NEURAL NETWORK PREDICTION OF SPM ACHIEVEMENT

A thesis submitted to the Graduate School in partial
fulfillment of the requirements for the degree
Master of Science (Information Technology),
Universiti Utara Malaysia

By
Robizah Binti Haji Sudin
October 2000

Sekolah Siswazah
(Graduate School)
Universiti Utara Malaysia

PERAKUAN KERJA KERTAS PROJEK
(Certification of Project Paper)

Saya, yang bertandatangan, memperakukan bahawa
(I, the undersigned, certify that)

ROBIZAH BT. HJ. SUDIN

calon untuk Ijazah
(candidate for the degree of) Sarjana Sains (Teknologi Maklumat)

telah mengemukakan kertas projek yang bertajuk
(has presented his/ her project paper of the following title)

NEURAL NETWORK PREDICTION OF SPM ACHIEVEMENT

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
(as it appears on the title page and front cover of project paper)

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan, dan meliputi bidang ilmu dengan memuaskan.
(that the project paper acceptable in form and content, and that a satisfactory knowledge of the field is covered by the project paper).

Nama Penyelia
(Name of Supervisor) : En. Roshidi bin Din

Tandatangan
(Signature) :

Tarikh
(Date) : 9 Oktober 2000

GRADUATE SCHOOL
UNIVERSITI UTARA MALAYSIA

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a postgraduate degree from the Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor(s) or, in their absence, by the Dean of the Graduate School. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or in part should be addressed to:

Dean of Graduate School
Universiti Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman

TABLE OF CONTENTS

CHAPTER THREE: NEURAL NETWORK FUNDAMENTALS

CHAPTER FOUR: RESEARCH METHODOLOGY

CHAPTER FIVE: FINDINGS AND RESULT

LIST OF TABLES

LIST OF FIGURES

To my mom and dad.
You have always been a source of inspiration

To Sofea, Hafiz, and Hakim.
Allah blessed me with wonderful kids

And to the memory of my beloved husband,
Mohd Zawawi Mat Nor
25th June 1970 – 19th May 1999

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor(s) or, in their absence, by the Dean of the Graduate School. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Graduate School

Universiti Utara Malaysia

06010 UUM Sintok

Kedah Darul Aman

ABSTRACT (BAHASA MALAYSIA)

Tujuan kajian ini adalah untuk membentuk sebuah model rangkaian neural bagi meramal pencapaian pelajar-pelajar sekolah menengah di Malaysia di dalam peperiksaan SPM. Rangkaian neural yang dibentuk menggunakan perceptron multi aras yang melibatkan algoritma "*backpropagation*" serta tangen sigmoid sebagai fungsi pindahan. Kajian ini bukan sahaja mengambil kira gred yang diperolehi oleh pelajar di dalam matapelajaran teras yang di ambil di dalam SPM tetapi juga mengambil kira jantina mereka. Berdasarkan kepada keputusan yang diperolehi dari model tersebut, prestasi sebenar pelajar boleh diramal. Kajian ini menunjukkan bahawa rangkaian neural boleh dilatih dengan data-data yang berkait dengan pelajar untuk meramal pencapaian mereka di dalam peperiksaan SPM.

TABLE OF CONTENTS

ABSTRACT (ENGLISH)

The purpose of this study is to build a neural network model for prediction of SPM achievement for the students in a Malaysian secondary school. The neural network model uses multi-layer perceptron involving a backpropagation algorithm and the tangent sigmoid as the transfer function. This study does not only consider the students' grades for the core subjects that they take in the SPM but also the student gender. Based on the model results, the real exam performance is to be predicted. This study shows that neural network can be trained with students' data to predict their achievement in the SPM examination.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Encik Roshidi Din for his help and advice in conducting this study.

Also thanks to the Headmaster of Dato' Wan Ahmad Rasdi Secondary School, Encik Yahya Abdul Rahman and staffs for their help in collecting the application data.

Thanks to my employer, Universiti Teknologi MARA (UiTM), for the financial support which had enabled me to further my studies.

Special thanks are due to my mother, Limah Mat Yaman and father, Sudin Ahmad for their precious loves and endless support.

My kids, Sofea Nabeela 6, Luqman Hafizi 5, and Azmir Nurhakim, 1½, get credit for their patience and understanding – for trying to stay quiet and well behaved while their mom busy with reading and writing.

Finally, I express my gratitude to the memory of my husband, Mohd Zawawi Mat Nor, whose love and spirit will reside inside me and kids forever

CHAPTER THREE: NEURAL NETWORK FUNDAMENTALS

CHAPTER FOUR: RESEARCH METHODOLOGY

CHAPTER FIVE: FINDINGS AND RESULT

LIST OF TABLES

LIST OF FIGURES

Chapter One: Introduction

1.1. The context of the study

Forecasting, at least intelligent forecasting, is predicting future events based on historical data (Blum, 1992). Predicting for the future is a very common problem in human life. Though, it is a very tough task (Angstenberger, 1996). In spite of that, with neural networks – also called artificial neural networks, or ANN – models, effective predictive applications can be developed (Ding *et al*, 1996).

It has been found from many studies that neural network is capable of forecasting and giving better results compared to statistical and traditional analytical methods (Adya *et al*, 1998; Muzaffer *et al*, 1999; Mahmood *et al*, 1999; Aiken, 1999; and Indro *et al*, 1999). In addition, as stated by Law (1998), what makes a neural network superior to traditional statistical methods in forecasting is that a neural network is better able to recognize the high level features, such as the intra-correlation or serial correlation, of a training set.

Artificial neural networks have been applied in many applications, which include pattern recognition, prediction, text retrieval, diagnostic and optimization. For the most part, ANNs have been used for their ability to capture arbitrary non-linear and highly complex relationships between the inputs and outputs of the processes.

Neural networks pattern the functions of the human brain and are not preprogrammed to perform statistical techniques in the conventional sense (Aiken, 1999). Rather neural networks discover relationships from repetitively presented data and "learn" through familiarization. McMenamin (1997) suggested that neural network is suitable for forecasting problems when there exist explanatory variables, which are interactive and nonlinear. Furthermore, Pattie and Synder (as cited by Law, 1998) claimed that a neural network has been demonstrated to outperform standard statistical models in forecasting with a small-sized training set and a high level of white noise (random errors in the samples).

Accurate, precise and reliable prediction is very important in any field of management. As a result of that, preprocessing of data and the choice of sampling data and inputs are very important to ensure good predicting performance (Ding *et al*, 1999). Besides that, as indicated by

Awad (1996), neural networks do not solve a problem by rules or procedures. They only work with data. So, the most vital aspect of neural processing is information.

In view of the notable forecasting performance of a neural network, this study attempts to model the SPM exam results forecast using a neural network. To date, there exists no published work that makes such an attempt. This is particularly true in the context of the selected rural school, namely the Dato' Wan Ahmad Rasdi Secondary School, Perak.

1.2. Problem Statements

Excellent performance of the students has always been a major issue in Malaysian education. Secondary education aims to promote the general development of students by helping them to acquire knowledge, insight and skills including the inculcation of values on the National Philosophy of Education. The ultimate goal of secondary education is to develop a strong foundation for life-long education. Besides receiving general education, students are introduced to the beginnings of specialization.

Being the main national examination, SPM (Sijil Pelajaran Malaysia) or the Malaysian Certificate of Education records the students' achievements upon finishing the secondary education. In many situations, the certificate is the minimum requirement qualification for entry to university or college. If the students' performance can be predicted, proper planning can be taken especially by the school to ensure a better performance in the future. It is about time for us to find and develop a new mechanism to perform this complex nonlinear forecasting problem.

1.3. The Goals And Objectives Of The Study

1.3.1. Goals

The goal of this study is to develop a neural network model for forecasting the students' SPM results, using historical SPM accomplishment of students in the school. The model is trained by using a feed forward multi-layer perceptron structure with backpropagation learning algorithm.

1.3.2. Specific Objectives

This study is anticipated to examine the applicability of a neural network model for a predicting task. In the study done by Huang (1999), he found that ANNs were well suited to do prediction tasks. It is believed that ANNs are also applicable in areas related to education.

This study seeks to determine the best network parameters to develop the most efficient or the best-fitting neural network model. This study addresses such issues, as whether or not an effective neural network model can be developed to reliably predict the SPM results of form five students. With regard to this, this study investigates the ability of feed-forward neural networks to predict the SPM result based on eight input parameters that reflect the students' past achievements.

1.4. Significance Of The Study

The use of artificial neural networks has been very limited in the field of education. It is believed that using a sophisticated model such as neural network model is one valuable tool, which can be used to help the educational administrator to obtain a best guess at certain outcomes based on case history. The hypothesis is that the use of neural networks will produce an accurate model of student exam score performance based on other exam scores. A school could use this model to provide teachers and counselors with the ability to make accurate early instructional interventions with certain students.

1.5. Scope Of The Study

Typically, this study involves a three-stage process in which:

- Decisions are made about what the input variables and learning parameters would be

- Training and testing the network

  The neural network model is then formed from the learning process done on the input data. The network is supplied with data from the old student performance files (input) together with the decision made or results i.e. overall SPM grades (output). The network gets it's training by repeatedly taking in each input (example or a set of training facts), guessing at its output, and comparing the results to the supplied output (Awad, 1996).

  The network is trained using a subset of the data until the average error between the forecast and the actual values is reduced to a minimum (Aiken, 1999). Once a neural network has been adequately trained, it is tested to see how well it has learned. Presenting the network with special examples or facts it has not encountered in the past does testing. Several important network parameters are identified to form a model that could deliver the best performance.

- Running the network

  Running the network means presenting it with new inputs with no known outputs (Awad, 1996). This study looks at the possibility of presenting the network with students' trial SPM exam results and determines the respective behavior of the network.

1.6. Organization Of The Report

This report consists of six chapters altogether. Chapter one presents the overall view of the study conducted, which includes the problem statement, the objectives of the study, the importance of the study being carried out, and the scope of the study. Then, chapter two follows with a related literature review regarding the curriculum development in Malaysia with focus on the SPM examination. Here, the process of awarding the SPM grade to the students is presented.

After that, chapter three discusses in details with reference to the neural network fundamentals. Starting with an introduction to neural network and its basic architecture, chapter three then proceeds with discussing how the neural network works. Next, modes of learning of the neural network are presented. This includes backpropagation and the algorithm, supervised and unsupervised learning and multi-layer perceptron. Chapter three ends with a brief discussion on generalization with ANN.

The research methodology carried out in this study is presented in chapter four. Here, the five major phases involved in building the neural network application are presented. Chapter four starts with discussion on the process of collecting and preparing data. After that the process of designing the network architecture and learning method is presented followed by the data preprocessing method. Next, the process of training, validating, testing and running the neural network are discussed.

Chapter five of the report presents the findings and result of the study, followed by a conclusion and recommendations in chapter six.

Chapter Two: Review Of Related Literature

In Malaysia, children enter secondary school in their 7^{th} year of schooling, at around the age of 13 years. Secondary education is basically an extension of primary level education. Education at this level is provided in national secondary schools. The medium of instruction in these schools is the Malay Language and English Language is taught as a second language.

