et al.</i>	An adaptive error-constrained least mean square (AECLMS) algorithm is derived and proposed using adaptive error-constrained optimization techniques. The cost function of the LMS algorithm is modified using augmented Lagrangian multipliers. Paper showed improved performance in terms of convergence speed and maladjustment.

2006	Ng <i>et al.</i>	Localized generalization error model for single layer perceptron neural network (SLPNN). This is an extensibility of the localized generalization error model for supervised learning with minimization of Mean Square Error. However, this work serves as the first step of investigating localized generalization error models of Multilayer perceptron Neural Network (MLPNN).
2006	Otair and Salameh	Three algorithms to improve the different versions of backpropagation training is proposed. The new modification consists of a simple change in the error signal function which create stable learning and robustness to oscillations. This algorithms tested on OR, encoding problem and character recognition.
2006	Rimer and Martinez	Classification-Based (CB) cost functions that attempt to guide the network directly to correct pattern classification. It reduces average test error but only applicable for Classification-based (CB) problems.
2007	Zhang <i>et.al</i>	An cost function of Wang <i>et al.</i> (2004) is introduced for element neural network. This method can avoid the local minima problem and accelerate the speed of the convergence.

## **2.7 Cost function**

The error calculations used to train a neural network are very important (Edward, 2004). Quantifying the output error provides a way for iteratively updating the network weights in order to minimize that error and thereby achieve more accurate output (Rimer and Martinez, 2006). This cost function is sometimes known as objectives function or error function and so on.

The derivative of the cost function is one of the factors in the weight update equations. It is therefore important for success of the application, to train the network with a cost function that resembles the objective of the problem at hand (Falas and Stafilopatis, 1999). Cost function that has been used in weight adjustment formulation in Standard Two Term and Three Term backpropagation is Mean Square Error.

### **2.7.1 Mean square error**

In statistics, the mean squared error or MSE of an estimator is the expected value of the square of the 'error'. The error is the amount by which the estimator differs from the quantity to be estimated. MSE is one of many ways to quantify the amount by which an estimator differs from the true value of the quantity being estimated (Wikipedia, 2007).

The MSE is the most popular and commonly used in error measure used in BP learning. The MSE function is defined as:

$$MSE = \frac{1}{2} \sum_k (Desired - Actual)^2 \quad (2.10)$$

where,

*Desired* is target value,

*Actual* is actual value generated by network

The error signal  $\delta_k$  is as shown below:

$$\delta_k = -(T_k - O_k)O_k(1 - O_k) \quad (2.11)$$

where,

$T_k$  is the target output value.

$O_k$  is the actual output of the network.

Mean Square error is continuous, monotonous, differentiable and has a single minimum. The derivative of mean square is also continuous and analogous to the sign and magnitude of the error (Falas and Stafilopatis, 1999).

Falas and Stafilopatis (1999) studied on impact of cost function in neural network classifier. Their result showed that a cost function other than the usual mean square gives a better performance, both in terms of the number of epochs needed for training, as well as the obtained generalization ability of the trained network.

Mean Square error gives more emphasis to reducing the larger errors as compared to the smaller errors, due to the squaring that takes place in the algorithm. This is desirable in many cases but not for all cases. It would be disadvantages in some cases. In addition, if a class is not well presented and happens to have small errors, it may be completely ignored by the learning algorithm due to the summation of the errors for all input patterns.

### 2.7.2 Bernoulli Cost Function (BL)

According to Chow *et al.* (1994), the Bernoulli cost function is more suitable for classification problem compared to Mean Square Error.

The Bernoulli error measure is defined as (Chow *et al.*, 1994):

$$BL_p = -\sum_i \{t_{pj} \log(y_{pj}) + (1 - t_{pj}) \log(1 - y_{pj})\} \quad (2.12)$$

where

$t_p = [t_{p1}, t_{p2}, \dots, t_{pm}]^T = \{t_1, t_2, t_3, \dots, t_p\}$  is the vector of desired target classes

$[y_{p1}, y_{p2}, \dots, y_{pm}]^T$  is output of the network.

As indicated in equation 2.12, even when the network output is same as the target value, the error  $e_p$  is not necessarily zero. For example when  $t_{pj} = 0.5$  and  $y_{pj} = 0.5$ , the error measure  $e_p$  is 0.3. However when y matches it does not give the optimal point of the cost function.

The error is backpropagated through  $\delta_{pj}$  (error signal) which is function of the chosen error measure as shown below:

$$\delta_{pj} = \left( \frac{1 - t_{pj}}{1 - y_{pj}} - \frac{t_{pj}}{y_{pj}} \right) (1 - y_{pj}) y_{pj} = (t_{pj} - y_{pj}) \quad (2.13)$$

According to the author, significant increase in training speed has been obtained by using the Bernoulli error measure instead Mean Square error measure. With

appropriate choice of training parameters, the Bernoulli error measures certainly a potentially appropriate choice to train a network to learn classification problems (Chow *et al.*, 1994).

### 2.7.3 Modified Cost Function

Many error calculations have been researched, trying to find a calculation with a short training time that is appropriate for the network's application (Edward, 2004). In year 2001, Shamsuddin *et al.* proposed a modified cost function for Two Term BP to improve the speed of convergence. This cost function is used for calculating the error signal instead of usual mean square error. So, in this study, the Modified Cost function (Shamsuddin *et al.*, 2001) will be employed in Three Term Backpropagation to measure the performance.

The modified cost function ( $mm$ ) is defined implicitly below as in (Shamsuddin *et al.*, 2001):

$$mm = \sum_K \rho_k \quad (2.14)$$

with,

$$\rho_k = \frac{E_k^2}{2a_k(1-a_k^2)} \quad (2.15)$$

where

$$E_k = t_k - a_k,$$

and ,

$E_k$  error at output unit  $k$ ,

$t_k$  target value of output unit  $k$ ,

$a_k$  an activation of unit  $k$ .

Delta rule is used in the updating weight of backpropagation model. It is proportional to the negative gradient of weight changes. The common representation is as follows:

$$\Delta W_{kj} = -\alpha \frac{\partial E}{\partial W_{kj}} \quad (2.16)$$

Thus, above relation can be written as follow:

$$\Delta W_{kj} = -\alpha \frac{\partial \rho}{\partial W_{kj}} \quad (2.17)$$

By Chain Rule,

$$\frac{\partial \rho}{\partial W_{kj}} = \frac{\partial \rho}{\partial Net_k} \cdot \frac{\partial Net_k}{\partial W_{kj}} \quad (2.18)$$

Since  $Net_k = \sum_j w_{kj} a_j + \theta_k$  , that  $\theta_k$  is bias. Partial derivative of it is as below:

$$\frac{\partial Net_k}{\partial W_{kj}} = a_j \quad (2.19)$$

Now, substitutes (2.17) into (2.16) gives

$$\frac{\partial \rho}{\partial W_{kj}} = \frac{\partial \rho}{\partial Net_{kj}} \cdot a_j \quad (2.20)$$

Let  $\frac{\partial \rho}{\partial Net_k} = \delta_k$ . Here error signal ( $\delta_k$ ) is from the output to the hidden layer.

By using chain rule,

$$\delta_k = \frac{\partial \rho_k}{\partial Net_k} = \frac{\partial \rho_k}{\partial a_k} \cdot \frac{\partial a_k}{\partial Net_k} \quad (2.21)$$

Now, since

$$a_k = f(Net_k) = \frac{1}{(1 + e^{-Net_k})} \quad (2.22)$$

By taking partial derivatives and simplify it in term of  $a_k$ , gives

$$f'(Net_k) = \frac{\partial a_k}{\partial Net_k} = a_k(1 - a_k) \quad (2.23)$$

So, now,  $\delta_k$  becomes,

$$\delta_k = \frac{\partial \rho_k}{\partial a_k} \cdot a_k(1 - a_k) \quad (2.24)$$

It is known that  $\rho_k = \frac{E_k^2}{2a_k(1 - a_k^2)}$ . By taking partial derivatives with respect to

$a_k$ , gives

$$\frac{\partial \rho_k}{\partial a_k} = \frac{-[4a_k(t_k - a_k)(1 - a_k^2) + 2(t_k - a_k)^2(1 - 3a_k^2)]}{4a_k^2(1 - a_k^2)} \quad (2.25)$$

Simplify equation (2.23) becomes,

$$\frac{\partial \rho_k}{\partial a_k} = - \left[ \frac{E + \rho(1 - 3a_k^2)}{a_k(1 - a_k^2)} \right] \quad (2.26)$$

By substituting (2.21) and (2.24) in (2.19) we have the improved error signal of BP for the output layer as,

$$\delta_k = \frac{\partial \rho_k}{\partial Net_k} = - \frac{(E + \rho(1 - 3a_k^2))}{a_k(1 - a_k^2)} \cdot a_k(1 - a_k)$$

$$\delta_k = \frac{\partial \rho_k}{\partial Net_k} = - \frac{(E + \rho(1 - 3a_k^2))}{1 + a_k} \quad (2.27)$$

By substituting (2.25) into (2.18), gives

$$\frac{\partial \rho}{\partial W_k} = - \frac{(E + \rho(1 - 3a_k^2))}{1 + a_k} \cdot a_j \quad (2.28)$$

The adaptation of weight between output layer and hidden layer is now:

$$\Delta W_{kj} = -\alpha \frac{\partial \rho}{\partial W_{kj}} + \beta \Delta W_{kj} + \gamma e(W_{kj}) \quad (2.29)$$

$$\Delta W_k = -\alpha \left[ - \frac{(E + \rho(1 - 3a_k^2))}{1 + a_k} \right] \cdot a_j + \beta \Delta W_{kj} + \gamma e(W_{kj}) \quad (2.30)$$

$$\Delta W_k = \alpha \left[ \frac{(E + \rho(1 - 3a_k^2))}{1 + a_k} \right] \cdot a_j + \beta \Delta W_{kj} + \gamma e(W_{kj}) \quad (2.31)$$

and an error signal for modified BP of the hidden layer is the same as standard BP,

$$\partial_j = \sum \delta_k w_j f'(a_j) \quad (2.32)$$

where,

$$f'(a_j) = a_j(1 - a_j) \quad (2.33)$$

#### 2.7.4 Improved Cost Function (IC)

Improved Cost Function has been introduced by Zhang *et al.* (2007). The author proposed the improved cost function the Element Neural Network. However Wang *et al.* have introduced the same error type for Two Term Backpropagation algorithm back in 2004. The successfulness of the cost function in Backpropagation has invited Zhang *et al.* to be proposing it to Element Neural Network.

According to the author, the neuron outputs in the output layer and those in the hidden layer should be considered together during the iterative update procedure. So, author adds one term embodying the neuron outputs in the hidden layer to the conventional cost function. In this way, it is believed that weights connected to the hidden layer and the output layer could be modified harmoniously. The Improved Cost Function (IC) algorithm is taken from Zhang *et al.* (2007).

$$IC = E_A + E_B \quad (2.34)$$

Where,

$$E_A = \frac{1}{2} \sum_{p=1}^P \sum_j^J (t_{pj} - o_{pj})^2 \text{ is conventional Mean Square Error}$$

and

$$E_B = \frac{1}{2} \sum_{p=1}^P \left( \sum_j^J (t_{pj} - o_{pj})^2 \right) * \left( \sum_j^H (y_{pj} - 0.5)^2 \right) \text{ is the added term for Cost function}$$

with,

$y_{pj}$  is the output of the  $j^{\text{th}}$  neuron in the hidden layer

0.5 is average value of activation function

H is the number of neurons in the hidden layer.

$\sum_{j=1}^H (y_{pj} - 0.5)^2$  can be defined as the degree of saturation in the hidden layer for pattern p. This added term is used to keep the degree of saturation of the hidden layer small while  $E_A$  is large where the output layer does not approximate to the desired signals. While the output layer approximates to the desired signals, the effect of term  $E_B$  will be diminished and will eventually become zero (Wang *et al.*, 2004).

## 2.8 Importance of Cost function

Researchers have identified that a good cost function can significantly accelerate the convergence rate than MSE cost function for BP algorithm by parameter optimization. Cost functions which was introduced by (Humpert,1994; Neelakanta, 1996; Dhiantravan, 1996; Oh and Lee, 1999; Taji *et al.*,1999; Shamsuddin *et al.*, 2001; Jiang *et al.*, 2003; Wang *et al.*, 2004; Lv and Yi, 2005; Choi *et al.*, 2005; Otair and Salameh, 2006; Zhang, 2007) has been proved to converge faster during experiments compared to conventional Backpropagation. So, the choice of proper cost function is being an important factor.

Besides that proper cost function also helps to improve generalization or accuracy. Quantifying the output error provides a way for iteratively updating the network weights in order to minimize that error and thereby achieve more accurate output (Telfer and Szu, 1994; Rimer and Martinez, 2006).

At the same time, some researchers stated that their cost function overcomes the problems of getting stuck into local minima (Telfer and Szu, 1994; Jiang *et al.*, 2003; Wang *et al.*, 2004; Bi *et al.*, 2004) while others stated that it tend to find local minima less often than MSE does (Oh and Lee, 1999; Zhang *et al.*, 2007). These obviously encourage more researches to explore cost function as one of significant way to overcome local minima.

Cost function is also can help to improve in terms convergence characteristics, such as stable learning and robustness to oscillations based on experiments that are conducted to compare and evaluate the convergence behavior (Otair and Salameh, 2006). A network needs to be stable to produce desires result. If network becomes unstable then

it will result in poor performance. So, good cost function is important to ensure stable learning of the network.

## 2.9 Comparison

Comparison can be defined as evaluation of two or more method or techniques in this study. Comparison must be carried out to investigate the performance of Three Term BP with different cost function. This study will compare the four cost functions mentioned earlier that are Three Terms BP with Mean Square Cost function, Three Term BP with Bernoulli Cost Function, Three Term BP with Modified Cost Function, Three Term BP with Improved Cost Function.

There are three type of comparison that will be carried out. They are error value, convergence time and accuracy rate. This three type of comparison is needed to measure the performance of the various cost function in Three Term BP. These three types of comparison criteria are compared according to the number of epochs (iteration). For example, the network is being run for 10 epochs and all the comparison criteria such as error value, convergence time and accuracy rate are taken down to do comparison.

Comparison in term of error refers to the error produced by the network at the end of specified epoch (iteration) size during training. For example, the error value that produced by the network of a Three Term BP with a particular cost function during training at a specified epoch is taken down and compared among Three Term BP with other cost function's error value at that specified epoch. The error value that is the smallest will be considered as better then others.

Besides error value, convergence time is needed to do comparison among different cost functions for Three Term BP. Convergence time is the time needed by the network to converge to the output. This convergence time is including training and testing period. Here the convergence time will be measured in milliseconds. Network that converges faster (less convergence time) will be considered better.

Lastly, accuracy rate is measured in percentage. Accuracy rate refers to number of correct output achieved by the network. Percentage is calculated as number of correct output over overall and multiply to hundreds. This criterion can be considered as the most important comparison criteria. The higher the accuracy rate, the better the network. So, a better cost function is weighted more by the higher percentage of accuracy rate.

These three criteria are needed to compare the performance of the different cost function for Three Term BP. All three criterions will be analyzed in detail to identify the better cost function for Three Term BP.

## **2.10 Classification**

Backpropagation neural network has been applied successfully in wide domain for different purposes. One of the purposes is classification. Classification can serve variety of domain such as medicine, textile, image processing and so on. It is applicable in basically every situation in which a relationship between the predictor variables called inputs and predicted variables called outputs exist.

It also able to compute many probable results across many parameters, if we have sufficient examples with a known outcome from the past. Traditionally, classification problems are learned through error backpropagation by providing a vector of strict (“hard”) 0/1 target values to represent the class label of a particular pattern (Rimer and Martinez, 2006).

Desai, Bandyopadhyay and Kane (2000) presented a method to classify and determine blend composition using neural networks. They proposed a method to automatically categorize an unknown fabric and determine blend composition with the help of artificial neural network with back propagation algorithm.

In 2005, neural network trained with the backpropagation algorithm with adaptive learning rate is used to classify the marble slabs in three categories, according to their quality. The results show very high performance compared with the traditional (manual) system. The successful classification rate was 98.9% (Martínez-Alajarín *et al.*, 2005). Poynton and McDaniel (2006) examined the ability of a backpropagation neural network classifier to distinguish between current and former smokers in the 2000 National Health Interview Survey (NHIS) sample adult and their study establishes the ability of backpropagation neural networks to classify a complex health behavior, smoking cessation (Poynton and McDaniel, 2006).

In 2007, backpropagation artificial neural network, was built from logarithmic sigmoid neurons and trained, has demonstrated to provide excellent Atrial fibrillation classifying results. Atrial fibrillation (AF) is an arrhythmia in which electrical activity in the atria is disorganized (Kara and Okandan, 2007).

All these studies have shown that ANN has been used extensively in classifications problems in past years. Thus, this study will employ Three Term Backpropagation algorithm with different cost function to classify classification problems, which are Balloon, Cancer, Diabetes and Pendigits.

## **CHAPTER 3**

### **RESEARCH METHODOLOGY**

#### **3.1 Introduction**

This chapter discusses the methodology that is used in this study. The first section describes the steps involve, follows by dataset representation. The third part describes the parameters that are required in BP learning and training. Subsequent sections explicate the conducted experiments and analysis to investigate the efficiency of Three Term Backpropagation with various cost functions: Mean Square Error, Bernoulli Cost Function, Modified Cost Function and Improved Cost Function.

This research study was conducted based on the methodology. This methodology plays an important role in implementing this research study accordingly. The details of the methodology are explained in detail in this chapter.

### 3.2 Methodology

The procedures involved in implementing Three Term BP algorithm with various cost functions studied and presented substantially. There are 11 major steps involved in implementing Three Term BP with various cost functions as shown below. These steps are applied for each dataset.

1. Defining dataset attributes
2. Characterization of network architecture (inputs, hidden and output nodes)
3. Determination of network parameter and formulation of Three Term BP with Mean Square Error (MSE).
4. Commencing Three Term BP training and testing with MSE using K+10 Increment Rule
5. Determination of network parameter and formulation of Three Term BP with Bernoulli Cost Function (BL).
6. Commencing Three Term BP training and testing with BL using K+100 Increment Rule
7. Determination of network parameter and formulation of Three Term BP with Modified Cost Function (MM).
8. Commencing Three Term BP training and testing with MM using K+100 Increment Rule
9. Determination of network parameter and formulation of Three Term BP with Improved Cost Function (IC).
10. Commencing Three Term BP training and testing with IC using K+100 Increment Rule
11. Substantial Comparison and Analysis.

The general framework of proposed study is as shown in Figure 3.1. The following section will discuss the processes in detail.

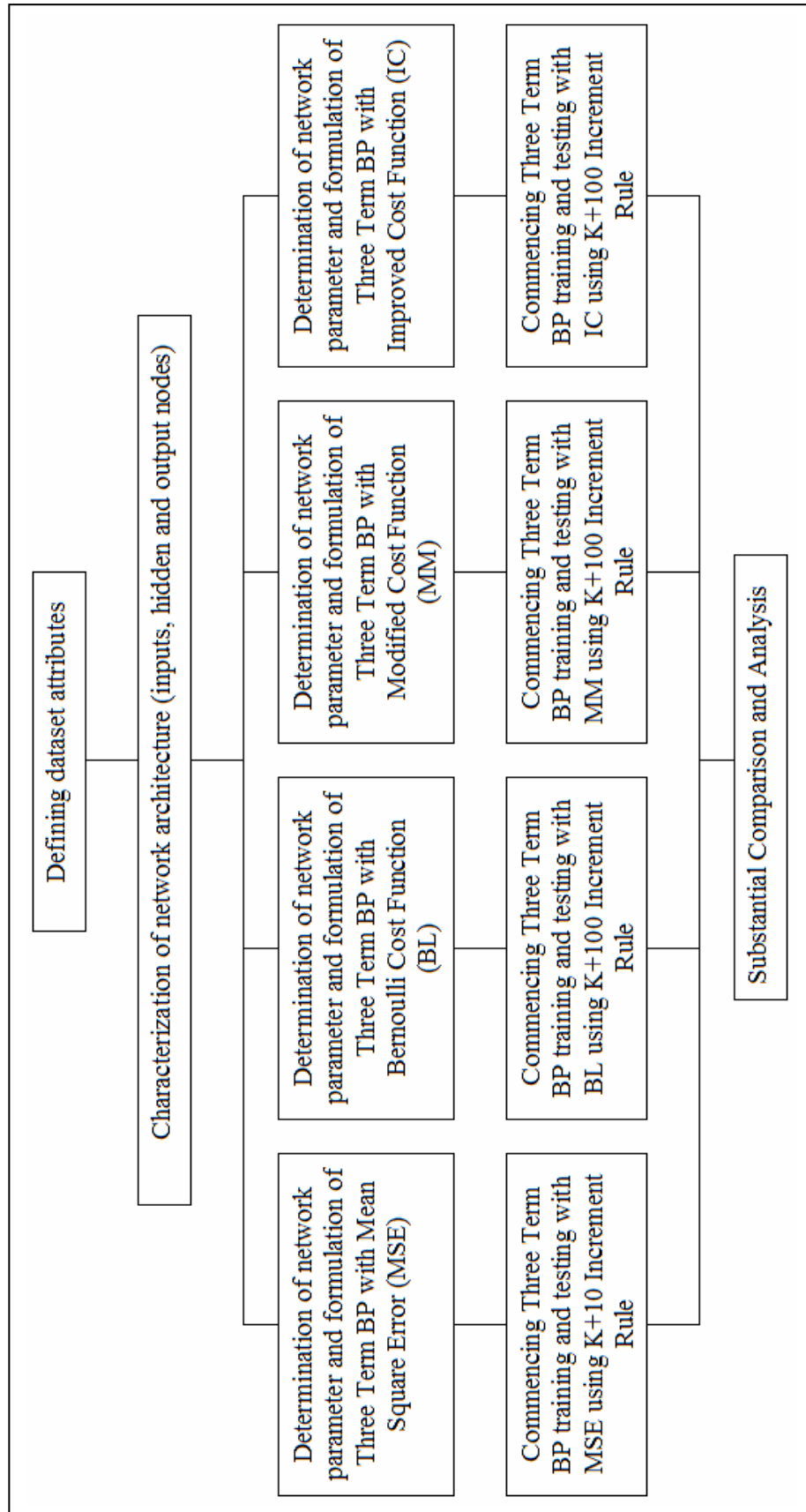


Figure 3.1: A general framework of the proposed study

### 3.3 Defining Dataset Attributes

Selection of datasets is important to benchmark the findings in this study. Hence, real world classification problem datasets have been chosen and these data are available from UCI Machine Learning Repository at *Center for Machine Learning and Intelligent Systems*. The datasets are Balloons, Cancer Diabetes and Pendigits.

#### 3.3.1 Balloons

Balloons dataset is used for classifying four balloons attributes. Data previously used in cognitive psychology experiment. Michael Pazzani donated this datasets (Asuncion and Newman, 2007). Balloons dataset contains of a class with 16 instances. The Network will have 4 inputs to represent information for each attribute and 1 output. The output is either inflated true (T) or inflated false (F). Classification of balloons attributes are colour (yellow, purple), size (large, small), act (stretch, dip) and age (adult, child) into a class, either inflated true (T) or inflated false (F).

Balloons dataset have four sets that represent different conditions of an experiment. All of them have the same attributes as follows:

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 (color=yellow and size = small) or
4. small-yellow+adult-stretch.data Inflated is true if (color=yellow and size = small) or (age=adult and act=stretch)

### **3.3.2 Cancer**

These dataset were generated by University of Wisconsin Madison, by Dr. William H. Wolberg (Asuncion and Newman, 2007). The dataset consist 9 inputs and 1 output with 500 instances. The inputs are Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses. They will have 1 output type of Cancer that is either benign or malignant. This datasets classify the data into two classes of breast lump's diagnosis: benign or malignant; based on automated microscopic examination of cells collected by needle aspiration.

### **3.3.3 Diabetes**

The source of this dataset is Michael Kahn from Washington University (Asuncion and Newman, 2007). The dataset consist 8 inputs and 1 output with 768 instances. Number of instances that was used for training is 512 while Testing 256. The inputs are number of times pregnant, plasma glucose concentration, diastolic blood pressure, triceps skin fold thickness, 2-hour serum insulin, BMI, diabetes pedigree function and age. The output is result of dividing the set into non-diabetes and diabetes subsets.

### 3.3.4 Pendigits

Pendigits datasets is actually Pen-Based Recognition of Handwritten Digits. Alpaydin and Alimoglu donated this dataset (Asuncion and Newman, 2007). Originally this digit database is formed by collecting 250 samples from 44 writers. Number of Instances used for Training is 7494 and Testing is 3498. But for this study, number of instances used is 1000 where 500 is for training and 500 is for testing. The dataset consist 16 inputs and 10 outputs.

### 3.3.5 Summary of Datasets

The datasets characterizations as discussed in section 3.3 are summarized in Table 3.1. This Table 3.1 shows that number of input, output and total instances for each datasets (Balloon, Cancer, Diabetes and Pendigits). For example, balloons datasets have 4 inputs, cancer has 9 inputs, diabetes has 8 inputs and pendigits has 16 inputs. Balloon, cancer and diabetes have 1 output while pendigits have 10 outputs. Total number of instances in the balloon, Cancer Diabetes and pendigits datasets are 16, 500 and 768 and 1000 accordingly.

Table 3.1 Summary of Datasets

	<b>Balloons</b>	<b>Cancer</b>	<b>Diabetes</b>	<b>Pendigits</b>
<b>Input</b>	4	9	8	16
<b>Output</b>	1	1	1	10
<b>Total</b>	16	500	768	1000

### 3.4 Characterization of Network Architecture

Subsequent to datasets presentation is to determine network architecture. It is very crucial to determine the architecture because it greatly influences the classification accuracy. Basically, the number of layer in neural network needs to be determined, follows by the number of nodes in each layer. In this study, three layers will be employed which consists of one input layer, one hidden layer and one output layer. This is to standardize the comparison criteria. The number of nodes on each layer depends on the datasets representation. The number of inputs and outputs for each datasets that will be used in this study is shown in Table 3.1. The number of hidden layer needs to be determined too. From previous studies, it is known that the hidden layer affects the time for training and capacity to generalize.

There are several ways to determine number of nodes in hidden layer. For example, Kolmogorov's Theorem was published in the 1950's describes how the neural network is to be constructed where the neurodes have a connection to each neurode in the hidden layer. The hidden layer has  $(2*n + 1)$  neurodes, where  $n$  is the number of inputs. Charytoniuk and Chen (2000) presented an equation of  $\sqrt{m*n}$ , where  $m$  is the number of input nodes and  $n$  is the number of output nodes.

There is no standard formula or procedure to determine the optimal hidden layer nodes. Hence, this study will employ the charytoniuk formula as below:

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

where,

$m$  is the number of input nodes,

$n$  is the number of output nodes.

### 3.4.1 Balloon Dataset

The Balloon Network will have 4 inputs: colour (yellow, purple), size (large, small), act (stretch, dip) and age (adult, child), and one output. The output is either inflated true (T) or inflated false (F). Number of hidden nodes is calculated using Charytoniuk formula. For this datasets there will be 2 hidden nodes. The network structure is shown in Figure 3.2.

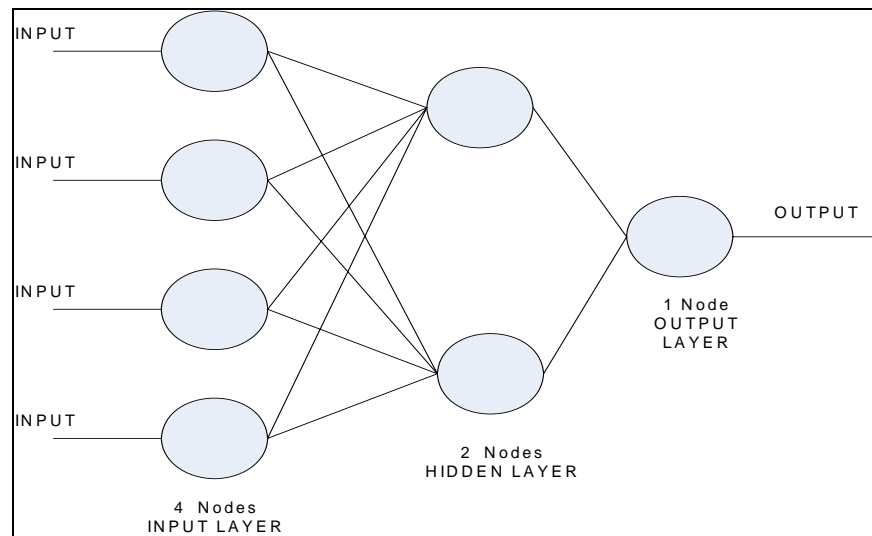


Figure 3.2 Balloon Datasets's Network Structure

### 3.4.2 Cancer Dataset

The Cancer dataset network will have 9 attributes as inputs and 1 output. Number of hidden nodes is calculated using Charytoniuk formula. For this datasets there will be three hidden nodes, and the network structure is shown in Figure 3.3.

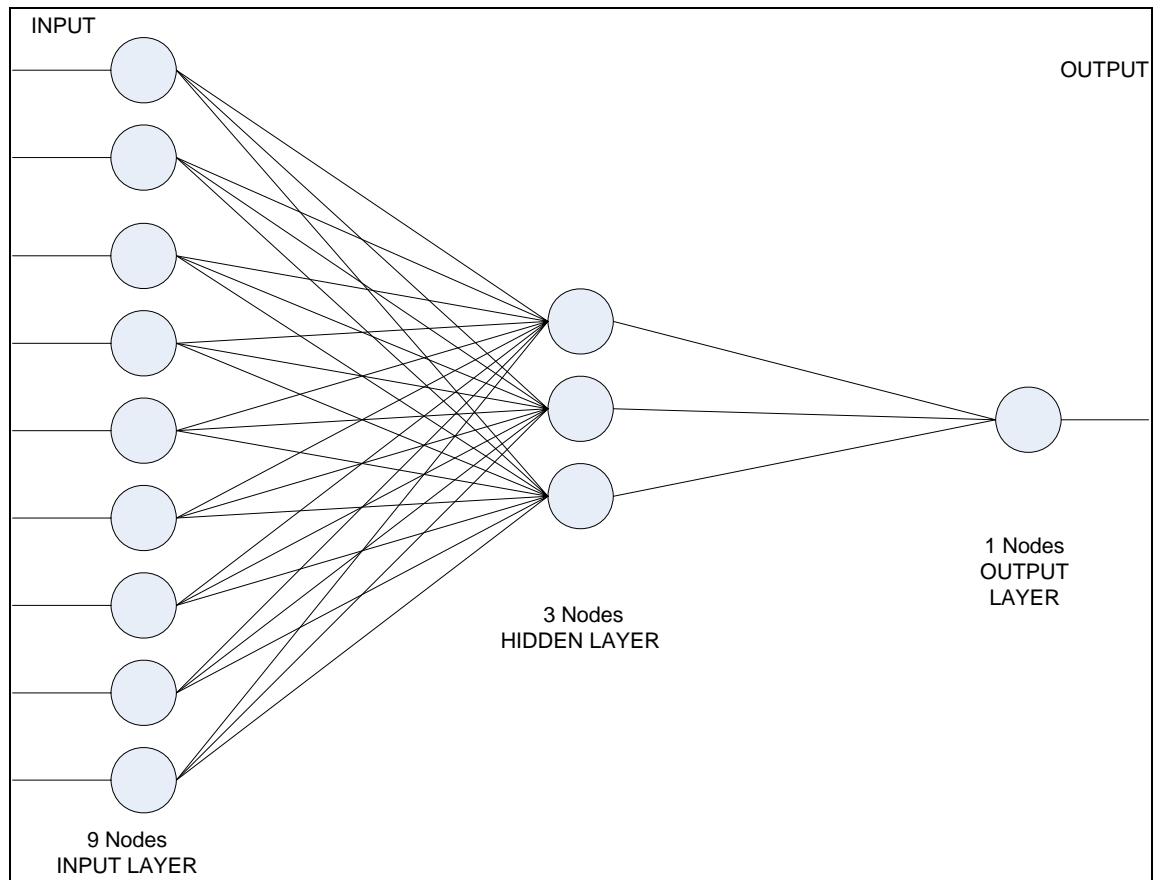


Figure 3.3 Cancer Datasets's Network Structure

### 3.4.3 Diabetes Dataset

The Diabetes dataset network have 8 inputs and 1 outputs. Number of hidden nodes is calculated using Charytoniuk formula. For this datasets there will be two hidden nodes. The network structure is shown in Figure 3.4.

