THE EFFECT OF DATA PREPROCESSING ON THE PERFORMANCE OF ARTIFICIAL NEURAL NETWORKS TECHNIQUES FOR CLASSIFICATION PROBLEMS

WALID HASEN ATOMI

A thesis submitted in
Fulfilment of the requirement for the award of the
Degree of Master of Computer Science

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

DECEMBER, 2012

ABSTRAK

Penggunaan rangkaian neural tiruan (ANN) semakin popular dan telah digunapakai dalam pelbagai bidang, seperti perubatan, kewangan, ekonomi, kejuruteraan dan sebagainya. Ini kerana ANN terkenal sebagai satu teknik pengkelasan bagi data tidak linear dan data keluaran bernombor. Walaubagaimanapun ANN masih menghadapi masalah dari segi pemprosesan yang agak lambat dan selalu terhenti di local minima. Kajian ke arah mempertingkatkan kecekapan latihan ANN telah menjadi satu bidang kajian yang aktif dengan penghasilan pelbagai kertas kajian daripada penyelidik ANN. Pada dasarnya, prestasi Perceptron Multi Lapisan (MLP) adalah banyak dipengaruhi dari saiz data dan teknik data pra pemprosesan yang akan digunakan. Kajian ini menganalisis kebaikan-kebaikan menggunakan data yang telah melalui pra pemprosesan dengan menggunakan teknik-teknik yang berbeza dalam kaedah untuk memperbaiki kecekapan latihan ANN. Min-maks, Z -skor dan teknik pra pemprosesan Pernormalan Penskalaan Perpuluhan telah dinilai secara spesifiknya. Keputusan simulasi menunjukkan bahawa ketepatan berkomputer bagi proses latihan ANN adalah sangat baik apabila digunakan bersama-sama dengan pelbagai jenis teknik pra pemprosesan.

ABSTRACT

The artificial neural network (ANN) has recently been applied in many areas, such as medical, biology, financial, economy, engineering and so on. It is known as an excellent classifier of nonlinear input and output numerical data. Improving training efficiency of ANN based algorithm is an active area of research and numerous papers have been reviewed in the literature. The performance of Multi-layer Perceptron (MLP) trained with back-propagation artificial neural network (BP-ANN) method is highly influenced by the size of the data-sets and the data-preprocessing techniques used. This work analyzes the advantages of using pre-processing datasets using different techniques in order to improve the ANN convergence. Specifically Min-Max, Z-Score and Decimal Scaling Normalization preprocessing techniques were evaluated. The simulation results showed that the computational efficiency of ANN training process is highly enhanced when coupled with different preprocessing techniques.

TABLE OF CONTENTS

LIST OF APPENDICES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	-	Artificial Neural Networks
BP	-	Back Propagation
MLP	-	Multi Layer Perceptron
PMMN1	-	Pre-processing Min Max Normalization
PZSN2	-	Pre-processing Z-Score Normalization
PDSN3	-	Pre-processing Decimal Scaling Normalization
CPU	-	Central Processing Unit
GD	-	Gradient Descent
BPNN	-	Back-Propagation Neural Network
WEKA	-	Waikato Environment for Knowledge Analysis
LS-SVM	-	Least Squares Support Vector Machine
MSE	-	Mean Squared Error
UCI	-	University of California Irvine
PCA	-	Principal Component Analysis
BBNN	-	Back-Propagation Neural Network
MLP-ANN	-	Multilayer Perceptron Artificial Neural Networks
MATLAB	-	Matrix Laboratory
FFNN	-	Feed Forward Neural Network
BPGD	-	Back Propagation Gradient Descent
BPGD-AG	-	Back Propagation Gradient Descent with Adaptive Gain
GDM	-	Gradient Descent with Momentum
GDG	-	Gradient Descent with Gain
GDL	-	Gradient Descent with Line Search
GDGL	-	Gradient Descent with Gain and Line Search
RAM	-	Random Access Memory
μ_A	-	Mean of Attribute
σ_A	-	Standard Deviation of Attribute
BPFM	-	Back-propagation with Fixed Momentum
EBP	-	Error Back Propagation

$\boldsymbol{o}^{\mathbf{0}}$	-	A particular Input Pattern
t	-	The desired output
$\boldsymbol{o}_{\boldsymbol{k}}^{\boldsymbol{L}}$	-	Actual Output
L	-	Denotes the Output Layer
E	-	The Error Function on the Pattern
$\eta^{(\boldsymbol{n})}$	-	The Learning Rate
$\boldsymbol{w}^{\boldsymbol{n}}$	-	The weight vector
IEEE	-	Institute of Electrical and Electronics Engineering
ACM	-	Association for Computing Machinery
UTHM	-	University Tun Hussein Onn Malaysia
cm	-	Centi Meter
GHz	-	Giga Hertz
DDR	-	Double Data Rate
CMC	-	Contraceptive Method Choice

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 An overview

Artificial neural networks (ANN) seemed to be one of the best approaches in machine learning methods. ANNs are software constructs designed to mimic the way of the human brain learns. The neural network can imitate the process of human's behavior and solve nonlinear matters, which has made it widely used in calculating and predicting complicated system, and achieved a non linearity mapped effect, which the conventional calculating way could not do. The nerve cell, which is the base of neural network, has the function of processing information. Neural networks extensively used in various applications such as image/signal processing, system identification, face detection and etc. Gradient based methods are one of the most widely used error minimization methods used to train back propagation networks. Back propagation algorithm is a classical domain dependent technique for supervised training. It works by measuring the output error calculating the gradient of this error, and adjusting weights and biases in the descending gradient direction. Back propagation is the most commonly used and the simplest feed forward algorithm used for classification.

ANNs are now globally recognized as the most effective and appropriate artificial intelligence technology for pattern recognition since ANN is nonlinear in nature and, thus, represents an exceptionally powerful method of analyzing real-world data that allows modeling extremely difficult dependencies [1]. ANNs have proven to be among the best methods of detecting hidden relations in a dataset (for example, in stock market or sales data). After ANN has analyzed the dataset (this process is called network training), it can make predictions and perform pattern recognition and categorization based on the found hidden dependencies. Supervised learning algorithms are trained with labeled data only. But labeling the data can be costly and hence the amount of labeled data available may be limited. Training the classifiers with limited amount of labeled data can lead to low classification accuracy.

Hence pre-processing the data is required for getting better classification accuracy. Furthermore, data pre-processing can has a significant impact on generalization performance of ANN learning algorithms.

1.2 Problem Statement

The feed-forward Backpropagation artificial neural networks (ANN) are widely used to control the various industrial processes, modelling and simulation of systems and forecasting purposes. Back propagation (BP) algorithm is known for its ability to learn and highly suitable for problems in which no relationship is found between the input and output [2]. The Backpropagation learning has various drawbacks such as slowness in learning, stuck in local minima, requires functional derivative of aggregation function and thresholding function to minimize error function. Moreover, it was found out that the techniques for Pre-processing the data also plays an important role in effecting the performance of BP in term of high quality accuracy and less processing time. These weaknesses make BP unreliable in solving real-world problems.

Improving training efficiency of neural network based algorithm particularly back propagation algorithm is an active area of research and numerous papers have been published until today to improving BP training.