Education at this level is general in nature and is divided into three main levels: lower secondary level (form 1 to form 3), upper secondary level (form 4 to form 5) and pre-university level.

Lower secondary education in Malaysia prepares students to develop skills needed in life and to be useful citizens of the country. Education at the upper secondary level covers a period of two years. At the end of the two-year period, the students will be assessed by a compulsory national examination, Sijil Pelajaran Malaysia (SPM) or Malaysian Certificate of Examination (MCE) or Sijil Pelajaran Malaysia Vokasional /Vocational Malaysian Certificate of Examination (SPMV/VMCE), as in the case of the vocational streaming. The SPM/MCE/SPMV/VMCE certificates are equivalent to O-level Cambridge University Examinations.

2.1. Curriculum Development: Independence To 1980

During this period of time, the system of education in Malaysia can be described as providing basic education at the elementary level, general comprehensive education at the lower secondary level, and semi-specialized at the upper secondary level (Rahimah, 1998).

After completing the third year of lower secondary education, the students are required to take a national assessment test, Sijil Rendah Pelajaran/Lower Certificate of Education (SRP/LCE). The students performances on SRP/LCE will determine their academic streaming to the upper secondary level i.e. whether to be in science, arts, technical or vocational streams.

At the upper secondary level, curriculum allows students to specialize into the Science or Arts streams in the academic schools, or to enter the technical or vocational streams as preparation for world of work.

The upper secondary curriculum consists of a number of core subjects taken by all students, providing basic general education, as well as an opportunity to specialize in one of the four streams. Core subjects consist of the two main languages (Malay Language and English), Mathematics, Science, Islamic/Moral Knowledge and the choice of at least one social science subject (History or Geography). Elective subjects for the Arts stream include Malay Literature, Art Education, and Literature in English.

Students choose between 7 – 9 subjects for the SPM examination, which students sit for at the end of the eleven years of schooling (six years elementary and five years secondary). To qualify for the certificate, or determine the grade obtained, certain conditions have to be met including the right combination of subjects and subject grade for Malay Language.

2.2. Educational Reform Since 1980s

The new curriculum prescribed for secondary schools is KBSM (Kurikulum Bersepadu Sekolah Menengah) or the Integrated Secondary School Curriculum. The new curriculum takes on a whole new approach. The syllabus is developed to suit the needs and aspirations of the country. The KBSM aims to continue providing general education. Students are no more streamed into specialized areas, although there is room for them to have subject concentration through their choice of elective subjects. Students are able to choose the electives according to their interests and talents, to be sufficiently prepared for specialization at a higher level.

After completing the third year of the secondary school, the students are required to take a new national assessment test, Penilaian Menengah Rendah/Lower Secondary Assessment (PMR/LSA). In this exam, all subjects are categorized as core subjects and are compulsory for all.

The new curriculum for upper secondary level i.e. the SPM level, consists of core subjects (Malay Language, English, Mathematics, Science, Islamic Knowledge and History), which are compulsory, and four groups of elective subjects from four areas (Humanities, Science, Technology and Vocational, and Islamic Studies).

Two types of certificates are awarded to the SPM candidate if the candidate meets certain stated criteria and conditions. The certificates are:

- The SPM (Sijil Pelajaran Malaysia): students are given this certificate only if they pass six subjects, which out of two must be credits.

- The SAP (Sijil Am Pelajaran): (i) Students are given this certificate if they do take the SPM wholly but are not qualified to get the SPM certificates and they obtain at least one credit for the subjects taken. (ii) The students who do not take the SPM wholly but obtain at least one credit for the subjects taken.

The SPM result is recorded in such a way that it contains the following information:

- The subjects' grades: 1 – 2 (Distinction); 3 – 6 (Credit); 7 – 8 (Pass); 9 (Fail)
- The overall SPM grade: Grade 1, 2 or 3

The grade is awarded to the candidate based on a certain set of criteria. To be awarded a grade 1, the candidate must get a credit in Malay Language and also a number of credits from the other subjects. If the candidate gets only a pass in Malay Language, the candidate could be awarded a grade 2 even if he or she gets credits for other subjects. If the student fails Malay Language he could be awarded SAP (if there is at least one credit) or fail to receive any certificate (if there is no credit at all). The grade awarded is also greatly dependent upon the total aggregates achieved by the student.

Chapter Three: Neural Network Fundamentals

3.1. Neural networks: An Introduction

According to Awad (1999), the first wave of neural networks – also called artificial neural networks, or ANNs – began in the 1940s and 1950s when scientists discovered that the brain's neurons are on-off switches just as the digital computer's bits are either on or off. From this idea, the analogy between the brain and the computer was developed. Awad (1996) further cited that in 1943, neurophysiologist Warren McCullough and Walter Pitts developed an ANN system showing how highly complex computations could be performed by a network of simple binary neurons.

Since 1987, artificial neural network applications have been one of the fastest growing in the history of science. ANN is functional, adaptive learning system. It can learn and is also capable of adapting itself with characteristics of the sampling data presented to it (Schalkoff, 1997; Awad, 1996; Haykin, 1994; Lisboa, 1992; Kohonen, 1988; Lipmann, 1987). The neural network can improve its performance from the learning process that it undergoes.

Ideally, the neural network will be more knowledgeable about its environment after every learning iteration process (Haykin, 1994). It can classify the data correctly even though there are noise, distortion or nonlinearities (Schalkoff, 1997; Lisboa, 1992).

Using a neural network, the historical data can be looked at as a set of pattern associations (Blum, 1992). The input patterns are the values of all of the chosen input factors for a particular time period. The output patterns are the outputs documented for the time period. Once the neural network is trained on the pattern associations of input and output factors for the historical data, it will "recall" output patterns when presented with input patterns. In this fashion, the trained neural network can predict future events based on new sets of input factors.

The neural network framework provides a flexible function that can approximate a wide range of nonlinear processes. In forecasting problems where nonlinearities and variable interactions are important, neural networks can provide significant advantages (McMenamin, 1997). Artificial neural network models are flexible nonlinear models (McMenamin, 1997).

Neural networks store data throughout the network in the pattern of weights, interconnections and states of the neurons. Unlike the Von Neumann digital computer which follows the fetch-execute-store process, neural network globally respond to input pattern stimuli (Caudill and Butler, 1990).

3.2. The Basic Architecture of a Neural Network

The human brain is a very complex system capable of thinking, remembering and problem solving. A neuron is the fundamental cellular unit of the brain's nervous system. It receives and combines signals from other neurons through input paths called dendrites. In the human brain, there are between 10^{10} and 10^{11} neurons (Clark, 1997). Biological neural systems are highly interconnected. A single neuron may receive input from as many as $8x10^4$ neighboring neurons (Clark, 1997).

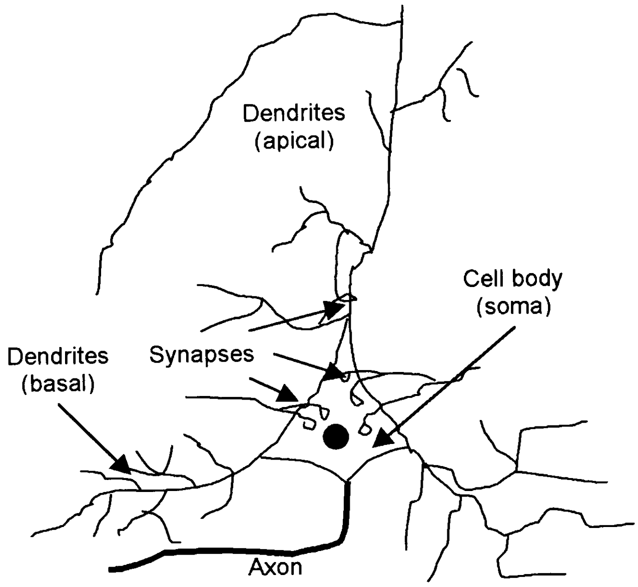

Figure 3.1: Sketch of a biological neuron showing components

Figure 3.1 above is a sketch of a neuron showing the various components. Every signal coming into a neuron along a dendrite passes through a synapse or synaptic junction. This junction is an extremely small gap in the dendrite that is filled with neurotransmitter fluid that either speed up or slow down the flow of electrical charges. The primary actions of the neuron are chemical in nature, and this neurotransmitter fluid generates electrical signals that go to the nucleus or soma of the neuron. The adjustment of the impedance or conductance of the synaptic gap lead to memory and learning.

An artificial neural network is a model that emulates a biological neural network (Turban, 1992). The architecture of the artificial neural network resembles in some respects that of the human brain, which was the original inspiration for them, although neural networks are not meant to constitute a model for the workings of the human brains (Weijters and Hoppenbrouwers, 1995). It consists of an interconnected network of simple processing units termed as neurons (Dutta, 1993). A key distinctive feature of the neural network is that the different units operate independently, and in parallel. Unlike biological systems, however, artificial neural network requires inordinate amounts of training (Kim, 1994).

An artificial neural network is represented by a set of nodes and arrows. A node corresponds to a neuron; an arrow corresponds to a connection, along with the direction of signal flow between neurons. Some nodes are designated as input units, others as output units. An artificial neural network is trained empirically on a data set by altering its weights, so a given input can be mapped to a target output. In this way, the network learns, or discovers, the knowledge embedded in the data. Its power stems from the network's own internal capability to adapt and self-organize (Fu, 1999).

3.3. How Does A Neural Network Work?

An artificial neural network is composed of processing elements (PEs) or neurons, organized in different ways to form the network's structure. The neurons, which receive inputs, process the inputs and deliver a single output, are group in layers.

3.3.1. Inputs and Outputs

A neural network is viewed as a self-programming system based on its inputs and outputs. Each neuron has a transfer function that computes the output signal from the input signals. A typical neuron building block, or node, is shown in Figure 3.2.

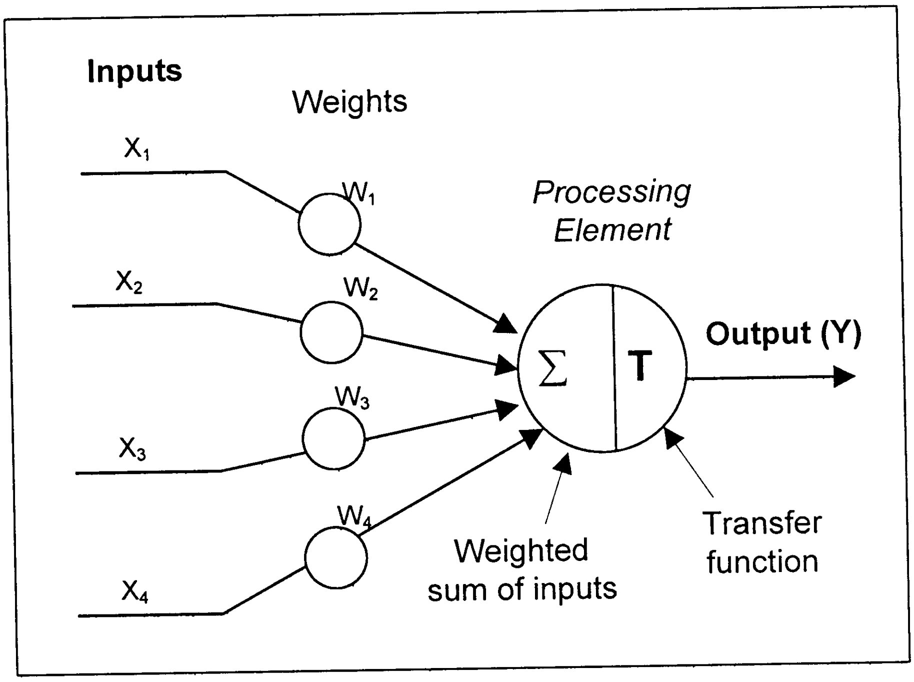

Figure 3.2: A single node with weighted inputs and transfer function

Interconnecting or combining neurons with other neurons forms a layer of nodes or a neural network. As shown in Figure 3.3, inputs can be connected to many nodes with different weights, which result in many outputs – one output per node.