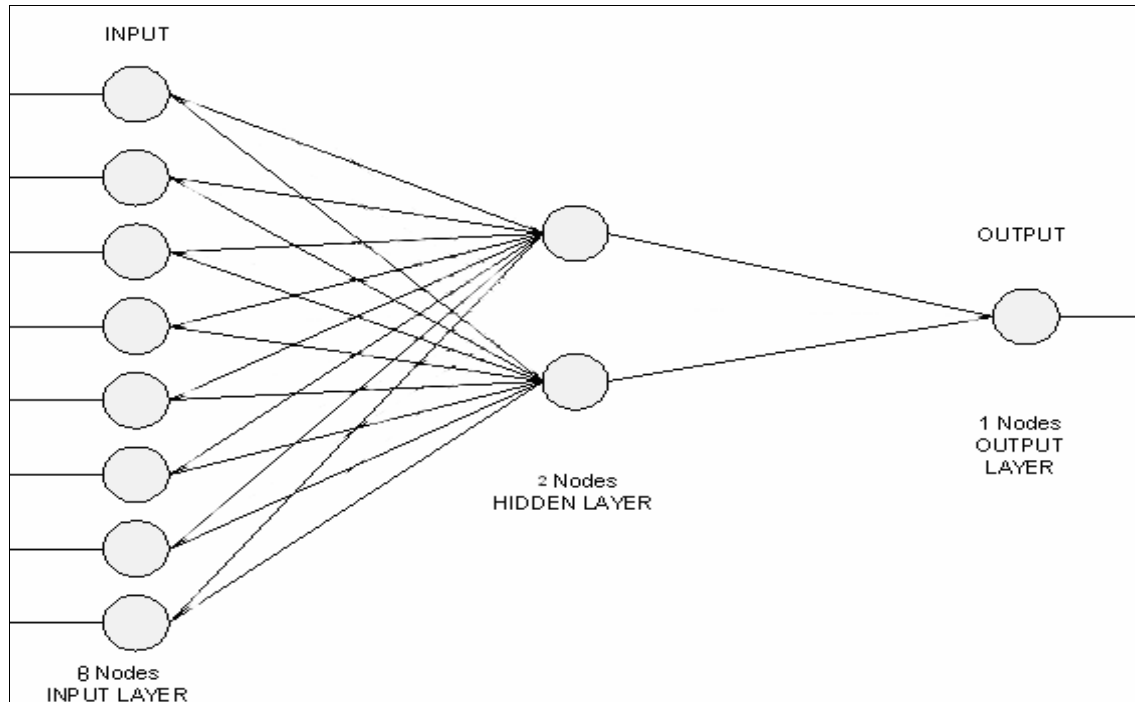


Figure 3.4 Diabetes Datasets's Network Structure

### 3.4.4 Pendigits Dataset

The Iris dataset network will have 16 inputs and 10 outputs. Number of hidden nodes is calculated using Charytoniuk formula. For this datasets there will be four hidden nodes as shown in Figure 3.5. Table 3.2 depicts the summary of network structure for all datasets to be used in this study.

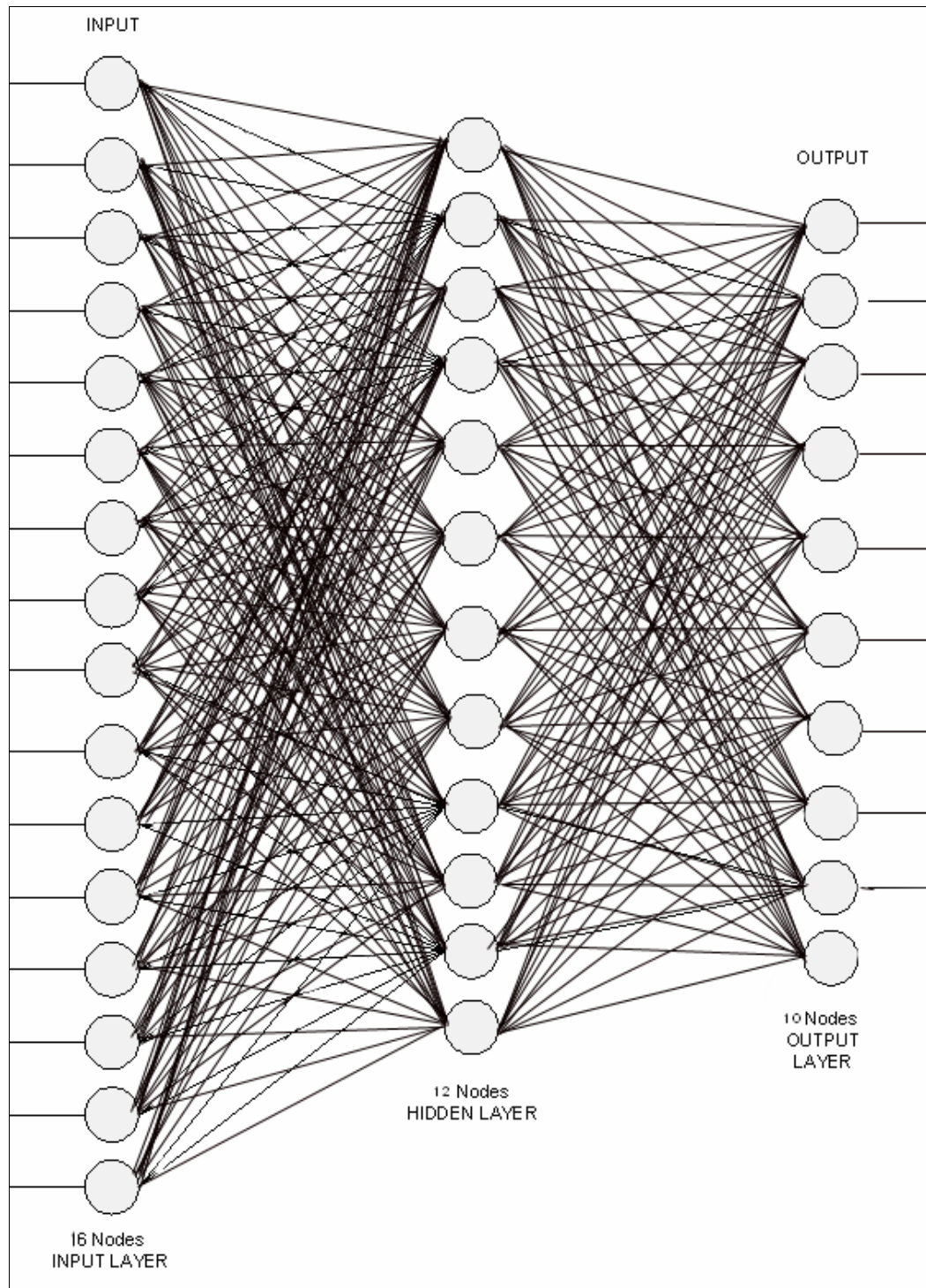


Figure 3.5 Pendigits Datasets's Network Structure

Table 3.2: Summary of different Dataset's Network Architecture

	<b>Balloons</b>	<b>Cancer</b>	<b>Diabetes</b>	<b>Pendigits</b>
<b>Input</b>	4	9	8	16
<b>Hidden</b>	2	3	2	12
<b>Output</b>	1	1	1	10

### 3.5 Determine Network Parameters and Formulation of MSE Cost Function

Network parameters that will be used for Three Term Backpropagation learning are learning rate, momentum Term and Proportional term. The assigned value for learning rate is 1.0, momentum factor is 0.75 and proportional term is 0.95. These values were taken from Zweri *et al.* (2003) best practice value for the best success rate. The same values will be used for all the dataset as standardization. The Cost Function that will be used in is mean square error as given in equation 3.1.

$$MSE = \frac{1}{2} \sum_k (Desired - Actual)^2 \quad (3.1)$$

where,

*Desired* is target value,

*Actual* is actual value generated by network

### 3.6 Determine Network Parameters and Formulation of Bernoulli Cost Function

The assigned value for learning rate is 1.0, momentum factor is 0.75 and proportional term is 0.95. It is the same values for each datasets too. The Cost Function that will be used is Bernoulli Cost Function (BL) as given in equation 3.2.

$$BL_p = -\sum_i \{t_{pj} \log(y_{pj}) + (1 - t_{pj}) \log(1 - y_{pj})\} \quad (3.2)$$

where

$t_p = [t_{p1}, t_{p2}, \dots, t_{pm}]^T = \{t_1, t_2, t_3, \dots, t_p\}$  is the vector of desired target classes

$y_p = [y_{p1}, y_{p2}, \dots, y_{pm}]^T$  is output of the network.

### 3.7 Determine Network Parameters and Formulation of Modified Cost Function

Network parameters that will be used for two term backpropagation are learning rate, momentum term and proportional term. The value for learning rate is 1.0 and momentum factor is 0.75. The proportional term takes 0.95 as its initial value. This term is proportional to  $e(W(k))$  which represents the difference between the output and the target at each iteration. The cost function that will be used is Modified Cost function (MM):

$$mm = \sum_K \rho_k \quad (3.3)$$

with

$$\rho_k = \frac{E_k^2}{2a_k(1-a_k^2)} \quad (3.4)$$

where

$$E_k = t_k - a_k,$$

and ,

$E_k$  error at output unit  $k$ ,

$t_k$  target value of output unit  $k$ ,

$a_k$  an activation of unit  $k$ .

### 3.8 Determine Network Parameters and Formulation of Improved Cost Function

The assigned value for learning rate is 1.0, momentum factor is 0.75 and proportional term is 0.95. The cost function that will be used is Improved Cost Function (IC) as given in equation 3.5.

$$IC = E_A + E_B = \frac{1}{2} \sum_{P=1}^P \sum_j (t_{pj} - o_{pj})^2 + \frac{1}{2} \sum_{P=1}^P \left( \sum_j (t_{pj} - o_{pj})^2 \right) * \left( \sum_j (y_{pj} - 0.5)^2 \right) \quad (3.5)$$

### 3.9 Training and Testing Three Term BP with Various Cost Function

Various steps will be involved in implementing Three Term Backpropagation, which are initialization, activation, weight training and iteration as discussed below. These steps are the same for various cost function MSE, BL, MM and IC that will be used in this study.

First, the network architecture is structured, and some parameters are initialized with random numbers while others are assigned to a constant value. Weights are initialized randomly but learning rate, momentum factor and proportional term is assigned constant values. Stopping criteria for the training is according to K+10 or K+100 Increment Rule epochs. For example, if K+10 increment rule is used then, the stopping criteria will range from epoch number 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. Each input parameter is applied to input node and the network will feed forward from input layer to hidden layer, then from hidden layer to output layer. The output will be using sigmoid activation function. The formulations are given as:

$$\begin{aligned}
 a &= output = f(net) \\
 f(net) &= \frac{1}{(1 + e^{-net})} \\
 net &= \sum W a + \theta
 \end{aligned} \tag{3.6}$$

where,

For output between input layer (i) and hidden layer (j) given as:

$W$  is the weight connected between node i and j,

$\theta$  is the bias of node j,

$a$  is the output of node i.

For output between hidden layer(j) and output layer(k) given as:

$W$  is the weight connected between node  $j$  and  $k$ ,  
 $\theta$  is the bias of node  $k$ ,  
 $a$  is the output of node  $j$ .

Subsequent to the feed forward network is calculation of cost function (MSE or BL or MM or IC) that will be used in the weight adaptation during backpropagation. The network will be propagated backward from the output layer to hidden layer of the network and weights adjustment is adapted as in equation 3.7. This weight adaptation algorithm which involve third term, proportional term will be fitted to the network to backpropagate:

$$\Delta W_{kj}(t+1) = -\alpha \delta_k a_j + \beta \Delta W_{kj}(t) + \gamma e(W_{kj}(t)) \quad (3.7)$$

with,

$$\delta_k = -\frac{(E + \rho(1 - 3a_k^2))}{1 + a_k},$$

where,

$\alpha$  is Learning Rate,  
 $\delta_k$  is error signal at output layer,  
 $a_j$  is output at hidden layer(j),  
 $\beta$  is Momentum term,  
 $\Delta W_{kj}(t)$  is Previous Weight Change,  
 $\gamma$  is Proportional term,  
 $e(W_{kj}(t))$  proportional to  $e_s$

For batch learning,  $e(W_{kj}(t))$  can be written as below:

$$e(W(t)) = [e_s e_s \dots e_s]^\tau \quad (3.8)$$

where,  $e$  is of appropriate dimension of  $\tau$ , and represent the difference between the output and the target at each iteration.

$$e_s = [t_k - a_k], \quad (3.9)$$

And new weights of  $W_{kj}$  is shown below:

$$W_{kj}(t+1) = W_{kj}(t) + \Delta W_{kj}(t+1) \quad (3.10)$$

For error changes from hidden layer to input layer is calculated as shown in equation 3.11.

$$\Delta W_{ji}(t+1) = -\alpha \delta_j a_i + \beta \Delta W_{ji}(t) + \gamma e(W_{ji}(t)) \quad (3.11)$$

with,

$$\delta_j = \sum \delta_k w_j f'(a_j)$$

where,

$$f'(a_j) = a_j(1 - a_j)$$

This learning process will be conducted iteratively by presenting training data to input layer of the network. The network will be trained to adjust the weight according to the cost function until it meets the stopping criteria. In this study, the stopping criterion is number of epoch. Subsequent to training process, the generalization process will

begin. In this stage, the network will be fed with testing data to validate its error value, convergence time and accuracy.

### **3.10 Implementation of K+10 or K+100 Increment Rule**

To observe the performance of the datasets closely, new rule was proposed after some research. The new rule known as K+10 and K+100 Increment Rule is introduced and applied in the experiments. K+10 and K+100 is actually referred to the networks number of epochs. This would be used as a stopping criterion instead of minimum error value. For small datasets consider below 100 instances should use K+10 Increment Rule where the epoch will be increased by 10 for each simulation and the current error, time elapsed and current accuracy rate will be evaluated. Then the stopping criteria will be increased to 20 and so on.

For medium and large scale datasets consider 100 instances and above should use K+100 Increment Rule. Medium and large datasets requires more iteration to converge to the local minima. Thus, 100 epochs increment is applied to observe the trend. For this study the observation is limited to 1000 epoch to do standardize comparison to each datasets.

This rule is proposed to observe the patterns of the performance of the datasets when the epoch size is increased. As a result, we can derive the comparison and generalization based on the given patterns. Hence, comparatively performance can be validated at various epochs accordingly.

### **3.11 Summary**

This chapter discusses the methodology that will be used to investigate the efficiency of cost functions for Three Term Backpropagation. Firstly, the framework of the study is proposed. Secondly, the architecture of the network is discussed followed by the parameters and cost function involved. Then, it is followed by training mechanism of Three Term Backpropagation. Finally, the experiment and comparison analysis are explained to probe for better cost functions for Three Term BP.

## **CHAPTER 4**

### **EXPERIMENTAL RESULT AND ANALYSIS**

#### **4.1 Introduction**

This chapter discusses the results of experiments implementing on four different cost functions in Three Term BP. The experiments are implemented using Balloon, Cancer, Diabetes and Pendigits datasets of universal data. The four cost functions involved are MSE, BL, MM and IC. The results for each dataset are studied based on the convergence speed, error generated and accuracy.

Besides that, t-test was conducted to validate the results more preciously. The t-test was conducted based on the comparison criteria such as convergence time, error generated and accuracy. Then, the results was summarized and concluded.

## 4.2 Experiment Setup

Four programs have been developed which are Three Term BP with Mean Square Error cost function (MSE), Three Term BP with Bernoulli cost function (BL), Three Term BP with Modified cost function (MM) and Three Term BP with Improved cost function (IC). In this section the convergence behavior of the Three Term BP with MSE, BL, MM and IC cost function are compared using four example problems. They are Balloon, Cancer, Diabetes and Pendigits datasets.

In all cases the learning rate, momentum factor and proportional factor for the Three Term BP algorithm are selected such that the algorithm could converge faster resulting to a best-fit values. For the comparisons Three Term BP algorithm with various cost functions use the same values of learning rate, momentum factor and proportional factor. These values were taken from Zweri *et al.* (2003) for best practice value of the best success rate. Learning rate would be 1.0 while momentum factor is 0.75 and proportional factor would be 0.95. These values were identified after a 20 trials of different combination of learning rate, momentum factor and proportional factor for each datasets from various researchers best practice values.

For all the tests, sigmoid activation function is used for the hidden and output nodes of the network. The stopping criterion is number of epochs. The error threshold for a successful trial is 0.005. The initial values of the weights are selected randomly between  $-1$  to  $1$ . The network maps the input pattern into the corresponding output target value. For Balloon datasets, all cost functions are executed from 10 to 100 epochs while for other datasets such as Cancer, diabetes and Pendigits, the cost functions are executed from 100 to 1000 epochs.

### 4.3 Implementation of various cost functions

There are two major roles of cost function. First, cost function will be used to quantify the output error. This will provide a way for iteratively updating the network weights in order to minimize that error and thereby achieve more accurate output. Secondly, the derivative of the cost function, which is known as error signal, is one of the parameters in the weight update equation between output layer and hidden layer.

Many cost functions were introduced in the past as described in problem background chapter 2. Those cost functions were studied in detail. Specific four types of cost functions was examined in detail and was chosen to be used for the experimental comparisons of various cost function in Three Term BP. Those are MSE, BL, MM and IC.

MSE is the most common cost function, which is in use for the decades for Two Term BP. The Three Term BP also employed MSE as its cost function. Cost function as in equation 2.10 is employed in the program for each datasets being tested. Second cost function that was studied in detail and tested is BL. Even tough this cost function was introduced quite some time ago (1994) but its simplicity and clarity of the function makes it a right candidates for the comparison experiments. Equation (2.12) was used for the experiments.

Third, new MM proposed by Shamsuddin *et al* (2001) was taken into consideration as well for the comparison purpose. MM was proved to improve convergence speed of Two Term BP and now being tested in Three Term BP. Equation (2.14) is employed in the program for each datasets being tested. Forth, IC of Zhang *et al*

(2007) is executed. This cost function was introduced recently for element network. This cost function adds extra term into the existing MSE Cost function (Equation 2.34).

#### 4.4 Implementation of T-Test

The objective of t-test is to learn how to test for the significance of the differences between two mean for independent data. Experimental research often tests hypotheses about differences between two means to determine if a specific experimental intervention caused observed difference, or if the difference could reasonably be ascribed to chance. The *t*-test evaluates the statistical significance of the observed difference between means at a specific probability level.

Formula

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\frac{SD_1}{\sqrt{N_1}} + \frac{SD_2}{\sqrt{N_2}}}$$

or

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{SD_1^2}{N_1} + \frac{SD_2^2}{N_2}}} \quad (4.1)$$

where  $\bar{X}_1$  = mean 1;  $\bar{X}_2$  = mean 2;  $SD_1$  and  $SD_2$  = standard deviations for groups 1 and 2, respectively;  $N_1$  and  $N_2$  = the respective number of subjects for each group.

In this study, the experiments results are shown in section 4.6. The comparison based on experimental results in term of error value, convergence time and accuracy is

shown in section 4.7. Then t-test is conducted to further validate the results to determine if this experimental intervention caused observed difference in performance of cost functions. It is shown in section 4.8.

For example, mean and standard deviation (SD) is calculated from each error value, convergence time and accuracy for each cost function in each dataset. Once done, t-value is calculated in pair comparison such MSE and BL, MSE and MM, MSE and IC, BL and MM, BL and IC, MM and IC. Then significance level and the critical value of t is determined meanwhile the critical region is calculated based on critical value. If the critical value is 1.734 then the critical region would be  $t \geq + 1.734$  or  $t < -1.734$ . After that, the mean and standard deviation is exploited to the equation as in 4.1 to produce the t-value. This t-value is then will be compared to critical region value. If the t-value is within the critical region value then there is no significance difference between those two cost functions. If the critical region value is beyond the critical value then there is significance difference between two cost function and this means sufficient evidence exists to support the original claim to determine the better cost function.

#### **4.5 Analysis of Comparison Parameters**

Experimental comparisons of various cost functions in Three Term BP include parameters such as epoch size, network error, and convergence time (execution time) and accuracy.

### 4.5.1 Epoch Size

Epoch is referred to number of iteration. This parameter is essential for comparison of performance trend accordingly. The trend could be noticed by the parameter of epoch. For example when the epoch size is increased, clear performance changes can be seen for each datasets. This parameter was used in papers such as Kandil *et al.* (1993), Chow *et al.* (1994), Fukuoka *et al.* (1998), Matsuoka Kiyotoshi and Yi Jianqiang (2000), Abid, Fnaiech, and Najim (2001), Lee, Chen and Huang (2001), In-Cheol Kim and Sung-II Chien (2002), Zweiri *et al.* (2002), Yu and Liu (2002), Mandischer (2002), Zweiri *et al.* (2003), Jiang *et al.* (2003), Bi *et al.* (2004), Wang *et al.* (2004), Pernia-Espinoza *et al.* (2005), Rimer and Martinez (2006) and Guijarro *et al.* (2007). For this study, epochs will be used as a stopping criterion. The epochs basically will not play a role as a comparison criterion but would be a benchmark for the stopping values. For example, at epoch 10, the error value, convergence time and the accuracy is noted down for analysis accordingly.

### 4.5.2 Network Error

Network Error is referred to the current error produced by the network during training. Sometimes, this error could be a stopping criterion, but in this study the error at the particular epoch is purposely studied to conduct the analysis. This parameter was used as a comparison criterion in papers such as Chow *et al.* (1994), Fukuoka *et al.* (1998), Falas and Stafylopatis (1999), Mandischer (2002), Wang *et al.* (2004) and Rimer and Martinez (2006). Eventough error does not serve as a stopping criteria but the error threshold is set to 0.0050. This is to help validating cost functions in more details.

### 4.5.3 Convergence Time

Convergence time is concerned to the time elapsed in milliseconds to complete the number of epochs (iteration) specified. This execution time is differing according to the complexity of input, hidden and output node. If the structure is huge and complex, more time is needed just to complete a small number of epochs. Lee, Chen and Huang (2001), Abid, Fnaiech and Najim (2001), In-Cheol Kim and Sung-Il Chien (2002) and Ng *et al.* (2003) papers used this as a comparison criterion.

### 4.5.4 Accuracy

Accuracy is referred to the number of correct output compared to the whole test data. Similarly it could be defined as the rate of successful convergence within an iteration limit. After the network learn through training data, and then the network is tested for the result accuracy. This will give reliability of the particular cost function in real environment. If a network could give high percentage of accuracy, then it should considered as a good network despite of other criteria. This parameter was used as a comparison criterion in papers such as Falas and Stafylopatis (1999), Zweiri *et al.* (2002), Yu and Liu (2002), Mandischer (2002), Zweiri *et al.* (2003), Wang *et al.* (2004) and Rimer and Martinez (2006).

## 4.6 Experimental Results

The experiments of Three Term BP with four cost functions are conducted. The results are as shown in the tables below.

### 4.6.1 Result of Three Terms BP for Balloon Dataset

First, Balloon dataset problem is investigated. This is a popular universal datasets of classification problems. The network architecture used for this problem consisted of four input units, two hidden units and one output unit. Total number of instances is 16.

All the tests for Balloon dataset have been grouped into ten; Group I until Group X. The epochs increase across the groups. These tests are carried out using constant learning rate 1.0, momentum factor 0.75 and proportional factor 0.95 through the simulation and the initial weights are selected randomly in the range of  $-1$  to  $1$ . These networks use “K+10” Increment Rule where the epochs range from 10 to 100. The network is tested on 50 trials where each group is tested 5 times and the best result is shown in the Table 4.1, 4.2, 4.3 and 4.4. The comparative results for the error (current error generated by network during training), time (convergence time of the network) and accuracy (percentage of test result) are summarized in Table 4.1 through Table 4.4, and the results are presented according to number of groups.

#### 4.6.1.1 Result of Three Term BP with MSE Cost Function for Balloon Dataset

Table 4.1 shows that the Three Term BP performed well with MSE cost function for Balloon dataset. The graph indicates that the error generated at epoch 10 is 0.2274 and it degrades to 0.0279 at epoch 100. This result shows that MSE cost function still requires more epochs to give an error threshold of 0.005 during training data. As Figure 4.2 shows, this cost function manages to help Balloon datasets to complete 100 epochs in 110 milliseconds. This is known as convergence speed. One of the reasons could be that Balloon dataset is considered a small datasets consists of 16 instances. Furthermore, the percentage of correct classification of the testing data remains 75% throughout the 100 epochs as illustrated in Figure 4.3. This percentage shows a good performance of the MSE cost function. Even though the accuracy is high, but it shows that the number epoch does not help the network to improve its accuracy for test data of balloon dataset.

Table 4.1 Testing results of Balloon dataset with MSE cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	10	20	30	40	50	60	70	80	90	100
Error	0.2274	0.1261	0.0875	0.0670	0.0543	0.0457	0.0394	0.0347	0.0309	0.0279
Time(ms)	50	50	60	50	60	110	110	110	110	110
Accuracy	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result): Percentage

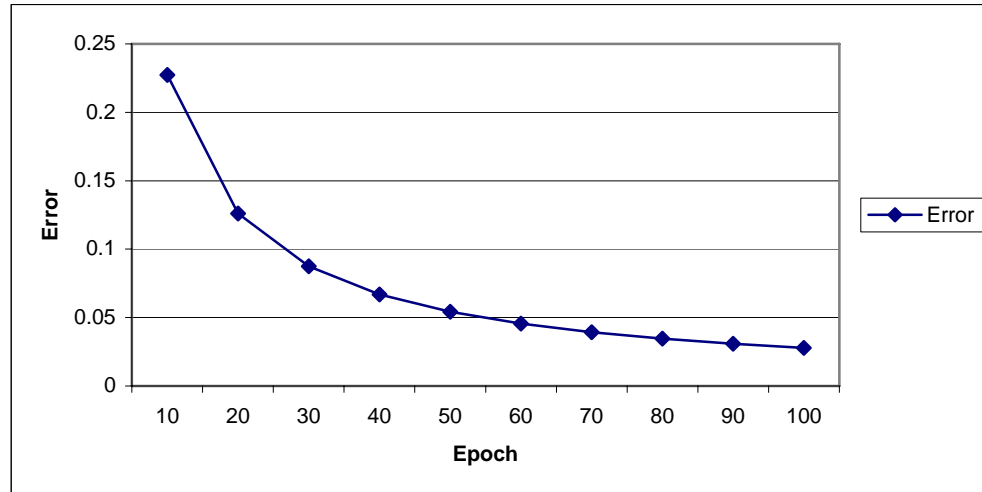


Figure 4.1: Error Convergence of Balloon Dataset with MSE Cost Function

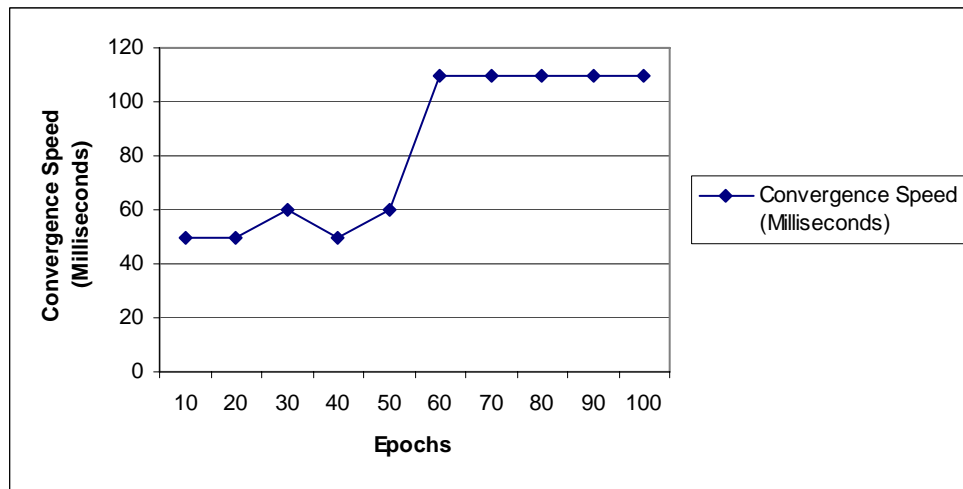


Figure 4.2: Convergence Time of Balloon dataset with MSE cost function

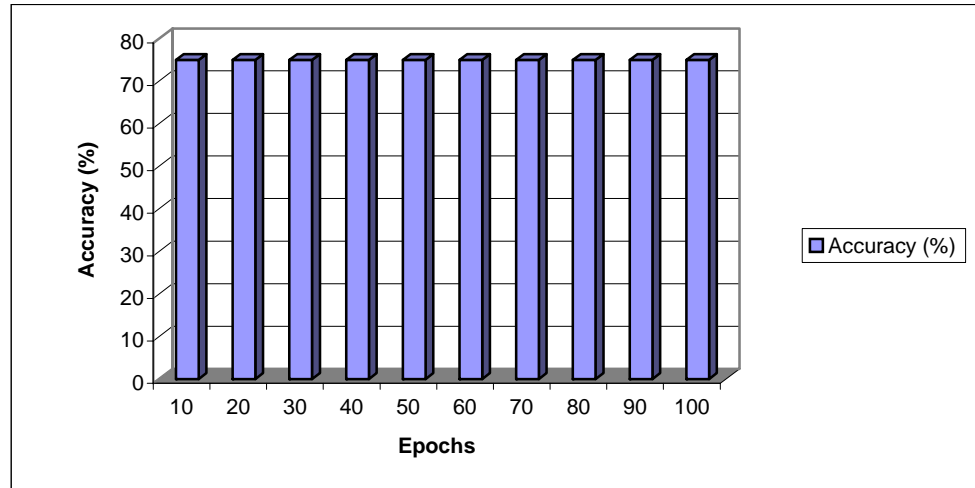


Figure 4.3: Accuracy Percentage (%) of Balloon dataset with MSE cost function

#### 4.6.1.2 Result of Three Term BP with BL Cost Function for Balloon Dataset

Table 4.2 shows that the Three Term BP with the BL Cost Function performed very well for Balloon dataset. These groups of test are carried out by epoch. The initial error for 10 epochs is 0.0419 while the error is 0.0058 for 100 epochs. As depicts in Figure 4.4, this error range is very small and promises a good performance of BL cost function with Balloon datasets, and approaching the error thresholds of 0.005. Another interesting criterion is to look at the speed. Figure 4.5 shows that the networks convergence time somewhat static for first 50 epochs and then for 70 to 100 epochs. For the first 50 epochs the Balloon dataset with BL cost function execution time remained as 50 milliseconds while for the 60 epochs the execution time is 60 milliseconds and for the following 30 more epochs (until 100 epochs) the execution time remained at 110 milliseconds. This shows that number of iteration does not have much impact on the execution time. Simultaneously, it does not increase with the number of epochs as in other cost functions. Finally, Balloon Dataset with BL cost function delivered 75%

accuracy at each epoch as shown in Figure 4.6. Overall, BL cost function gives the best performance compared to other cost functions for Balloon dataset.