In this research will investigate the following issues which affect the performance of BP algorithm:

(i) Data is not properly pre-process

Data preprocessing often neglected by researchers, although, it is an important step in data mining process. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data have quality if they satisfy the requirements of the intended use, Low-quality data will lead to low-quality mining Results. There are many techniques in pre-process data such as; Min-Max, Z-Score and Decimal Scaling Normalization preprocessing techniques. By knowing which technique that can influence BP the most it can help to enhance BP training.

(ii) Some parameters that influence on the performance of BP

The performance of BP algorithm is control by some parameters such as momentum, learning rate, activation function, number of hidden nodes, gain parameter. Those parameters are used for controlling the weight adjustment along the descent direction. Finding those initial and fixed parameters must be done with great care. The activation function must be differentiable, and it helps if the function is bounded, the sigmoid functions such as logistic and tanh and the Gaussian function are the most common choices. Therefore, by knowing the best parameters for BP algorithm, the training process is highly enhanced when coupled with those different preprocessing techniques.

1.3 Research Objectives

This research was intended to do the following objectives:

(i) To clarify the effect of some parameters in back propagation algorithm with different pre-processing techniques in improving the classification accuracy on some classification problems.

(ii) To apply a combination of data pre-processing technique with optimal parameters in BP training algorithm.

(iii)To compare the performance of the applying combination with other traditional techniques in classifying some benchmark problems.

1.4 Scope of Research

The research will focus only on enhancing the current working back propagation algorithm proposed by Nazri [3] with optimal parameters and combine with optimal data pre-processing techniques on some classification problems. Three pre-processing techniques will be used in this research for the comparison proposes which are:

(i) Min-Max Normalization.

(ii) Z-Score Normalization.

(iii)Decimal Scaling Normalization.

The performances of the integration method will be compare and analyze in terms of processing time, number of epochs and accuracy modeling on five dataset benchmark problems.

1.5 Research Methodology

The purpose of this research is to provide empirical evidence on the comparative study of different data preprocessing method in MLP model for classification problems, with four different algorithms.

The research methodology will be as presented in Figure 1.1 will be describe as follows:

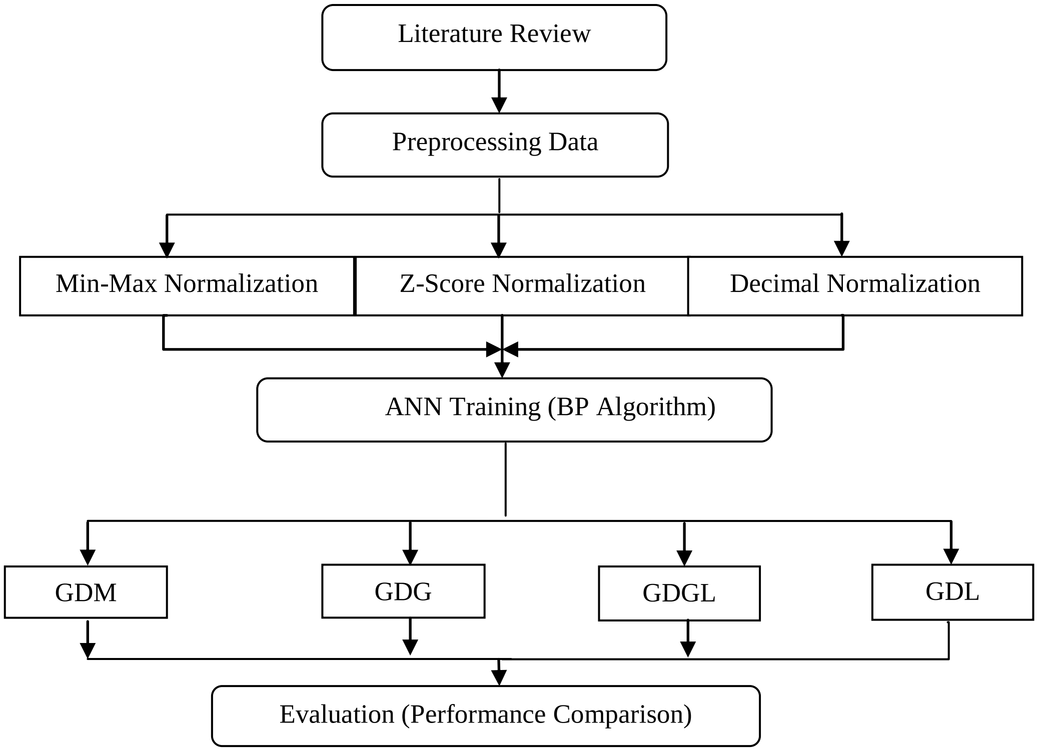

Figure 1.1: Research Methodology

(a) Literature Review

This phase provides the context of the research work which is relevant to the previous and the current work of the research and the relationship between different works. In this chapter, literature review discussed about back-propagation algorithm and some improvements by other researchers. Some modifications and variations of BP algorithm proposed Nazri [3] [2] which have been used to solve classification problems and will be applied throughout this thesis.

(b) Pre-processing Data

Today the real world data are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources[4] [5]. Low-quality data will lead to low-quality mining results. This phase will discuss several data preprocessing techniques that can clean and remove noise and correct inconsistencies in data.

(c) Artificial Neural Network (ANN)

ANNs are mathematical techniques designed to deal with different problems. The research in the field has a history of several decades, but the interest started to grow in the early 1980s [2]. Today, Neural Networks can be configured in various arrangements to perform a range of tasks including pattern recognition, data mining, classification, forecasting and process modeling. ANNs are composed of attributes that lead to perfect solutions in applications where we need to learn a linear or nonlinear mapping. Some of these attributes are learning ability, generalization, parallel processing, and error endurance. These attributes would enable ANNs to solve complex problem methods precisely and flexibly. ANNs consist of an inter-connection number of neurons. There are many varieties of connections under study [4]. There are lot connection topologies to construct Multi Layer Perception (MLP) network as shown in Figure 1.2. The input nodes are the previous lagged observations, while the output provides the forecast for the future value. In order to process the information received by the input nodes, hidden nodes with appropriate nonlinear transfer functions are used.

The MLP's most popular learning rule is the error back propagation algorithm and can solve problems which are not linearly separable.

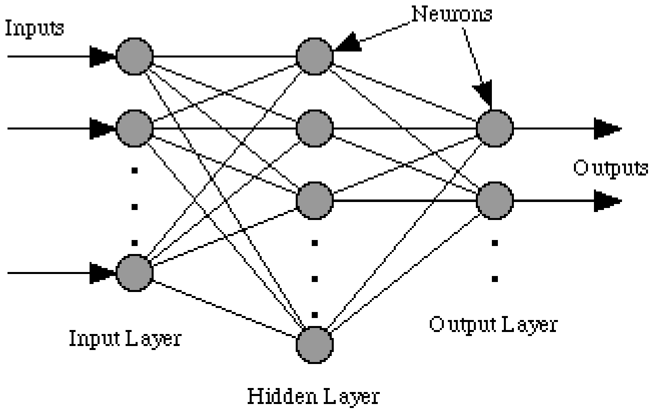

Figure 1.2: Multilayer Perceptron Architecture

(i) Backpropagation Algorithm

Backpropagation (BP) is a common method of training ANN to minimize the objective function. BP described it as a *Multi-stage Dynamic System* optimization method [5]. BP algorithm is a supervised learning method, and is a generalization of the delta rule [6]. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). The term is an abbreviation for "backward propagation of errors". BP requires that the activation function used by the artificial neurons or "nodes" be differentiable [7].