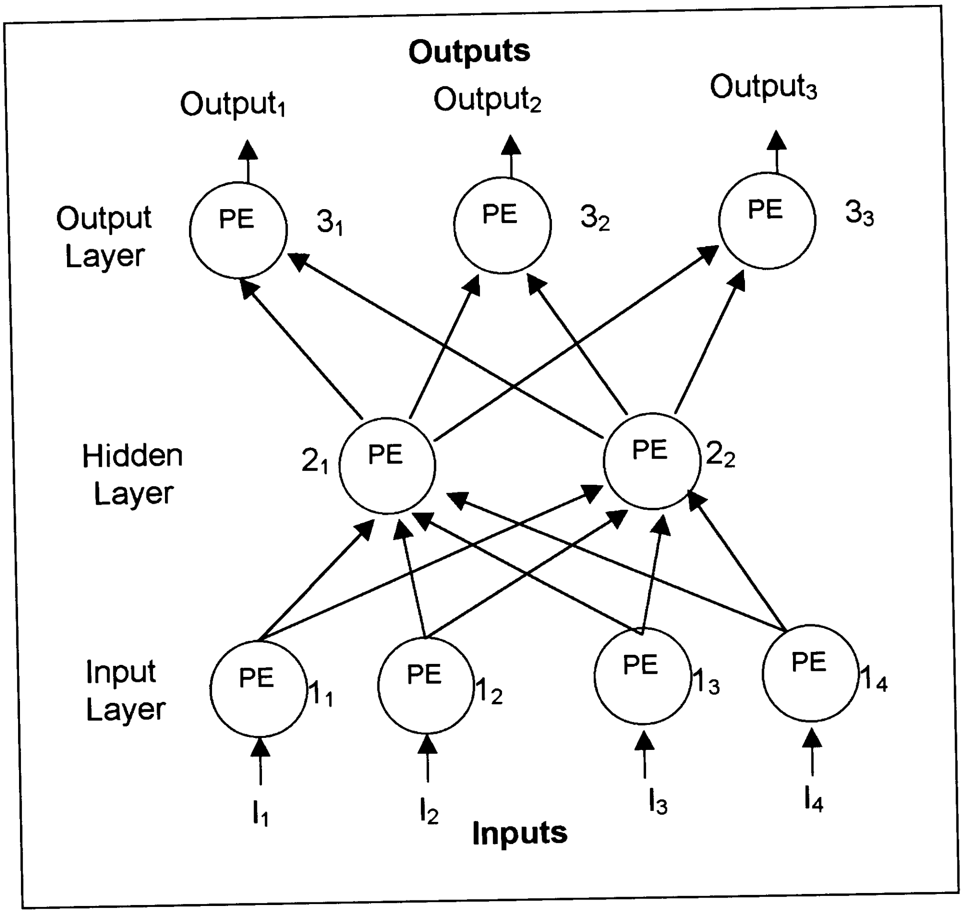

Figure 3.3: Neural network with one hidden layer

3.3.2. Processing Information In The Network

Inputs. Every input corresponds to a single attribute.

Outputs. The output of the network is the solution to a problem.

Weights. Weights express the relative strength (or mathematical value) of the preliminary entering data or the various connections that transfer data from layer to layer.

Summation functions. This function finds the weighted average of all the input elements to each processing element. It multiplies each input value (X_i) by its weight (W_i) and totals them together for a weighted sum, Y. For several (*j*) processing neurons, the formula is:

$$Y_j = \sum_{i}^{n} X_i W_{ij} \qquad (3.1)$$

Transformation (transfer) function. This function presents the relationship between the internal activation level and the output, which may be linear or nonlinear. The nonlinear transfer function used in this study is called a sigmoid function:

$$Y_T = \frac{1}{1 + e^{-Y}} \qquad (3.2)$$

Where Y_T is the transformed (or normalized) value of Y. The purpose of this transformation is to modify the output levels to a reasonable value (e.g., between zero and one).

According to McMenamin (1997), in the most general form, the type of neural network model typically used in forecasting can be written as follows:

$$Y = Ft\ H(X), H2(X), \ldots, HN(X) + u \qquad (3.3)$$

Where

Y is a dependent variable,

X is a set of explanatory variables

F and the H's are the neural network functions

And u is the model error term.

In the neural network language: The X's are called inputs, Y is called output, the H functions are called the hidden layer activation functions, and F is called the output layer activation function.

McMenamin (1997) further pointed out that in the specific form, that is normally used, F is the linear in the H functions. The H functions are specified to be S-shaped curves using the "logistic" (or the sigmoid) function. In this case, the neural network model is described as follows:

- Single output feed forward neural network
- With one hidden layer and with multiple nodes in the hidden layer
- With logistic activation functions in the hidden layer
- With a linear activation function at the output layer

A logistic activation function refers to the S-shaped functions are typically used in the hidden layer. By construction, logistic functions vary smoothly between zero and one. If a variable has a positive slope parameter in the function, then when it increases in value, it will slide upward on the S curve. This upward movement is called activation (McMenamin, 1997). As shown in figure 3.4 (a), the output is a continuous monotonic function of the input. Both the function and its derivatives are continuous everywhere.

Figure 3.4 (a) shows the most popular activation function for multi-layer perceptrons network i.e. the sigmoid function (Dutta, 1993). This function is commonly used in perceptrons using the backpropagation training algorithm. The sigmoid function is closely related to the tanh(x) function as shown in figure 3.4 (b). The tanh(x) function is particularly useful when network output should range between –1 and +1 (Henseler, 1995).

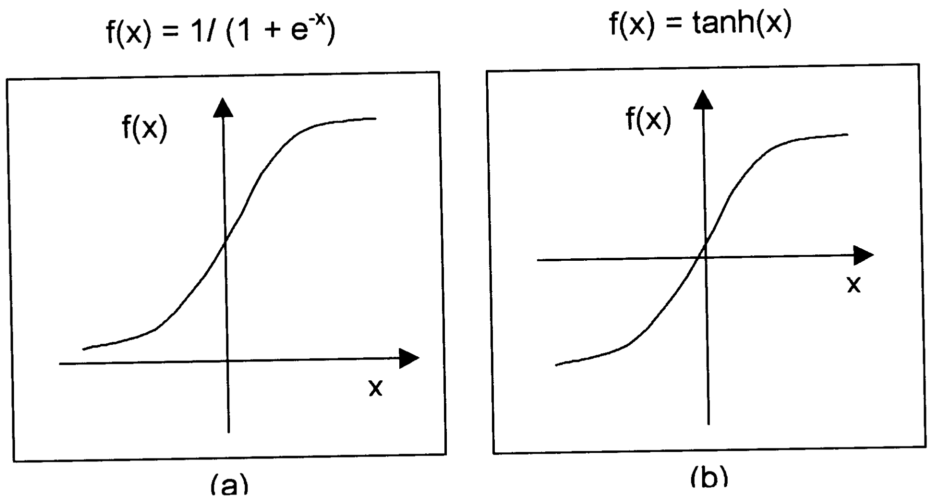

Figure 3.4: Two typical output functions used in multi-layer networks. (a) Sigmoid function, (b) Tanh(x) function.

According to Schürmann (1996), in the context of its use in the neuron model, the sigmoid function has two important properties:

- Nonlinear
- Differentiable

The nonlinearity is crucial since without this property the multi-layer feed-forward networks would collapse into single-layer networks (Schürmann, 1996; Henseler, 1995).

3.4 Learning Modes

3.4.1 Backpropagation and Training Algorithm

According to Awad (1996), the learning function takes place within the neural network's ability to change the weights and allow the neuron to modify its activity in response to its inputs. Awad (1996) further acknowledged that making weight adjustments by backing up from the output is called backpropagation. In the forward pass of backpropagation, the input pattern is applied to the network and allows the resulting activity to spread through the network to the output layer. The weights in the network are adjusted based upon the error between the expected output and the computed network output until this error is minimized (Silvert *et al*, 1998).

The backpropagation algorithm is the most popular method for neural networks training and it has been used to solve numerous real life problems (Davalo and Naïm, 1991; Moreira and Fiesler, 1995; Awad, 1996; Tsoukalas and Uhrig, 1997). This method of learning is used in neural network with more than two layers of neurons, i.e. with hidden layers (Raggett and Bains, 1992; Tsoukalas and Uhrig, 1997).

According to Raggett and Bains (1992) again, for a given input pattern, the actual output is compared with the desired target output. The dissimilarity between the two is then "propagated" back into whatever connections were used to get that output in the first place. If the match is superior, then the units, which contributed to the output, get their connections strengthened. If the match is awful, then the connections between the units concerned are reduced in strength; next time those particular input units fire up, they will have fewer effect on the hidden and output units than before.

Backpropagation is a training procedure for feedforward neural networks that consists in an iterative optimization of a so-called error function representing a measure of the performance of this network (Moriera and Fiesler, 1995). This error function E is defined as the mean square sum of differences between the values of the output units of the network and the desired target values, calculated for the whole pattern set:

$$E = 1/2 \sum_{P=1}^{P} \sum_{j=1}^{N_L} (t_j - a_j)^2 \tag{3.4}$$

t_j and a_j are the target and actual response values of output neuron j and N_L is the number of output neurons; L being the number of layers.

As cited by Tsoukalas and Uhrig (1997), the backpropagation training objective is to adjust the weights so that the application of a set of inputs produces the desired outputs. Dutta (1993) suggested the following basic steps of the backpropagation algorithm:

Step 1 - Assign small and random weights to the various inter-neuron connections in the network

Step 2 - Feed a set of inputs from an input-output training example to the input layer of the network

Step 3 - Compute the output of the network given the current connection weights and the inputs of step 2. The computation inside each individual neuron occurs as depicted in figure 3.5.

Step 4 - Compare the output of the network (step 3) with the desired output (from the input-output example of step 2) and compute the error

Step 5 - Use a recursive algorithm starting at the output nodes and working backwards to the input layer nodes, to adjust the weights on the various connections in a manner so as to reduce the error (step 4)

Step 6 - Repeat steps 2 – 5 with the input-output pairs from the training set till all (input, output) pairs in the training set are passed through the network

Step 7 - Repeat steps 2 – 6 till no changes are required in the weights on the various connections (step 5) for any example input-output pair. Thus, the algorithm cycles through the entire training set repeatedly till the weights on the various connections converge to a stable value.