Table 4.2 Testing results of Balloon datasets with BL cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	10	20	30	40	50	60	70	80	90	100
Error	0.0419	0.0251	0.0178	0.0138	0.0112	0.0094	0.0081	0.0072	0.0064	0.0058
Time(ms)	50	50	50	50	50	60	110	110	110	110
Accuracy	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

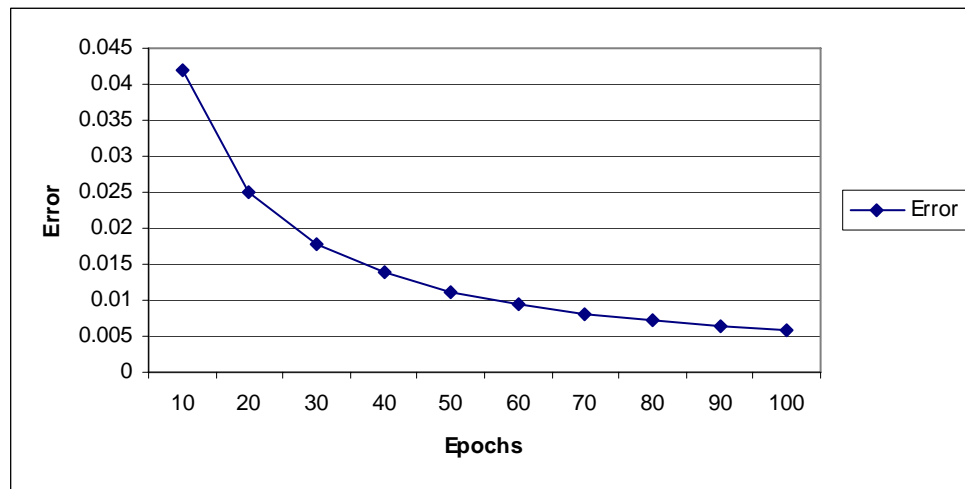


Figure 4.4: Error Convergence of Balloon Dataset with BL Cost Function

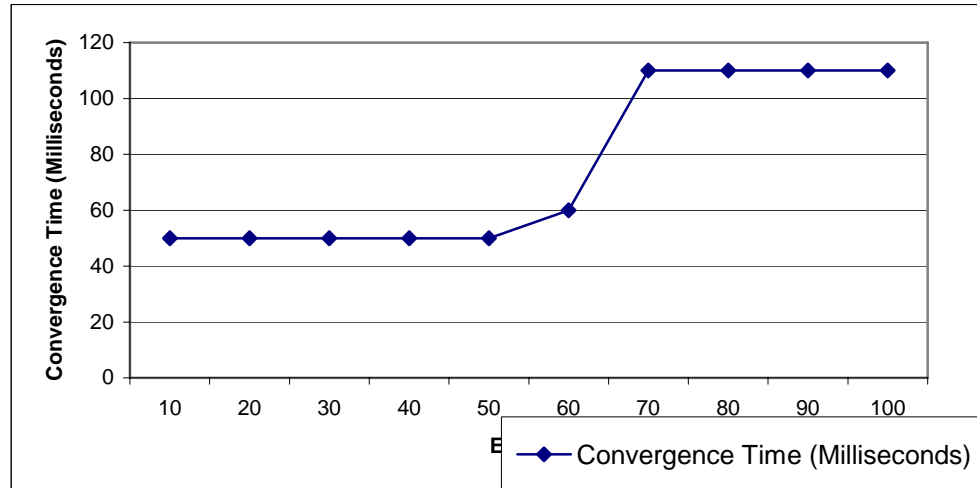


Figure 4.5: Convergence Time of Balloon Dataset with BL cost function

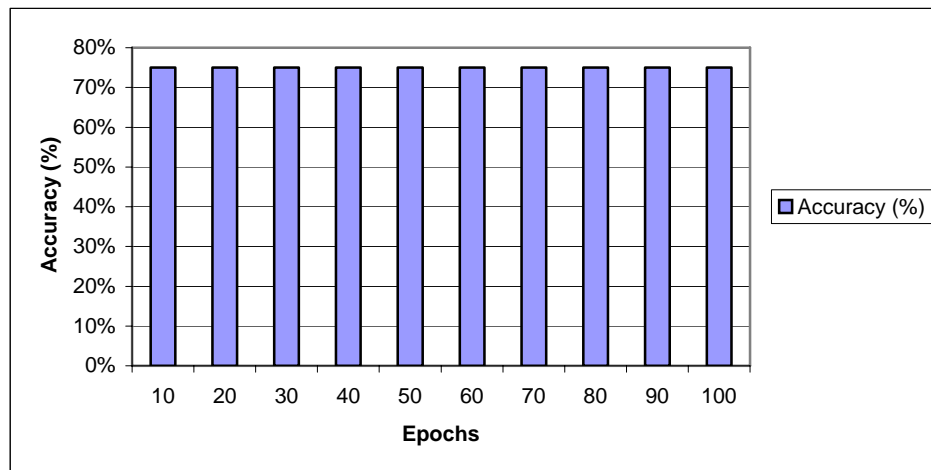


Figure 4.6: Accuracy Percentage (%) of Balloon Datasets with BL cost function

#### 4.6.1.3 Result of Three Term BP with MM Cost Function for Balloon Dataset

Table 4.3 depicts that the performance of Three Term BP with the MM Cost Function for Balloon dataset. Figure 4.7 indicates that the error generated at epoch 10 is 0.0538 and it degrades to 0.0061 at epoch 100 for Three Term BP with MM Cost

Function. This result shows that MM cost function doesn't reach the error threshold of 0.005 at even 100 epochs. As Figure 4.8 shows, this cost function completes 100 epochs in 110 milliseconds too as MSE cost function. Furthermore, the percentage of correct classification for the testing data remains 75% throughout 100 epochs as illustrated in Figure 4.9. This percentage shows good performance of the MM cost function. The accuracy is high; however, the number of epoch does not contribute to the improvement of balloon dataset.

Table 4.3 Testing results of Balloon datasets with MM cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	10	20	30	40	50	60	70	80	90	100
Error	0.0538	0.0296	0.0203	0.0153	0.0123	0.0103	0.0088	0.0077	0.0068	0.0061
Time(ms)	60	50	60	60	50	50	110	110	110	110
Accuracy	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

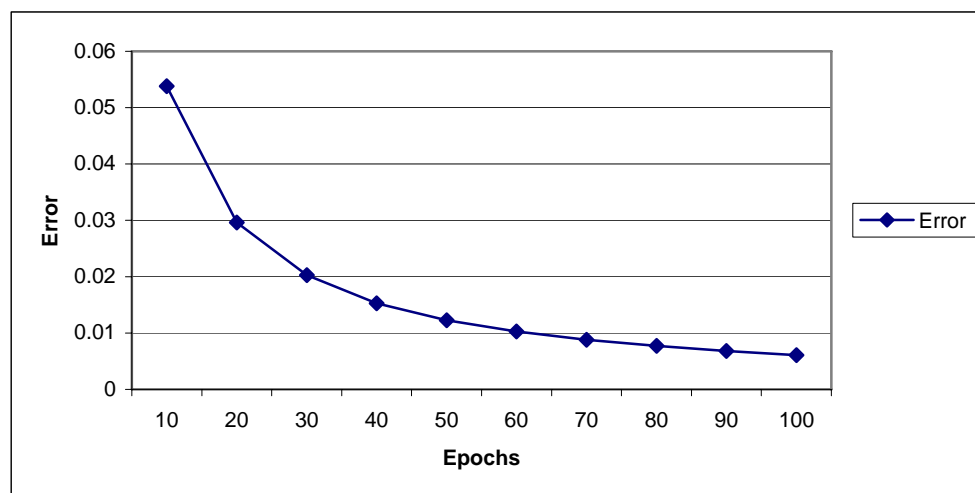


Figure 4.7: Error Convergence of Balloon Dataset with MM Cost Function

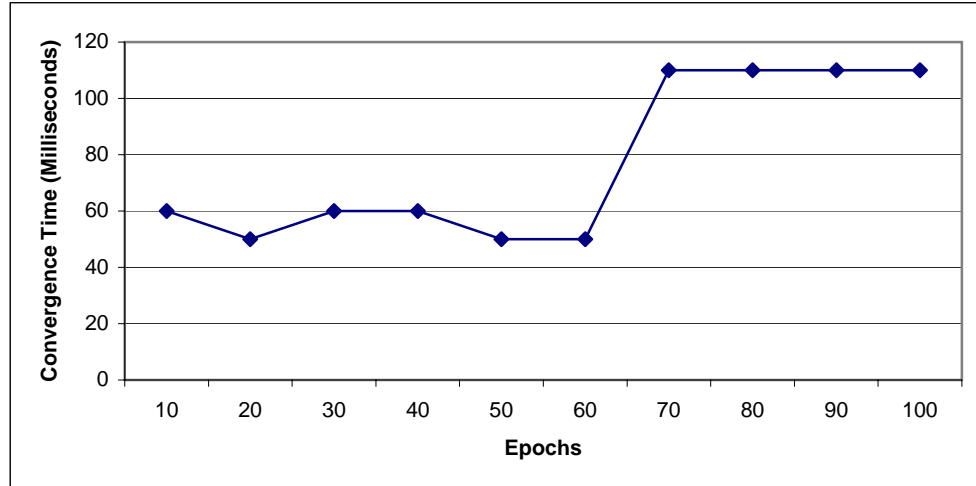


Figure 4.8: Convergence Time of Balloon Dataset with MM cost function

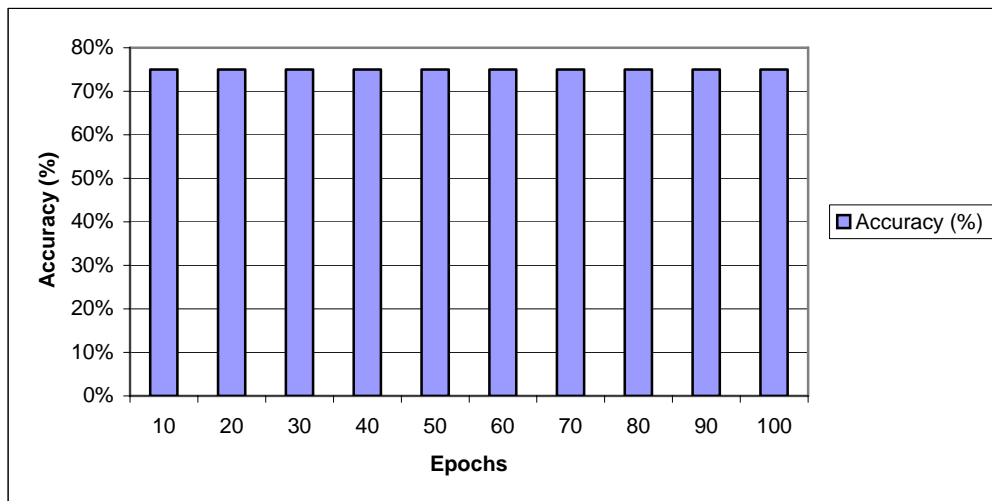


Figure 4.9: Accuracy Percentage (%) of Balloon Dataset with MM cost function

#### 4.6.1.4 Result of Three Term BP with IC Cost Function for Balloon Dataset

Table 4.4 shows that the performance of Three Term BP with the IC cost function for Balloon dataset. The simulation is tested for 10 trials for 100 epochs. The testing results indicate that the generated errors ranging between 0.2345 and 0.0282 (Figure 4.10). These error values of IC cost function are considered large compared to other cost functions. At epoch 100, the network could only manage to achieve minimum error of 0.0282. This value is far from the threshold error value. Contrastively, the execution time of the Balloon datasets with IC cost function is among the minimum as shown in the Figure 4.11. The average execution time for Balloon with IC cost function is 76 milliseconds. This low execution time shows that this cost function is quite simple to be implemented. Alternatively, IC cost function still maintaining the same accuracy rate as others (Figure 4.12).

Table 4.4 Testing results of Balloon datasets with IC cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	10	20	30	40	50	60	70	80	90	100
Error	0.2345	0.1291	0.0894	0.0683	0.0553	0.0469	0.0410	0.0352	0.0313	0.0282
Time(ms)	60	50	50	50	50	60	110	110	110	110
Accuracy	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

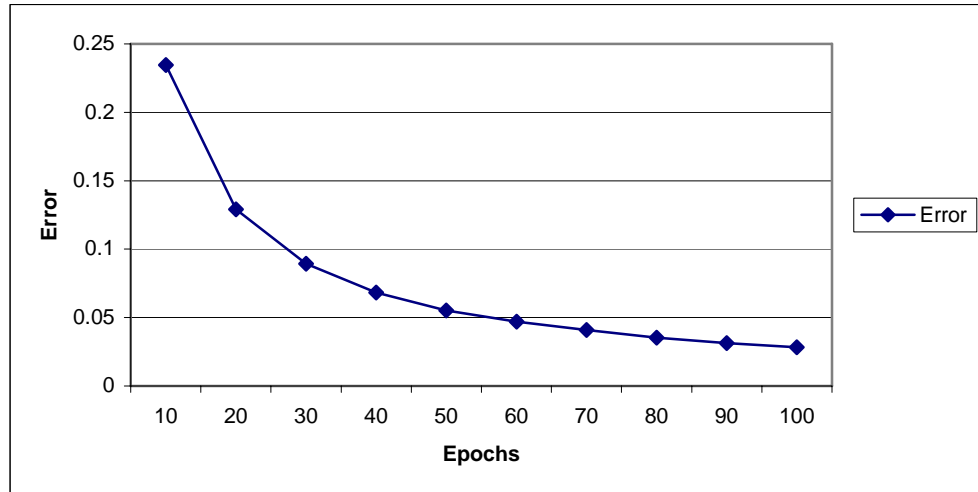


Figure 4.10: Error Convergence of Balloon Dataset with IC Cost Function

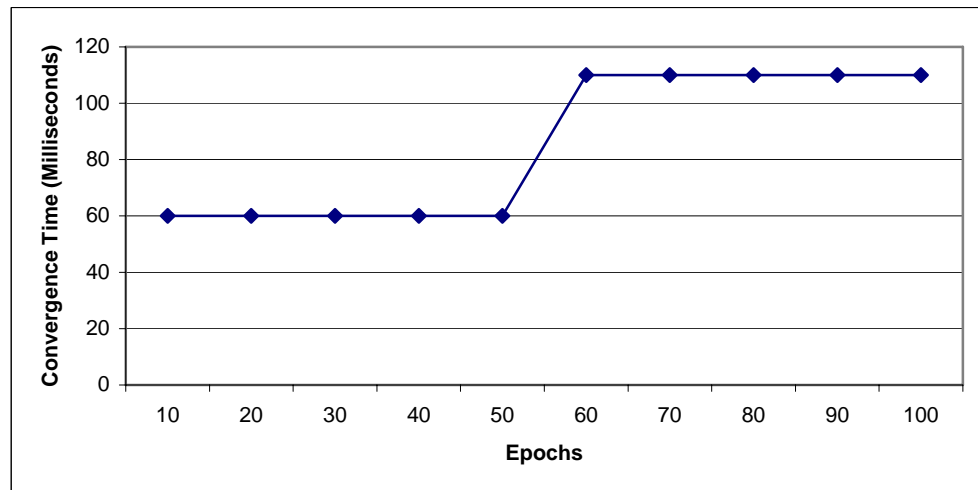


Figure 4.11: Convergence Time of Balloon Datasets with IC cost function

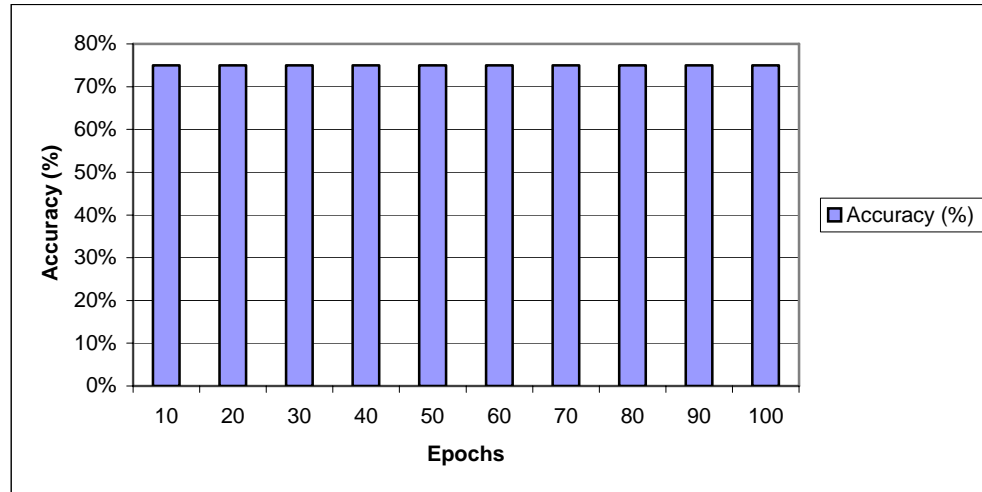


Figure 4.12: Accuracy Percentage (%) of Balloon Dataset with IC cost function

#### 4.6.2 Result of Three Terms BP for Cancer Dataset

The Cancer dataset was presented to the network which consists of nine input units, three hidden units and one output unit. Total number of instances is 500. From these 500 instances, 400 were used for training and 100 were used for the testing purposes. The original datasets of cancer database is 600 instances. When it was presented to network, the network could not converge at all even after a 50 trials. So, the datasets was filtered to remove anomalies data. Finally, only 500 instances were chosen in the testing phase.

All the tests for Cancer datasets have been grouped into ten; Group I through Group X. The epoch increases across the groups. These tests are carried out using constant learning rate 1.0, momentum factor 0.75 and proportional factor 0.95 through the simulation and the initial weights are selected randomly in the range  $-1$  to  $1$ . These networks use “K+100” Increment Rule where the epochs are range from 100 to 1000.

This is because the cancer datasets is large and “K+10” rule is not sufficient to be able to produce good result. For example the error of cancer datasets is still very large at 100 epochs, and the accuracy rate was very low. The network needed more epochs to perform well. The network is tested on 50 trials where each group is tested 5 times and the best result is shown in the Table 4.5 through Table 4.8. The comparative results for the error (current error generated by network), time ( execution/ convergence time of the network ) and accuracy ( percentage of test result ) are summarized in Table 4.5 through Table 4.8.

#### **4.6.2.1 Result of Three Terms BP with MSE Cost Function for Cancer Dataset**

Table 4.5 shows Three Term BP performance with MSE cost function for Cancer dataset. Figure 4.13 indicates that the error generated at epoch 100 is 0.0500 and it degrades to 0.0050 at epoch 1000. This result shows that MSE cost function managed to achieve error threshold of 0.005. As Figure 4.14 shows, this cost function requires 2520 milliseconds to complete 1000 epoch. This is known as convergence speed. The 2520 are reasonable for 1000 epochs compared to the convergence speed of other cost function. In addition, the accuracy of the testing data is boosting as epoch increases, resulting from a range of 33% at 100 epoch to 83% at 1000 epoch (Figure 4.15). This indicates a good progress in the accuracy. Unlike balloon dataset, Cancer dataset has proven that the number of epoch has a significant effect on the performance of accuracy for Three Term BP with MSE cost function. The accuracy is high at 1000 epoch despite its lowest percentage compared to other cost function for Cancer dataset.

Table 4.5 Testing results of Cancer datasets with MSE cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.0500	0.0260	0.0170	0.0130	0.0100	0.0090	0.0070	0.0060	0.0060	0.0050
Time(ms)	1120	1340	1070	1260	1480	1590	1650	2000	2260	2520
Accuracy	33%	62%	71%	75%	78%	79%	81%	82%	82%	83%

## Indicator

Time (Convergence Time) : Milliseconds,

Accuracy (Test Result) : Percentage

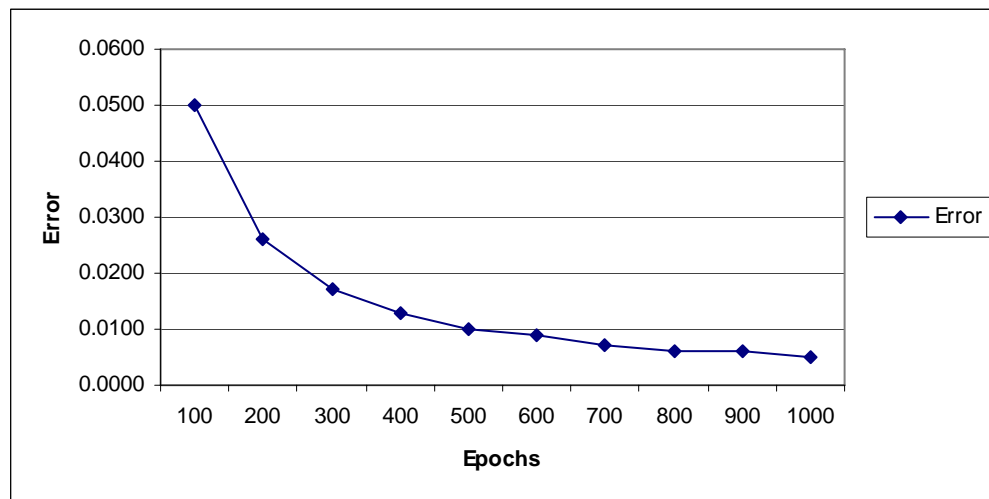


Figure 4.13: Error Convergence of Cancer Dataset with MSE Cost Function

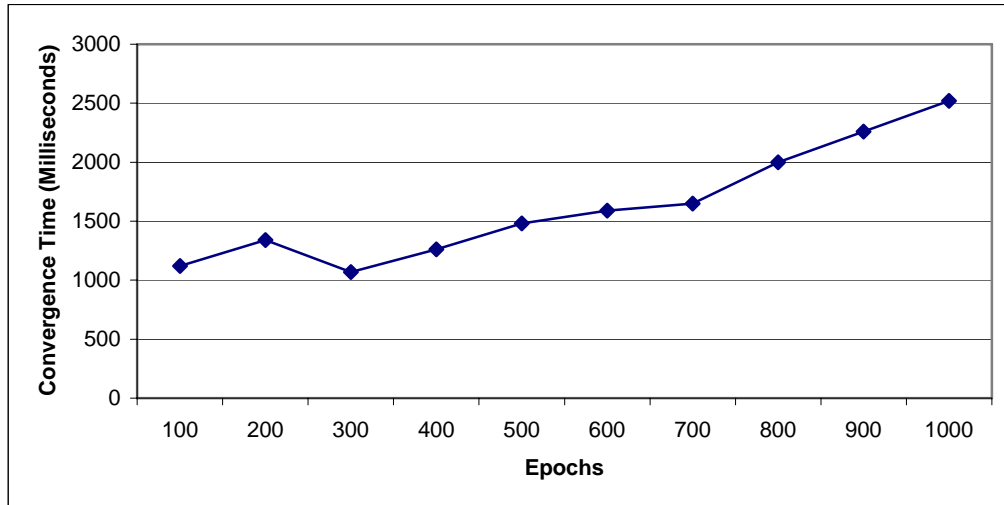


Figure 4.14: Convergence Time of Cancer Dataset with MSE cost function

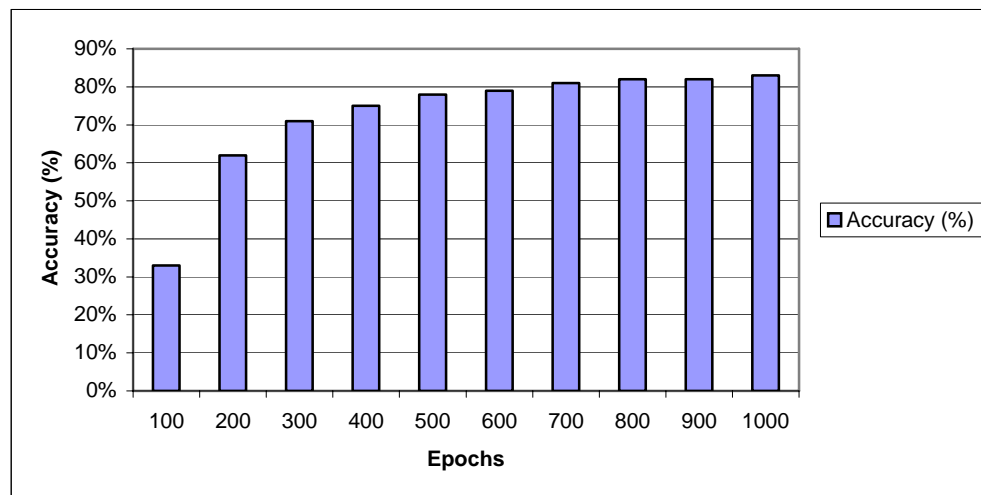


Figure 4.15: Accuracy Percentage (%) of Cancer Dataset with MSE cost function

#### 4.6.2.2 Result of Three Terms BP with BL Cost Function for Cancer Dataset

Table 4.6 shows Three Term BP performance with BL cost function for Cancer dataset. The comparison parameters showed a significance changes in this group of tests.

The Figure 4.16 shows the errors for each group. 100 epoch produced 0.3940 error value and decreases significantly to 0.0420 at 1000 epochs. This significance decrease shows that number of epoch also plays an important role in the performance of a cost function. Figure 4.17 depicts the convergence time from group I to group X. The convergence time of group I with 100 epochs is 440 milliseconds and increased dramatically to 3070 for 1000 epochs. The accuracy of 100 epochs is 49% and increases to 84% at 1000 epochs. Even though the accuracy rate is low, but the network is managed to reach 84% at 1000 epochs (Figure 4.18). Overall, the BL cost function has shown significant changes as the number of epoch increases, but the performance is quite low compared to other cost function.

Table 4.6 Testing result of Cancer datasets with BL cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.3940	0.2080	0.1400	0.1060	0.0850	0.0700	0.0600	0.0630	0.0470	0.0420
Time(ms)	440	700	960	1190	1470	1670	1930	2120	2380	3070
Accuracy	49%	68%	74%	78%	80%	81%	82%	83%	84%	84%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

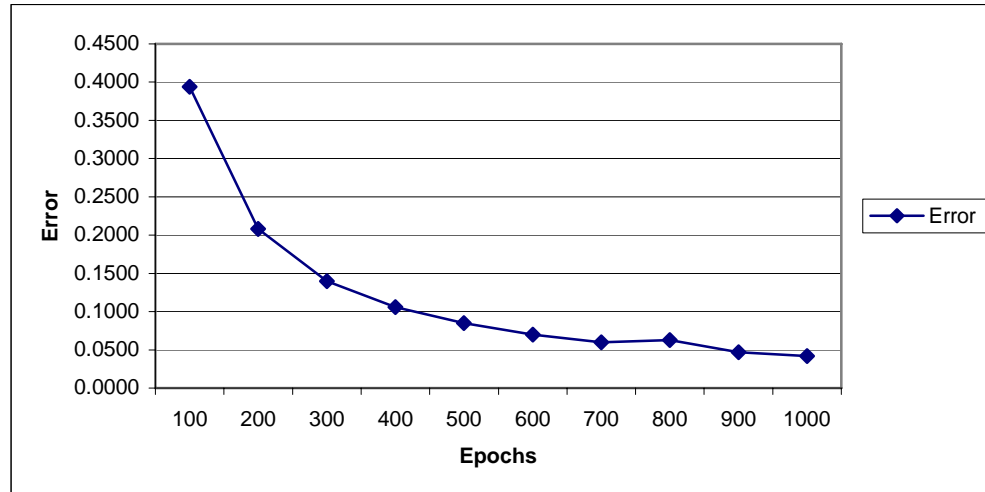


Figure 4.16: Error Convergence of Cancer Dataset with BL Cost Function

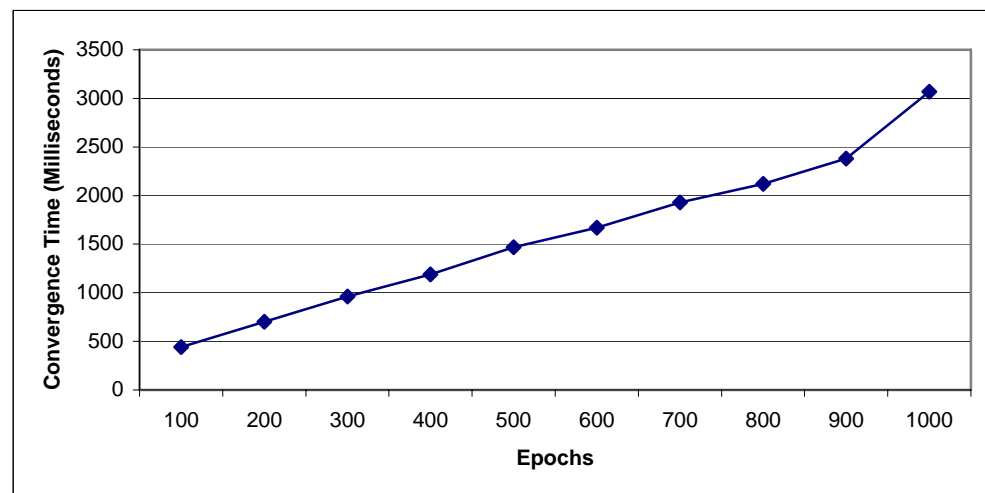


Figure 4.17: Convergence Time of Cancer Dataset with BL cost function

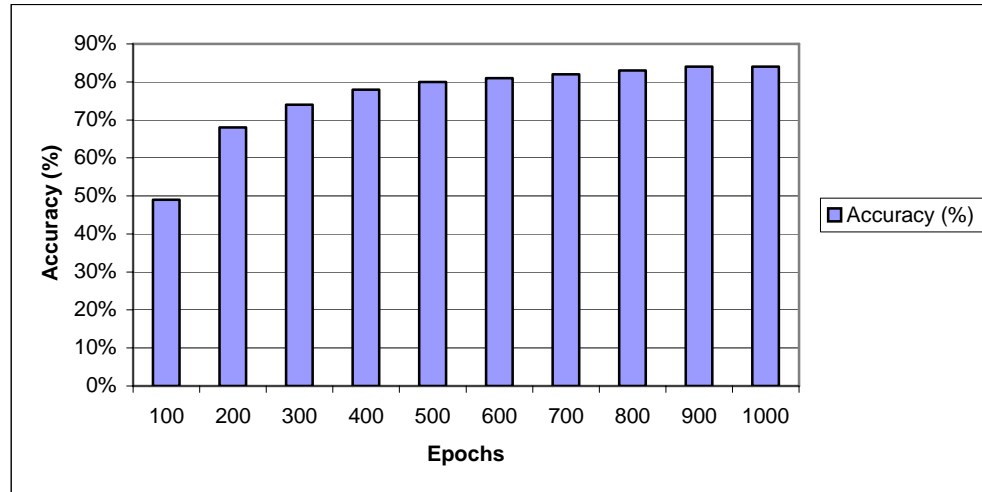


Figure 4.18: Accuracy Percentage (%) of Cancer Dataset with BL cost function

#### 4.6.2.3 Result of Three Terms BP with MM Cost Function for Cancer Dataset

Table 4.7 shows Three Term BP performance with MM Cost Function for Cancer dataset. It shows that MM cost function performed very well with Cancer dataset. It can be observed from Figure 4.19, the error is very low since the beginning of the group test. The error is 0.0140 for 100 epochs and decreased to 0.0010 for 1000 epochs. This is a good achievement. Even though it performs in term of error but the convergence time is 660 milliseconds for 100 epochs and gradually increasing to 3240 milliseconds for 1000 epochs as illustrated in Figure 4.20. This network can be considered as slow in convergence speed. MM cost function also produced high accuracy rate as depicted in Figure 4.21. The accuracy rate is high for cancer datasets. This rate is increasing to 86% for the 1000 epochs. In conclusion, MM cost function performed well in Cancer dataset.