BP is very good at recognizing of patterns (in fact it is usually better than a human). The data are presented directly to the network with each pattern well positioned on a grid and correctly sized. However, BP cannot handle patterns in a noisy “scene”, like recognizing a face in a crowd or a letter in a page of print [8]. Since the algorithm gets confused or this phenomena is known as an “unconstrained environment”. So it is very crucial to pre-process the data before presenting the data to BP algorithm.

(d) Evaluation

The effect of pre-processing techniques with ANN training algorithm is evaluate based on simulations on some benchmark datasets. A classification model is tested by applying it to test data with known target values and comparing the predicted values with the known values [9]. The test data must be compatible with the data used to build the model and must be prepared in the same way that the build data was prepared. Typically the build data and test data come from the same historical data set.

A percentage of the records are used to build the model, the remaining records are used to test the model. Test metrics are used to assess how accurately the model predicts the known values. If the model performs well and meets the targets, it can then be applied to new data to predict the future.

1.6 Research Outcome

This research outcome is to evaluate the effect of some pre-processing techniques and optimal parameters in BP training algorithm. Based on BP techniques, the best model was obtained from the selection and adjusting the optimal parameter settings which are built in BP algorithm settings, such as learning rate, momentum term, activation function, gain value and the number of hidden nodes in achieving a good performance in classification which are measured based on *Accuracy*, *CPU time* and the number of *Epochs*.

1.7 Research Significance

The research significance is improving and enhancing BP training by integrate or combine the optimal data pre-processing technique with optimal parameters which control BP to get good accuracy for classification problem on some benchmark dataset.

1.8 Thesis Outline

The thesis is subdivided into six chapters, including the introduction and conclusion chapters. The following is the synopsis of each chapter:

- ***Chapter 1:*** *Introduction.* Apart from providing an outline of the thesis, this chapter contains an overview of the background to research work, research problem, objectives, research scope and methodologies in conducting this research.

- ***Chapter2:*** *Literature Review*. The Back-propagation (BP) is one of the most successful computational intelligence techniques and widely used learning algorithm for Artificial Neural Networks (ANN). However, it convergence rate can be very slow. Researchers had tried to improve its computational efficiency by adjusting parameters such as learning rate, momentum, gain of activation function, network topology and different learning algorithms. Moreover, the proper choices of pre-process technique also play a big role in improving the BP learning process. This chapter reviews the research contribution made by various researchers to improve the training efficiency of BP. This chapter demonstrates the effect of using pre-process technique to the BP learning process. At the end of this chapter, some of the advantages posed by the proper choice of pre-process technique are outlined. This chapter lays a foundation for introducing a proper technique for improving the learning efficiency as described in Chapter Three.

- ***Chapter3:*** *Research Methodology*. This chapter extends the work on using pre-process technique as proposed in Chapter Two. It was discovered that the use of pre-process technique influences the BP performance. Since most of the data mining technique employed during the training process of BP networks use pre-process technique. An efficient of work flow has been presented that properly select pre-process technique together with BP algorithms. The proposed workflow is programmed in MATLAB programming language and is tested for its correctness on selected benchmark data sets. The results of the proposed workflow were compared to facilitate further testing and validation in the next chapter.

- ***Chapter4:*** *Results and Discussions*. The efficient workflow proposed in Chapter Three is further validated for its efficiency and accuracy on a variety of benchmark datasets. The performance of the proposed workflow is tested in three ways: (a) the speed of convergence measured in number of iterations and CPU time; and (b) the classification accuracy on testing data from benchmark datasets. The results are then discussed for their interpretation and implementation through various ANN models.

- ***Chapter5:*** *Conclusion and future work*. The research contributions are summarized and recommendations are made for further continuation of work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Artificial Neural Networks (ANNs) particularly Back-propagation neural network (BPNN) is one of the most successful computational intelligence techniques used in various applications nowadays [10]. However, the training process for a BPNN require the designers to arbitrarily select parameters such as network topology, initial weights and biases, learning rate value, the activation function, value for gain in activation function and momentum. An improper choice of any of these parameters can result in the slow convergence or even network paralysis, where the training process comes to a standstill or get stuck at local minima. Furthermore, the effect of selecting a proper pre-process technique can also contribute a significant improvement to the BP learning process [11]. Therefore, this chapter focuses on the previous literature work that suggested certain improvements on BPNN model together with the effect of using pre-processing techniques for classification problems.

2.2 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) is analytical techniques modelled on the learning processes of human cognitive system and the neurological functions of the brain. ANN works by processing information like biological neurons in the brain and consists of small processing units known as Artificial Neurons, which can be trained to perform complex calculations [12].

An Artificial Neuron can be trained to store, recognize, estimate and adapt to new patterns without having the prior information of the function it receives. This ability of learning and adaption has made ANN superior to the conventional methods used in the past. Due to its ability to solve complex time critical problems, it has been widely used in the engineering fields such as biological modelling, financial forecasting, weather forecasting, decision modelling, control systems, manufacturing, health and medicine, ocean and space exploration etc [13] [14] .

2.3 Back Propagation Neural Network

Back-Propagation Neural Network (BPNN) is one of the most novel supervised-learning Multilayer Perceptron Artificial Neural Networks (MLP-ANN) model proposed by Rumelhart, Hinton, & Williams (1986). The BPNN learns by calculating the errors of the output layer to find the errors in the hidden layers. This qualitative ability makes it highly suitable to be applied on problems in which no relationship is found between the output and the inputs. Due to its high rate of plasticity and learning capabilities, it has been successfully implemented in wide range of applications (Lee, Booth, & Alam, 2005). Previous works have confirmed that the performance of MLP in many cases outperform conventional statistic techniques [16]. The main power of ANNs lies in their ability to approximate complex and non-linear relationship between input and output data by learning from example [17].

Despite providing successful solutions BPNN has some limitations. Since, it uses gradient descent learning which requires careful selection of parameters such as network topology, initial weights and biases, learning rate, activation function, and value for the gain in the activation function. An improper use of these parameters can lead to slow network convergence or even network stagnancy. Previous researchers have suggested some modifications to improve the training time of the network. Some of the variations suggested are the use of learning rate and momentum to stop network stagnancy and to speed-up the network convergence to global minima. These two parameters are frequently used in the control of weight adjustments along the steepest descent and for controlling oscillations (Zaweri, Seneviratne, & Althoefer, 2005).

Recently, researches had found out that the techniques for pre-processing the data also plays an important role in effecting the performance of MLP, where pre-processing data encourage the high quality accurate and less time.

There are many data pre-processing techniques have been discussed in literatures. Some of work has been done on the real world data and the others on benchmark dataset problems. The researchers used various model of ANN techniques and combined with the pre-processing techniques to get high accurate. It is important to pre-process data due to noisy data, errors, inconsistencies, outliers and lack of variable values [21] [23]. Different data pre-processing techniques like cleaning method, outlier detection, data integration and transformation can be carried out before classification process to achieve successful analysis. Normalization is an important pre-processing step in data mining to standardize the values of all variables from dynamic range into specific range. Outliers can significantly affect data mining performance, so outlier detection and removal is an important task in wide variety of data mining applications.