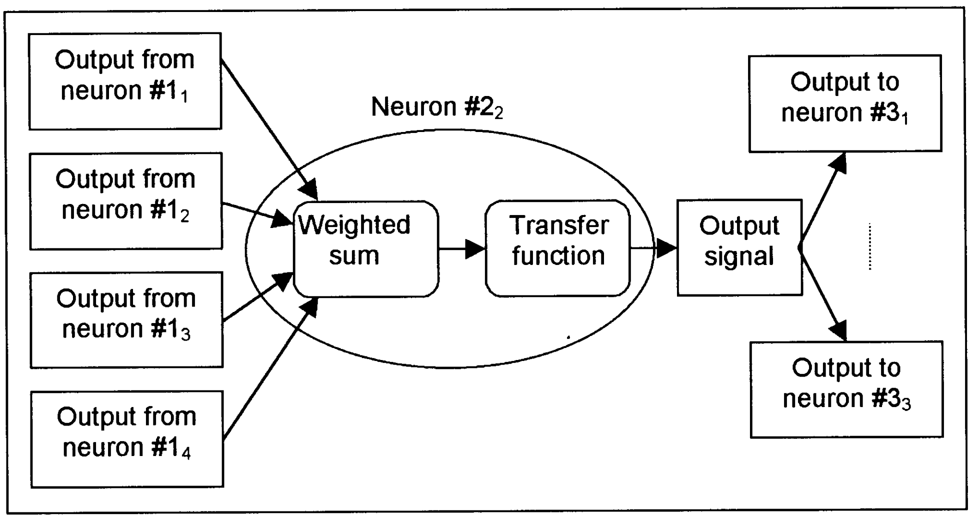

Figure 3.5: A schematic representation of processing with a neuron

3.4.2. Supervised and Unsupervised Learning

Learning may be supervised or unsupervised. In supervised learning, the neural network needs a training set of examples of input and output. Each element in a training set is paired with an acceptable response. That is, the actual output of a neural network is compared to the desired target output. They essentially learn to associate one set of input data with the corresponding set of output data. In unsupervised learning, no external factors influence the adjustment of the input's weights. The neural network has no advanced indication of correct or incorrect answers. It adjusts solely through direct confrontation with new experiences.

Tsoukalas and Uhrig (1997) pointed out that supervised neural networks are generally used for prediction, evaluation, or generalization. They also stated that, in general, the more example sets that are presented to a network for training and testing, the better the training would be. However, there must be enough examples of a sufficient variety for training so that the network will be able to make valid correlations and generalizations for unfamiliar cases.

3.4.3. Multi-layer Perceptron

According to Dutta (1993) and Gomm *et al* (1993), one of the most popular neural network architectures is the multi-layer perceptron network. Multi-layer perceptrons are a class of artificial neural networks that have been researched quite extensively (Bode, 1998). This architecture basically implements on a computer a complex mathematical function (network function) between input and output signals of the whole neural network.

According to Bode (1998), there are two properties that make multi-layer perceptrons attractive to many applications. First, the parameters of the network function can be changed easily. Second, it is possible to automatically control this change of parameters so that the function approximates a given set of input and output signals.

This offers the chance to present the neural network with a number of inputs and output data (called the training set) and let the network find the function that describes the relationship between them. The perceptron then can be fed with new input data, and it will work out an output according to the approximation function obtained. This capability is referred to as "learning" because the network automatically adapts to the situation prescribed by the training set. In other words, all "learning" occurs within the network by the manipulation of weights (Dutta, 1993).

The multi-layer perceptron network consists of an input layer, a number of hidden layers (typically only one or two hidden layers are used) and an output layer. The inputs form the *input nodes* of the network; the outputs are taken from the *output nodes*. The middle layer of nodes, visible to neither the inputs nor the outputs, is termed the hidden layer, and unlike the input and output layers, its size is not fixed (Rohwer *et al*, 1994). The network is trained in a supervised fashion.

Bode (1998) emphasized that the topology of multi-layer perceptrons has to be designed carefully. If there are too many neurons, the network might select a very complex function that approximates the training data extremely well but fails to generalize the situation between and beyond training data ("overlearning"). As said by Bode again, research to date is quite experimental because neural network theory does not yet provide applicable rules for the optimal setting of control variables and topologies.

3.5. Generalization with ANN

Prediction of future trends application is an example of a generalization problem (Dutta, 1993). According to Davalo and Naïm (1991), the generalization capacity of a neural network is its capacity to give a satisfactory response for an input, which is not part of the set of examples on which it was trained.

In generalization problems, the trained network is tested with input In+1, which is distinct from the inputs I1, I2, ..., In used for training the network. The network is expected to correctly predict the output On+1 for the (previously unseen) input In+1 from the model of domain it has learned from the training input-output pairs. Figure 3.6 depicts this process.

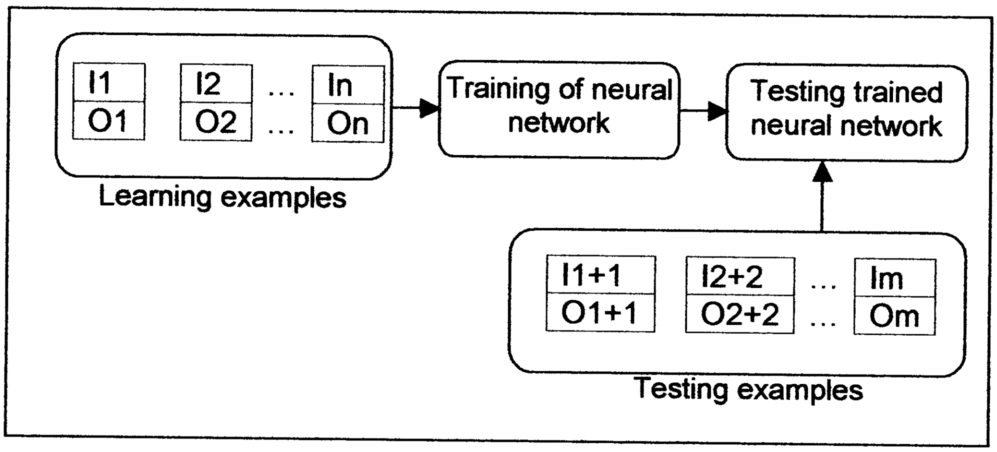

Figure 3.6: Generalization problem

Chapter Four: Research Methodology

The development process for the ANN application in this study follows loosely the nine steps suggested by Turban (1992) as shown in figure 4.1 below:

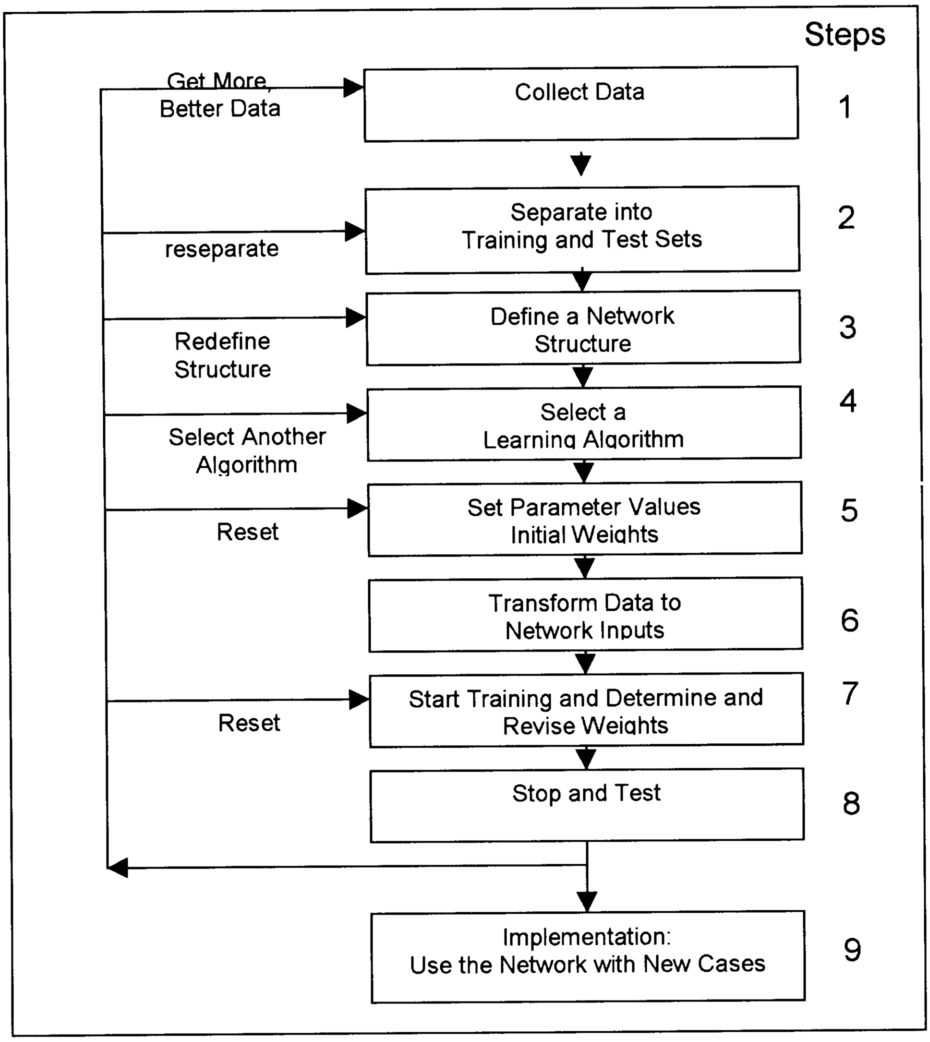

Figure 4.1: Flow diagram of the development process of a neural network

In general, in this study, there are five major phases involved in building the neural network application:

- Data collection and preparation
- Design of network architecture and learning method
- Data preprocessing
- Training, validating and testing the network
- Running the network

4.1 Data Collection And Preparation

Even though one of the great strength of neural networks is that they work well in nonlinear situations, linear relationships are the easiest for neural networks to learn and emulate. Therefore, minimizing the effects of nonlinearity of a problem pays off in terms of faster training, a less complicated network, and better overall performance. Hence, one goal of data preparation is to reduce nonlinearity when its character is known and let the network resolve the hidden nonlinearities that are not understood.

Neural network models rely on the availability of data. For training the neural network, examples for which correct answers are known are gathered. The examples are presented as facts, which are a collection of inputs with corresponding outputs. The set of examples is called the training set. The rule of thumb is to have too many examples rather than too few (Awad, 1996). Large training sets are required to train the network models so that acceptable performance can be achieved (Lam, 1999). The examples should also be of sufficient variety to allow the network to make generalizations from them (Awad, 1996).

All that is needed to train a neural network is an adequate amount of the kind of information that is important in solving a problem. If there is uncertainty whether specific data are important, it is usually best to include it because a neural network can learn to ignore inputs that have little or nothing to do with the problem, provided that enough examples are provided (Tsoukalas and Uhrig, 1997). Using too much or too many kinds of data is seldom a problem if there is adequate data. If inadequate data are used, correlations become difficult to find.