Table 4.7 Testing results of Cancer datasets with MM cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.0140	0.0070	0.0050	0.0030	0.0030	0.0020	0.0020	0.0020	0.0010	0.0010
Time(ms)	660	910	1160	1410	1920	2170	2620	2670	2920	3240
Accuracy	74%	81%	83%	85%	85%	86%	86%	86%	87%	86%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

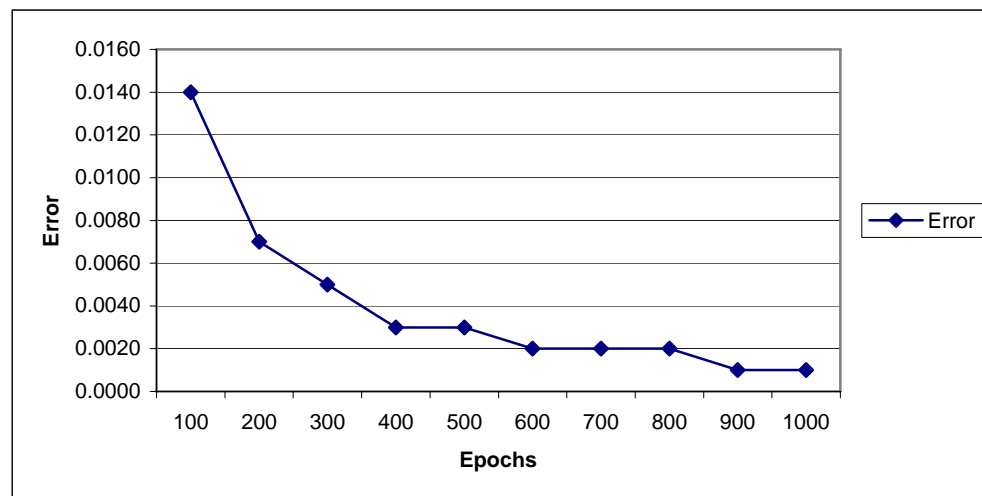


Figure 4.19: Error Convergence of Cancer Dataset with MM Cost Function

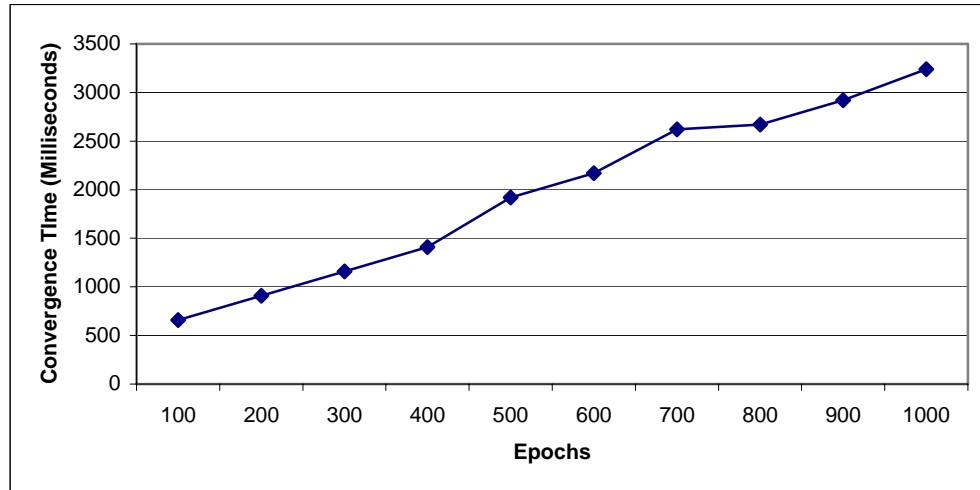


Figure 4.20: Convergence Time of Cancer Dataset with MM cost function

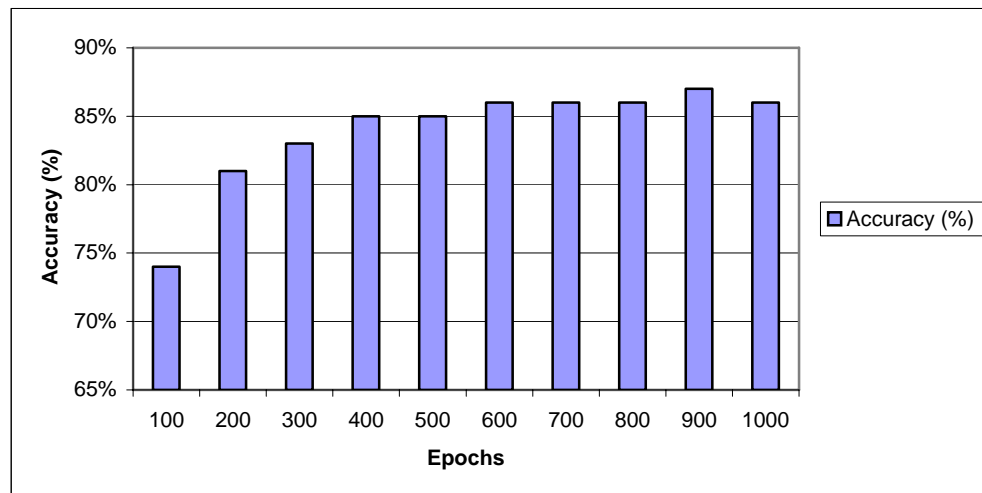


Figure 4.21: Accuracy Percentage (%) of Cancer Dataset with MM cost function

#### 4.6.2.4 Result of Three Terms BP with IC Cost Function for Cancer Dataset

Table 4.8 shows Three Term BP performance with IC cost function for Cancer dataset. The IC cost function achieved a balanced performance for Cancer dataset. The IC cost function's error values are illustrated in the Figure 4.22. The error value for 100 epochs is 0.0630 and decreased to 0.0070 at 700 epochs. Then it showed a slow decrease to 0.0050 for 1000 epochs. Besides that, Figure 4.23 shows the convergence time of the IC cost function. The convergence time for IC cost function is the best timing for Cancer datasets This is the shortest timing required by Cancer dataset to complete a 1000 epochs which is 2250 milliseconds. Furthermore, the accuracy rate of the IC cost function is illustrated in the Figure 4.24. This IC cost function performed poorly at 100 epochs where the accuracy rate was 25% only. But as the epoch increases, the accuracy rate also begins to increase to 85% at 1000 epochs.

Table 4.8 Testing results of Cancer datasets with IC cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.0630	0.0300	0.01780	0.0140	0.0110	0.0100	0.0070	0.0060	0.0060	0.0050
Time(ms)	380	560	710	890	1090	1270	1450	1600	1810	2250
Accuracy	25%	58%	70%	74%	77%	78%	81%	82%	83%	85%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

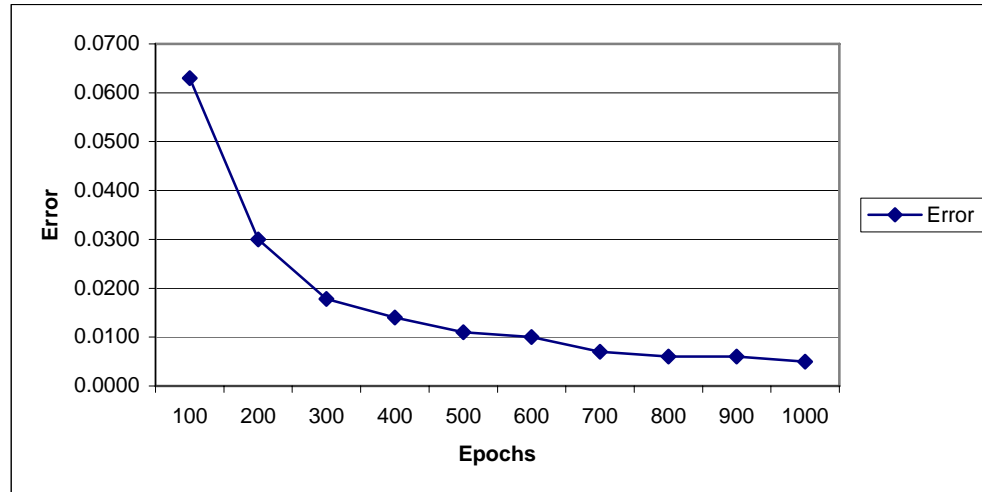


Figure 4.22: Error Convergence of Cancer Dataset with IC Cost Function

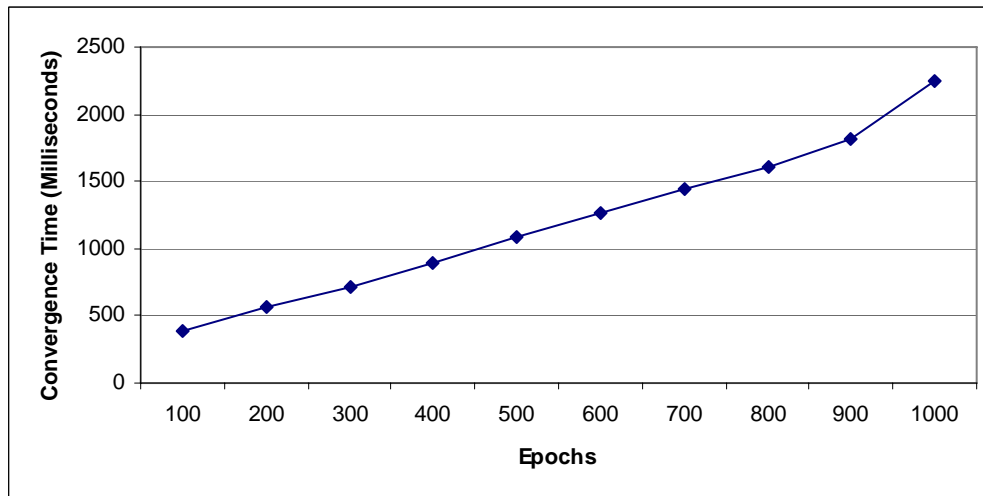


Figure 4.23: Convergence Time of Cancer Dataset with IC cost function

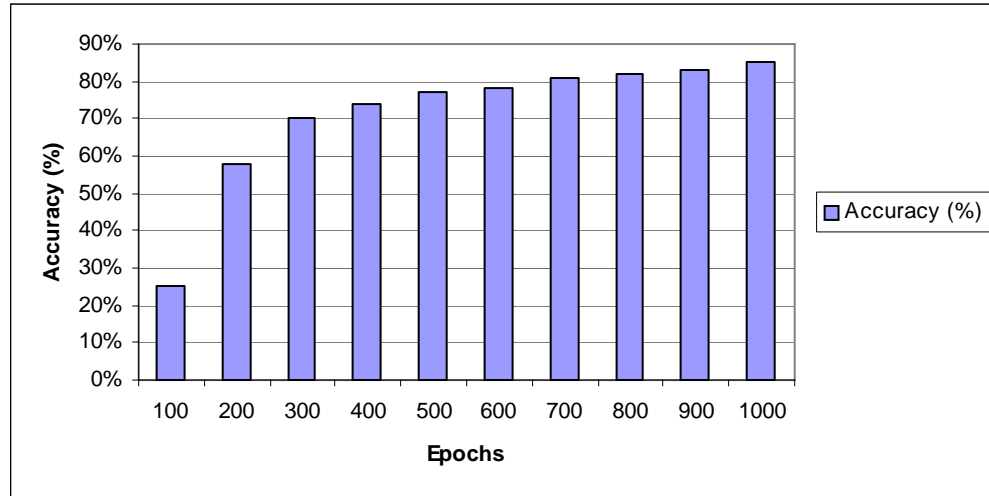


Figure 4.24: Accuracy Percentage (%) of Cancer Dataset with IC cost function

#### 4.6.3 Result of Three Terms BP for Diabetes Dataset

Diabetes problem were examined and to be presented to the Three Term BP network. The Diabetes dataset consists of eight input units, two hidden units and one output unit with total number of instances of 768.

All testing for Diabetes dataset are grouped into ten: group I through group X. These testing datasets are carried out using constant learning rate 1.0, momentum factor 0.75 and proportional factor 0.95 through the simulation. The initial weights are selected randomly in the range of  $-1$  to  $1$ . These networks use “K+100 Increment Rule” where the epochs size are ranging from 100 to 1000. This is because the Diabetes datasets is large and “K+10 Increment Rule” is insufficient to produce good result. As we can see in the Table 4.9 through Table 4.12, the error of Diabetes dataset is still very large at 100 epochs and the accuracy rate was very low. The network needed more epochs to perform well. Therefore, the network is tested on 50 trials where each group is tested 5 times and

the best result is shown in Table 4.9 through Table 4.12. The comparative results for the error (current error generated by network ), time ( execution / convergence time of the network ) and accuracy ( percentage of test result) are summarized in Table 4.9 through 4.12.

#### 4.6.3.1 Result of Three Term BP with MSE Cost function for Diabetes dataset

Table 4.9 shows Three Term BP performance with MSE cost function for Diabetes dataset. The results are presented according to groups and epoch limitation from 100 to 1000. Figure 4.25 indicates that the error generated at epoch 100 is 0.0560 and it degrades to 0.0060 at epoch 1000. MSE cost function for Diabetes dataset fail to attain the error threshold of 0.0050. In addition, as illustrates in Figure 4.26, the convergence speed of MSE cost function is 320 milliseconds for the completion of 100 epochs. The duration went up to 3090 for 1000 epochs. This would be nearly 10 times higher and this duration is among the longest compared to other cost functions. Hence, it would be the least performed cost function for Diabetes dataset if we compared in terms of convergence speed. The accuracy level of MSE cost function is in the range of 17% to 34 % (Figure 4.27). This range is quite small indeed. Even after 1000 epoch, the network could only manage to accomplish 34% accuracy. Hence, the network may require more epochs for better performance.

Table 4.9 Testing results of Diabetes dataset with MSE cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.0560	0.0280	0.0190	0.0140	0.0110	0.0090	0.0080	0.0070	0.0060	0.0060
Time(ms)	320	540	740	960	1430	1760	1940	2270	2620	3090
Accuracy	17%	11%	21%	26%	29%	31%	32%	33%	34%	34%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

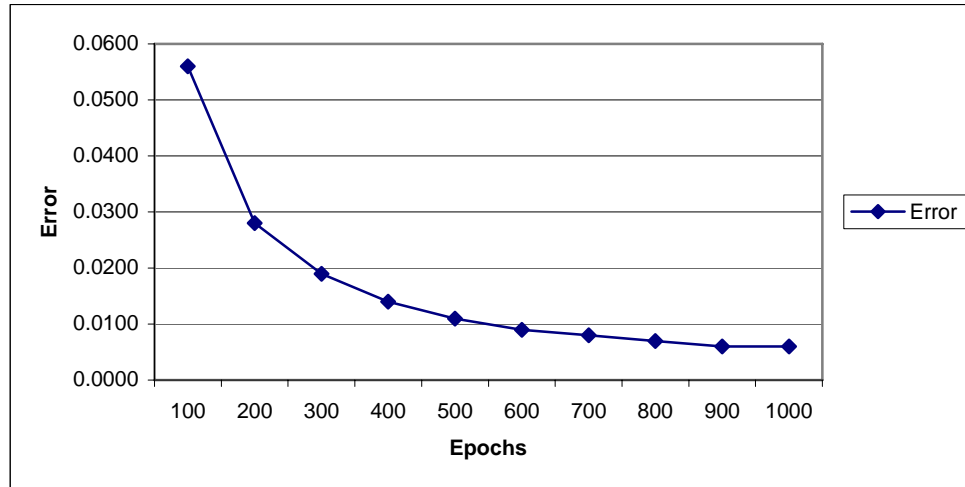


Figure 4.25: Error Convergence of Diabetes Dataset with MSE Cost Function

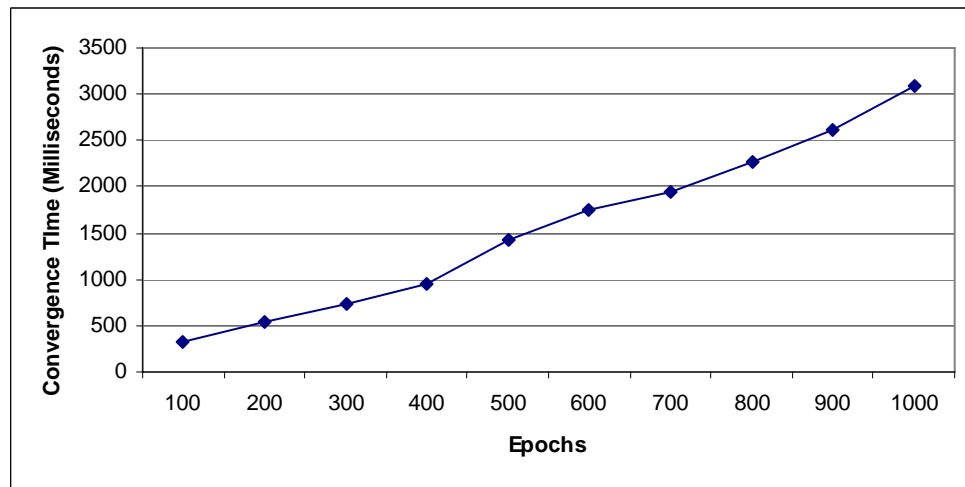


Figure 4.26: Convergence Time of Diabetes Dataset with MSE Cost Function

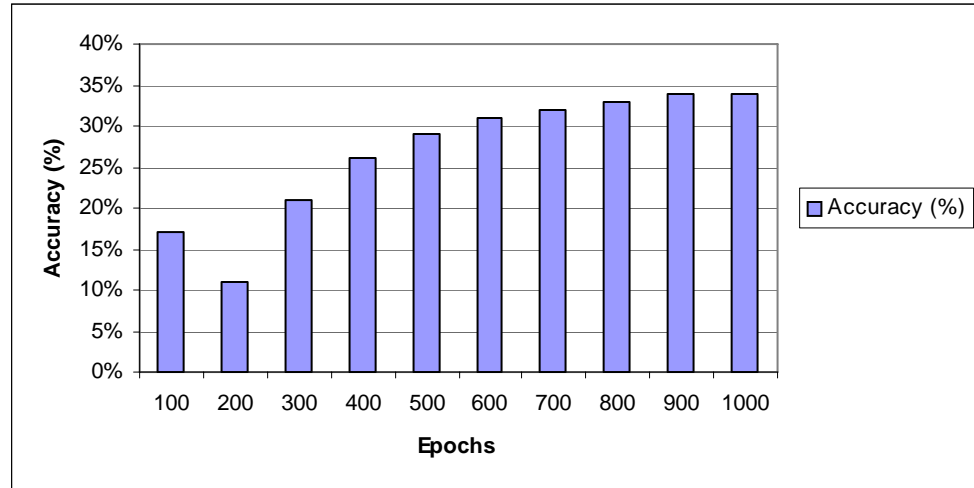


Figure 4.27: Accuracy Percentage (%) of Diabetes Dataset with MSE Cost Function

#### 4.6.3.2 Result of Three Terms BP with BL Cost Function for Diabetes Dataset

Table 4.10 shows Three Term BP performance with BL cost function for Diabetes dataset. BL cost function performed an average level for the Diabetes dataset. The Figure 4.28 shows the errors produced by network according to group. The error value for 100 epochs would be 0.0544 and decrease slowly to 0.0112 for 500 epochs and finally reaches 0.0056 for 1000 epochs. Besides that the Figure 4.29 is showing the convergence time for the Three Term BP network with BL cost function. The convergence time for 100 epochs is 440 milliseconds and increased to 2690 for 1000 epochs. The convergence time for 100 epochs for BL cost function of Diabetes dataset is same as 100 epochs for BL cost function for Cancer dataset. The accuracy rate of the network can be observed from Figure 4.30. The accuracy of 100 epochs is 14% only and increasing slightly to 34% at 1000 epoch with low accuracy. Overall, BL cost function with Diabetes performed low accuracy.

Table 4.10 Testing results of Diabetes dataset with BL Cost Function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.0544	0.0278	0.0186	0.0140	0.0112	0.0093	0.0080	0.0070	0.0062	0.0056
Time(ms)	440	680	920	1160	1650	1910	2170	2430	2690	2690
Accuracy	14%	13%	22%	26%	29%	31%	32%	33%	34%	34%

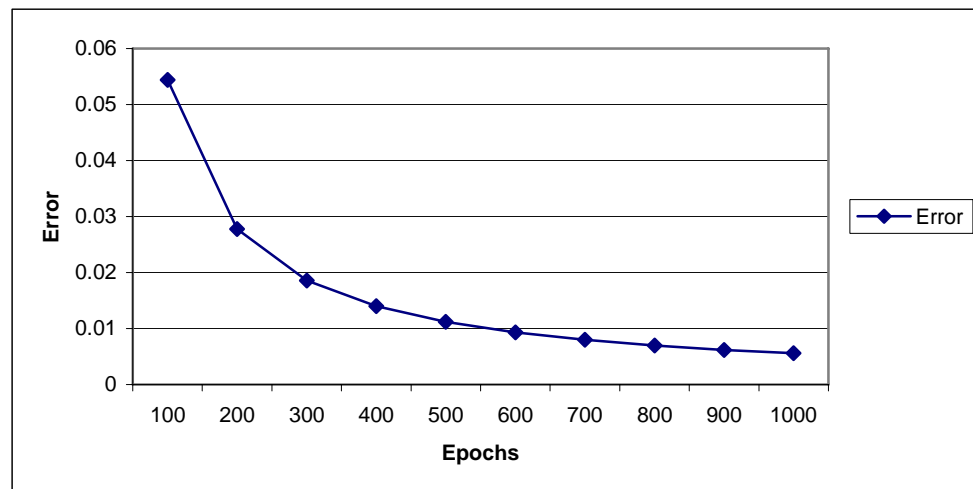


Figure 4.28: Error Convergence of Diabetes Dataset with BL Cost Function

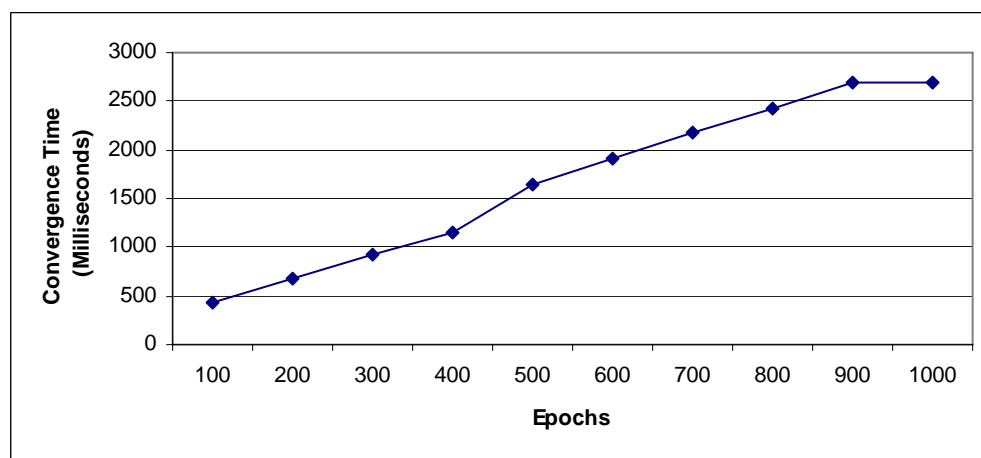


Figure 4.29: Convergence Time of Diabetes Dataset with BL cost function

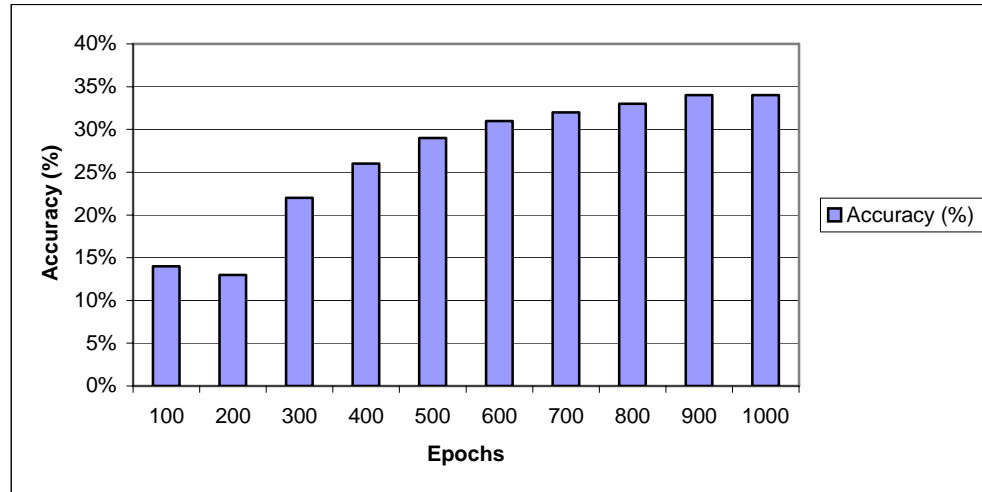


Figure 4.30: Accuracy Percentage (%) of Diabetes Dataset with BL cost function

#### 4.6.3.3 Result of Three Term BP with MM Cost Function for Diabetes Dataset

Table 4.11 shows Three Term BP performance with MM Cost Function for Diabetes dataset. MM cost function achieved the best result compared to other cost function for Diabetes dataset. Figure 4.31 shows the error convergence throughout the groups. The error is small since 100 epochs. The error value for 100 epochs is 0.0130 and for 1000 epoch is 0.0020. This cost function able to achieve threshold error 0.005 at 300 epochs itself. Besides that, as can be derived from Figure 4.32 the network's convergence time is 500 milliseconds to accomplish 100 epochs. It increased to 2200 milliseconds for 500 epochs and finally 3030 milliseconds to complete 1000 epochs. This shows that this network with MM cost function converge slowly. Moreover, the accuracy rate for MM cost function is still low. This can be observed from Figure 4.33. For the 100 epochs, the network produced 26% accuracy and increased slightly to 39% for 1000 epoch. Diabetes dataset showed better results with MM cost function compared to others.