2.4 Back Propagation and Pre-processing Techniques

Concern with the effect of pre-processing data, Suhartono, et la.2006, provided an empirical evidence on the comparative study of many data preprocessing method in NN model for forecasting trend and seasonal time series. Three types of data preprocessing based on the decomposition method are applied and compare to the airline data. Those are de-trend, de-seasonal, and combination detrend-deseasonal. All of these data preprocessing are implemented by using MINITAB software. To determine the best hybrid model, that is combination data preprocessing based on the decomposition method and NN model, an experiment is conducted with the basic cross validation method. The available training data is used to estimate the weights for any specific model architecture. The testing set is the used to select the best model among all models considered. In this work, the number of hidden nodes varies from 1 to 10 with an increment of 1. The lags of 1, 12 and 13 are included due to the results of Faraway and Chatfield (1998), Atok and Suhartono (2000), and Suhartono *et al*. (2005a). The FFNN model used in this empirical study is the standard FFNN with single-hidden-layer. They also use the standard data preprocessing in NN for the airline data by transform detrend, deseasonal, and combination detrend-deseasonal data to range (0,1) scaling . The performance of in-sample fit (training data) and out sample forecast (testing data) is judged by the commonly used error measures, the mean squared error (MSE) and ratio MSE to ARIMA model. The simulation and real data contain 144 month observations. The first 120 data observations are used for model selection and parameter estimation (training data in term of NN model) and the last 24 points are reserved as the test for forecasting evaluation and comparison (testing data). The ratio of testing samples comparison that combination detrend-deseasonal as data preprocessing and transformation range (0,1) on FFNN with 5 unit nodes (for simulation data) and 4 unit nodes for the airline data in hidden layer yield the best MSE. The reduction of MSE is highly significant if compare to the result of FFNN without detrend-deseasonal as data preprocessing, those are 52.8% for simulation data and 61.8% for the airline data. Based on the results can conclude that the combination detrend-deseasonal based on the decomposition method as data preprocessing in FFNN yields a great impact on the increasing accuracy of forecasting trend and seasonal time series.

The research followed by Tudu, et la. 2008 where he presents a comparative study of different normalization techniques for enhancing pattern classification of black tea using electronic nose. For this study black tea samples were collected from different tea gardens in India. At first Principal Component Analysis (PCA) was used to investigate presence of clusters in the sensors responses in multidimensional space. Then different normalization techniques were used on the black tea data. Finally classification performances were done using BP-MLP. BP-MLP algorithm for black tea classifications using normalized data marginally enhances the pattern recognition accuracy of electronic nose system. Three-layer Back propagation Multilayer Perceptron (BP-MLP) model with one input layer, one hidden layer and one output layer has been considered. Experimentations with electronic nose have been performed with 194 finished tea samples with six different taster scores of tea samples and sensor output signatures are logged in the computer. Initial PCA analysis was done on raw data set as well as on normalized date set, and improves the separable nature on normalized data set. Further BP-MLP was tested on normalized data set. Different normalization techniques and BP-MLP algorithm were done on MATLAB platform. With raw data, the simulation results for testing demonstrated that BP-MLP algorithm achieved accuracy of 60.25%. Whereas, with normalized data set, the classification accuracy marginally improves. It is noted that from this study for classification of black tea using electronic nose based on BP-MLP algorithm, normalization technique of Range Scale2 gives the better classification rate more than 93% [24].

In the same year 2008, Paulin, et la. In his research on feed forward neural network and the Back propagation algorithm are used to train the network. The proposed algorithm is tested on a real life problem, the Wisconsin Breast Cancer Diagnosis problem. The data set was partitioned into two sets: training and testing set. The testing set was not seen by any neural network during the training phase and it is only used for testing the generalization of neural network ensembles after they are trained. The research used the 80% examples for the training set, and the rest 20% examples for the testing set. There are three layers in the back propagation network, including an input layer containing nine units, a hidden layer containing six units and an output layer containing only one unit. The value of the unit in the output layer shows whether the input is a normal cell or not. At the start of training, all connection weights in network are set to random values. All input vectors are normalized so that the minimum and maximum are 0 and 1, respectively. The architecture used in these applications consisted of tan-sigmoid hidden units and one purelin output unit. The learning rate is 0.7 was used. Number of maximum allowable epochs was 1000. In this work six training algorithms are used, among these six methods, Levenberg Marquardt method gave the good result of 99.28%. Preprocessing using min-max normalization is used in this diagnosis.

Later, Jayalakshmi et la. [15], proposed various normalization methods used in back propagation neural networks to enhance the reliability of the trained network. Jayalakshmi proposed a network with eight inputs and four inputs and the results obtained are compared in terms of error. The highest performance is obtained when the network consists of eight inputs with three hidden layers with 15, 14, 14, 1 neurons respectively. The network parameters such as learning rate, momentum constant, training error and number of epochs can be considered as 0.9, 0.9, 1e-008 and 100 respectively. Before training the weights are initialized to random values. The main reason to initialize weights with small values is to prevent saturation. The performance of the network models were evaluated with the entire sample which was randomly divided into training and test sample. All models are tested using the standard rule of 80/20, where 80% of the samples are used for training and 20% are used for testing. The experimental results showed that the performance of the diabetes data classification model using the neural networks was dependent on the normalization methods [25].

In the same year 2011, Jain et la, proposed a privacy preserving data distortion method based on min max normalization transformation. They use WEKA (Waikato Environment for Knowledge Analysis) software to test the accuracy of distorted method. The privacy parameters are measured by a separate Java program. They have constructed the classifier for NB Tree classification, and a 10-fold cross validation to obtain the classification results .In his paper, Jain [27] conducted the experiment on four real life datasets and the experimental result show that the min max normalization transformation based data distortion method is effective for privacy preserving data mining. The privacy parameters used in this paper show the degree of privacy protection by the proposed method. Furthermore, the proposed method also maintain the performance of data mining technique after data distortion. Jain also suggested that it is interesting to use the other normalization methods like Z-score normalization and Normalization by decimal scaling and compare its result with Min-max normalization.

Mustaffa et al.[28] investigated the use of three normalization techniques in predicting dengue outbreak where all of the input and output data were normalized before training and testing processes in order to ensure that data are not overwhelmed by each other in terms of distance measure. In this paper three types of normalization techniques has been applied separately; Min-Max Normalization, Z-Score Normalization and Decimal Point Normalization, these techniques are incorporated in the LS-SVM and Neural Network (NNM) prediction model respectively, the purpose of using three different normalization techniques is to identify the best normalization to be used in this research. Comparisons of results are made based on prediction accuracy and mean squared error MSE, the data divided proportionally as stated; 70% Training, 350 instance data for training and 30% testing, 150 instance data for testing. From the undertaken experiments it shown that LS-SVM and NNM can achieved better accuracy and MSE by using Decimal Point Normalization compared to the other two techniques Min-Max and Z-Score Normalization. Furthermore, the simulations results obtained indicate that the LS-SVM is a better prediction model as compared to the NNM.