The source of the data for this study was obtained from the Headmaster of SMK Dato' Wan Ahmad Rasdi. The first batch of students at the school took SPM exam in 1995. This school is a coeducational school and situated in a rural locality. In the past, it had almost 100% of Malay students. Table 4.1 below shows an analysis of students' achievement in their SPM examination from the year of 1995 to 1999. Table 4.1 displays total number of students who received grade 1, or grade 2 and so on for the SPM.

	Grade 1	Grade 2	Grade 3	SAP	Fail
1995	14	21	21	12	18
1996	10	22	20	28	32
1997	16	31	25	37	6
1998	30	37	27	32	12
1999	18	38	36	22	12

Table 4.1: Analysis of the SPM results at Dato' Wan Ahmad Rasdi Secondary School

The raw data for this study were taken from the students' SPM result slips kept in the school office. Data for the year of 1996 to 1999 were obtained which comprised of the student gender and the grades/points achieved for every subjects taken in SPM exam.

The data for the year of 1995 could not be gathered since the file that contained the exam slips had been misplaced somewhere in the school office and therefore could not be found. The school did not have a proper documentation of the previous students' SPM results. They had only kept copies of the exam slips for each individual student. The exam slips could be regarded as temporary SPM transcripts. Table 4.2

below shows the total records gathered from the school. While table 4.3 presents the detail attribute information for the data set collected.

Exam	Year	Sets of Input Patterns
SPM	1999	128
	1998	138
	1997	114
	1996	84
	Total	464
TRIAL SPM	1999	129

Table 4.2: Breakdown of the raw data gathered

# Attribute	Domain
1. Year	1996 - 1999
2. Exam Slip No.	Student Id. Number
3. Gender	Male/Female
4. Malay Language	1 – 9
5. English	1 – 9
6. Islamic Knowledge	1 – 9
7. History	1 – 9
8. Mathematics	1 – 9
9. Science	1 – 9
10. Elective 1	1 – 9
11. Elective 2	1 – 9
12. Elective 3	1 – 9
13. Aggregate	6 - 44
14. Grade (Class attribute)	1 / 2 / 3 / SAP / Fail

Table 4.3: Attributes and Domain values

Data were collected from four different groups of students starting from those who took the SPM in 1996 up to those who took the SPM in 1999. In addition to this, the results for the trial SPM exam for the year of 1999 were also collected. In the year range mentioned above, it is found that the students had taken six core subjects and three different elective subjects. The input variables considered in this study are listed below in table 4.4:

Var. #	Definition
1	Year
2	Gender
3	Malay Language
4	English
5	Islamic Knowledge
6	History
7	Mathematics
8	Science

Table 4.4: Input variables used to predict SPM result

The other three elective subjects were varied differently from year to year and among the students themselves. Among the elective subjects that had been taken by the different groups of students at the school were: Malay Literature, Art Education, Geography, Basic Economics, Home Economics, Additional Mathematics and Principles of Accounts. Hence, the elective subjects would not be considered or included as the input variables.

4.2 Design Of Network Architecture And Learning Method

In this study, a specific configuration for the neural network model was decided regarding especially to the important network parameters. This study believed that proper neural network design was critical for generating good forecasts as suggested by Awad (1996). These decisions that affect the performance of the multi-layer perceptron network include the following:

- Inputs, outputs, hidden layer and hidden nodes
- The transfer function used by the nodes
- The learning algorithm
- The initial values of the weights between nodes

(i). Inputs, outputs, hidden layer and hidden nodes

As mentioned earlier on, this study selected only six core subjects out of the nine subjects taken by the students in the SPM exam. Since the other three elective subjects differed very much from year to year, they were excluded from the training set. It had also been decided to include the student gender and the year the student took the exam as the input patterns. Since there were eight input attributes, hence, the input layer of the neural network consisted of 8 neurons corresponding to the stated inputs above.

Regarding the outputs used in this study, as explained before there were five outputs altogether corresponding to the students' overall grades for the SPM. Therefore, as suggested by Tsoukalas and Uhrig (1997), five neurons were used.

As for the number of hidden layer of the neural network model, since this study dealt with a continuous mapping, hence, only one hidden layer was needed (Tsoukalas and Uhrig, 1997; Gallant, 1993; Dutta, 1993). Moreover, it was believed that a second hidden layer was not necessary since in most cases it did not produce a large improvement in performance (Hudson and Postma, 1995).

With respect to the number of hidden neurons in the hidden layer, this study opted for performing trial and error runs to ensure the optimum numbers of hidden neurons were identified (Hrycej, 1992; Henseler, 1995; Tsoukalas and Uhrig, 1997; McMenamin, 1997). This study first chose two most efficient numbers of hidden neurons and after that the best number of hidden neuron was determined. Here, in order to identify the best number of hidden neuron, the seed for the random number generation was altered ten times and the average result was calculated.

This study believed that it was very important to determine a suitable number of hidden nodes. Too few neurons in the hidden layer prevent it from correctly mapping inputs to outputs, while too many impede generalization and add to training time. In other words, excessive number of neurons may allow the network to "memorize" the patterns presented to it without taking out significant features for generalization (Tsoukalas and Uhrig, 1997).

(ii). The Transfer Function Used By The Nodes

The activation function chosen in this study was the sigmoid function. Nevertheless this study also tried to determine the best activation function by testing and training the neural network model involving other two common activation functions i.e. the linear function and the hyperbolic tangent (tanh) function.

(iii). The Learning Algorithm

This study used backpropagation learning algorithm since it was the best choice and the most applicable to the problem under study (Dutta, 1993). Furthermore, this architecture was selected since it satisfied the mapping requirement of the application.

In this study, steepest descent algorithm was used to change the weights of the connections between the neurons. The steepest descent method measured the gradient of the error surface after each iteration and changed the weights in the direction of the steepest gradient. When a minimum was reached, a new gradient was measured and the weights were changed in the new direction.

As for this study, in order to improve the usefulness of the steepest descent method, two parameters were altered, namely the learning coefficient and the momentum coefficient. The learning coefficient was used to specify whether the network was going to make major adjustment after each learning trial. The best value of learning coefficient fell in between 0 and 1 Blum (1992). This study performed several trial and errors to determine the best learning coefficient for the neural network model.

While the momentum coefficient was needed to control possible oscillation in the weights, which could be caused by alternately signed error signals. This study opted to use 0.9 as the momentum coefficient as suggested by Gallant (1993), Henseler, (1995), and Tsoukalas and Uhrig (1997). However, in order to ensure the best value had been chosen, this study also performed several trial and error runs to confirm the choice made.

(iv). The Initial Values Of The Weights Between Neurons

With the use of steepest descent learning algorithm, the node input weights were selected for updating after an entire pass of patterns or epoch. In this study, a uniform distribution function had been used whereby the initial weights were randomly selected and whose values did not exceed the range set.

4.3 Data Preprocessing

According to Turban (1992), the way the application data is represented and ordered determines the efficiency and possibly the accuracy of results. As for this study, the raw data acquired from the school were not in a proper arrangement to be used by the neural network. Therefore the data were preprocessed and transformed into the type and format accepted by the neural network.

The process of preprocessing the data to make it more meaningful was important since this was extremely beneficial (Tsoukalas and Uhrig, 1997). The raw data collected were reviewed and those, which were unreliable and impractical, were purged for technical and economic reasons (Tsoukalas and Uhrig, 1997).

In view of the fact that preprocessing methods were critical to the neural network's performance (Mendelsohn, 1993), data for this project had to undergo several processes namely data cleansing, data representation, data scaling and normalization, data randomization and data segmentation. After all of these processes had been carried out on the data appropriately, only then the data were really ready for training and testing.

(i). Data Cleansing

Only suitable and reliable data that contain no error would be included in the training set. There were students who had registered to sit for the SPM exam but they did not turn up during the exam period. In other words, they were absent during the exam. So this kind of data would definitely be taken out from the input patterns.

In total, there are 464 students' records. Four records were afraid to contain errors and hence, had to be deleted and excluded from the overall data set. Then another ten records were deleted to ensure a proper and evenly distribution of data. Finally, the remaining 450 instances were used in the input patterns.

The minimum amount of data required for training is 10(M + N) where M equals the number of inputs and N equals the number of outputs. Another rule of thumb is that the number of training cases should be 10 times the number of model weights (Q & A, 1999).

The subjects' points (i.e. results), the student gender and the year were defined as the input vectors to the network. And the overall achievements of the students i.e. the overall grades were defined as the output vector of the network.

(ii). Data Representation

The task of representing the data in a meaningful way is an essential stage in the successful application of a neural approach (Beale *et al*, 1992; Tsoukalas and Uhrig, 1997). Usually the raw data in the training set are in different formats. Some of them are not acceptable or understandable by the neural network.

Consequently, the inappropriate data has to be transformed into network understandable format, particularly numeric form. In this study, the student gender was converted into meaningful forms to a neural network. Numerical code was assigned in such a way that:

Gender	Assigned code
Male	0
Female	1

Table 4.5: Numerical representation for student gender

As for the outputs, the following numerical codes had been assigned to represent the overall grades:

SPM result	Assigned code
Grade 1	1
Grade 2	2
Grade 3	3
"SAP" (Grade 4)	4
Not Qualified (Fail)	5

Table 4.6: Numerical representation for SPM grades

(iii). Data Scaling And Normalization

After representing the data properly, all the data were scaled or normalized accordingly. Since the natural range of the data in this study was different from the network's operating range, the numeric data were normalized or scaled as appropriate (Tsoukalas and Uhrig, 1997). Furthermore, the data in this study were converted to numbers so that it was easy for the neural network to do its job which usually only in the form of scaling and normalization (Eklund, 1994). In addition, as stressed by Mendelsohn, (1993), Awad (1996) and Tsoukalas and Uhrig (1997), neural networks understand only numbers that fall within the range of the neuron activation function i.e. usually 0 to 1, or –1 to +1.

Data in this study were normalized to make sure all input variables were treated equally and to encourage faster convergence. This was very important to ensure that the statistical distribution of values for each net input and output is roughly uniform (Mendelsohn, 1993).

In this study, the result or point obtained for each subject taken by the student was normalized as follows:

$$\text{New X value} = \frac{\text{Old X value}}{\text{Maximum value}} \tag{4.1}$$

Maximum value was used in the formula to limit the maximum value to unity (Tsoukalas and Uhrig, 1997).

From the training set of this study, the year attribute was scaled into the range of 0.1 and 0.9. Scaling of the variable between 0.1 and 0.9 was used to limit the amount of the sigmoid activation function used in the representation of the variables, in order to get out of "network paralysis" in the training process (Tsoukalas and Uhrig, 1997). The formula used to scale the variable mentioned above was:

$$[0.8/(x_{max} - x_{min})]x + [0.9 - 0.8\, x_{max} /(x_{max} - x_{min})]$$

or

$$\left[\frac{(x - x_{min}) * 0.8}{(x_{max} - x_{min})} \right] + 0.1 \qquad (4.2)$$

where

x_{max} and x_{min} are the maximum and the minimum values in the list.

(iv). Data Randomization

Because of the possibility of hidden order correlations in the data, it is very significant to randomize the training file data. Moreover, as for this study, the training set collected mostly consists of ordered examples, i.e. according to the overall SPM grades achieved. Without adjustment, the data could not be presented to the network and if so, the network might not be able to make a generalization. To avoid getting poor results, thus, the training set were randomized.