Table 4.11 Testing results of Diabetes datasets with MM cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.0130	0.0060	0.0050	0.0050	0.0040	0.0040	0.0030	0.0020	0.0020	0.0020
Time(ms)	500	840	1180	1520	2200	2360	2530	2680	2840	3030
Accuracy	26%	33%	35%	37%	37%	38%	38%	36%	37%	39%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

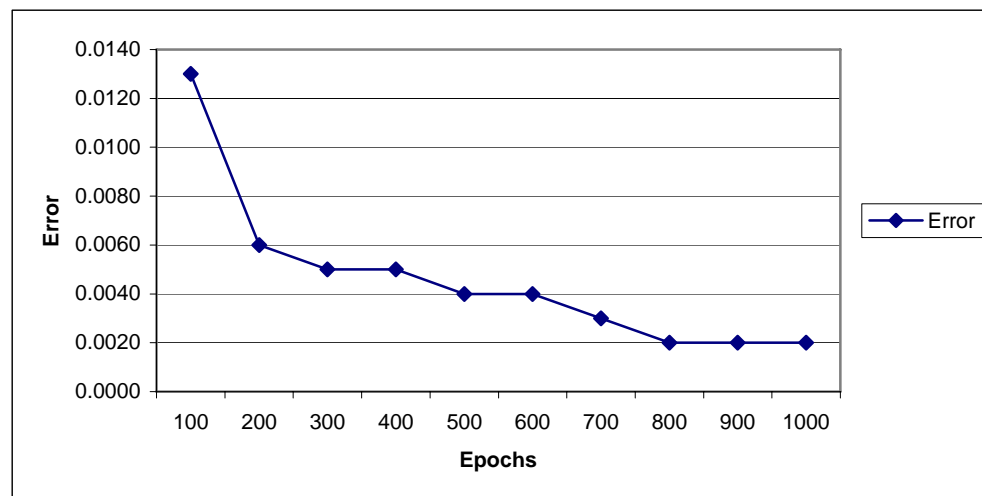


Figure 4.31: Error Convergence of Diabetes Dataset with MM Cost Function

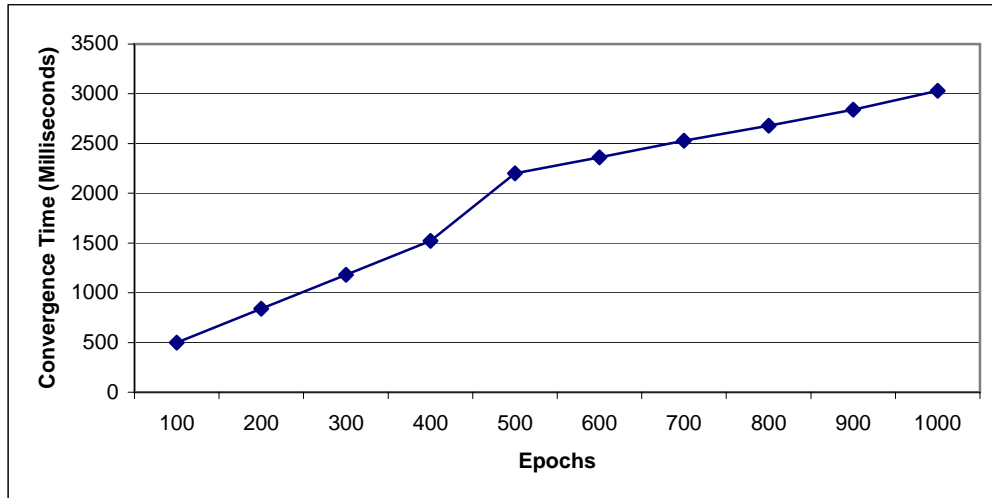


Figure 4.32: Convergence Time of Diabetes Dataset with MM cost function

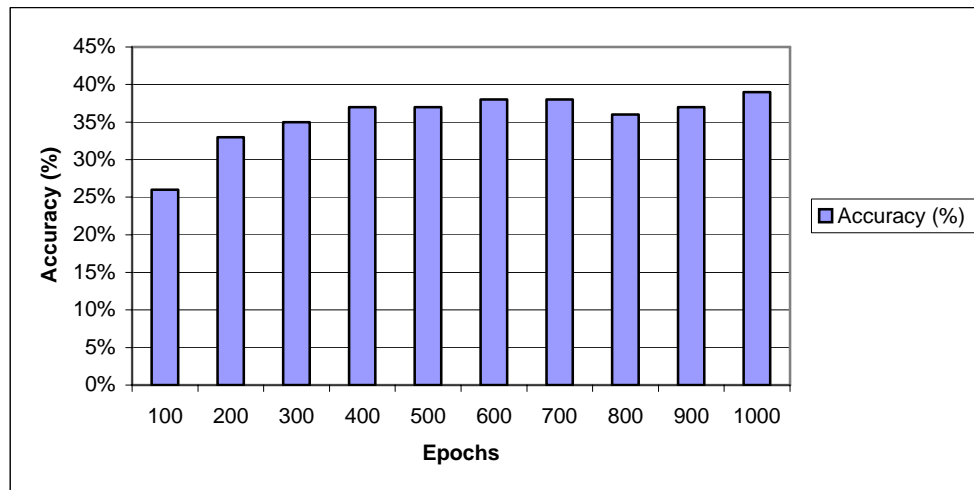


Figure 4.33: Accuracy Percentage (%) of Diabetes Dataset with MM cost function

#### 4.6.3.4 Result of Three Terms BP with IC Cost Function for Diabetes Dataset

Table 4.12 shows Three Term BP performance with IC cost function for Diabetes dataset. The IC cost function achieved a best result in terms of convergence time but performed badly in term of error and accuracy rate. Figure 4.34 shows the IC cost function's error values. The error value for 100 epochs is 0.0610 and decreased to 0.0060 at 1000 epochs. Besides that Figure 4.35 shows the convergence time of the IC cost function for Diabetes dataset. The convergence time for IC cost function of Diabetes dataset is 2200 milliseconds for 1000 epochs. This speed is fastest compared to other cost functions. Finally the accuracy rate can be observed in the Figure 4.36 for IC cost function in Three Term BP of Diabetes dataset. The accuracy rate for 100 epochs would be 19% and increased slightly to 33% in 1000 epochs. The accuracy percentage of IC cost function is low.

Table 4.12 Testing results of Diabetes datasets with IC cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.0610	0.0420	0.0260	0.0180	0.0140	0.0110	0.0090	0.0080	0.0060	0.0060
Time(ms)	330	520	710	990	1100	1370	1650	1920	2100	2200
Accuracy	19%	20%	14%	22%	26%	28%	30%	31%	33%	33%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

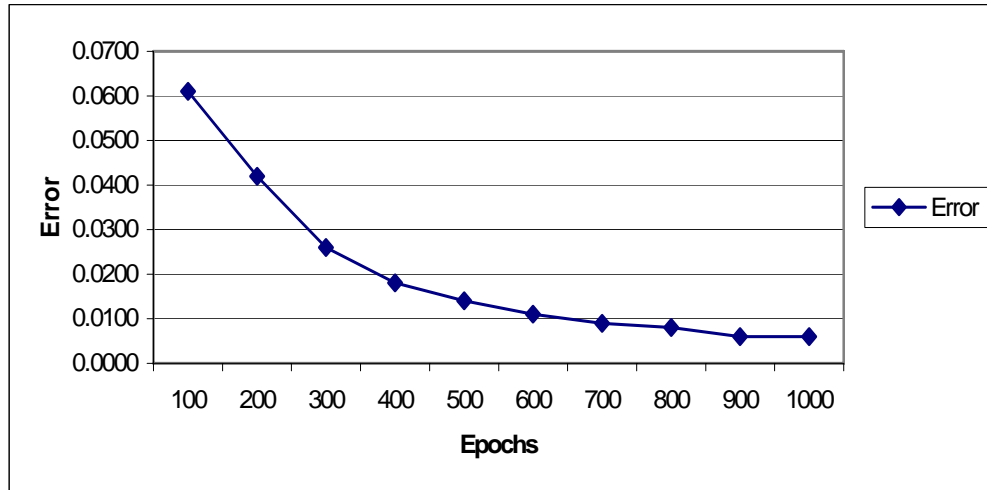


Figure 4.34: Error Convergence of Diabetes Dataset with IC Cost Function

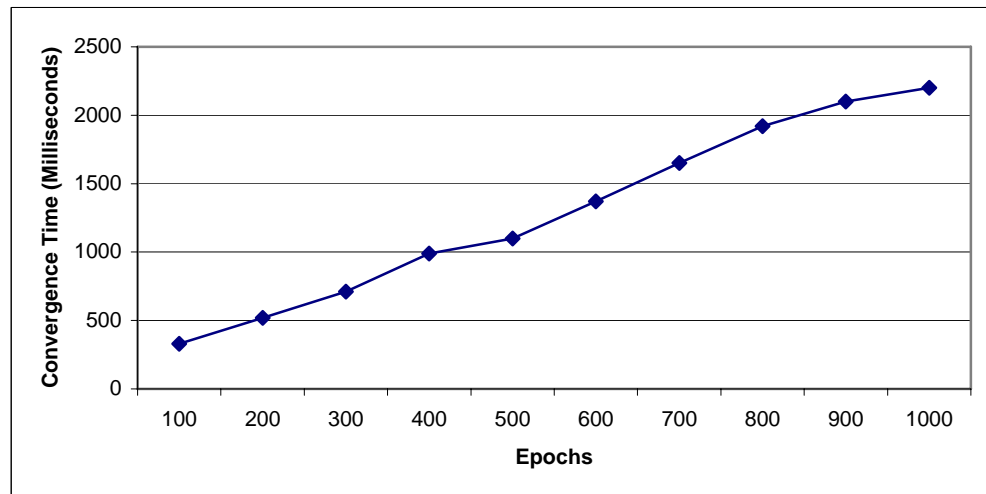


Figure 4.35: Convergence Time of Diabetes Dataset with IC cost function

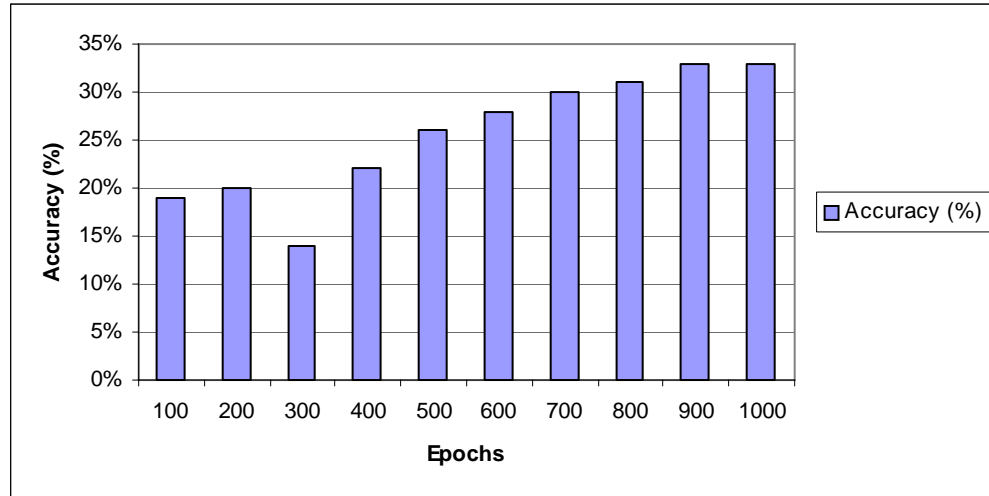


Figure 4.36: Accuracy Percentage (%) of Diabetes Dataset with IC cost function

#### 4.6.4 Result of Three Terms BP for Pendigits Dataset

Finally, Pendigits dataset was feed into the Three Term BP network with MSE, BL, MM and IC cost function. The purpose of treating this problem was to establish mapping between 16 input nodes, 12 hidden nodes and 10 output nodes. Total number of instances is 1000 instances. The test result is shown in the following texts.

All the testing data for Pendigits are grouped into ten: group I through group X. These testing datasets are carried out using constant learning rate 1.0, Momentum Factor 0.75 and Proportional Factor 0.95 through the simulation and the initial weights are selected randomly in the range  $-1$  to  $1$ . This complex and large network uses “K+100 Increment Rule” where the epochs are range from 100 to 1000. This is because the Pendigits dataset is large and complex and “K+10 Increment Rule” is insufficient to produce good result. The network is tested on 50 trials where each group is tested 5 times and the best result is shown in the Table 4.13 through Table 4.16. The comparative

results for the error (current error generated by network), time (convergence time of the network ) and accuracy ( percentage of test result) are summarized in Table 4.13 through Table 4.16.

#### 4.6.4.1 Result of Three Term BP with MSE Cost function for Pendigits dataset

Table 4.13 shows Three Term BP performance with MSE Cost Function for Pendigits dataset. Figure 4.37 indicates that the error generated at epoch 100 is 0.0800 and it degrades to 0.1000 at epoch 1000. This result shows that MSE cost function performed poorly. As illustrated in Figure 4.38, this cost function requires 37003 milliseconds to complete 1000 epoch. This convergence speed also slow compared to other cost functions for the same dataset. Besides that, as shown in the Figure 4.39 an accuracy of the testing data resulting from a range of 83% at 100 epochs to 90% at 1000 epochs. This indicates that high accuracy has been obtained despite poor performance in terms of error and convergence speed.

Table 4.13 Testing results of Pendigits datasets with MSE cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.8000	0.4000	0.3000	0.2000	0.2000	0.1000	0.1000	0.1000	0.1000	0.1000
Time(ms)	3950	6460	8470	10480	14010	16610	19210	21810	22210	37003
Accuracy	83%	87%	88%	89%	89%	90%	90%	90%	90%	90%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage



Figure 4.37: Error Convergence of Pendigits Dataset with MSE Cost Function

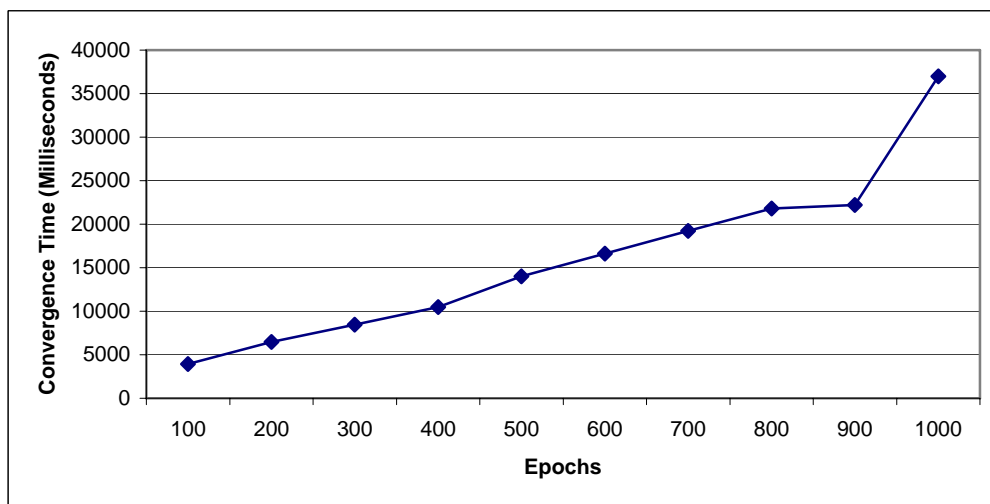


Figure 4.38: Convergence Time of Pendigits Dataset with MSE cost function

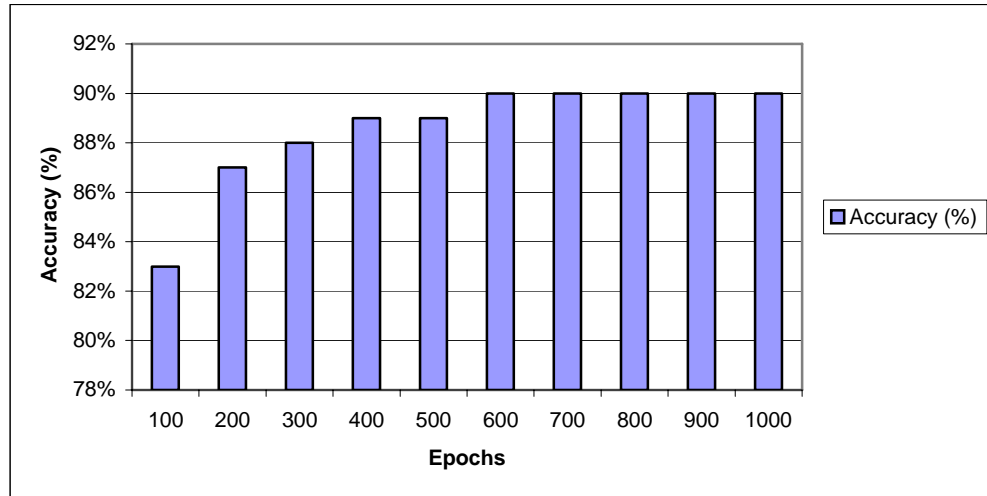


Figure 4.39: Accuracy Percentage (%) of Pendigits Dataset with MSE cost function

#### 4.6.4.2 Result of Three Term BP with BL Cost function for Pendigits dataset

Table 4.14 shows Three Term BP performance with BL cost function for Pendigits dataset. Performance of BL cost function is poor compared to other cost functions. The Figure 4.40 shows the errors for each group according to the epochs limits. 100 epoch produced 0.7100 error value and decreases significantly to 0.0580 at 1000 epochs. This significance decrease shows that number of epoch also plays an important role in the performance for the BL cost function. The Figure 4.41 is showing the convergence time from group I to X. The convergence time of group I, which will be 100 epochs is 4000 milliseconds and increased dramatically to 27020 for 1000 epochs. This networks taking a long time to converge because of the complex structure of the Pendigits dataset. The accuracy of 100 epochs is 20% and increased up to 84% at 1000 epochs. Overall BL cost function for Pendigits dataset performed quite low compared to other cost function (Figure 4.42).

Table 4.14 Testing results of Pendigits Dataset with BL cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.7100	0.5800	0.1780	0.1380	0.1120	0.0940	0.0810	0.0720	0.0640	0.0580
Time(ms)	4000	6300	8580	10880	13180	16020	18860	21700	24540	27020
Accuracy	20%	33%	74%	78%	80%	82%	83%	83%	84%	84%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

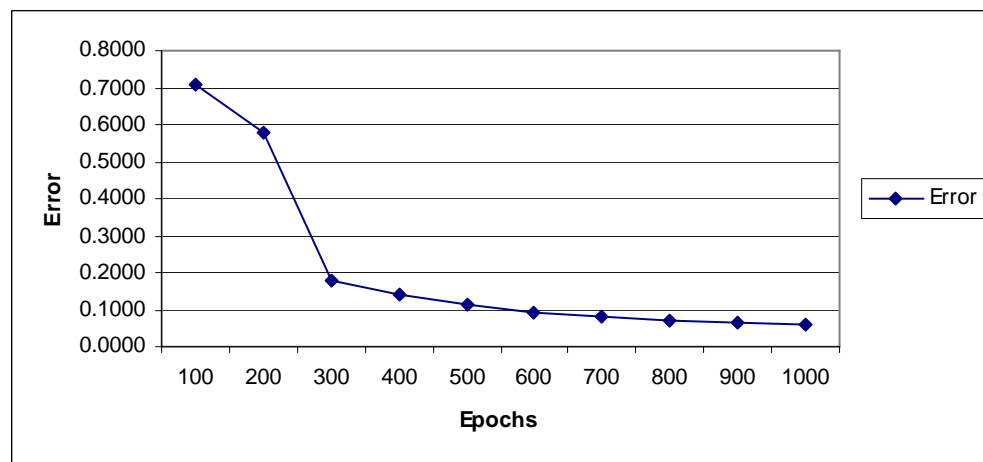


Figure 4.40: Error Convergence of Pendigit Dataset with BL Cost Function

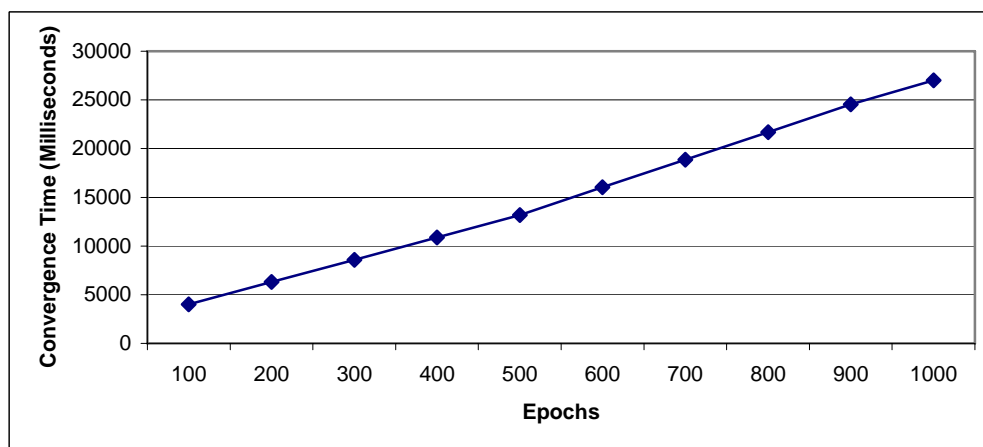


Figure 4.41: Convergence Time of Pendigits Dataset with BL cost function

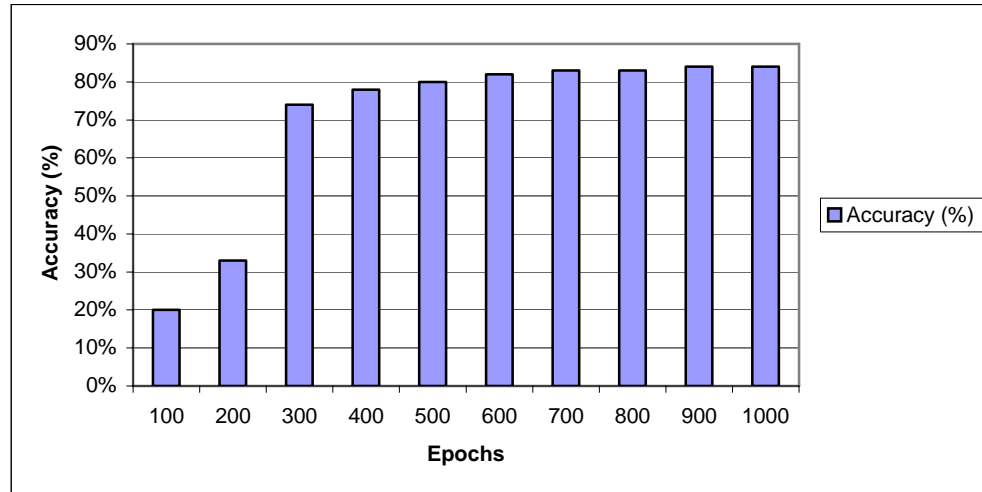


Figure 4.42: Accuracy Percentage (%) of of Pendigits Dataset with BL cost function

#### 4.6.4.3 Result of Three Term BP with MM Cost Function for Pendigits Dataset

Table 4.15 shows Three Term BP performance with MM cost function for Pendigits dataset. It shows that MM cost function performed well with Pendigits dataset too. It can be observed from Figure 4.43 that error decreased from 100 epochs to 1000 epochs in the result. The error is 0.2000 for the 100 epochs and decreased to 0.0050 for 1000 epochs. This error range indicates that MM cost function well performed on the Pendigits dataset where error is the best minimum value among other cost functions. Even though it performs in term of error but the convergence time needed 5270 milliseconds for 100 epochs increased through group to 37030 milliseconds for 1000 epochs. This result can be observed from Figure 4.44. This network can be considered as slow in convergence speed. On the other side, this MM cost function produced high accuracy rate as can be seen in the Figure 4.45. The accuracy rate is high for Pendigits dataset since the 100 epochs itself is 71%. This rate increased to 91% for the 1000 epochs.

Table 4.15 Result of Three Term BP with MM Cost function for Pendigits Dataset

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.2000	0.1000	0.3000	0.1000	0.2000	0.1000	0.0090	0.0070	0.0060	0.0050
Time(ms)	5270	9800	12720	16460	24000	23600	27000	30300	33550	37030
Accuracy	71%	81%	61%	81%	88%	80%	90%	90%	91%	91%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

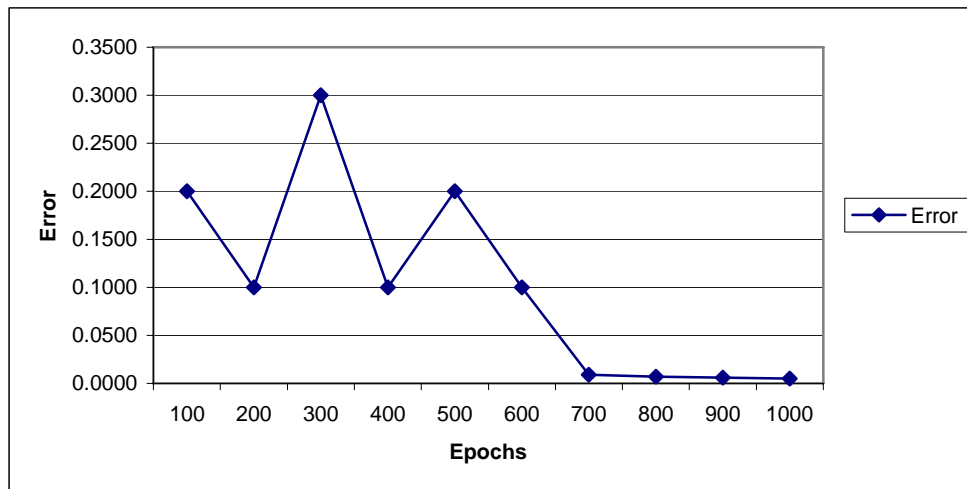


Figure 4.43: Error Convergence of Pendigits Dataset with MM Cost Function

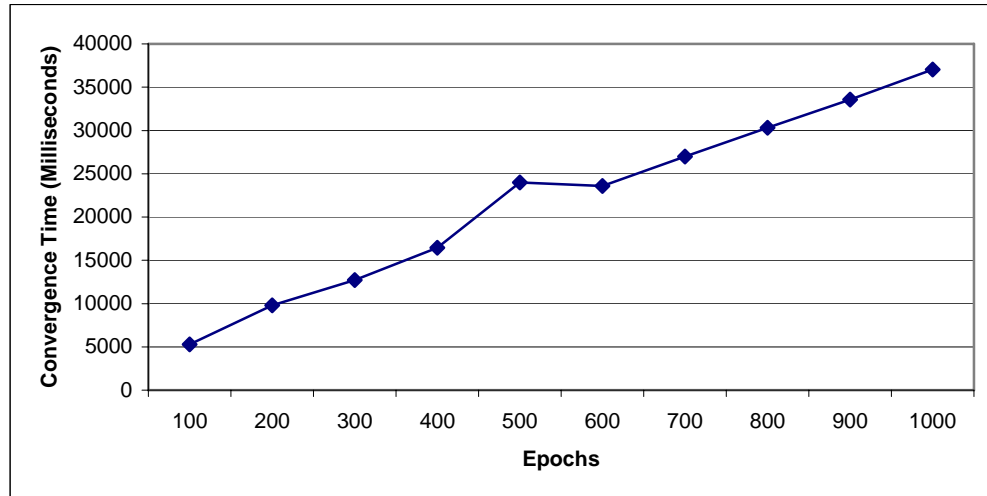


Figure 4.44: Convergence Time of Pendigits Dataset with MM Cost Function

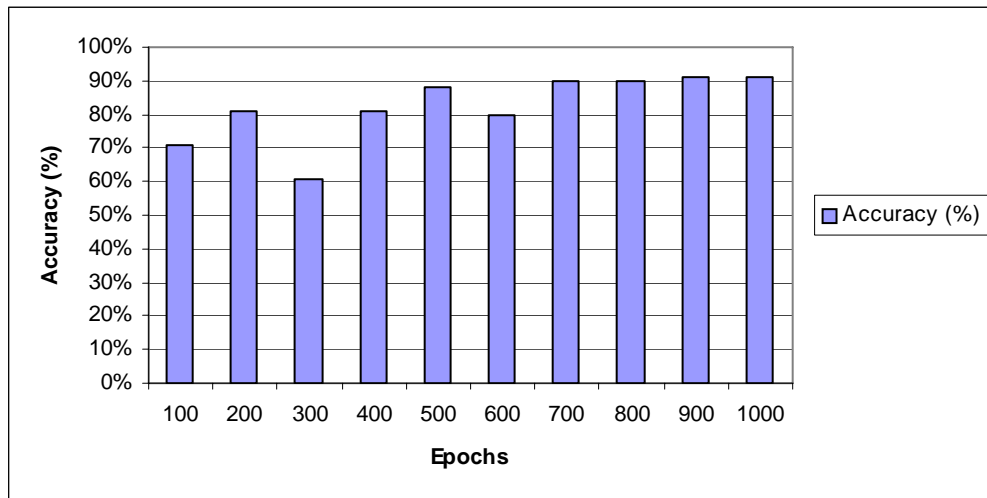


Figure 4.45: Accuracy Percentage (%) of Pendigits Dataset with MM Cost Function

#### 4.6.4.4 Result of Three Terms BP with IC Cost function for Pendigits dataset

Table 4.16 shows Three Term BP performance with IC cost function for Pendigits dataset. The IC cost function achieved a very good performance for Pendigits dataset. The IC cost function's error values are illustrated in the Figure 4.46. The error value for 100 epochs is 0.4400 and decreased to 0.0080 at 800 epochs. Then it remained at 0.0080 until 1000 epochs. Besides that Figure 4.47 shows the convergence time of the IC cost function. This IC cost function produced the shortest timing required by Pendigit dataset to complete 1000 epochs that is 24710 milliseconds. Furthermore, the accuracy rate of the IC cost function is illustrated in the Figure 4.48. This IC cost function performed well where at 100 epochs the accuracy rate was 60% and increased to 91% at 1000 epochs. This convergence percentage is the best successful rate for Pendigits dataset.

Table 4.16 Testing results of Pendigits dataset with IC cost function

Group	I	II	III	IV	V	VI	VII	VIII	IX	X
Epoch	100	200	300	400	500	600	700	800	900	1000
Error	0.4000	0.1500	0.0800	0.0500	0.0300	0.0200	0.0200	0.0080	0.0080	0.0080
Time(ms)	4400	7130	9860	12590	15330	17200	19070	20940	22810	24710
Accuracy	60%	75%	80%	83%	81%	88%	88%	91%	91%	91%

Indicator:-

Time (Convergence Time) : Milliseconds

Accuracy (Test Result) : Percentage

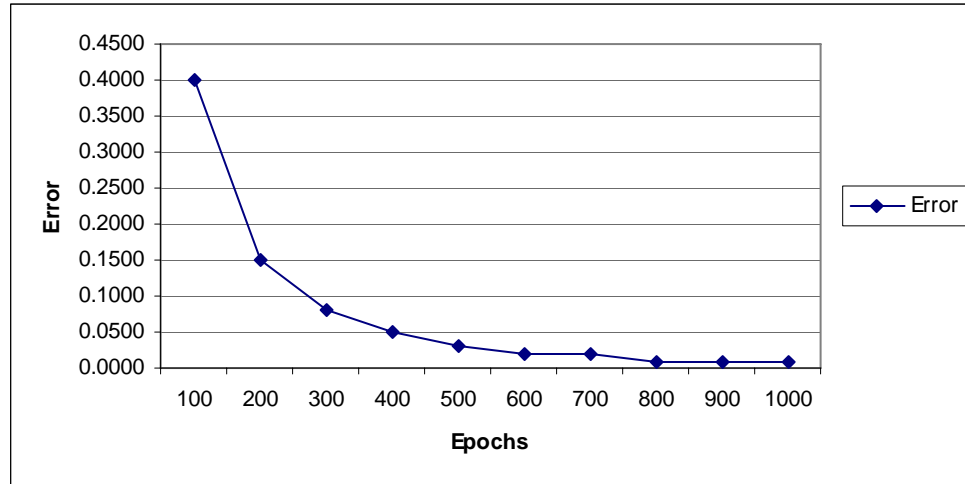


Figure 4.46: Error Convergence of Pendigits Dataset with IC Cost Function

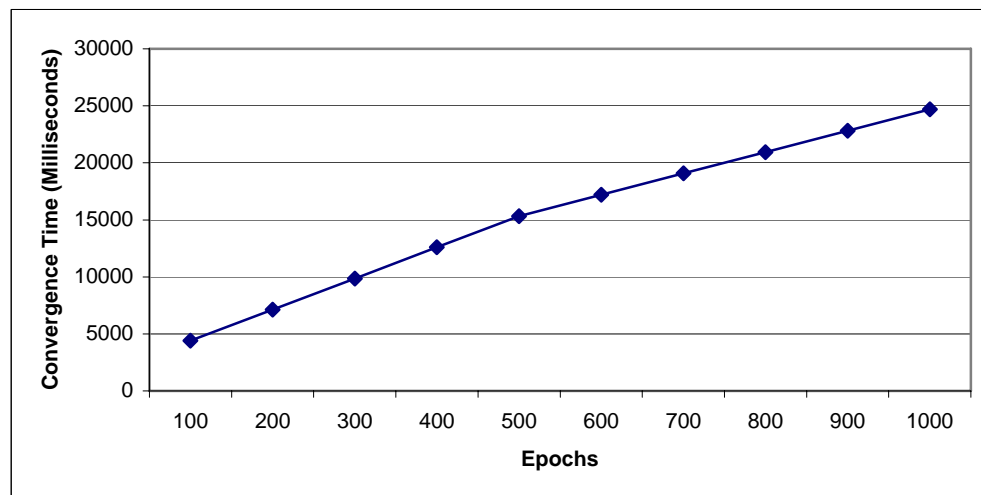


Figure 4.47: Convergence Time of Pendigit Dataset with IC Cost Function

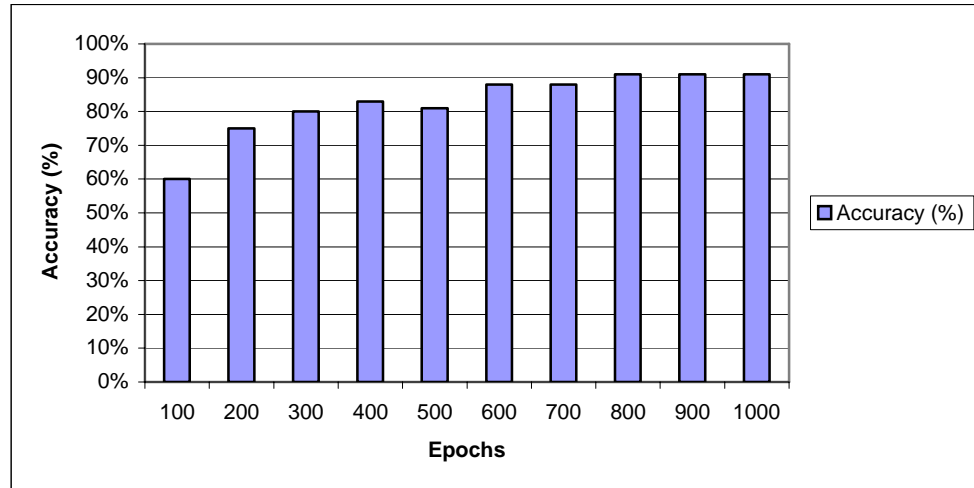


Figure 4.48: Accuracy Percentage (%) of Pendigits Dataset with IC Cost Function

#### 4.7 Performance Comparison for Three Term BP with Various Cost Function

Referring back on the objectives, experimental comparisons of various costs function in Three Term BP have been conducted. All the cost functions have been compared based on three comparison parameters and the summary are provided in the next section. The comparison has been carried out in terms of comparison parameters and the cost functions are presented in order with their performances accordingly. For example, in terms of convergence time, MM cost function has the shortest time followed by MSE, BL and IC. Hence, MM cost function will be put in the 1<sup>st</sup> position and will be denoted as 1 point followed by MSE with 2 points, BL with 3 points and IC with 4 points. The same approach is applied for other comparison criteria as well. At the end of the calculation, the cost function with the lesser points will be the best cost function for that particular Datasets.