Then Patel et al. [29], analyzed the performance of modified k-Means clustering algorithm with data preprocessing technique includes cleaning method, normalization approach and outlier detection with automatic initialization of seed values on datasets from UCI dataset repository. Data pre-processing techniques are applied on raw data to make the data clean, noise free and consistent. Data Normalization standardize the raw data by converting them into specific range using linear transformation which can generate good quality clusters and improve the accuracy of clustering algorithms. The performance analysis of modified k-Means clustering algorithm shows that decimal scaling normalization technique gives the best results for the modified k-means clustering algorithm and secondly min-max data normalization generates the best results for modified k-means clustering algorithm. The analysis shows that outlier detection and removal with generates the best and most effective and accurate results than other techniques used in this paper. The paper also compares the MSE performance of the proposed Mk-Means algorithm with MSE of Data Normalization techniques on River dataset. The analysis result shows the best result for Mk-Means with normalization approach.

Recently, in the year 2011, Norhamreeza and Nazri et la.2011, proposed a new modified BP learning algorithm by introducing adaptive gain together with adaptive momentum and adaptive learning rate into weight update process. By computer simulations, they demonstrated that the proposed algorithm achieved a better convergence rate and can find a good solution in early time as compared to the conventional BP. The research used four common benchmark classification problems to illustrate the improvement in convergence time The proposed algorithm adaptively changes the gain parameter of the activation function together with momentum coefficient and learning rate to improve the learning speed. The effectiveness of the proposed algorithm has been compared with the conventional Back Propagation Gradient Descent BPGD and Back Propagation Gradient Descent with Adaptive Gain BPGD-AG [3]. The three algorithms were been verified by means of simulation on four classification problems including iris dataset with an improvement ratio nearly 2.8 seconds for the BPGD and 1.33 seconds better for the BPGD-AG in terms of total time to converge; card dataset indicates almost 92.5% and 12.92% faster compared to BPGD and BPGD-AG respectively; glass almost 10.2 seconds less time to converge than BPGD, whilst BPGD-AG nearly 2 seconds; and thyroid is 90.65% faster than BPGD and almost 16.2% faster than BPGD-AG in learning the patterns. The results show that the proposed algorithm BPGD-AGAMAL has a better convergence rate and learning efficiency as compared to conventional BPGD and BPGD-AG.

The research continue by Norhamreeza and Nazri Nawi, et la.[18] by proposing an algorithm for improving the performance of the current working back propagation algorithm which is Gradient Descent Method with Adaptive Gain by changing the momentum coefficient adaptively for each node. The influence of the adaptive momentum together with adaptive gain on the learning ability of a neural network is analyzed. Multilayer feed forward neural networks have been assessed. Physical interpretation of the relationship between the momentum value, the learning rate and weight values is given. The efficiency of the proposed algorithm is compared with conventional Gradient Descent Method and current Gradient Descent Method with Adaptive Gain was verified by means of simulation on three benchmark problems. In learning the patterns, the simulations result demonstrate that the proposed algorithm converged faster on Wisconsin breast cancer with an improvement ratio of nearly 1.8, 6.6 on Mushroom problem and 36% better on Soybean data sets. The results clearly show that the proposed algorithm significantly improves the learning speed of the current gradient descent Backpropagation algorithm. The result shows that the proposed algorithm GDM/AGAM has a better convergence rate and learning efficiency as compared to conventional Gradient Descent Method GDM and Gradient Descent Method with Adaptive Gain GDM/AG .

Later, Nazri Nawi, et la.[39], proposed an algorithm for improving the performance of the back propagation algorithm by introducing the adaptive gain of the activation function. The efficiency of the proposed algorithm is compared with conventional Gradient Descent Method and verified by means of simulation on four classification problems. The performance criterion used in this research focuses on the speed of convergence, measured in number of iterations and CPU time. The benchmark problems used to verify the proposed algorithm are taken from the open literature. Four classification problems have been tested including Wisconsin breast cancer, thyroid, IRIS classification and diabetes classification problem. The simulations have been carried out on a Pentium IV with 3 GHz PC Dell, 1 GB RAM and using MATLAB version 6.5.0 (R13). On each problem, the following three algorithms were analyzed and simulated. For all problems the neural network had one hidden layer with five hidden nodes and sigmoid activation function was used for all nodes. All algorithms were tested using the same initial weights, initialized randomly from range, the learning rate value is 0.3 and the momentum term value is 0.7 with the initial value used for the gain parameter is one and received the input patterns for training in the same sequence. The number of iterations until convergence is accumulated for each algorithm from which the mean, the standard deviation and the number of failures are calculated. The networks that fail to converge are obviously excluded from the calculations of the mean and standard deviation but are reported as failures. For each problem, 100 different trials were run, each with different initial random set of weights. The effectiveness of the proposed algorithm has been compared with the Gradient Descent Method GDM, verified by means of simulation on four classification problems including Wisconsin breast cancer and diabetes classification problem with an improvement ratio nearly 2.8 and 1.2.for the total time of converge, thyroid took almost 65% less time to converge and IRIS the proposed algorithm outperformed the traditional GDM with 97% success in learning the patterns. Moreover, when comparing the proposed algorithm with GDM, it has been empirically demonstrated that the proposed algorithm GDM/AG performed highest accuracy than GDM. This conclusion enforces the usage of the proposed algorithm as alternative training algorithm of back propagation neural networks.

2.5 The Pre-processing Technique

Data transformation such as normalization is one of the data pre-processing techniques used in most of Data Mining System. An attribute of a dataset is normalized by scaling its values so that they fall within a small-specified range, such as 0.0 to 1.0.

Normalization is particularly useful for classification algorithms involving neural networks, or distance measurements such as nearest neighbor classification and clustering. Normalization performs data smoothing and data normalization preparatory to modeling. The technique is easy to apply by using standard mathematical transformations such as min-max normalization to numerical columns, z-score normalization, log normalization, or decimal scaling normalization. Extreme values in data can make it difficult to detect patterns. If the data is very irregular, has very high or low values, or values are scattered or do not follow a Gaussian distribution, normalizing the data can help fit the data to a distribution that better supports any modeling. There are many methods for data normalization includes min-max normalization, z-score normalization and normalization by decimal scaling.

2.5.1 Min-Max Normalization

This method rescales the features or outputs from one range of values to a new range of values. More often, the features are rescaled to lie within a range of [0,1] or from [-1, 1]. The rescaling is often accomplished by using a linear interpretation formula, such as:

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \qquad (1)$$

Where max_A is the maximum value of attribute, min_A is the minimum value of attribute for ($new\_max_A - new\_min_A$) = 0, When ($max_A - min_A$) = 0, it indicates a constant value for that feature in the data. When a feature value is found in the data with a constant value, it should be removed because it does not provide any information to the neural network. When the min-max normalization is applied, each feature will lie within the new range of values will remain the same. Min-max normalization has the advantage of preserving exactly all relationships in the data.