(v). Data Segmentation

In this study, the training set was split into several segmentations to ensure the neural network performed at its best and to ensure generalization (Roberts and Penny, 1997). Basically, the original data were split into different training and validation data sets. A validation data set was used to assess the performance of the network on data unseen at training time. The network that, on average, generalized best (i.e. has the lowest validation-set error) was chosen. The performance of the optimal network was further tested on a test set, which is independent from either training, or validation sets and was unused in the network development procedure.

At this stage, the sample data were segmented into three sets:

- One set for training
- One set for validation
- And another one set for testing

In the first attempt to train and test the network, the data (referred as Data Set A) were distributed as follows:

- 80% of the data was allocated for training
- Another 10% for validation
- And the remaining 10% was for testing (Roshidi, 1999)

So, at this stage, instead of training on all of the available data, only 80% of the total data were trained. The other portions were reserved for testing and validating the trained network.

In order to determine the best allocation for data segmentation, another different approach was done on the data. Here, the data set (referred as Data Set B) were distributed differently in such a way that:

- 70% of the data was allocated for training
- Another 20% for validation
- And the remaining 10% was for testing (Roshidi, 1999)

4.4. Training, Validating And Testing The Network

At this stage, the neural network was trained and the trained network was tested and validated. Here the actual process was performed to obtain the best neural network model. As cited by Dutta (1993), prior to the application of a neural network, it had to be trained to recognize patterns from the problem domain. The goal of the training process was to find network weights that made the model errors small (McMenamin, 1997). And the neural network was tested to verify accuracy of the patterns learnt from the training data.

During the process of training and testing the neural network, its performance was examined thoroughly. The best model would be the one that was stable and displayed a consistent set of weights after all the data had been trained (Haykin, 1994). As there were no formal guidelines for structuring and training networks, several trial and error runs with different structures, different training sets, and variations of learning algorithms were done till desirable performance was achieved (Dutta, 1993).

In this study, with alteration to some parameters and maintaining the default values of some other parameters, the model was trained and tested both sequentially and randomly. From that, the model that gave the best result was chosen for further training and testing to identify a few other important network learning parameters. This is done with the aim to identify the best-fitting neural network model.

The network was first trained using one hidden layer with one hidden neuron. The network was then retrained with two hidden neurons. The process is repeated by adding the number of hidden neurons one by one until the number of hidden neuron becomes ten. The experiments were performed several times in order to identify the best number of hidden neurons in the layer.

In this first training run, sigmoid function with steepest descent algorithm was used. Other parameters like the learning and momentum coefficients and the weight seed were not being altered i.e. the network used default values. At first, the Data Set A were trained and tested sequentially. Here the percentage of correctness for the training done was written down.

The experiments were then performed on the Data Set B. The data were also trained and tested sequentially employing the same transfer function and algorithm. The experiments done were also started with one hidden neuron and then the process was repeated by incrementing the number of hidden neurons. Again, here the default parameters values were used just like in the first attempt.

In the third attempt, the Data Set A were trained and tested not sequentially but rather randomly. The same procedures with the first two attempts were also performed on the Data Set A. For the fourth attempt, the Data Set B were trained and tested randomly.

From all of the four attempts that had been done, the approach that gave the highest percentage of correct testing was chosen for further training and testing to determine optimum values for other important parameters. As mentioned and explained before, the performance of the neural network model did depend on several factors or parameters. Therefore, it is very important to identify the best parameters' values.

The best two hidden neurons were chosen from the best approach for further training and testing to identify the optimum number of hidden neurons for the model. Here the experiments were performed in such a way that, the weight seed was changed from one, two and up to ten. The average results were then calculated and the one that gave the highest percentage of correct testing was selected.

After getting the best number of hidden neurons, then the best value of learning coefficient and momentum coefficient were determined by trial and error too. Here, different values of both learning and momentum coefficients were tested on the network. Values, which gave the highest percentage of correctness, were then selected.

As for this study, attempts had also been made to determine the best stopping criteria and also the best activation function for the neural network model.

4.5. Running The Network

Neural networks learn by example the same way humans do. They have a mind of their own; they give opinions but follow no particular rules. They are best at pattern recognition but are not noted for precision (Awad, 1996).

After getting the best-fitting neural network model, then the network was ready to run to do real prediction. This study used the SPM trial exam's results for running on the network generated. A small group of data from the preprocessed trial exam data was presented to the network for prediction. The network was expected to produce a prediction for the SPM grade. In this operational phase, the network was more efficient, since it was no longer subjected to the iterative process of training.

Chapter Five: Findings And Result

As explained in Chapter Four, in Data Set A, 80% of the total sample data were assigned for training, whereas in Data Set B, only 70% of the total data were assigned for training. The neural network used in this study was a multi-layer perceptron with a sigmoidal activation function in the hidden layer and a linear activation function in the output layer.

Several experiments were performed to identify the best network architecture. Learning was stopped when the error on the cross-validation set was no longer decreasing. The trained network was then tested with the testing data set. The performance of each network was measured by looking at correctness percentage of the testing. For each experiment, the best result was retained.

5.1 Determining The Best Number Of Hidden Neurons

As explained in Chapter Four, the input patterns were trained and tested both sequentially and randomly. The results of the application are shown in Table 5.1. Table 5.1 summarizes the results of the learning and testing phases respectively. The % entries in Table 5.1 represent the % of correct prediction of the two approaches (sequential and random).

Experimental Approach	Data Set	Best Two Hidden Neurons	Average Training Result %	Average Testing Result %
Sequential	A	4	75.00	71.11
		9	77.61	75.11
	B	4	69.21	70.67
		6	73.21	69.33
Random	A	4	75.83	76.45
		5	75.43	76.00
	B	5	75.43	75.11
		8	76.95	76.00

Table 5.1: Results from the learning and testing phases for different approach and data set

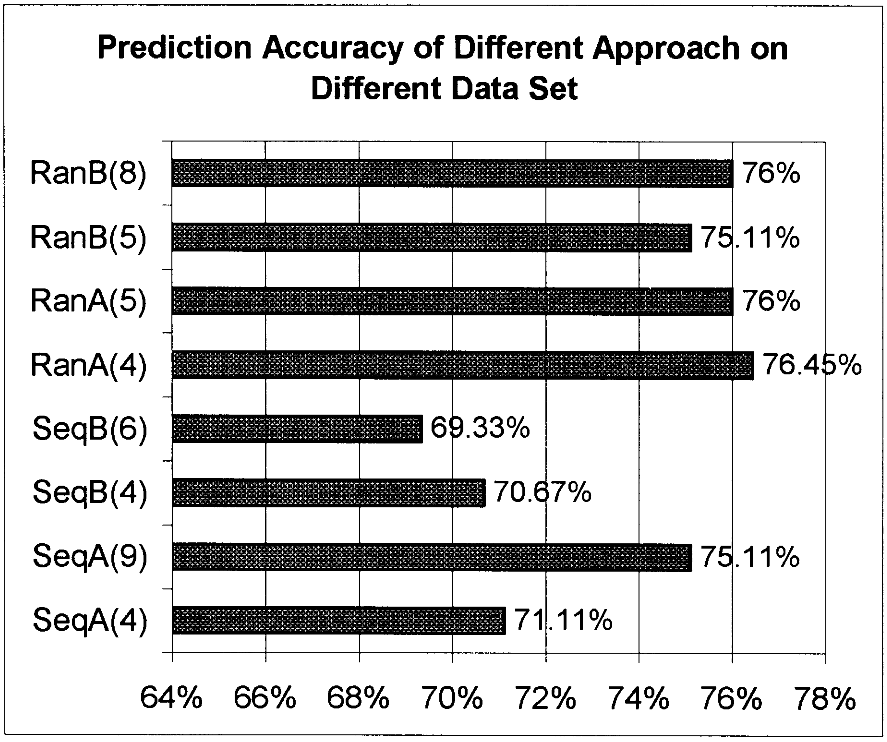

Figure 5.1: The prediction accuracy of two best numbers of hidden neurons for each approach

As can be seen from the Figure 5.1 above, the Data Set A with random testing gave the best result where its % of correct testing were the highest of all (see Experiment 1, 2, 3, and 4 in appendix A for detailed results). This shows that the neural network model requires more training data to be more accurate and effective. Furthermore this also shows that the random approach is much better than the sequential approach. Hence, the two best numbers of hidden neurons were found to be 4 and 5.

Out of the two values obtained above, the best one would be chosen as the most fitting number of hidden neurons of the neural network structure. Networks with 4 and 5 hidden neurons were then tested with 10 different weights initialization. Values obtained with 4 and 5 hidden neurons are presented in the following table (see Experiment 5 in appendix A for detailed results):

	Hidden neurons = 4		Hidden neurons = 5	
Weight Seed	Average Training Result %	Average Testing Result %	Average Training Result %	Average Testing Result %
1	75.83	76.45	75.83	76.00
2	75.22	76.00	76.00	77.33
3	76.05	76.89	76.33	76.89
4	61.56	60.44	75.61	76.44
5	76.17	77.33	73.56	72.44
6	72.50	73.33	71.89	72.45
7	72.11	71.11	76.22	76.89
8	69.22	68.45	76.11	76.89
9	69.06	68.00	73.50	74.22
10	72.89	73.33	75.17	75.56
AVERAGE	72.06	72.13	75.02	75.51

Table 5.2: Results from the training and testing phases for different values of weight seed.

Results from the table 5.2 above shows that the best numbers of hidden neurons are 5 since its average % of correct testing is higher than of 4. So from this point onwards, to complete the experimentations in identifying the best-fitting neural network model that utilized the best parameters, the model employed 5 hidden neurons at the hidden layer of the three-layered network structure.