### 4.7.1 Balloon Dataset

For Balloon dataset, the comparison is carried out and summarized in the Table 4.17. BL cost function outperforms others in term of error value and convergence time. All the cost function performs equally in terms of accuracy. The MM cost function performed well too by being in a second position for error value and convergence time. Besides that, MSE cost function does not performed well while IC cost function performed the worst in term of error value and convergence time for Balloon dataset.

As we could observed from the Table 4.17, MSE cost function gained 7 points, MM cost function 5 points, BL cost function 3 points while IC cost function is 9 points. Therefore these results indicate that BL cost function is best for Balloon datasets. However the MM cost function also can be considered because the performance is near to BL cost function with 5 points. The third place would be for MSE cost function followed by IC cost function.

Table 4.17 Cost functions position in term of comparison parameters

Position	1	2	3	4
Error	BL	MM	MSE	IC
Convergence Time	BL	MM	MSE	IC
Accuracy	MSE, MM, BL, IC			

The following subsection will analyze the cost function with various parameters such as error value, convergence time and accuracy.

### 4.7.1.1 Error

For balloon datasets, the error of convergence was compared among the all four cost functions. Table 4.18 shows the error produced by different cost function for Balloon dataset. The error value between BL and MM is not far while the MSE and IC cost function error value is in its own same range. The graph illustration of the result can be seen from the Figure 4.49. The result shows that BL cost function outperform all and produce sufficiently smallest error compared to other cost function. This followed by MM cost function, MSE cost function and IC cost function.

Table 4.18 Error produced by different cost function for Balloon Dataset

Epochs	10	20	30	40	50	60	70	80	90	100
MSE	0.2274	0.1261	0.0875	0.067	0.0543	0.0457	0.0394	0.0347	0.0309	0.0279
BL	0.0419	0.0251	0.0178	0.0138	0.0112	0.0094	0.0081	0.0072	0.0064	0.0058
MM	0.0538	0.0296	0.0203	0.0153	0.0123	0.0103	0.0088	0.0077	0.0068	0.0061
IC	0.2345	0.1291	0.0894	0.0683	0.0553	0.0469	0.041	0.0352	0.0313	0.0282

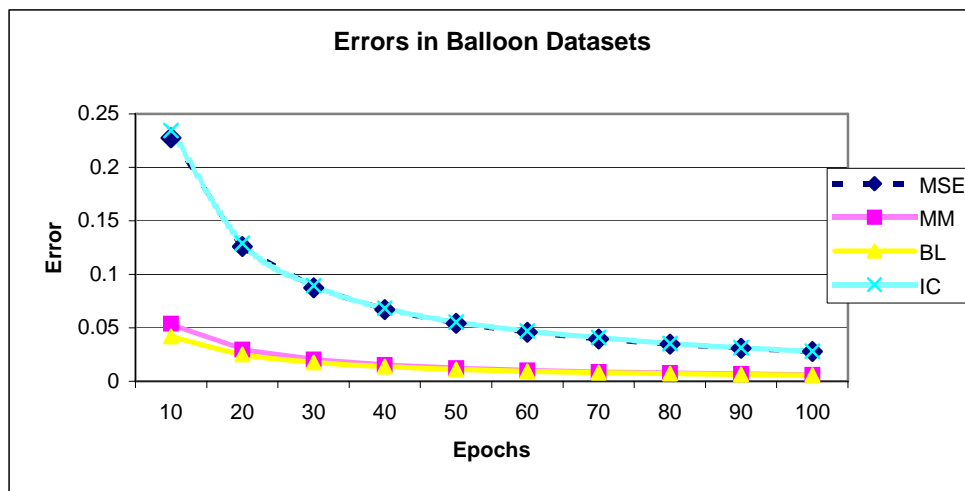


Figure 4.49 Error produced by different cost functions for Balloon dataset

### 4.7.1.2 Convergence Time

Table 4.19 shows the convergence time of cost functions for Balloon Dataset. The convergence time for 10 epochs is 50 milliseconds for MSE and BL cost function while 60 milliseconds for MM and IC cost functions. The Figure 4.50 indicates that the early epochs such as 10 to 50 epochs, the convergence time for all the cost function fluctuate except for BL cost function but later it became static from 60 epochs for MSE and IC cost function while from 70 epochs for MM and BL cost functions. This shows that basically all the cost function produced good result but since BL cost function's convergence time was short since the early epochs, so BL cost function considered performing the best for convergence time.

Table 4.19 Convergence Time by different cost function for Balloon Dataset

Epoch	10	20	30	40	50	60	70	80	90	100
MSE	50	50	60	50	60	110	110	110	110	110
BL	50	50	50	50	50	60	110	110	110	110
MM	60	50	60	60	50	50	110	110	110	110
IC	60	60	60	60	60	110	110	110	110	110

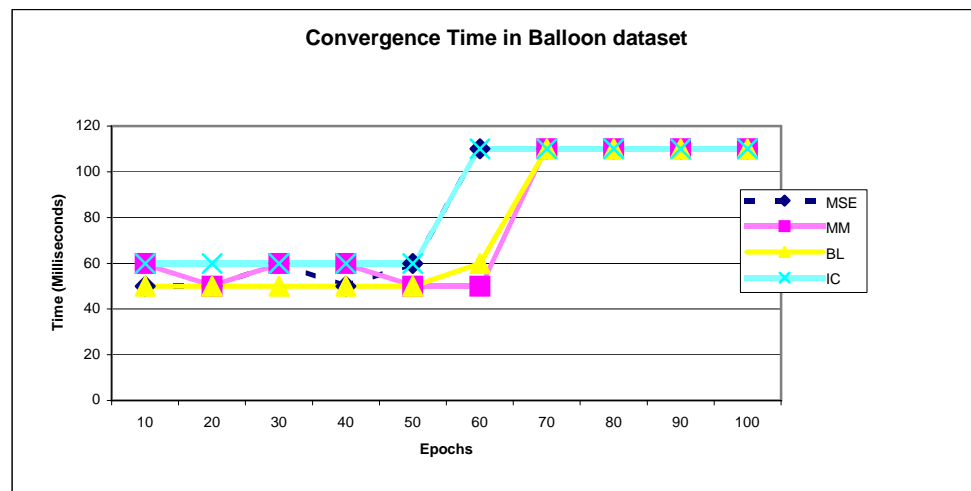


Figure 4.50 Convergence Time by different cost functions for Balloon Dataset

### 4.7.1.3 Accuracy Percentage

Table 4.20 shows the accuracy percentage of cost functions for Balloon dataset. The accuracy rate for all cost function produced the same result that is 75%. This is due to the testing data is only 4 and all the four cost functions produced 1 wrong answer eventough the epoch is increased up to 100 epochs. Thus, cost function could not be compared in this accuracy percentage. The illustration of the result could be found in Figure 4.51.

Table 4.20 Accuracy Percentage by different cost function for Balloon Dataset

Epoch	10	20	30	40	50	60	70	80	90	100
MSE	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%
BL	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%
MM	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%
IC	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%

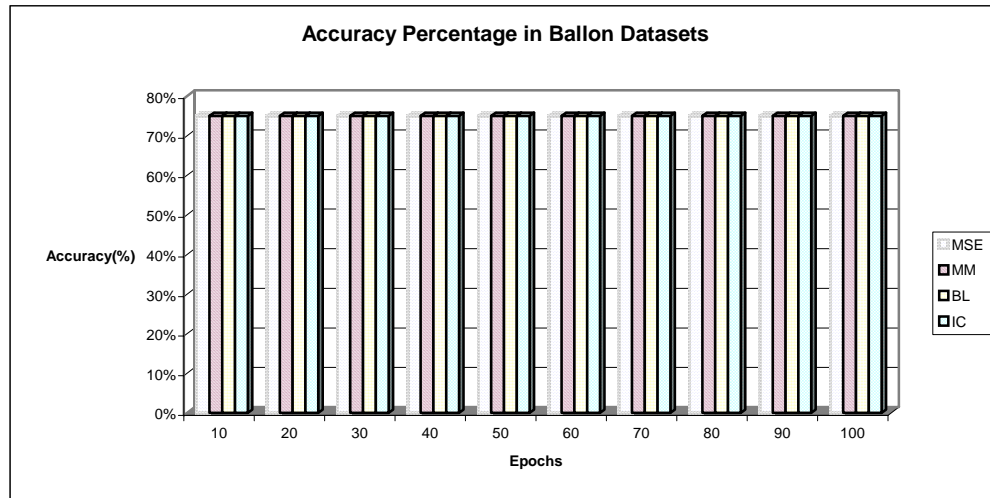


Figure 4.51 Accuracy Percentage by different cost functions for Balloon dataset

### 4.7.2 Cancer Dataset

For Cancer dataset, the comparison is carried out and summarized in the Table 4.21. MM cost function performed well by producing smallest error and high accuracy. On the other hand MM cost function requires more time compared to other cost function to complete 1000 epochs. In mean while IC cost function outperform others in term of convergence time. IC cost function also performed well in term of accuracy but produced quite high error value. The MSE cost function performed well too by being in a second position for error value and Convergence time but remained last in terms of accuracy percentage. Finally BL cost function could not performed well in cancer dataset. This may be that BL cost function is suitable for small scale datasets.

As we could observed from the Table 4.21 MSE cost function gained 8 points, MM cost function 6 points, BL cost function 10 points while IC cost function is 6 points. Therefore these results indicate that IC and MM cost functions are equally performed well for Cancer datasets. However the MM cost function is a better cost function for Cancer dataset because eventough the convergence time is longer but the accuracy which would be the significant comparison parameter is high and the error produced is the smallest. Therefore, The best cost function for Cancer dataset is MM cost function and followed by IC cost function. The MSE cost function takes third place followed by BL cost function. The following subsection examines cost function with various parameters such as error value, convergence time and accuracy.

Table 4.21 Cost Functions Position in term of comparison parameters

Position	1	2	3	4
Error	MM	MSE	IC	BL
Time	IC	MSE	BL	MM
Accuracy	MM	IC	BL	MSE

### 4.7.2.1 Error

The error value of convergence was compared among the all four cost functions. Table 4.22 shows the error produced by different cost function for cancer dataset. The graph illustration of the result can be seen from the Figure 4.52. The result shows that MM cost function outperform all and produce sufficiently smallest error for 100 epochs compared to other cost functions. The error value is 0.0010, the smallest error ever produced during the experiments. This followed by MSE cost function and IC cost function. The error value of MSE and IC for 100 epoch is exactly same that is 0.0050. This followed by BL cost function. Therefore, it could be concluded that the MM cost function is the best cost function for Cancer dataset in term of error value.

Table 4.22 Error produced by different cost function for Cancer Dataset

Epochs	10	20	30	40	50	60	70	80	90	100
MSE	0.050	0.026	0.017	0.013	0.01	0.009	0.007	0.006	0.006	0.005
BL	0.394	0.208	0.14	0.106	0.085	0.07	0.06	0.063	0.047	0.042
MM	0.014	0.007	0.005	0.003	0.003	0.002	0.002	0.002	0.001	0.001
IC	0.063	0.03	0.0178	0.014	0.011	0.01	0.007	0.006	0.006	0.005

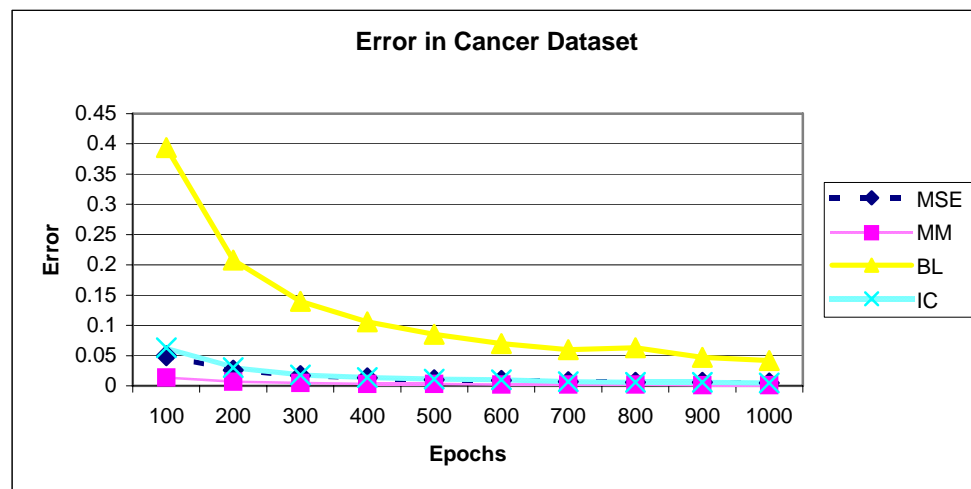


Figure 4.52 Error produced by different cost function for Cancer dataset

#### 4.7.2.2 Convergence Time

Table 4.23 shows the convergence time of cost functions for Cancer dataset. The convergence time for cancer dataset would be from 100 epochs to 1000 epochs for all the cost functions. The Figure 4.53 illustration helps to see the pattern of the cost function's convergence time. It indicates that IC cost function gives a good performance by producing the shortest time to complete 100 to 1000 epochs. This followed by MSE cost function. The BL and MM cost function needs longer times up to 3070 milliseconds and 3240 milliseconds each. Therefore we could conclude that IC cost function is the best cost function in term of convergence time. The MSE cost function also produce a good timing. MM cost function and BL cost function does not perform well in term of convergence time for Cancer dataset.

Table 4.23 Convergence Time by different cost function for Cancer Dataset

Epoch	100	200	300	400	500	600	700	800	900	1000
MSE	1120	1340	1070	1260	1480	1590	1650	2000	2260	2520
BL	440	700	960	1190	1470	1670	1930	2120	2380	3070
MM	660	910	1160	1410	1920	2170	2620	2670	2920	3240
IC	380	560	710	890	1090	1270	1450	1600	1810	2250

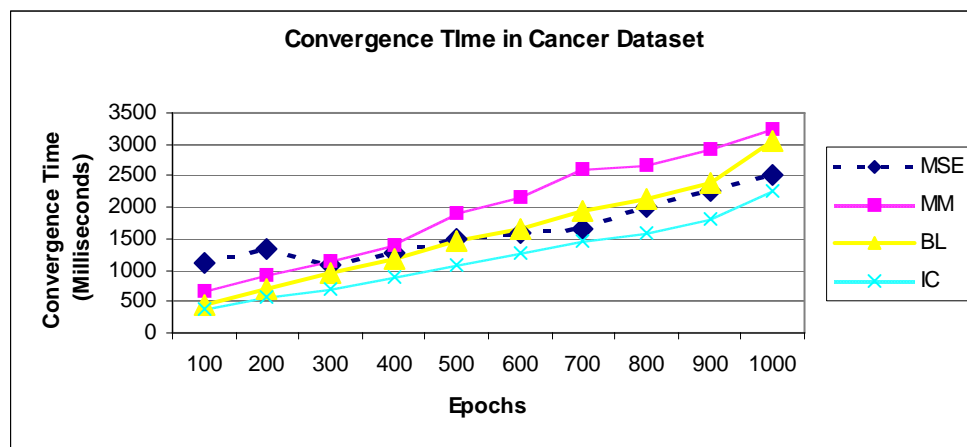


Figure 4.53 Convergence Time by different cost function for Cancer Dataset

### 4.7.2.3 Accuracy Percentage

Table 4.24 shows the accuracy percentage of cost functions for Cancer dataset. The accuracy percentage was low at the beginning for most of the cost function such as MSE, BL and IC. But MM cost function produced good accuracy even during 100 epochs. Finally for 1000 epochs the MM cost function proved to perform the best when it reached 86% of accuracy. As the Figure 4.54 show, basically all the cost functions produced good result because the accuracy difference between all the cost function is 1% only.

Table 4.24 Accuracy Percentage by different cost function for Cancer Dataset

Epoch	100	200	300	400	500	600	700	800	900	1000
MSE	33%	62%	71%	75%	78%	79%	81%	82%	82%	83%
BL	49%	68%	74%	78%	80%	81%	82%	83%	84%	84%
MM	74%	81%	83%	85%	85%	86%	86%	86%	87%	86%
IC	25%	58%	70%	74%	77%	78%	81%	82%	83%	85%

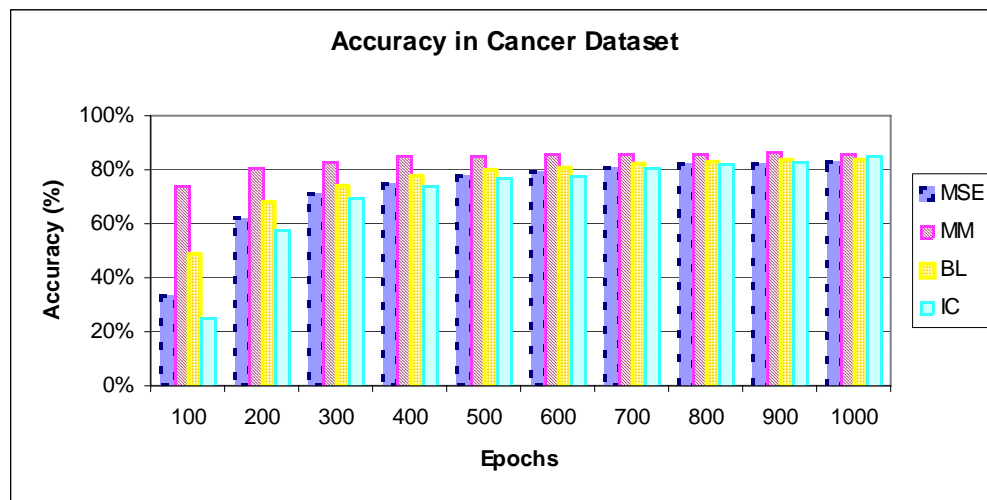


Figure 4.54 Accuracy Percentage by different cost function for Cancer dataset

### 4.7.3 Diabetes Dataset

For Diabetes dataset, the comparison is carried out and summarized in the Table 4.25. As can be observed from Table 4.25, MM cost function performed well by producing smallest error and high accuracy. On the other hand IC cost function also performed well in term of convergence time but produced quite high error value during training. The BL cost function performed well too by being in a second position for error value and Convergence time but not for accuracy. Finally MSE cost function could not performed well in Diabetes dataset except for accuracy percentage.

As we could observed from the Table 4.25, MSE cost function gained 9 points, MM cost function 5 points, BL cost function 7 points while IC cost function is 9 points. Therefore these results indicate that MM cost function performed the best for Diabetes datasets. This will be followed by BL cost function. Furthermore, the IC and MSE function performed badly with a score of 9 each. Therefore, The best cost function for Diabetes dataset is MM cost function and followed by BL cost function. The MSE cost function and IC cost function sharing the third place.

Table 4.25 Cost Functions Position in term of comparison parameters

Position	1	2	3	4
Error	MM	BL	MSE	IC
Time	IC	BL	MM	MSE
Accuracy	MM	MSE	BL	IC

The following subsections evaluate the cost function with various parameters such as error value, convergence time and accuracy.

### 4.7.3.1 Error convergence

The error value of convergence was compared among the all four cost function. Table 4.26 shows the error produced by different cost function for Diabetes dataset. The graph illustration of the result can be seen from the Figure 4.55. The result shows that MM cost function outperform all and produce sufficiently smallest error compared to other cost function. The error value is 0.0020. This followed by BL cost function. The error value of MSE and IC for 100 epoch is exactly same that is 0.0060. Therefore, it could be concluded that the MM cost function is the best cost function for Diabetes dataset in term of error value.

Table 4.26 Error produced by different cost function for Diabetes Dataset

Epochs	100	200	300	400	500	600	700	800	900	1000
MSE	0.0560	0.0280	0.0190	0.0140	0.0110	0.0090	0.0080	0.0070	0.0060	0.0060
BL	0.0544	0.0278	0.0186	0.0140	0.0112	0.0093	0.0080	0.0070	0.0062	0.0056
MM	0.0130	0.0060	0.0050	0.0050	0.0040	0.0040	0.0030	0.0020	0.0020	0.0020
IC	0.0610	0.0420	0.0260	0.0180	0.0140	0.0110	0.0090	0.0080	0.0060	0.0060

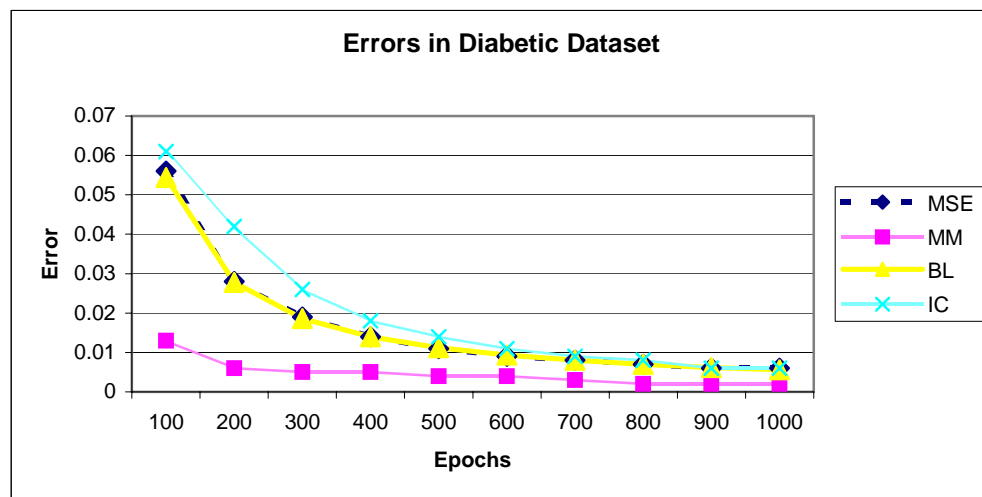


Figure 4.55 Error produced by different cost function for Diabetes Dataset

### 4.7.3.2 Convergence Time

Table 4.27 shows the convergence time of cost functions for Diabetes dataset. The convergence time gradually increased when the epochs increases. The Figure 4.56 illustration shows graphically how the convergence time increase by groups. It indicates that IC cost function gives a good performance by producing the shortest time to complete 100 to 1000 epochs. This followed by BL cost function, MM cost function and lastly MSE cost function. Therefore we could conclude that IC cost function is the best cost function in term of convergence time. This IC error function is not complicated, thus the convergence time is short. The BL cost function also produce a good timing and grabbed second place. MM cost function and MSE cost function does not perform well in term of convergence time.

Table 4.27 Convergence Time by different cost function for Diabetes dataset

Epoch	100	200	300	400	500	600	700	800	900	1000
MSE	320	540	740	960	1430	1760	1940	2270	2620	3090
BL	440	680	920	1160	1650	1910	2170	2430	2690	2690
MM	500	840	1180	1520	2200	2360	2530	2680	2840	3030
IC	330	520	710	990	1100	1370	1650	1920	2100	2200

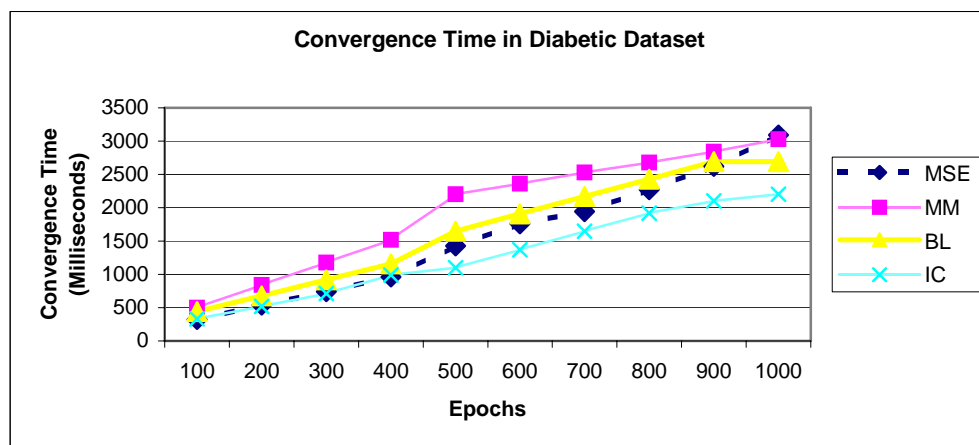


Figure 4.56 Convergence Time by different cost function for Diabetes dataset

### 4.7.3.3 Accuracy Percentage

Table 4.28 shows the accuracy percentage of cost functions for Diabetes Dataset. Overall, the accuracy percentage is low for all the cost function such as MSE, BL, MM and IC. As illustrated in the Figure 4.57 the range of accuracy rate for Diabetes dataset is from 11% to 39%. This percentage is small compared to other datasets. Even though MM cost function produced 39% accuracy but it is considered good for Diabetes dataset. Besides that, MSE and BL cost function produced a 34% sharing the second best status here. Finally IC cost function could achieve 33% of accuracy only for 1000 epochs.

Table 4.28 Accuracy Percentage by different cost function for Diabetes dataset

Epoch	100	200	300	400	500	600	700	800	900	1000
MSE	17%	11%	21%	26%	29%	31%	32%	33%	34%	34%
BL	14%	13%	22%	26%	29%	31%	32%	33%	34%	34%
MM	26%	33%	35%	37%	37%	38%	38%	36%	37%	39%
IC	19%	20%	14%	22%	26%	28%	30%	31%	33%	33%

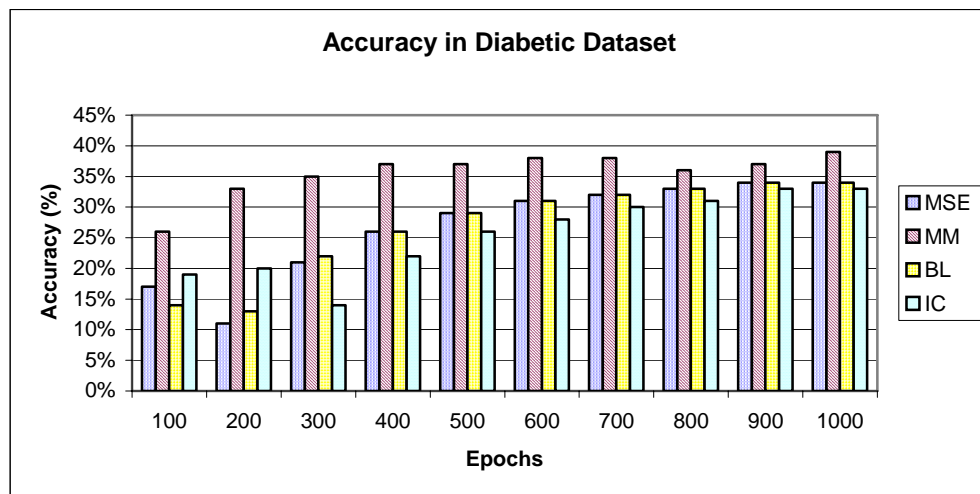


Figure 4.57 Accuracy Percentage by different cost function for Diabetes dataset

#### 4.7.4 Pendigits Dataset

For Pendigits dataset, the comparison is carried out and summarized in the Table 4.29. The performance of IC cost function and MM cost function are good for Pendigits dataset. Both cost function produced good result and the difference between both cost functions are not much. The MM cost function performed well by producing smallest error and high accuracy. On the other hand IC cost function also produced shortest convergence time and high accuracy. In contra, The MM cost function just like in other datasets requires long convergence time. Meanwhile, BL cost function and MSE cost function performed badly for Pendigits dataset. Those functions are not suitable for Pendigits dataset. MSE once again performed badly for Pendigits dataset.

As we could observed from the Table 4.29, MSE cost function gained 10 points, MM cost function 6 points, BL cost function 9 points while IC cost function is 5 points. Therefore these results indicate that IC cost functions is the best cost function for Pendigits dataset. Besides that MM cost function also would be suitable for Pendigits dataset.

Table 4.29 Cost Functions Position in term of comparison parameters for Pendigits

Position	1	2	3	4
Error	MM	IC	BL	MSE
Time	IC	BL	MSE	MM
Accuracy	MM IC	-	MSE	BL

The following subsection analyzes the cost function with various parameters such as error value, convergence time and accuracy rate.

#### 4.7.4.1 Error

The error value of convergence was compared among the all four cost function. Table 4.30 shows the error produced by different cost function for Pendigits dataset. The graph illustration of the result can be seen from the Figure 4.58. The result shows that MM cost function outperform all and produce sufficiently smallest error compared to other cost function. The error value of 1000 epochs is 0.0050. This followed by IC cost function with 0.0080 for 1000 epochs. Besides that, MSE and BL cost function does not produce good results in term of error value. Therefore, it could be concluded that the MM cost function is the best cost function for Pendigits dataset in term of error value.