2.5.2 Z-Score Normalization (Statistical)

Z-score normalization is also called zero-mean normalization; this technique uses the mean and standard deviation for each feature across a set of training data to normalize each input feature vector. The mean and standard deviation are computed for each feature. The transformation is given in the general formula:

$$v' = \frac{v - \mu_A}{\sigma_A} \qquad (2)$$

Where μ_A is mean of attribute A and σ_A = standard deviation of attribute A. This produces data where each feature has a zero mean and a unit variance. Z-Score normalization technique is applied to all the feature vectors in the data set first; creating a new training set and then training is commenced. Once the means and standard deviations are computed for each feature over a set of training data, they must be retained and used as weights in the final system design. It is a preprocessing layer in the neural network structure. Otherwise, the performance of the neural network will vary significantly because it was trained on a different data representation than the un-normalized data.

The advantage of this statistical normalization is that it reduces the effects of outliers in the data.

2.5.3 Decimal Scaling Normalization

Normalization by decimal scaling normalizes by moving the decimal point of value of attribute A. The number of decimal points moved depends on the maximum absolute value of A. A value A of A is normalized to A' by the following formula:

$$A' = \frac{A}{10^{m}} \qquad (3)$$

Where m is the smallest integer such that Max |A'|<1.

2.6 Fundamentals of Gradient Descent

Multilayer feed-forward Neural Network training using gradient descent BPNN requires parameters such as network topology, initial weights and biases, learning rate value, activation function, and value for the gain in the activation function should be selected carefully. An improper choice of these parameters can lead to slow network convergence, network error or failure. Seeing these problems, many variations in gradient descent BPNN algorithm have been proposed by previous researchers to increase the training efficiency. Some of the variations are the use of learning rate and momentum to speed-up the network convergence and avoid getting stuck at local minima. These two parameters are frequently used in the control of weight adjustments along the steepest direction and for controlling oscillations.

Momentum coefficient is a modification based on the observation that convergence might be improved if the oscillation in the trajectory is smoothed out, by adding a fraction of the previous weight change [32] [33]. It has been revealed through various studies that Back-propagation with Fixed Momentum Coefficient BPFM shows acceleration results when the current downhill gradient of the error function and the last change in weights are in the similar directions, when the current gradient is in an opposing direction to the previous update, BPFM will cause the weight direction to be updated in the upward direction instead of down the slope as desired, so in that case it is necessary that the momentum coefficient should be varied adaptively instead of being kept fixed.

Gradient descent is a first-order optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient or of the approximate gradient of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent. Gradient descent is also known as steepest descent, or the method of steepest descent.

2.6.1 Gradient Descent Training

Gradient descent is one of the oldest optimization methods known. The use of the method as a basis for multivariate function minimization dates back to Cauchy in 1847 [34], and has been the subject of intense analysis. A popular method of training Multilayer Perceptron MLP is the error back propagation EBP algorithm which is a gradient descent with fixed learning rate. In order to accelerate the EBP algorithm, which has a drawback with slow convergence, many proposed a modified error function which showed improved convergence. However, they are still based on the gradient descent with non-optimal learning rates.

2.6.2 The Gradient Descent Modification

Nowadays, the Multilayer Perceptron (MLP) trained with the back propagation (BP) is one of the most popular methods used for classification purposes [34]. This method has the capacity of organizing the representation of the knowledge in the hidden layers and their high power of generalization. Typical architecture has three sequential layers: input, hidden and output layer [35-36]. Such that, a MLP with one layer can build a linear hyper-plane, a MLP with two layers can build convex hyper-plane and a MLP with three layers can build any hyper-plane.

An MLP training using gradient descent back propagation requires careful selection of parameters such as; network topology, initial weights and biases, learning rate value, activation function, and value for the gain in the activation function. An improper choice of these parameters can lead to slow network convergence, network error or failure [37-38]. Seeing these problems, many variations in gradient descent algorithm have been proposed by previous researchers. Lately, Nazri [39] proposed a modification by changing the gradient based search direction using a gain value. For the comparison purposes, this study will apply the gradient descent with gain (GDG) algorithm proposed by Nazri [39] due to its successful implementation and proven not to suffer the local minimum as compared to the traditional gradient descent. The following iterative algorithm has been proposed by Nazri [39] for changing the gradient based search direction using a gain value:

> *Initializes the initial weight vector with random values and the vector of gain values with unit values. Repeat the following steps 1 and 2 on an epoch-by-epoch basis until the given error minimization criteria are satisfied.*
>
> ***Step 1*** *By introducing gain value into activation function; calculate the gradient of error with respect to weights by using Equation (6), and gradient of error with respect to the gain parameter by using Equation (7)*
>
> ***Step 2*** *Use the gradient weight vector and gradient of gain vector calculated in step 1 to calculate the new weight vector and vector of new gain values for use in the next epoch.*

2.6.3 Derivation of the Expression to Calculate Gain Value

Consider a multilayer feed-forward network, as used in standard back propagation algorithm [37]. Suppose that for a particular input pattern o^0 , the desired output is the teacher pattern $t = [t_1 \dots t_n]^T$, and the actual output is o_k^L , where L denotes the output layer. The error function on that pattern is defined as;

$$E = \frac{1}{2} \sum_k (t_k - o_k^L)^2 \tag{4}$$

Let o_k^s be the activation values for the k^{th} node of layer *s*, and let $o^s = [o_1^s \dots o_n^s]^T$, be the column vector of activation values in the layer *s* and the input layer as layer 0. Let w_{ij}^s be the weight values for the connecting link between the i^{th} node in layer *s* -1 and the j^{th} node in layer *s* , and let $w_{ij}^s = \left[w_{1j}^s \dots w_{nj}^s\right]^T$ be the column vector of weights from layer *s* -1to the j^{th} node of layer *s* . The net input to the j^{th} node of layer *s* is defined as; $net_j^s = (w_j^s, o^{s-1}) = \sum_k (w_{j,k}^s - o_k^{s-1})$,and let $net^s = [net_1^s \dots net_n^s]^T$ be the column vector of the net input values in layer *s* . The activation value for a node is given by a function of its net inputs and the gain parameter c_j^s ;

$$o_j^s = f\left(c_j^s net_j^s\right) \qquad (5)$$

, where *f* is any function with bounded derivative.

This information is now used to derive an expression for modifying gain values for the next epoch.

Most of gradient based optimization methods use the following gradient descent rule:

$$\boldsymbol{\Delta}_{ij}^{(n)} = -\boldsymbol{\eta}^{(n)} \frac{\partial E}{\partial w_{ij}^{(n)}} \qquad (6)$$

, where $\boldsymbol{\eta}^{(n)}$ is the learning rate value at step *n* and the gradient based search direction at step n is $d^{(n)} = -\frac{\partial E}{\partial w_{ij}^{(n)}} = g^{(n)}$.