5.2 Determining The Best Learning Coefficient

Learning Coefficient	Average Training Result %	Average Testing Result %
0.1	75.61	75.56
0.2	75.78	75.56
0.3	75.61	76.00
0.4	75.67	75.56
0.5	75.45	75.56
0.6	76.50	75.56
0.7	76.05	76.00
0.8	76.00	75.56
0.9	75.83	76.00
1.0	72.56	72.44

Table 5.3: Results from the training and testing phase for different learning coefficients

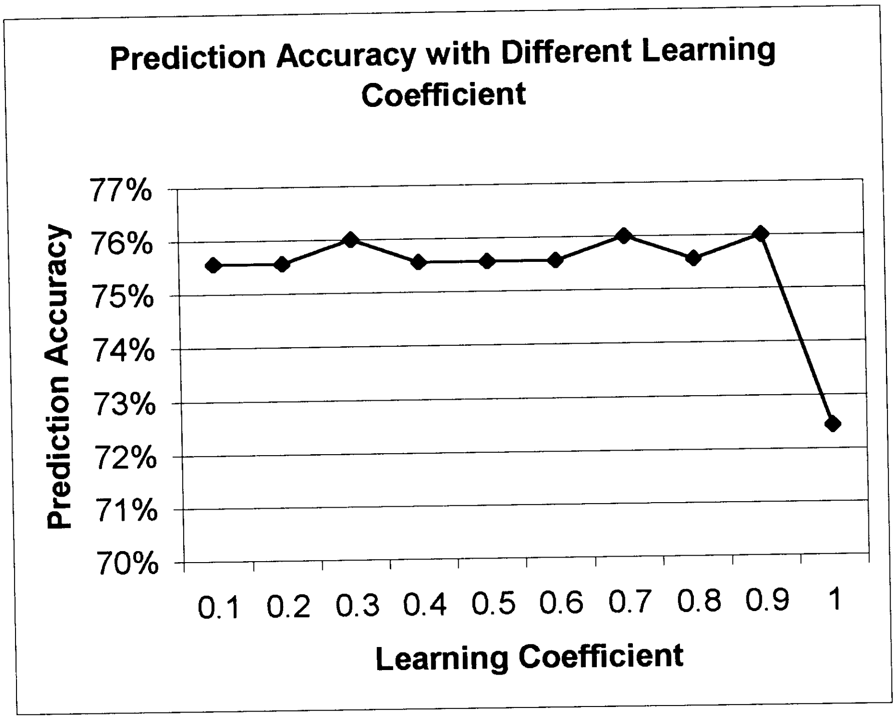

Figure 5.2: The prediction accuracy from testing phase with different learning coefficients

Experiments were performed to choose the optimum value of learning coefficient. Starting with value 0.1, the experiments were repeated until the value was 1.0. The most effective learning coefficient is the one that gave the highest testing percentage but with lowest training percentage. It can be seen from the table 5.3 and figure 5.2 above that there are several coefficients that gave the same percentage of correct testing (see Experiment 6 in appendix A for detailed results). Therefore, learning coefficient with value 0.3 would be the best choice.

5.3. Determining The Best Momentum Coefficient

Momentum Coefficient	Average Training Result %	Average Testing Result %
0.1	75.61	76.00
0.2	75.61	76.00
0.3	75.67	75.56
0.4	75.50	75.56
0.5	75.06	75.56
0.6	76.50	76.00
0.7	75.61	75.56
0.8	75.72	75.56
0.9	77.00	77.33
1.0	69.28	71.11

Table 5.4: The results from training and testing phases with different momentum coefficients

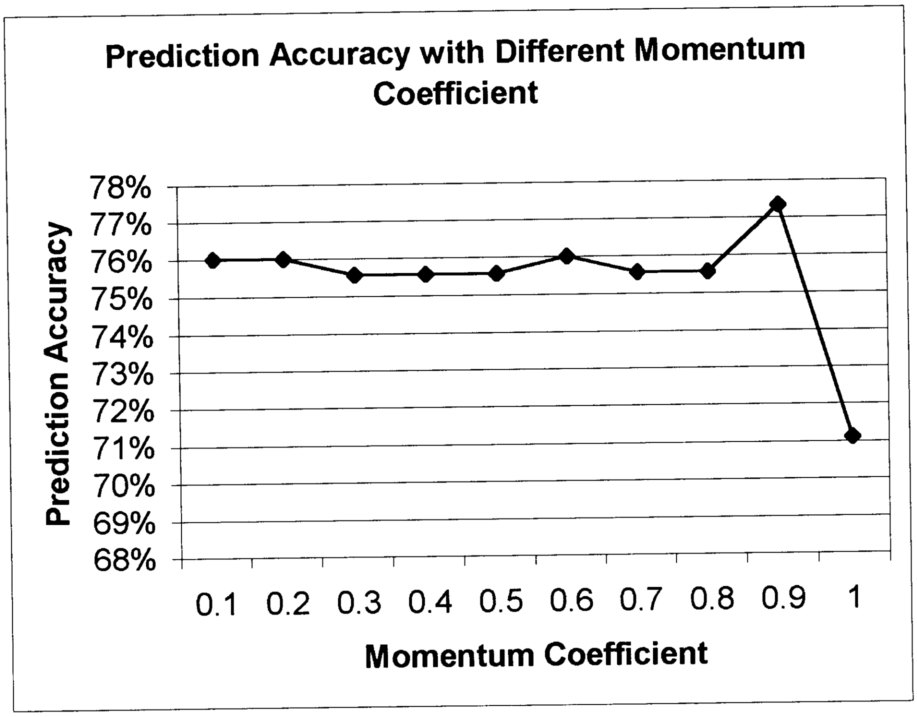

Figure 5.3: The prediction accuracy from testing phase with different momentum coefficient.

From the table 5.4 above, the best momentum coefficient for the neural network model is 0.9 (see Experiment 7 in appendix A for detailed results). With this coefficient the network was able to correctly predict 77.33%.

5.4 Determining The Best Stopping Criteria

Stopping criteria refer to the minimum percentage correct of training and validation sets to cease training. The minimum stopping criteria with the highest testing result would be the best choice.

Stopping Criteria	Average Training Result %	Average Testing Result %
80%	77.00	76.00
85%	77.00	77.33
90%	77.00	77.33
95%	77.00	77.33

Table 5.5: Results from different values of stopping criteria

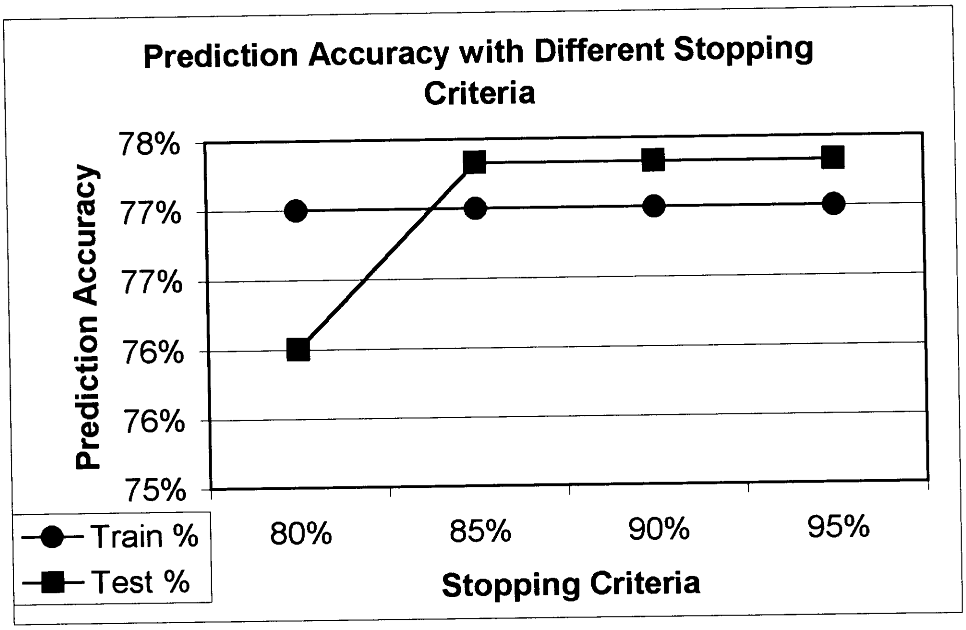

Figure 5.4: A comparison between different stopping criteria

From table 5.5, the best stopping criteria for the neural network model in this study is 85% (see Experiment 8 in appendix A for detailed results). Meaning that, the network is able to predict correctly with only 85% of its data being trained. In other words, it is not necessary to train the whole 100% data from the training set in order to achieve the best result. It is more economical to stop training after 85% of the data has been trained rather than spending more time to train more data. Figure 5.4 depicts the comparison graphically.

5.5. Determining The Best Activation Function

Table 5.6 shows the result of applying different activation functions for the neural network model.

Activation Function	Average Training Result %	Average Testing Result %
LINEAR	50.33	49.78
TANH	77.72	76.44
SIGMOID	77.00	77.33

Table 5.6: Results from training and testing phases with different activation functions

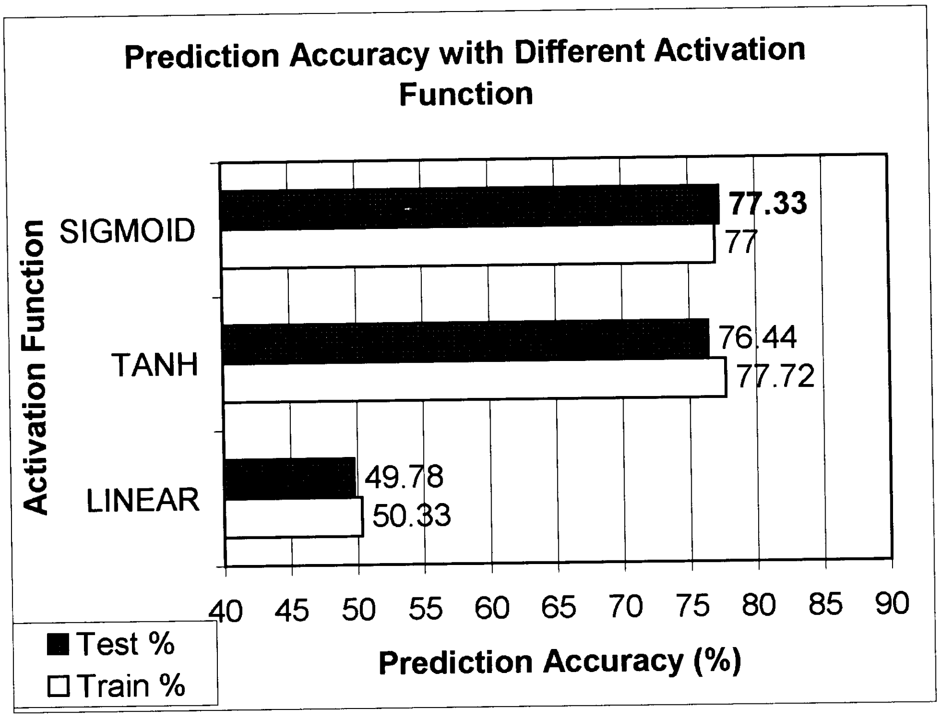

Figure 5.5: A comparison of forecasting model with different activation function

The results shown in table 5.6 and as depicted in figure 5.5 indicate that that Linear function is definitely not suitable to be used at the hidden layer. And as we can see the performance of Sigmoid function outperforms the performance of Tanh function (see Experiment 9 in appendix A for detailed results). In conclusion, the best function is the Sigmoid function.

5.6. Overall Choice Of Architecture

The best network architecture was found to consist of 8 input neurons, 5 in a hidden layer and 5 output neurons. As mentioned previously five units of neurons were in the output layer corresponding to the SPM overall achievements i.e. grades. The analysis of the training data revealed the network was able to identify 77.00% of the training data. And testing on the new data (i.e. the test data set) achieved an accuracy rate of 77.33%. Table 5.7 summarizes the best neural network model built in this study.

PARAMETER	RESULT
Problem type	Prediction
Network architecture	Multi-layer perceptron with backpropagation learning algorithm
Training pattern	Data set A (80% - training, 10% - testing, 10% - validation); random

Number of neurons in the input layer	8
Number of neurons in the output layer	5
Number of hidden layer	1
Number of neurons in the hidden layer	5
Error decent method	Steepest descent
Weights updating method	Update after each pattern
Learning coefficient	0.3
Momentum coefficient	0.9
Stopping criteria	85.00%
Transfer function	Sigmoid function
Correctness of training	77.00%
Correctness of testing	77.33%

Table 5.7: Overall Network Configuration

The final architecture of the network is shown in Figure 5.6. The network contained a single hidden neuron layer of 5 nodes.