Table 4.30 Error produced by different cost function for Pendigits dataset

Epochs	100	200	300	400	500	600	700	800	900	1000
MSE	0.8000	0.4000	0.3000	0.2000	0.2000	0.1000	0.1000	0.1000	0.1000	0.1000
BL	0.7100	0.5800	0.1780	0.1380	0.1120	0.0940	0.0810	0.0720	0.0640	0.0580
MM	0.2000	0.1000	0.3000	0.1000	0.2000	0.1000	0.0090	0.0070	0.0060	0.0050
IC	0.4000	0.1500	0.0800	0.0500	0.0300	0.0200	0.0200	0.0080	0.0080	0.0080

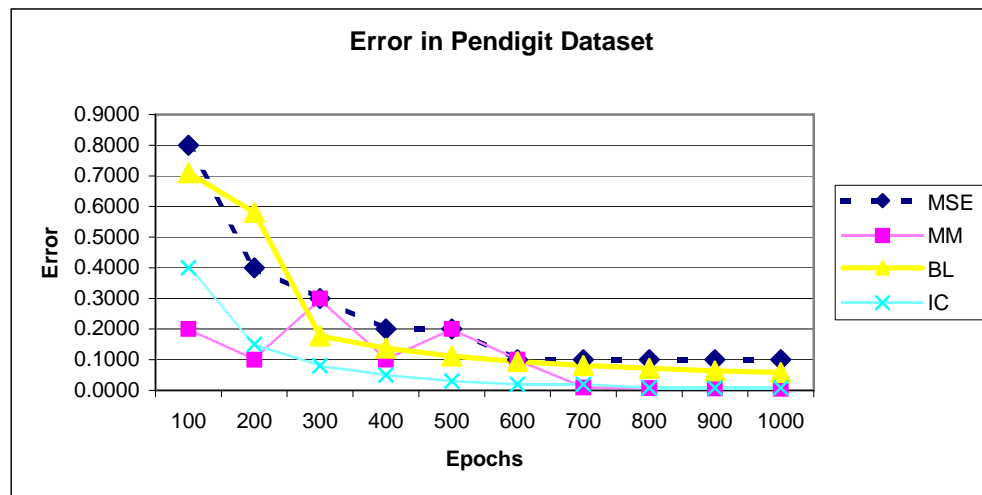


Figure 4.58 Error produced by different cost function for Pendigits Dataset

#### 4.7.4.2 Convergence Time

Table 4.31 shows the convergence time of cost functions for Pendigits dataset. MSE, BL and IC cost functions showed steady and almost same increase along with the number of epochs. MM cost function showed steady increase to but needs longer time compared to other cost functions. The Figure 4.59 illustration all the cost function's convergence time. It indicates that IC cost function gives a good performance by producing the shortest time to complete 100 to 1000 epochs. This followed by BL cost function. The MSE and MM cost function needs longer times up to 37003 milliseconds and 37030 milliseconds each to complete 1000 epochs. Therefore we could conclude that IC cost function is the best cost function in term of convergence time. The BL cost function also produce a good timing. MSE cost function and MM cost function does not perform well in term of convergence time.

Table 4.31 Convergence Time by different cost function for Pendigits dataset

Epoch	100	200	300	400	500	600	700	800	900	1000
MSE	3950	6460	8470	10480	14010	16610	19210	21810	22210	37003
BL	4000	6300	8580	10880	13180	16020	18860	21700	24540	27020
MM	5270	9800	12720	16460	24000	23600	27000	30300	33550	37030
IC	4400	7130	9860	12590	15330	17200	19070	20940	22810	24710

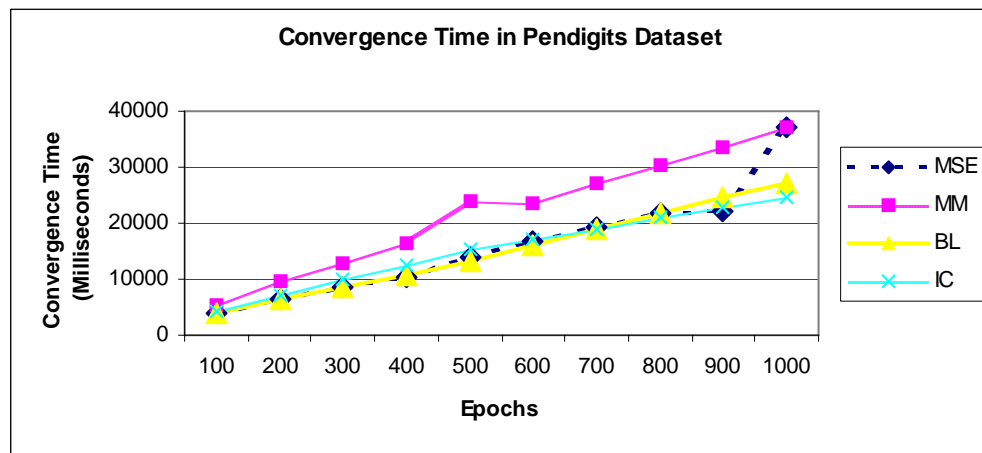


Figure 4.59 Convergence Time by different cost function for Pendigits dataset

#### 4.7.4.3 Accuracy Percentage

Table 4.32 shows the accuracy percentage of cost functions for Pendigits dataset. The accuracy percentage was low at the beginning for BL and IC. But MSE and MM cost functions showed good accuracy even during 100 epochs. Finally for 1000 epochs the MM cost function and IC cost function proved to perform the best when it reached 91% of accuracy. But when compared these two cost function MM cost function outstanding because it could produce high accuracy even at 100 epochs. This followed closely by MSE cost function. The least accuracy for Pendigits is produced by BL cost function. As the Figure 4.60 show, basically all the cost functions produced good result. But looking closely at the result, MM, IC and MSE will be good cost functions to be used for Pendigits dataset.

Table 4.32 Accuracy Percentage by different cost function for Pendigits dataset

Epoch	100	200	300	400	500	600	700	800	900	1000
MSE	83%	87%	88%	89%	89%	90%	90%	90%	90%	90%
BL	20%	33%	74%	78%	80%	82%	83%	83%	34%	84%
MM	71%	81%	61%	81%	88%	80%	90%	90%	91%	91%
IC	60%	75%	80%	83%	81%	88%	88%	91%	91%	91%

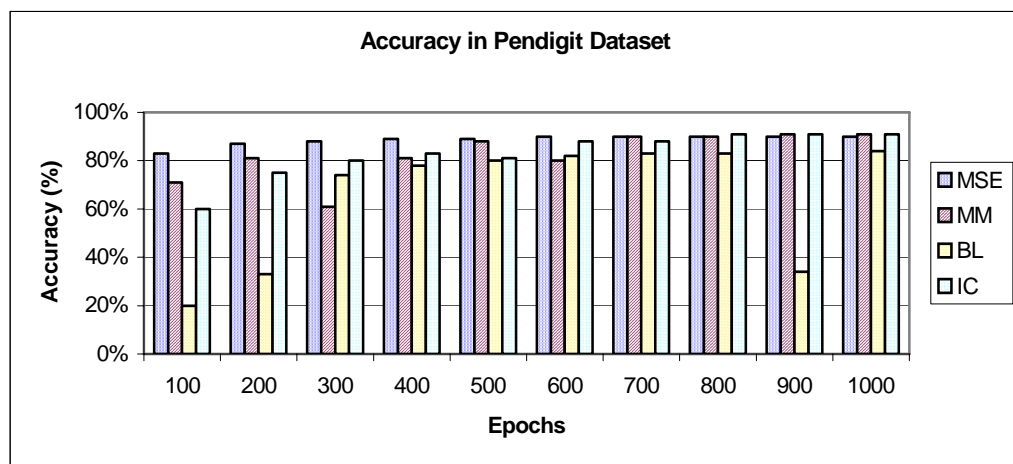


Figure 4.60 Accuracy Percentage by different cost function for Pendigits Dataset

## 4.8 T-Test

T-test was conducted in terms of error value, convergence time and accuracy

### 4.8.1 T-test for Error Value

T-Test for error value was conducted in Balloon, Cancer, Diabetes and Pendigits datasets. The following subsections describes in detail on test for error value for each dataset.

#### 4.8.1.1 Balloon Data

Table 4.33: Comparison mean and standard deviation Balloon data error value.

Balloon dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	0.0740	0.0146	0.0171	0.0759
Standard Deviation ( $SD$ )	0.0586	0.0112	0.0147	0.0637

Based on Table 4.33, difference mean between MSE cost function and BL cost function is 0.0594, difference mean between MSE cost function and MM cost function is 0.0569, difference mean between MSE cost function and IC cost function is 0.0019,

difference mean between BL cost function and MM cost function is 0.0025, difference mean between BL cost function and IC cost function is 0.0613, while the difference mean between MM cost function and IC cost function is 0.0588. Based on the calculation below, t value for the MSE and BL is 2.7, t value for the MSE and MM is 2.4632, t value for the MSE and IC is -0.0492, t value for the BL and MM is -0.3086, t value for the BL and IC is -2.5975, while t value for the MM and IC is -2.3806. We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

$H_0$ : population means are the same,  $\mu_1 = \mu_2$  in term of error value

$H_1$ : population means are not the same,  $\mu_1 \neq \mu_2$  in term of error value

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(n_1+n_2)-2] = [(10+10)-2] = 18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\frac{SD_1}{\sqrt{N_1}} + \frac{SD_2}{\sqrt{N_2}}} \quad \text{or} \quad t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{SD_1^2}{N_1} + \frac{SD_2^2}{N_2}}}$$

$$t = \frac{0.0740 - 0.0146}{\frac{0.0586}{\sqrt{10}} + \frac{0.0112}{\sqrt{10}}} \quad \text{or} \quad t = \frac{0.0740 - 0.0146}{\sqrt{\frac{0.0586^2}{10} + \frac{0.0112^2}{10}}}$$

$$t = \frac{0.0594}{0.0185 + 0.0035} = \frac{0.0594}{0.0220}$$

$$t = 2.7$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{0.0740 - 0.0171}{\frac{0.0586}{\sqrt{10}} + \frac{0.0147}{\sqrt{10}}} = \frac{0.0569}{0.0185 + 0.0046}$$

$$t = 2.4632$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{0.0740 - 0.0759}{\frac{0.0586}{\sqrt{10}} + \frac{0.0637}{\sqrt{10}}} = \frac{-0.0019}{0.0185 + 0.0201}$$

$$t = -0.0492$$

While t-test for BL and MM is calculated as below:

$$t = \frac{0.0146 - 0.0171}{\frac{0.0112}{\sqrt{10}} + \frac{0.0147}{\sqrt{10}}} = \frac{-0.0025}{0.0035 + 0.0046}$$

$$t = -0.3086$$

While t-test for BL and IC is calculated as below:

$$t = \frac{0.0146 - 0.0759}{\frac{0.0112}{\sqrt{10}} + \frac{0.0637}{\sqrt{10}}} = \frac{-0.0613}{0.0035 + 0.0201}$$

$$t = -2.5975$$

And t-test for MM and IC is calculated as below:

$$t = \frac{0.0171 - 0.0759}{\frac{0.0147}{\sqrt{10}} + \frac{0.0637}{\sqrt{10}}} = \frac{-0.0588}{0.0046 + 0.0201}$$

$$t = -2.3806$$

Through experiments, t value for MSE and IC, also BL and MM are within the range of the critical region. t value for MSE and BL is outside the range of the critical region ( $2.7 > 1.734$ ), t value for MSE and MM is also outside the range of critical region ( $2.4632 > 1.734$ ), t value for BL and IC is also outside the range of critical region ( $-2.5975 < -1.734$ ) and t value for MM and IC is also outside the range of critical region ( $-2.3806 < -1.734$ ). Therefore, it shows that there is significant difference between the data pattern and hypothesis  $H_0$  is rejected.

#### 4.8.1.2 Cancer data

Table 4.34: Comparison mean and standard deviation Cancer data error value.

Cancer dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	0.0149	0.1215	0.0040	0.0170
Standard Deviation ( $SD$ )	0.0139	0.1082	0.0039	0.0178

Based on Table 4.34, difference mean between MSE cost function and BL cost function is 0.1066, difference mean between MSE cost function and MM cost function is 0.0109, difference mean between MSE cost function and IC cost function is 0.0021,

difference mean between BL cost function and MM cost function is 0.1175, difference mean between BL cost function and IC cost function is 0.1045, while the difference mean between MM cost function and IC cost function is 0.013. Based on the calculation below, t value for the MSE and BL is -2.7617, t value for the MSE and MM is 1.9464, t value for the MSE and IC is -0.21, t value for the BL and MM is 3.3192, t value for the BL and IC is 2.6256, while t value for the MM and IC is -1.9118. We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

$H_0$ : population means are the same,  $\mu_1 = \mu_2$  in term of error value

$H_1$ : population means are not the same,  $\mu_1 \neq \mu_2$  in term of error value

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(10+10)-2]=18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{0.0149 - 0.1215}{\frac{0.0139}{\sqrt{10}} + \frac{0.1082}{\sqrt{10}}} = \frac{-0.1066}{0.0044 + 0.0342}$$

$$t = -2.7617$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{0.0149 - 0.0040}{\frac{0.0139}{\sqrt{10}} + \frac{0.0039}{\sqrt{10}}} = \frac{0.0109}{0.0044 + 0.0012}$$

$$t = 1.9464$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{0.0149 - 0.0170}{\frac{0.0139}{\sqrt{10}} + \frac{0.0178}{\sqrt{10}}} = \frac{-0.0021}{0.0044 + 0.0056}$$

$$t = -0.21$$

While t-test for BL and MM is calculated as below:

$$t = \frac{0.1215 - 0.0040}{\frac{0.1082}{\sqrt{10}} + \frac{0.0039}{\sqrt{10}}} = \frac{0.1175}{0.0342 + 0.0012}$$

$$t = 3.3192$$

While t-test for BL and IC is calculated as below:

$$t = \frac{0.1215 - 0.0170}{\frac{0.1082}{\sqrt{10}} + \frac{0.0178}{\sqrt{10}}} = \frac{0.1045}{0.0342 + 0.0056}$$

$$t = 2.6256$$

And t-test for MM and IC is calculated as below:

$$t = \frac{0.0040 - 0.0170}{\frac{0.0039}{\sqrt{10}} + \frac{0.0178}{\sqrt{10}}} = \frac{-0.013}{0.0012 + 0.0056}$$

$$t = -1.9118$$

Through experiments,  $t$  value for MSE and IC is within the range of the critical region.  $t$  value for MSE and BL is outside the range of the critical region ( $-2.7617 < -1.734$ ),  $t$  value for MSE and MM is also outside the range of critical region ( $1.9464 > 1.734$ ),  $t$  value for BL and MM is also outside the range of critical region  $3.3192 > 1.734$ ),  $t$  value for BL and IC is also outside the range of critical region  $2.6256 > 1.734$ ) and  $t$  value for MM and IC is also outside the range of critical region ( $-1.9118 < -1.734$ ). Therefore, it shows that there is significant difference between the data pattern and hypothesis  $H_0$  is rejected.

#### 4.8.1.3 Diabetes data

Table 4.35: Comparison mean and standard deviation Diabetes data error value.

Diabetes dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	0.0164	0.0162	0.0046	0.0201
Standard Deviation ( $SD$ )	0.0155	0.0151	0.0033	0.0182

Based on Table 4.35, difference mean between MSE cost function and BL cost function is 0.0002, difference mean between MSE cost function and MM cost function is 0.0118, difference mean between MSE cost function and IC cost function is 0.0037, difference mean between BL cost function and MM cost function is 0.0116, difference mean between BL cost function and IC cost function is 0.0039, while the difference mean between MM cost function and IC cost function is 0.0155. Based on the calculation below,  $t$  value for the MSE and BL is 0.0206,  $t$  value for the MSE and MM is 2.0,  $t$  value for the MSE and IC is  $-0.3458$ ,  $t$  value for the BL and MM is 2.0,  $t$  value for the BL and IC is  $-0.3679$ , while  $t$  value for the MM and IC is  $-2.2794$ . We want to

test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

$H_0$ : population means are the same,  $\mu_1 = \mu_2$  in term of error value

$H_1$ : population means are not the same,  $\mu_1 \neq \mu_2$  in term of error value

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $(n_1+n_2)-2 = (10+10)-2=18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{0.0164 - 0.0162}{\frac{0.0155}{\sqrt{10}} + \frac{0.0151}{\sqrt{10}}} = \frac{0.0002}{0.0049 + 0.0048}$$

$$t = 0.0206$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{0.0164 - 0.0046}{\frac{0.0155}{\sqrt{10}} + \frac{0.0033}{\sqrt{10}}} = \frac{0.0118}{0.0049 + 0.0010}$$

$$t = 2.0$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{0.0164 - 0.0201}{\frac{0.0155}{\sqrt{10}} + \frac{0.0182}{\sqrt{10}}} = \frac{-0.0037}{0.0049 + 0.0058}$$

$$t = -0.3458$$

While t-test for BL and MM is calculated as below:

$$t = \frac{0.0162 - 0.0046}{\frac{0.0151}{\sqrt{10}} + \frac{0.0033}{\sqrt{10}}} = \frac{0.0116}{0.0048 + 0.0010}$$

$$t = 2.0$$

While t-test for BL and IC is calculated as below:

$$t = \frac{0.0162 - 0.0201}{\frac{0.0151}{\sqrt{10}} + \frac{0.0182}{\sqrt{10}}} = \frac{-0.0039}{0.0048 + 0.0058}$$

$$t = -0.3679$$

And t-test for MM and IC is calculated as below:

$$t = \frac{0.0046 - 0.0201}{\frac{0.0033}{\sqrt{10}} + \frac{0.0182}{\sqrt{10}}} = \frac{-0.0155}{0.0010 + 0.0058}$$

$$t = -2.2794$$

Through experiments, t value for MSE and BL, MSE and IC, BL and IC are within the range of the critical region. t value for MSE and MM, BL and MM are outside the range of critical region with  $(2.0 > 1.734)$ . t value for MM and IC is also outside the range of critical region  $(-2.2794 < -1.734)$ . It shows that we could not draw to a conclusion to accept or reject hypothesis  $H_0$  for Diabetes dataset.

#### 4.8.1.4 Pendigits data

Table 4.36 Comparison mean and standard deviation Pendigits data error value.

Pendigits dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	0.2400	0.2087	0.1027	0.0774
Standard Deviation ( $SD$ )	0.2221	0.2348	0.1026	0.1217

Based on Table 4.36, difference mean between MSE cost function and BL cost function is 0.0313, difference mean between MSE cost function and MM cost function is 0.1373, difference mean between MSE cost function and IC cost function is 0.1626, difference mean between BL cost function and MM cost function is 0.106, difference mean between BL cost function and IC cost function is 0.1313, while the difference mean between MM cost function and IC cost function is 0.0253. Based on the calculation below, t value for the MSE and BL is 0.2166, t value for the MSE and MM is 1.3382, t value for the MSE and IC is 1.4959, t value for the BL and MM is 0.9934, t value for the BL and IC is 1.1640, and while t value for the MM and IC is 0.3568. We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

$H_0$ : population means are the same,  $\mu_1 = \mu_2$  in term of error value

$H_1$ : population means are not the same,  $\mu_1 \neq \mu_2$  in term of error value

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $(n_1+n_2)-2 = (10+10)-2=18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{0.2400 - 0.2087}{\frac{0.2221}{\sqrt{10}} + \frac{0.2348}{\sqrt{10}}} = \frac{0.0313}{0.0702 + 0.0743}$$

$$t = 0.2166$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{0.2400 - 0.1027}{\frac{0.2221}{\sqrt{10}} + \frac{0.1026}{\sqrt{10}}} = \frac{0.1373}{0.0702 + 0.0324}$$

$$t = 1.3382$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{0.2400 - 0.0774}{\frac{0.2221}{\sqrt{10}} + \frac{0.1217}{\sqrt{10}}} = \frac{0.1626}{0.0702 + 0.0385}$$

$$t = 1.4959$$

While t-test for BL and MM is calculated as below:

$$t = \frac{0.2084 - 0.1027}{\frac{0.2348}{\sqrt{10}} + \frac{0.1026}{\sqrt{10}}} = \frac{0.106}{0.0742 + 0.0324}$$

$$t = 0.9934$$

While t-test for BL and IC is calculated as below:

$$t = \frac{0.2084 - 0.0774}{\frac{0.2348}{\sqrt{10}} + \frac{0.1217}{\sqrt{10}}} = \frac{0.1313}{0.0742 + 0.0385}$$

$$t = 1.1640$$

And t-test for MM and IC is calculated as below:

$$t = \frac{0.1027 - 0.0774}{\frac{0.1026}{\sqrt{10}} + \frac{0.1217}{\sqrt{10}}} = \frac{0.0253}{0.0324 + 0.0385}$$

$$t = 0.3568$$

Through experiments, t value is within the range of the critical region. It shows that there is no significant difference since t value from table of critical values is 1.734. Hence, null hypothesis  $H_0$  is accepted.

#### 4.8.1.5 Overall T-Test Results for Error Value

Based on the T-test, conclusion could not be drawn by Diabetes dataset because three comparisons showing  $H_0$  accepted and other three comparisons showing  $H_0$  rejected results. Besides that, based on the T-test, hypothesis  $H_0$  is accepted for Pendigits dataset. This shows that the cost functions doesn't really have significant different for Pendigits dataset. Thus, we could not draw to a conclusion from Pendigits dataset too.

Meanwhile, Balloon datasets and cancer dataset can be used to conclude based on the T-test. It is because significant difference between the data pattern was shown and hypothesis  $H_0$  was rejected. For Balloon dataset MSE and BL, MSE and MM, BL and IC, MM and IC are outside the range of the critical region. This means there are significant differences between those cost functions. The MSE and IC, BL and MM are within the range of the critical region meaning that there is no significant difference between those cost functions.

For cancer dataset MSE and BL, MSE and MM, BL and MM, BL and IC, MM and IC are outside the range of the critical region. Therefore, it shows that there is significant difference between the data pattern and hypothesis null is rejected. Only MSE and IC are within the range of the critical region.

## 4.8.2 T-test for Convergence Time

T-Test for convergence time was conducted in Balloon, Cancer, Diabetes and Pendigits datasets. The following subsections describes in detail on test for convergence time for each dataset.

### 4.8.2.1 Balloon Data

Table 4.37: Comparison mean and standard deviation Balloon data Convergence Time

Balloon dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	82	75	77	85
Standard Deviation ( $SD$ )	29.7396	30.2765	28.6938	26.3523

Based on Table 4.37, difference mean between MSE cost function and BL cost function is 7, difference mean between MSE cost function and MM cost function is 5, difference mean between MSE cost function and IC cost function is 3, difference mean between BL cost function and MM cost function is 2, difference mean between BL cost function and IC cost function is 10, while the difference mean between MM cost function and IC cost function is 8. Based on the calculation below, t value for the MSE and BL is 0.3688, t value for the MSE and MM is 0.27059, t value for the MSE and IC is  $-0.1691$ , t value for the BL and MM is  $-0.1072$ , t value for the BL and IC is  $-0.5584$ , while t value for the MM and IC is  $-0.4596$ . We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

$H_0$ : population means are the same,  $\mu_1 = \mu_2$  in term of convergence time

$H_1$ : population means are not the same,  $\mu_1 \neq \mu_2$  in term of convergence time

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(n_1+n_2)-2] = [(10+10)-2] = 18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{82 - 75}{\frac{29.7396}{\sqrt{10}} + \frac{30.2765}{\sqrt{10}}} = \frac{7}{9.4045 + 9.5743}$$

$$t = 0.3688$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{82 - 77}{\frac{29.7396}{\sqrt{10}} + \frac{28.6938}{\sqrt{10}}} = \frac{5}{9.4045 + 9.0738}$$

$$t = 0.27059$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{82 - 85}{\frac{29.7396}{\sqrt{10}} + \frac{26.3523}{\sqrt{10}}} = \frac{-3}{9.4045 + 8.3333}$$

$$t = -0.1691$$

While t-test for BL and MM is calculated as below:

$$t = \frac{75 - 77}{\frac{30.2765}{\sqrt{10}} + \frac{28.6938}{\sqrt{10}}} = \frac{-2}{9.5743 + 9.0738}$$

$$t = -0.1072$$

While t-test for BL and IC is calculated as below:

$$t = \frac{75 - 85}{\frac{30.2765}{\sqrt{10}} + \frac{26.3523}{\sqrt{10}}} = \frac{-10}{9.5743 + 8.3333}$$

$$t = -0.5584$$

And t-test for MM and IC is calculated as below:

$$t = \frac{77 - 85}{\frac{28.6938}{\sqrt{10}} + \frac{26.3523}{\sqrt{10}}} = \frac{-8}{9.0738 + 8.3333}$$

$$t = -0.4596$$

Through experiments, t value is within the range of the critical region. It shows that there is no significant difference since t value from table of critical values is 1.734. Hence, null hypothesis  $H_0$  is accepted.

### 4.8.2.2 Cancer Data

Table 4.38: Comparison mean and standard deviation Cancer data Convergence Time

Cancer dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	1629	1593	1968	1201
Standard Deviation ( $SD$ )	488.2042	809.8841	899.3677	589.9426

Based on Table 4.38, difference mean between MSE cost function and BL cost function is 36, difference mean between MSE cost function and MM cost function is 336, difference mean between MSE cost function and IC cost function is 428, difference mean between BL cost function and MM cost function is 375, difference mean between BL cost function and IC cost function is 394, while the difference mean between MM cost function and IC cost function is 767. Based on the calculation below, t value for the MSE and BL is 0.0877, t value for the MSE and MM is  $-0.7726$ , t value for the MSE and IC is 1.2553, t value for the BL and MM is  $-0.69379$ , t value for the BL and IC is 0.8901, while t value for the MM and IC is 1.6286. We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

$H_0$ : population means are the same,  $\mu_1 = \mu_2$  in term of convergence time

$H_1$ : population means are not the same,  $\mu_1 \neq \mu_2$  in term of convergence time

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(n_1+n_2)-2] = [(10+10)-2] = 18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{1629-1593}{\frac{488.2042}{\sqrt{10}} + \frac{809.8841}{\sqrt{10}}} = \frac{36}{154.3837 + 256.1078}$$

$$t = 0.0877$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{1629-1968}{\frac{488.2042}{\sqrt{10}} + \frac{899.3677}{\sqrt{10}}} = \frac{-339}{154.3837 + 284.4050}$$

$$t = -0.7726$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{1629-1201}{\frac{488.2042}{\sqrt{10}} + \frac{589.9426}{\sqrt{10}}} = \frac{428}{154.3837 + 186.5562}$$

$$t = 1.2553$$

While t-test for BL and MM is calculated as below:

$$t = \frac{1593-1968}{\frac{809.8841}{\sqrt{10}} + \frac{899.3677}{\sqrt{10}}} = \frac{-375}{256.1078 + 284.4050}$$

$$t = -0.69379$$

While t-test for BL and IC is calculated as below:

$$t = \frac{1593 - 1201}{\frac{809.8841}{\sqrt{10}} + \frac{589.9426}{\sqrt{10}}} = \frac{394}{256.1078 + 186.5562}$$

$$t = 0.8901$$

And t-test for MM and IC is calculated as below:

$$t = \frac{1968 - 1201}{\frac{899.3677}{\sqrt{10}} + \frac{589.9426}{\sqrt{10}}} = \frac{767}{284.4050 + 186.5562}$$

$$t = 1.6286$$

Through experiments, t value is within the range of the critical region. It shows that there is no significant difference since t value from table of critical values is 1.734. Hence, null hypothesis  $H_0$  is accepted.

#### 4.8.2.3 Diabetes Data

Table 4.39: Comparison mean and standard deviation Diabetes data Convergence Time

Diabetes dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	1567	1674	1968	1289
Standard Deviation ( $SD$ )	929.7437	835.4533	892.4100	666.1740

Based on Table 4.39, difference mean between MSE cost function and BL cost function is 107, difference mean between MSE cost function and MM cost function is

401, difference mean between MSE cost function and IC cost function is 278, difference mean between BL cost function and MM cost function is 294, difference mean between BL cost function and IC cost function is 385, while the difference mean between MM cost function and IC cost function is 679. Based on the calculation below, t value for the MSE and BL is  $-0.1917$ , t value for the MSE and MM is  $0.6959$ , t value for the MSE and IC is  $0.5509$ , t value for the BL and MM is  $-0.5381$ , t value for the BL and IC is  $0.8108$ , while t value for the MM and IC is  $1.4299$ . We want to test this experiment at significance level of  $0.05$ . So, the confidence interval (CI) is equal to  $95\%$  and when we check with significant table, the critical value of t is  $1.734$ .

$H_0$ : population means are the same,  $\mu_1 = \mu_2$  in term of convergence time

$H_1$ : population means are not the same,  $\mu_1 \neq \mu_2$  in term of convergence time

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(n_1+n_2)-2] = [(10+10)-2] = 18$

Critical region:  $t \geq +1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{1567 - 1674}{\frac{929.7437}{\sqrt{10}} + \frac{835.4533}{\sqrt{10}}} = \frac{-107}{294.0108 + 264.1935}$$

$$t = -0.1917$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{1567 - 1968}{\frac{929.7437}{\sqrt{10}} + \frac{892.41}{\sqrt{10}}} = \frac{-401}{294.0108 + 282.2048}$$

$$t = 0.6959$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{1567 - 1289}{\frac{929.7437}{\sqrt{10}} + \frac{666.1740}{\sqrt{10}}} = \frac{278}{294.0108 + 210.6627}$$

$$t = 0.5509$$

While t-test for BL and MM is calculated as below:

$$t = \frac{1674 - 1968}{\frac{835.4533}{\sqrt{10}} + \frac{892.41}{\sqrt{10}}} = \frac{-294}{264.1935 + 282.2048}$$

$$t = -0.5381$$

While t-test for BL and IC is calculated as below:

$$t = \frac{1674 - 1289}{\frac{835.4533}{\sqrt{10}} + \frac{666.1740}{\sqrt{10}}} = \frac{385}{264.1935 + 210.6627}$$

$$t = 0.8108$$

And t-test for MM and IC is calculated as below:

$$t = \frac{1968 - 1289}{\frac{892.4100}{\sqrt{10}} + \frac{666.1740}{\sqrt{10}}} = \frac{679}{282.2048 + 210.6627}$$

$$t = 1.4299$$

Through experiments, t value is within the range of the critical region. It shows that there is no significant difference since t value from table of critical values is 1.734. Hence, hypothesis  $H_0$  is accepted.