In the proposed method the gradient based search direction is modified by including the variation of gain value to yield

$$d^{(n)} = -\frac{\partial E}{\partial w_{ij}^{(n)}} \left(c_j^{(n)}\right) = g^{(n)}\left(c_j^{(n)}\right) \qquad (7)$$

The derivation of the procedure for calculating the gain value is based on the gradient descent algorithm. The error function as defined in Equation (4) is differentiated with respect to the weight value w_{ij}^s . The chain rule yields,

$$\frac{\partial E}{\partial w_{ij}^s} = \frac{\partial E}{\partial net^{s+1}} . \frac{\partial net^{s+1}}{\partial o_j^s} . \frac{\partial o_j^s}{\partial net_j^s} . \frac{\partial net_j^s}{\partial w_{ij}^s}$$

$$= [-\delta_{\mathbf{1}}^{s+1} \;\ldots\ldots\; -\delta_n^{s+1}]. \begin{bmatrix} w_{\mathbf{1}j}^{s+1} \\ . \\ . \\ . \\ w_{nj}^{s+1} \end{bmatrix} . f'(c_j^s net_j^s)\, c_j^s . o_j^{s-1} \qquad (8)$$

where $\delta_j^s = \frac{\partial E}{\partial net_j^s}$. In particular, the first three factors of Equation (8) indicate that the following equation holds:

$$\delta_{\mathbf{1}}^s = \left(\sum_k \delta_k^{s+1} \delta_{k,j}^{s+1} \right) . f'\left(c_j^s net_j^s \right) c_j^s \qquad (9)$$

It should be noted that, the iterative formula as described in Equation (9) to calculate δ_i^s is the same as used in the standard back propagation algorithms except for the appearance of the gain value in the expression. The learning rule for calculating weight values as given in Equation (6) is derived by combining (8) and (9).

In this approach, the gradient of error with respect to the gain parameter can also be calculated by using the chain rule as previously described; it is easy to compute as:

$$\frac{\partial E}{\partial c_j^s} = (\sum_k \delta_k^{s+1} w_{k,j}^{s+1}) f'(c_j^s net_j^s) net_j^s \qquad (10)$$

Then the gradient descent rule for the gain value becomes,

$$\Delta c_j^s = \eta \quad \delta_j^s \frac{net_j^s}{c_j^s} \qquad (11)$$

At the end of every epoch the new gain value is updated using a simple gradient based method as given by the following formula,

$$c_j^{new} = c_j^{old} - \Delta c_j^s \qquad (12)$$

2.6.4 The implementation on Gradient Descent Method with Adaptive Gain Variation Algorithm (GD/AG)

In gradient descent method, the search direction at each step is given by the local negative gradient of the error function, and the step size is determined by a learning rate parameter. Suppose at step *n* in gradient descent algorithm, the current weight vector is w^n , and a particular gradient based search direction is d^n .

The weight vector at step n+1 is computed by the following expression:

$$w^{n+1} = w^n + \eta^{(n)} d^n \qquad (13)$$

where, $\eta^{(n)}$ is the learning rate value at step *n* .

By using the proposed method, the gradient based search direction is calculated at each step by using Equation (7).

The complete GD/AG algorithm works as follows;

Step 1 - Initialize the weight vectors randomly, the gradient vector g_0 to zero and gain vector to unit values. Let the first search direction d_0 be g_0, Set β_0 *=0, epoch =1* and *n=1* . Let Nt be the total number of weight values. Select a convergence tolerance value as *CT*.

Step 2 - At step *n*, evaluate the gradient vector g_n (c_n).

Step 3 - Evaluate *E* (w_n), if *E* (w_n) < CT then STOP training ELSE go to step4.

Step 4 - Calculate a new gradient based search direction which is a function of gain parameter:

$d_n = -g_n\ (c_n)\beta_{n-1}d_{n-1}$

Step 5 - If *n* > 1 THEN,

Step 6 - Update w_n : $w_{n+1} = w_n - \eta_n^* d_n$.

Step 7 - Evaluate the new gradient vector $g_{n+1}(c_{n+1})$ with respect to gain value c_{n+1}

Step 8 - Calculate the new gradient based search direction:

$$d_{n+1} = -g_{n+1}(c_{n+1}) + \beta_n(c_n)d_n$$

Step 9 - Set *n* = *n* +1 and go to step 2.

2.7 Variation on gradient descent training

A simple idea that can improve convergence of gradient descent is to include at the each iteration a proportion of the change from the previous iteration. Basic BP without momentum is a gradient descent algorithm and it was popularized by Rumelhart et al although similar ideas had been developed previously by Werbos [13]. The simplest implementation of this algorithm is by first calculating the network error for all patterns, then adjusting the network weights and biases along the negative of the gradient of the error function, one iteration of this algorithm is shown in Equation14,

$$w_{k+1} = w_k - \mu g_k \qquad (14)$$

Where, w_k is the vector of current weight, μ is the learning rate and g_k is the current gradient of the error with respect to the weight vector. The negative sign indicates that the new weight vector w_{k+1} is moving in a direction opposite to that of the gradient.

2.7.1 Gradient Descent BP with Momentum (GDM)

Momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Momentum allows the network to ignore small features in the error surface. Without momentum a network may get stuck in ash allow local minimum. With momentum a network can slide through such a minimum [14], as shown in Figure 2.1.

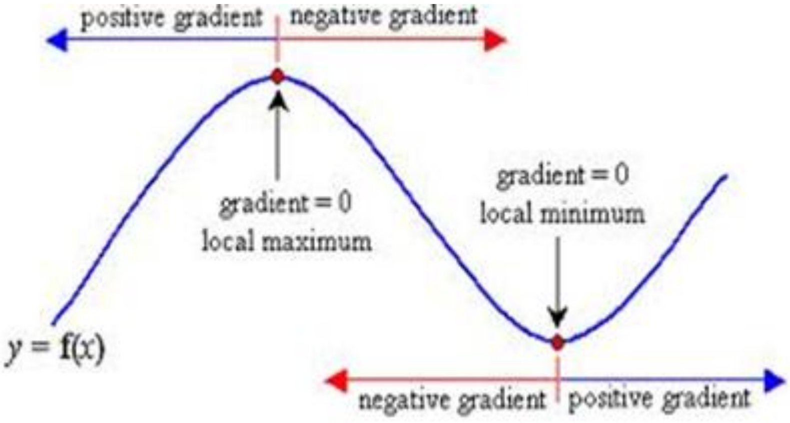

Figure 2.1: The local (minima, maximum)

Momentum can be added to BP method learning by making weight changes equal to the sum of a fraction of the last weight change and the new change suggested by the gradient descent BP rule. The magnitude of the effect that the last weight change is allowed to have is mediated by a momentum constant, μ , which can be any number between *0* and *1* when, the momentum constant is *0* a weight change is based solely on the gradient as shown in Figure 2.2. When the momentum constant is *1* the new weight change is set to equal the last weight change and the gradient is simply ignored, see Figures 2.3.

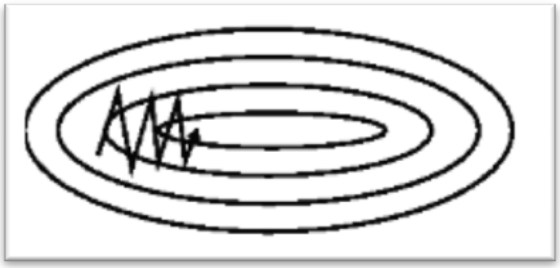

Figure 2.2: High oscillation when momentum is zero

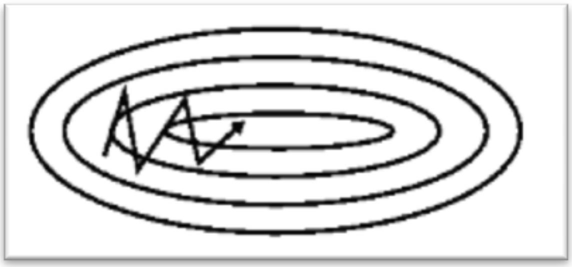

Figure 2.3: Lower oscillation when momentum is one

REFERENCES

[1] Kamarthi , Pittner. Accelerating Neural Network Training using Weight Extrapolations. Neural Networks, vol. 12, pp. 1285-1299 (1999).

[2] Rehman, Nawi, The Effect of Adaptive Momentum in Improving the Accuracy of Gradient Descent Back Propagation Algorithm on Classification Problems. 2011. CCIS, Vol. 179, Issue No. 6, Springer Heidelberg.

[3] Hongjun, Rudy, and Huan . Effective Data Mining Using Neural Networks, IEEE transactions on knowledge and data engineering, vol. 8, no. 6, december 1996.

[4] John, Premier Reference Source Data Warehousing and Mining, Concepts, Methodologies, tools and Applications , Volume (1), 2008, Information Science References.

[5] Abid, Fnaiech and Najim , 2001, “A fast feedforward training algorithm using a modified form of the standard backpropagation algorithm,” IEEE Trans. Neural Networks 12 424-430.

[6] Thimm, Moerland and Fiesler. The Interchangeability of Learning Rate and Gain in Backpropagation Neural Networks. Neural Computation, vol. 8(2) pp. 451- 460 (1996).

[7] Eom, Jung and Sirisena. Performance Improvement of Backpropagation Algorithm by Automatic Activation Function Gain Tuning Using Fuzzy Logic. Neurocomputing, vol. 50, pp. 439-460 (2003).

[8] Bishop. Neural Networks for Pattern Recognition. 1995: Oxford University Press.

[9] Weiss and Kulikowski. Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems. Morgan Kaufman, 1991.

[10] Khalid, Alrajeh, Tamer and Alzohairy, Date Fruits Classification using MLP and RBF Neural Networks, International Journal of Computer Applications (0975 – 8887) Volume 41– No.10, March 2012.

[11] Witten and Frank. Data Mining Practical Machine Learning Tools and Techniques with Java Implementations . 2000, Morgan Kaufmann Publisher .

[12] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature 323*, pp. 533-536, 1986.

[13] Lera and Pinzolas: Neighborhood based Levenberg-Marquardt Algorithm for Neural Network Training. IEEE Transaction on Neural Networks, vol. 13(5), pp. 1200-1203 (2002).

[14] Boulle and Khiops: A Statistical Discretization Method of Continuous Attributes. Machine Learning 55:1 (2004) 53-69.

[15] Jayalakshmi and Santhakumaran, Statistical Normalization and Back Propagation for Classification. International Journal of Computer Theory and Engineering, Vol.3, No.1, February (2011), 1793-8201.

[16] Moller , A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks, vol. 6(4), pp. 525-533 (1993).

[17] Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Phd Thesis, Harvard University, 1974.

[18] Nawi, Ransing and Ransing: An Improved Conjugate Gradient Based Learning Algorithm for Back Propagation Neural Networks. International Journal of Computational Intelligence, vol. 4, No. 1, pp. 46-55 (2007).

[19] Ooyen and Nienhuis, (1992) “Improving the convergence of the Backpropagation algorithm, Neural networks,” 5 465-471.

[20] S.Jeyaseeli Subavathia and T.Kathirvalavakumarb, Adaptive modified Backpropagation algorithm based on differential errors, International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.1, No.5, October 2011.

[21] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, ‘Data Pre-processing for Supervised Learning’ Computer Science, vol. 1, 2006, pp.1306-4428.

[22] M. Singh and G. M. Provan. Efficient learning of selective Bayesian network classifiers. In Machine Learning: Proceedings of the Thirteenth International Conference on Machine Learning. Morgan Kaufmann, 1996.

[23] J. Han, and M. Kamber, “Data Mining: Concepts and Techniques”,Second edition, 2006, Morgan Kaufmann, USA.

[24] K. Chen and L. Liu, “Privacy preserving data classification with rotation perturbation”, Proceeding of the 5th IEEE International Conference on Data Mining (ICDM 2005), pp. 589-592, 2005.

[25] Prachitee, Shekhawat, Sheetal and Dhande, A Classification Technique using Associative Classification, International Journal of Computer Applications (0975 – 8887) Volume 20– No.5, April 2011.

[26] Mehta, Agrawal, and Rissanen. SLIQ : A fast scalable classifier for data mining. In Proc. 1996 Int. Conf. Extending Database Technology (EDBT'96), Avignon, France, March 1996.

[27] Jain and Bhandare: Min Max Normalization Based Data Perturbation Method for Privacy Protection, Min Max Normalization Based Data Perturbation Method for Privacy Protection. International Journal of Computer & communication Technology Volume-2 Issue-VIII, 2011.

[28] Mustaffa and Yusof. A Comparison of Normalization Techiques in Predicting Dengue Outbreak. International Cinference on Business and Economics Resaerch, vol.1(2011)©(2011) LACSIT Press, Kuala Lumpur, Malaysia.

[29] Patel and Mehta. Impact of Outlier Removal and Normalization Approach in Modified k-Means Clustering Algorithm. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 2, September 2011, ISSN (Online): 1694-0814.

[30] UCI Machine Learning Database Repository University of California Irvine (*http://archive.ics.uci.edu/ml/*).

[31] Tudu and Bhattacharyya. Comparison of Multivariate Normalization Techniques as Applied to Electronic Nose Based Pattern Classification for Black Tea. 3rd International Conference on Sensing Technology, Nov. 30 – Dec. 3, 2008, Tainan, Taiwan.

[32] Suhartono and Subanar. The Effect of Decomposition Method as Data Preprocessing on Neural Networks Model for Forecasting Trend and Seasonal Time Series. Jurnal Teknik Industri vol. 8, no. 2, Dec. 2006: 156-164.

[33] Rumelhart, Hinton and Williams, “Learning Internal Representations by error Propagation,” Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, 1986.

[34] Fkirin, Badwai and Mohamed, “Change Detection Using Neural Network in Toshka Area,” Proc. NSRC, 2009, Cairo, Egypt, 17-19 March, pp.1-10.

[35] Haykin . Neural Networks, A Comprehensive Foundation. 2nd ed. Prentice Hall, New Jersey, 1999.

[36] Cristianini and Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge UK, 2000.

[37] Shertinsky and Rosalind. On the efficiency of the orthogonal Least Squares Training Method for Radial Function Networks, IEEE Transaction on Neural Networks. 1996. 7(1).

[38] Lakatos, Carson and Benyo. Artificial neural network approach to diabetic management, Proceedings of the Annual International Conference of the IEEE, EMBS, pp.1010–1011, 1992.

[39] Rehman, Nawi, and Ghazali, Predicting Noise-Induced Hearing Loss (NIHL) and Hearing Deterioration Index (HDI) in Malaysian Industrial Workers using GDAM Algorithm. 2012. Journal of Engineering and Technology, Vol. 3, Issue No. 1.

[40] Nawi, Ghazali and Mohd Salleh, The development of improved back-propagation neural networks algorithm for predicting patients with heart disease. 2010. LNCS, Vol. 6377, Issue No. M4D, p.p. 317-324.