#### 4.8.2.4 Pendigits Data

Table 4.40: Comparison mean and standard deviation Pendigits data Convergence Time

Pendigits dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	16021.3	15108	21973	15404
Standard Deviation ( $SD$ )	9726.6151	7848.5566	10568.7643	6806.6148

Based on Table 4.40, difference mean between MSE cost function and BL cost function is 913.3, difference mean between MSE cost function and MM cost function is 5951.7, difference mean between MSE cost function and IC cost function is 617.3, difference mean between BL cost function and MM cost function is 6865, difference mean between BL cost function and IC cost function is 296, while the difference mean between MM cost function and IC cost function is 6569. Based on the calculation below, t value for the MSE and BL is 0.1643, t value for the MSE and MM is 0.9274, t value for the MSE and IC is 0.1181, t value for the BL and MM is  $-1.1787$ , t value for the BL and IC is 0.0639, while t value for the MM and IC is 1.1955. We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

H0: population means are the same,  $\mu_1 = \mu_2$  in term of convergence time

H1: population means are not the same,  $\mu_1 \neq \mu_2$  in term of convergence time

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(n_1+n_2)-2] = [(10+10)-2] = 18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{16021.3 - 15108}{\frac{9726.6151}{\sqrt{10}} + \frac{7848.5566}{\sqrt{10}}} = \frac{913.3}{3075.8258 + 2481.9315}$$

$$t = 0.1643$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{16021.3 - 21973}{\frac{9726.6151}{\sqrt{10}} + \frac{10568.7643}{\sqrt{10}}} = \frac{-5951.7}{3075.8258 + 3342.1367}$$

$$t = 0.9274$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{16021.3 - 15404}{\frac{9726.6151}{\sqrt{10}} + \frac{6806.6148}{\sqrt{10}}} = \frac{617.3}{3075.8258 + 2152.4406}$$

$$t = 0.1181$$

While t-test for BL and MM is calculated as below:

$$t = \frac{15108 - 21973}{\frac{7848.5566}{\sqrt{10}} + \frac{10568.7643}{\sqrt{10}}} = \frac{-6865}{2481.9315 + 3342.1367}$$

$$t = -1.1787$$

While t-test for BL and IC is calculated as below:

$$t = \frac{15108 - 15404}{\frac{7848.5566}{\sqrt{10}} + \frac{6806.6148}{\sqrt{10}}} = \frac{-296}{2481.9315 + 2152.4415}$$

$$t = 0.0639$$

And t-test for MM and IC is calculated as below:

$$t = \frac{21973 - 15404}{\frac{10568.7643}{\sqrt{10}} + \frac{6806.6148}{\sqrt{10}}} = \frac{6569}{3342.1367 + 2152.4415}$$

$$t = 1.1955$$

Through experiments, t value is within the range of the critical region. It shows that there is no significant difference since t value from table of critical values is 1.734. Hence, hypothesis  $H_0$  is accepted.

#### **4.8.2.5 Overall T-Test Results for Convergence Time**

Based on the T-test, conclusion could not be drawn for convergence time because the all the datasets (Balloon, Cancer, Diabetes and Pendigits) are showing hypothesis  $H_0$  accepted. This means that the cost functions doesn't really have significant different in term of convergence speed. Thus, we could not draw to a conclusion from convergence time.

### 4.8.3 T-test for Accuracy

T-Test for accuracy was conducted in Balloon, Cancer, Diabetes and Pendigits datasets. The following subsections describes in detail on test for accuracy for each dataset.

#### 4.8.3.1 Balloon Data

T-test was not conducted for Balloon dataset for accuracy. It is because the accuracy is all same for all the datasets throughout all the samples. So, We could conclude that hypothesis  $H_0$  accepted. Thus, we could not draw to a conclusion from Balloon dataset from accuracy.

#### 4.8.3.2 Cancer Data

Table 4.41: Comparison mean and standard deviation Cancer data Accuracy

Cancer dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	0.7260	0.7630	0.8390	0.7130
Standard Deviation ( $SD$ )	0.1533	0.1082	0.0390	0.1807

Based on Table 4.41, difference mean between MSE cost function and BL cost function is 0.037, difference mean between MSE cost function and MM cost function is 0.113, difference mean between MSE cost function and IC cost function is 0.013, difference mean between BL cost function and MM cost function is 0.076, difference mean between BL cost function and IC cost function is 0.05, while the difference mean between MM cost function and IC cost function is 0.126. Based on the calculation below, t value for the MSE and BL is  $-0.4474$ , t value for the MSE and MM is  $-1.8586$ , t value for the MSE and IC is  $0.1231$ , t value for the BL and MM is  $-1.6344$ , t value for the BL and IC is  $0.5476$ , while t value for the MM and IC is  $1.8156$ . We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

H0: population means are the same,  $\mu_1 = \mu_2$  in term of data accuracy

H1: population means are not the same,  $\mu_1 \neq \mu_2$  in term of data accuracy

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(n_1+n_2)-2] = [(10+10)-2] = 18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{0.7260 - 0.7630}{\frac{0.1533}{\sqrt{10}} + \frac{0.1082}{\sqrt{10}}} = \frac{-0.037}{0.0485 + 0.0342}$$

$$t = -0.4474$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{0.7260 - 0.8390}{\frac{0.1533}{\sqrt{10}} + \frac{0.0390}{\sqrt{10}}} = \frac{-0.113}{0.0485 + 0.0123}$$

$$t = -1.8586$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{0.7260 - 0.7130}{\frac{0.1533}{\sqrt{10}} + \frac{0.1807}{\sqrt{10}}} = \frac{0.013}{0.0485 + 0.0571}$$

$$t = 0.1231$$

While t-test for BL and MM is calculated as below:

$$t = \frac{0.7630 - 0.8390}{\frac{0.1082}{\sqrt{10}} + \frac{0.0390}{\sqrt{10}}} = \frac{-0.076}{0.0342 + 0.0123}$$

$$t = -1.6344$$

While t-test for BL and IC is calculated as below:

$$t = \frac{0.7630 - 0.7130}{\frac{0.1082}{\sqrt{10}} + \frac{0.1807}{\sqrt{10}}} = \frac{0.05}{0.0342 + 0.0571}$$

$$t = 0.5476$$

And t-test for MM and IC is calculated as below:

$$t = \frac{0.8390 - 0.7130}{\frac{0.0390}{\sqrt{10}} + \frac{0.1807}{\sqrt{10}}} = \frac{0.126}{0.0123 + 0.0571}$$

$$t = 1.8156$$

Through experiments, t value for MSE and BL, MSE and IC, BL and MM, BL and IC are within the range of the critical region. t value for MSE and MM, MM and IC, are outside the range of critical region with  $(-1.8586 < -1.734)$ ,  $(1.8156 > 1.734)$  each. It shows that MSE and IC show significance difference with MM.

#### 4.8.3.3 Diabetes Data

Table 4.42: Comparison mean and standard deviation Diabetes data Accuracy

Diabetes dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	0.2680	0.2680	0.3560	0.2560
Standard Deviation ( $SD$ )	0.0797	0.0796	0.0378	0.0655

Based on Table 4.42, difference mean between MSE cost function and BL cost function is 0.0, difference mean between MSE cost function and MM cost function is 0.088, difference mean between MSE cost function and IC cost function is 0.012, difference mean between BL cost function and MM cost function is 0.088, difference mean between BL cost function and IC cost function is 0.12, while the difference mean between MM cost function and IC cost function is 0.1. Based on the calculation below, t value for the MSE and BL is 0, t value for the MSE and MM is 2.3656, t value for the

MSE and IC is 0.2614, t value for the BL and MM is 2.3656, t value for the BL and IC is 0.2614, while t value for the MM and IC is 3.0581. We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

H0: population means are the same,  $\mu_1 = \mu_2$  in term of data accuracy

H1: population means are not the same,  $\mu_1 \neq \mu_2$  in term of data accuracy

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(n_1+n_2)-2] = [(10+10)-2] = 18$

Critical region:  $t \geq +1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{0.2680 - 0.2680}{\frac{0.0797}{\sqrt{10}} + \frac{0.0796}{\sqrt{10}}}$$

$$t = 0.0$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{0.2680 - 0.3560}{\frac{0.0797}{\sqrt{10}} + \frac{0.0378}{\sqrt{10}}} = \frac{-0.088}{0.0252 + 0.0120}$$

$$t = 2.3656$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{0.2680 - 0.2560}{\frac{0.0797}{\sqrt{10}} + \frac{0.0655}{\sqrt{10}}} = \frac{0.012}{0.0252 + 0.0207}$$

$$t = 0.2614$$

While t-test for BL and MM is calculated as below:

$$t = \frac{0.2680 - 0.3560}{\frac{0.0796}{\sqrt{10}} + \frac{0.0378}{\sqrt{10}}} = \frac{-0.088}{0.0252 + 0.0120}$$

$$t = 2.3656$$

While t-test for BL and IC is calculated as below:

$$t = \frac{0.2680 - 0.2560}{\frac{0.0796}{\sqrt{10}} + \frac{0.0655}{\sqrt{10}}} = \frac{0.012}{0.0252 + 0.0207}$$

$$t = 0.2614$$

And t-test for MM and IC is calculated as below:

$$t = \frac{0.3560 - 0.2560}{\frac{0.0378}{\sqrt{10}} + \frac{0.0655}{\sqrt{10}}} = \frac{0.1}{0.0120 + 0.0207}$$

$$t = 3.0581$$

Through experiments, t value for MSE and BL, MSE and IC, BL and IC are within the range of the critical region. t value for MSE and MM, BL and MM, MM and IC , are outside the range of critical region with  $(2.3656 > 1.734)$ ,  $(2.3656 > 1.734)$ ,  $(3.0581 > 1.734)$  each. It shows that MSE and IC and BL showed significance difference with MM. Obviously MM cost function perform significantly better then other cost functions for Diabetes data. Besides that MSE and MM, BL and MM showed the same t value. Moreover MSE and BL doesn't have any differences because the t value is 0.

#### 4.8.3.4 Pendigits Data

Table 4.43: Comparison mean and standard deviation Pendigits data Accuracy

Pendigits dataset	MSE	BL	MM	IC
Number of Observations ( $N$ )	10	10	10	10
Mean ( $\bar{x}$ )	0.8860	0.7010	0.8240	0.8280
Standard Deviation ( $SD$ )	0.0222	0.2339	0.0996	0.0970

Based on Table 4.43, difference mean between MSE cost function and BL cost function is 0.185, difference mean between MSE cost function and MM cost function is 0.062, difference mean between MSE cost function and IC cost function is 0.058, difference mean between BL cost function and MM cost function is 0.123, difference mean between BL cost function and IC cost function is 0.127, while the difference mean between MM cost function and IC cost function is 0.004. Based on the calculation below, t value for the MSE and BL is 2.2840, t value for the MSE and MM is 1.6104, t value for the MSE and IC is 1.5397, t value for the BL and MM is 1.1659, t value for the BL and IC is 1.2133, while t value for the MM and IC is 0.0643. We want to test this experiment at significance level of 0.05. So, the confidence interval (CI) is equal to 95% and when we check with significant table, the critical value of t is 1.734.

H0: population means are the same,  $\mu_1 = \mu_2$  in term of data accuracy

H1: population means are not the same,  $\mu_1 \neq \mu_2$  in term of data accuracy

Significance level  $\alpha = 0.05$

Degree of freedom (DoF) =  $[(n_1+n_2)-2] = [(10+10)-2] = 18$

Critical region:  $t \geq + 1.734$  or  $t < -1.734$

When apply the t-test, we assumed the sample is come from a normally distributed population. t-test for MSE and BL is calculated as below:

$$t = \frac{0.8860 - 0.7010}{\frac{0.0222}{\sqrt{10}} + \frac{0.2339}{\sqrt{10}}} = \frac{0.185}{0.0070 + 0.0740}$$

$$t = 2.2840$$

While t-test for MSE and MM is calculated as below:

$$t = \frac{0.8860 - 0.8240}{\frac{0.0222}{\sqrt{10}} + \frac{0.0996}{\sqrt{10}}} = \frac{0.062}{0.0070 + 0.0315}$$

$$t = 1.6104$$

While t-test for MSE and IC is calculated as below:

$$t = \frac{0.8860 - 0.8280}{\frac{0.0222}{\sqrt{10}} + \frac{0.0970}{\sqrt{10}}} = \frac{0.058}{0.0070 + 0.0307}$$

$$t = 1.5385$$

While t-test for BL and MM is calculated as below:

$$t = \frac{0.7010 - 0.8240}{\frac{0.2339}{\sqrt{10}} + \frac{0.0996}{\sqrt{10}}} = \frac{-0.123}{0.0740 + 0.0315}$$

$$t = 1.1659$$

While t-test for BL and IC is calculated as below:

$$t = \frac{0.7010 - 0.8280}{\frac{0.2339}{\sqrt{10}} + \frac{0.0970}{\sqrt{10}}} = \frac{-0.127}{0.0740 + 0.0307}$$

$$t = 1.2130$$

And t-test for MM and IC is calculated as below:

$$t = \frac{0.8240 - 0.8280}{\frac{0.0996}{\sqrt{10}} + \frac{0.0970}{\sqrt{10}}} = \frac{-0.004}{0.0315 + 0.0307}$$

$$t = 0.0643$$

Through experiments, t value for MSE and MM, MSE and IC, BL and MM, BL and IC, MM and IC are within the range of the critical region. t value for MSE and BL is outside the range of critical region with ( $2.2840 > 1.734$ ). It shows that MSE and BL cost functions only have showed significance difference. And all other cost function doesn't have significance difference. Hence, hypothesis  $H_0$  is accepted

#### 4.8.3.5 Overall T-Test Results for Accuracy

Based on the T-test, conclusion could be drawn for accuracy in Cancer dataset where MSE and IC cost function showed significance difference with MM cost function. MM cost function can be concluded to perform better compared to MSE and also IC cost function. For Diabetes dataset, we could draw to a conclusion that MM is better cost function compared to MSE, BL and IC as a overall since there is significance difference between those cost unction and MM cost function. MM cost function obviously a better cost function in term of accuracy compared to MSE BL and IC cost functions.

## 4.9 Summary

Three Term BP with MSE Cost Function, Three Term BP with Bernouli Cost Function, Three Term BP with Modified Cost Function and Three Term BP with Improved Cost Function was tested on Balloon, Cancer, Diabetes and Pendigits datasets.

The results have shown that Three Term Backpropagation with other cost function instead of Mean Square error performed better. Consequently, it shows that the Mean Square Error is not always the best cost function for Three Term BP network. We could observe this conclusion from each dataset. MSE cost function performed badly in all the dataset. When T-test was conducted, MSE cost function performed significantly bad in these datasets compared to other cost functions.

In term of comparison parameter such as error value, MM cost function able to produce minimum error value compared to the rest of cost function. This is true for three datasets that are Cancer, Diabetes and Pendigits dataset. For balloon datasets BL cost function produced a minimum error value. Even though, T-test showed that Diabetes and Pendigits could not be used to draw a conclusion because the data falls within the critical region, which means the hypothesis ( $H_0$ ) is accepted. But based on Balloon and Cancer dataset MM cost function can be considered a good cost function when user requires less error value to be produced which more prone to high accuracy of dataset. It is because there is significant difference between BL and MM for Cancer dataset where the t value is outside the range of the critical region but not for Balloon Dataset. This means BL cost function that performed well in Balloon dataset doesn't have significance difference with MM cost function.

In term of convergence time, the BL cost function performed well in the Balloon dataset meanwhile IC cost function can successfully complete the epochs with the shortest time for Cancer, Diabetes and Pendigits. Through t-test experiments, all the t values for Balloon, Cancer, Diabetes and are within the range of the critical region. Hence, conclusion could not be drawn for convergence time because all the datasets are showing hypothesis  $H_0$  accepted. This means that the cost functions doesn't really have significant different in term of convergence time. Thus, we could not draw to a conclusion from convergence time.

The results also show that the percentage of successful test result for the MM cost function was significantly higher compared all other three cost function. For each problems that have been tested, MM cost function outperform all other cost function in term of accuracy percentage. For example, the MM cost function produced 75% accurate results for Balloon dataset, 87% for Cancer dataset, 39% for Diabetes dataset and 91% for Pendigits dataset. Based on the T-test, MM cost function showed significance better performance difference with MSE and IC cost function in Cancer dataset. It also showed better significance performance difference compared to MSE, BL and IC in Diabetes dataset. This helps us to draw a conclusion that MM obviously a better cost function in term of accuracy.

A summary of outcomes of this study is given as follow:

- The first outcome of this study is that MSE is not an ideal cost function to be used for Three Term BP.
- Second outcome of this study would be that the IC cost function suitable for those situations when the speed of the method is important in reaching a good solution, although this could not be always the best one in term of significance speed achievement.

- Finally, MM cost function is offering interesting combination of accuracy and reliability. So, MM cost function is the best cost function compared BL, MM and IC cost function. It is suitable for most of the real world problems that requires high accuracy but with a moderate speed.

## **CHAPTER 5**

### **CONCLUSION AND FUTURE WORK**

#### **5.1 Introduction**

This chapter discusses the summary of study that has been conducted to achieve the objective of the study. This chapter also discusses the conclusion about this study and suggestion for future work.

In chapter 1, the objective of the study has been identified together with scope of the study. In chapter 2, four particular cost functions have been studied in detail. Those are Mean Square Error, Bernoulli Cost Function (1994), Modified Cost Function (2001) and Improved Cost Function (2007). In Chap 3, the methodology of this study been developed to do a comparison of those cost function for Three Term BP. The methodology has been followed to conduct this study. Thus, in chapter 4, MSE, BL, MM and IC cost function was exploited in Three Term BP to probe the error, convergence time and accuracy.

## 5.2 Contribution of the Study

Both objectives of the study have been achieved from this study. The study of cost functions of previous researches has been done from 1993 to 2007. Altogether there are 24 cost functions studied. Table 2.4, summarizes all the cost function studied. Mean Square Error (MSE) cost function, Bernoulli (BL) cost function, Modified (MM) cost function and Improved (IC) cost function are especially studied in detail and exploited in this study. Subsection 2.7, contain detail explanation on these four cost functions.

The experimental comparisons of MSE cost function, BL cost function, MM cost function and IC cost function in Three Term BP was been accomplished in this study. These experiments were carried out with four classification problem datasets such as Balloon, Cancer, Diabetes and Pendigits. Comparison study also was carried out and can be found in chapter 4. The overall summarization or outcomes of this study is as below:

- MSE is not an ideal cost function to be used for Three Term BP.
- IC cost function suitable for those situations when the speed of the method is important in reaching a good solution, although this could not be always the best one in term of significance speed achievement.
- MM cost function is offering interesting combination of accuracy and reliability. So, MM cost function is the best cost function compared BL, MM and IC cost function. It is suitable for most of the real world problems that requires high accuracy but with a moderate speed.

### **5.3 Suggestion for future works**

There are several suggestions that can be done to improve Three Term BP algorithm for future work of this project.

- a) Implement Three Term BP with various cost function in simulated datasets for better evaluation and comparison of the cost functions.
- b) Employ other cost function that has not been studied in this study to evaluate the effectiveness of those cost functions.
- c) Various other cost functions can be studied in detail to identify the strength and weakness of each and modification can be done to overcome its weakness and this will increase its performance in future.

## REFERENCE

- Abid, S., Fnaiech, F., and Najim, M. (2001). A Fast Feedforward Training Algorithm Using a Modified Form of the Standard Backpropagation Algorithm. *IEEE Transactions On Neural Networks*, 12(2):424-430.
- Asuncion, A. & Newman, D.J. (2007). UCI Machine Learning Repository [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, School of Information and Computer Science.
- Bi, W., Wang, X., Zong, Z., and Tang, Z.(2004). Modified Error Function with Added Terms for the Backpropagation Algorithm. ISNN 2004, *LNCS 3173*, pp. 338–343, 2004.
- Bossan, M.C., Seixas, J.M., Caloba, L.P., Penha, R.S., and Nadal, J. (1995) A Modified Backpropagation Algorithm For Neural Classifiers. *38th IEEE Midwest Symposium on Circuits and Systems*, 1995. Rio de Janeiro, 562-565.
- Brian A. Telfer and Harold H. Szu.(1994). Energy functions for minimizing misclassification error with minimum-complexity networks. *Neural Networks*.7(5): 809-817
- Charytoniuk, W. and Chen, M.S (2000). Neural Network Design for Short-term Load Forecasting. *International Conference on Electric Utility Deregulation and Restructuring and Power Technologies 2000*. 4-7 April 2000. City University, London. 554-561.
- Chen, Y.Q., Yin, T., and Babri, H.A.(1997). A Stochastic Backpropagation Algorithm for Training Neural Networks. *International Conference on Information, Communications and Signal Processing*, 1997. 9-12 September 1997. Singapore, 703-707.
- Choi, S., Lee, T-W., and Hong, D.(2005).Adaptive error-constrained method for LMS algorithms and applications. *Signal Processing*. 85 (2005):1875–1897
- Chow, M-Y., Menozzi, A., Teeter, J., and Thrower, J.P. (1994).Bernoulli error measure approach to train feedforward Artificial Neural Networks for Classification problems.
- Dhiantravan, Y., and Priemer, R. (1996). Error phenomena of backpropagation

- learning. *Intelligent Engineering Systems Through Artificial Neural Networks*. 6:155-160.
- Drago, G.P., Morando, M., and Ridella, S. (1995). An Adaptive Momentum Back Propagation (AMBP). *Neural Comput & Application*. 3: 213-221.
- Edward, R. J. (2004). *An Introduction to Neural Networks A White Paper*. United States of America: Visual Numerics Inc.
- Fadhlina Izzah Binti Saman (2007). *Three-Term Backpropagation Algorithm For Classification Problem*. Master. Thesis. Universiti Teknologi Malaysia, Skudai.
- Falas, T. and Stafylopatis, A-G. (1999). The Impact of The Error Function Selection in Neural Network-based Classifiers. *International Joint Conference on Neural Network*. 3: 1799-1804.
- Fukuoka, Y., Matsuki, H., Minamitani, H., and Ishida A. (1998). A Modified Backpropagation Method To Avoid False Local Minima. *Neural Networks*. 11: 1059-1072.
- Guijarro-Berdinas, B., Fontenla-Romero, O., Perez-Sanchez, B., and Fraguera, P.(2007). A Linear Learning Method for Multilayer Perceptrons Using Least-Squares. *Lecture Notes in Computer Science 4881*. Berlin Heidelberg: Springer. 365–374.
- Guijarro-Berdi, B., Fontenla-Romero, O., Perez-Sanchez, B., and Fraguera, P.(2007). A Linear Learning Method for Multilayer Perceptrons Using Least-Squares. *Lecture Notes In Computer Science*. 365-374.
- Hahn-Ming Lee, Chih-Ming Chen, Tzong-Ching Huang. (2001). Learning science improvement of back-propagation algorithm by error saturation prevention method. *Neurocomputing*. 41 (2001) 125-143.
- Hauger, S.R.B. (2003). *Ensemble Learned Neural Networks Using Error-Correcting Output Codes and Boosting*. Master Thesis. University of Surrey.
- Herself's Artificial intelligence.<http://herselfsai.com/2007/02/neural-networks.html>.  
Date Accessed:18/12/2007.
- Humpert, B.K.(1994). Improving Back Propagation With A New Error Function. *Neural Networks*.7(8):1191-1192.
- Ibrahim, M. E. and Al-Shams, A.A.M (1997). Transient stability assessment using artificial neural networks. *Electric Power Systems Research*, 40, 7-16.
- In-Cheol Kim, Sung-II Chien. (2002). Speed-up of error ackpropagation algorithm with class-selective relevance. *Neurocomputing*. 48 (2002) 1009– 1014.

- Jiang, M., Deng, B., Wang, B. and Zhong, B.(2003). A Fast Learning Algorithm Of Neural Networks By Changing Error Functions. *IEEE International Conference Neural Networks Signal Processing*. December 14-17, 2003, Nanjing. China. 249-252
- Kandil N., Khorasani, Patel R.V. and Seed V.K. (1993). Optimum Learning Rate For backpropagation Neural Network. *IEEE*. 465-468.
- Kathirvalavakumar, T., and Thangavel, P. (2006). A Modified Backpropagation Training Algorithm for Feedforward Neural Networks. *Neural Processing Letters*. 23:111-119.
- Keogh, E. (2006). The UCR Time Series Data Mining Archive [<http://www.cs.ucr.edu/~eamonn/TSDMA/index.html>]. Riverside CA. University of California - Computer Science & Engineering Department
- Liu, C-S., and Tseng, C-H.(1999).Quadratic optimization method for multilayer neural networks with local error-backpropagation. *International Journal of Systems Science*. 30(8):889 - 898.
- Lv, J., and Yi, Z.(2005). An Improved Backpropagation Algorithm Using Absolute Error Function. *ISNN 2005, LNCS 3496*, pp. 585–590, 2005.Lv et al(2005)
- Mandischer M. (2002). A comparison of evolution strategies and backpropagation for neural network training. *Neurocomputing*. 42 (2002) 87–117.
- Matsuoka Kiyotoshi and Yi Jianqiang (2000). Backpropagation Based on the Logarithmic Error function and Elimination of Local Minima. *IEEE* . 1117-1122.
- Neelakanta, P. S. (1996). Csiszar's Generalized Error Measures for Gradient-descent-based Optimizations in Neural Networks Using the Backpropagation Algorithm. *Connection Science*. 8(1): 79 - 114.
- Neural Networks . Statistica is a trademark of StatSoft, Inc. Date accessed 8/12/2007. <http://www.statsoft.com/textbook/stneunet.html#multilayer>.
- Ng, S.C., Leung, S.H., and Luk, A. (1999). Fast Convergent Generalized Back-Propagation Algorithm with Constant Learning Rate. *Neural Processing Letters*. 9:13-23.
- Ng S. C., Cheung C. C, Leung S. H., Luk A. (2003). Fast Convergence for Back-Propagation Network with Magnified Gradient Function. *IEEE*. 1903-1908.
- Ng, W.W.Y., Yeung, D.S., and Tsang, E.C.C.(2006).Pilot Study On The Localized Generalization Error Model For Single Layer Perceptron Neural Network. *Proceedings of the Fifth International Conference on Machine Learning and*

- Cybernetics*, 13-16 August 2006. Dalian. 3078-3082.
- Nii O. Attoh-Okine. (1999). Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance.
- Advances in Engineering Software*. 30 (1999): 291–302.
- Oh, S.H., and Lee, Y.(1995). A Modified Error Function to Improve the Error Back-Propagation Algorithm for Multi-Layer Perceptrons. *ETRI Journal*. 17(1):11-22.
- Oh, S-H.(1997).Improving the Error Backpropagation Algorithm with a Modified Error Function. *IEEE Transactions On Neural Networks*.8(3): 799-803.
- Oh, S.H., and Lee, S-Y. (1999). A New Error Function at Hidden Layers for Fast Training of Multilayer Perceptrons. *IEEE Transactions On Neural Networks*. 10(4): 960-964.
- Otaïr M. A., Salameh W. A. (2006). Efficient training of backpropagation neural networks. *Neural Network World*. 16 (4):291-311.
- Pernia-Espinoza, A.V., Joaquin B., Martinez-de-Pison, O-M.F.J., and Gonzalez-Marcos, A.(2005). TAO-Robust Backpropagation Learning Algorithm. *Neural Networks*. 18:191-204.
- Rimer, M., and Martinez, T. (2006).CB3: An Adaptive Error Function for Backpropagation Training. *Neural Processing Letters*. 24:81–92
- Rumelhart, D.E. and McClelland, J.L. (1986). *Parallel Distributed Processing: Explorations in The Microstructure of Cognition*. Vol 1. MIT press, Cambridge,MA.
- Rydvan and Milan. (1999). Biquadratic error functions for the BP-networks. *Neural Network World*. 9(1):17-24.
- Salem, M. M., Malik, O. P., Zaki, A. M., Mahgoub, O. A., and El-Zahab, E. A. (2000). On-Line Trained Neuro-Controller with a Modified Error Function. *Proceedings, Canadian Conference on Electrical and Computer Engineering*, May 5-7, 2000, Halifax, 83-87.
- Saroja. Neural network. Date Accesed:18/12/2007.  
[www.cse.iitd.ernet.in/~saroj/nnet.ppt](http://www.cse.iitd.ernet.in/~saroj/nnet.ppt)
- Shamsuddin, S.M., Sulaiman, M.N. and Darus, M. (2001). An Improved Error Signal For Bacpropagation Model For Classification Problems. *Intern. J. Computer Mathematics*. 76(1-2): 297-305.

- Shamsuddin S. M., Darus M. and Saman. (2007). Three term backpropagation algorithm for classification problem. *Neural Network World* .17 (2007): 363-376
- Sridhar Narayan (1997). The generalized sigmoid activation function: Competitive supervised learning. *Information Sciences*, 1-2(99). 69-82
- Taji, K., Miyake, T., and Tamura, H.(1999). On error Backpropagation Algorithm Using Absolute Error Function. *IEEE International Conference, IEEE SMC '99 Conference Proceedings, 1999*. 12-15 October 1999. Tokyo, 5(1999):401-406
- Verma B.K. and Mulawka J.J. (1994). A Modified Backpropagation Algorithm. *IEEE*, 840-844
- Wang, X.G., Tang, Z., Tamura, H., Ishii, M., and Sun, W.D. (2004). An Improved Backpropagation Algorithm To Avoid The Local Minima Problem. *Neurocomputing*. 56:455 - 460.
- Wang, X.G., Tang, Z., Tamura, H., and Ishii, M.(2004). A modified error function for the backpropagation algorithm. *Neurocomputing*. 57 (2004):477 – 484
- Wang, C.H., Kao, C.H. and Lee W.H. (2007). A new interactive model for improving the learning performance of back propagation neural network. *Automation in Construction* . 16(6): 745-758.
- Wen, J.W., Zhao, J.L., Luo, S.W., and Han, Z. (2000). The Improvements of BP Neural Network Learning Algorithm. *Proceedings of ICSP2000*. 1647-1649
- Widder, D.R., and Fiddy, M.A. (1993). High Performance Learning by Modified Error Backpropagation. *Neural Computer Application*. 1:183-187
- Xu, L. (1993). Least Mean Square Error Reconstruction Principle For Self-Organizing Neural-Nets. *Neural Networks*. 6(5): 627-648. - only in html abstract
- Yam Y.F. and Chow T.W.S. (1993). Extended backpropagation algorithm. *Electronics Letters*. 29(19), 1701-1702.
- Yu, C-C., and Liu, B-D.(2002). A Backpropagation Algorithm with Adaptive Learning Rate and Momentum Coefficient. *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2002*. May 2002. 2:1218-1223.
- Zhiqiang, Z., Zheng, T., GuoFeng, T., Vairappan, C., XuGang, W., and RunQun, X. (2007). An Improved Algorithm for Eleman Neural Network by Adding a Modified Error Function. *Lecture Notes in Computer Science 4492*. Berlin Heidelberg: Springer. 465–473.
- 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.

Zweiri, Y. H., Whidborne, J. F., Althoefer, K and Seneviratne, L.D. (2005). Stability Analysis Of A Three-Term Backpropagation Algorithm. *Neural Networks*. 18 (2005) 1341–1347.