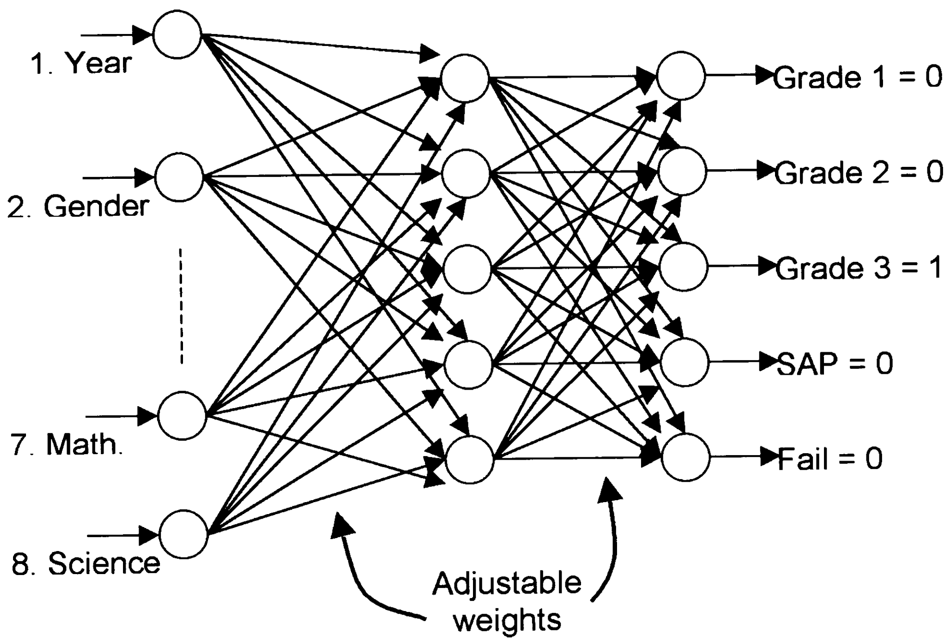

Figure 5.6: The neural network architecture for SPM forecast model

5.7. Testing The Finished Model

For testing the accuracy of the model built, 30 records from the 1999 trial SPM sample data were fed into the finished model to assess the accuracy of the model. Six records from each group of input patterns were selected randomly from the total sample data as the depicted in table 5.8. Besides that, table 5.8 also shows the total correct predicted grades made by the neural network model in comparing with the actual grade. The summary result shows that the model can easily predict the students with grade 1 and grade 2. However, the model does have a problem especially in predicting the SAP students where, out of six records, the model can only predict one record correctly. As explained in previous chapter, determination of SPM grade does not only depend on the six core subjects but also on the other three elective subjects.

SPM Grade	Occurrence of actual grade	Occurrence of correct predicted grade
1	6	6
2	6	6
3	6	2
SAP	6	1
Fail	6	3
Total	30	18

Table 5.8: Summary of actual and correct predicted grades

Nevertheless, as can be seen from table 5.9 that shows sample prediction results, the estimated grades from neural network are very close to the actual grades. In other words, the forecasting output from a neural network is accurate, with an acceptable amount of error. Please refer to Appendix B for in depth results of the actual and predicted SPM grades.

Year	Actual grade	Forecasted grade
1999	SAP	3
1999	SAP	3
1999	SAP	Fail
1999	Fail	Fail
1999	1	1
1999	Fail	Fail
1999	2	2
1999	Fail	SAP
1999	SAP	Fail
1999	3	2

Table 5.9: Comparison of actual and forecasted SPM grades

This study shows that neural network can be trained with students' data to predict their achievement in the examination.

CHAPTER SIX: CONCLUSION AND RECOMMENDATIONS

6.1. Conclusion

In this thesis, the procedures of SPM result forecasting for the Dato' Wan Ahmad Rasdi Secondary School, using a neural network, are presented. The sample data of real SPM results were divided into a training data set, a testing data set, and a validating data set. The input nodes of the neural network held variables for factors that determine the SPM grades. The output nodes corresponded to the overall SPM grades for the students.

The model developed in this study tries to predict individual student performance in the SPM exam. Other, more considerable input variables may exist, and undeniably, all of the chosen input variables in this study may not be helpful for the forecast. However, the resulting neural network from this study that uses these input variables does produce good results. Accordingly this tends to confirm the choice of variables as sufficient, but of course not necessarily the minimum or optimum set.

Based on the results of this study, there is enough evidence to suggest that the application of neural network in predicting the students' performance is worth the effort. As have been explained, in the previous chapters, the grade awarded to the student is based on certain criteria and conditions. The grade achieved is not just based on the core subjects but also on the elective subjects taken by the student. However, this study has not considered the elective subjects due to the various elective subjects being offered by the school. The choices made by the students are very much different from each other. The elective subjects are not included in the training pattern to avoid inconsistencies.

Actually, building a neural network model of student performance is not that easy due to the fact that there is such a large variation among students. Furthermore, there are a lot of outside factors, which can't be measured and fed into the model. Factors like society and social influences, family and home background, health, student characteristics, extracurricular activities and even the condition of the school building play a role in the student's performance. The task of developing a model even seems more daunting when there are continuing changes in the curriculum itself.

For example, Ministry of Education is yet to introduce a new improved system. Starting from the year 2000, a new SPM exam system, which is called "an open certification", will be put into practice. Under this system students are allowed to choose from 12 packages. The system is more flexible and would cater to the individual interests of students. In this new system, students will be given certificates as long as they pass Malay Language.

In prediction problem like predicting the student performance (not just in SPM but also in other kind of examination) or even the school performance, the choice of relevant inputs is the major issue. To get the input that contain as much relevant information as possible would be more difficult than to present this information to the neural network.

Many factors mentioned earlier on are not recorded by the school or even by the students themselves. The things happen and just go by. The only things being recorded were their grades and results achieved from their exams. Other things remain unrecorded. This especially difficult when dealing with neural network where what we require is historical data about something. We cannot get the historical data if the data are not recorded and kept somewhere. This could hinder the attempts to apply neural network model in area related to education.

Neural network research and development may succeed or may fail. In either event, we will certainly be the richer for trying. If all we accomplish with our efforts is to have a better understanding and appreciation of the marvelous complexity of our own brain, the journey may well be worth the price. In any event, the quest will be exciting and interesting (Caudill and Butler, 1990).

In conclusion, this study has been successful in developing a neural network model to predict the SPM grades for the form five students. It is hoped that this study could serve as the starting point for future research in areas related to education involving artificial neural network model. Despite all the problems, the neural network approach is still much better than the conventional statistical and analytical approach.

6.2. Recommendations

A future research possibility is to include more predictive variables to improve forecasts further and to determine the SPM grade forecasting efficiency of a neural network. For instance, factors like the student's ethnic and religion, the student's previous achievement in major exam like PMR could play important roles in determining the overall student's achievement in the SPM.

Another potential area for future research would be to perform the SPM forecast, using a neural network, for different types of school. Malaysian education supports various schools namely, religious school, academic school, and also technical school. The different schools are also located in different areas. There are schools which are located in rural areas and there are others which are located in urban areas. Therefore, these factors i.e. type and location of school could also contribute to the students' performance in their SPM.

Reference:

Adya, M. and Collopy, F. (1998), "How Effective Are Neural Networks At Forecasting And Prediction? A Review And Evaluation", *Journal of Forecasting,* Sep – Nov, Vol. 17, Issue 5 & 6, pp. 481 – 495.

Aiken, M. (1999), "Competitive Intelligence Through Neural Networks", *Competitive Intelligence Review.* 1st Quarter, Vol. 10, Issue 1, pp. 49 – 53.

Aleksander, I. and Morton, H. (1995), *An Introduction to Neural Computing* (2nd ed.), International Thomson Computer Press.

Angstenberger, J. (1996), "Prediction of the S&P 500 Index with Neural Networks", in J. G. Taylor (Ed.), *Neural Networks And Their Applications,* pp. 143 – 152, John Wiley & Sons Ltd.

Armand M. de Callatay (1992), *Natural and Artificial Intelligence: Misconceptions about Brains and Neural Networks*, Elsevier Science Publishers.

Awad, E. M. (1996), *Building Expert Systems: Principles, Procedures, and Applications*, West Publishing Company.

Bahrom Sanugi and Shang, S. W. (1996), "Neural Network Approach in Predicting KLSE Composite Index", *Laporan Teknik Jabatan Matematik,* Universiti Teknologi Malaysia.

Beale, R. and Edwards, A. D. N. (1992), Recognising Postures and Gestures Using Neural Network, in Beale, R. and Finlay, J. (Eds.), *Neural Networks and Pattern Recognition in Human-Computer Interaction*, pp. 163 – 172, Ellis Horwood.

Bengio, Y. (1995), *Neural Networks for Speech and Sequence Recognition,* International Thomson Computer Press.

Blum, A. (1992), *Neural Networks in C++: An Object-Oriented Framework For Building Connectionist Systems*, John Wiley & Sons, Inc.

Bode, J. (1998), "Neural Networks For Cost Estimation", *Cost Engineering*, Jan, Vol. 40, Issue 1, pp. 25 – 30.

Bose, N. K. and Liang, P. (1996), *Neural Network Fundamentals with Graphs, Algorithms and Applications*, McGraw-Hill, Inc.

Caudill, M. and Butler, C. (1990), *Naturally Intelligent Systems*, The MIT Press.

Clark, D. W. (1997), An Introduction to Neural Networks, URL for this article is http://members.home.net/neuralnet/introtonn/index.htm

Davalo, E. and Naïm, P. (1991), *Neural Networks*, Macmillan.

Ding, X., Canu, S., and Denoeux, T. (1996), "Neural Network Based Models for Forecasting", in J. G. Taylor (Ed.), *Neural Networks And Their Applications*, pp. 153 – 167, John Wiley & Sons Ltd.

Dutta, S. (1993), *Knowledge Processing and Applied Artificial Intelligence*, Butterworth Heinemann.

Eklund, P. (1994), Network Size Versus Preprocessing, in Yager, R. R. and Zadeh, L. A. (Eds.), *Fuzzy Sets, Neural Networks, and Soft Computing*, pp. 250 – 264, Van Nostrand Reinhold.

Fu, LiMin (1999), "Knowledge Discovery Based On Neural Network", *Association For Computing Machinery, Communications of the ACM*, Nov, Vol. 42, Issue 11, pp. 47 – 50.

Gallant, S. I. (1993), *Neural Network Learning and Expert Systems*, The MIT Press.

Gomm, J. B., Page, G. F. and Williams, D. (1993), Introduction to Neural Networks, in Page, G. F., Gomm, J. B., and Williams, D. (Eds), *Application of Neural Networks to Modelling and Control,* pp. 1 – 8, Chapman & Hall.

Haykin, S. (1994), *Neural Networks: A comprehensive Foundation*, Macmillan College Publishing Company.

Henseler, J. (1995), "Back Propagation", in Braspenning, P. J., Thuijsman, F. and Weijters, A. J. M. M. (Eds.), *Artificial Neural Networks: An Introduction to ANN Theory and Practice*, pp. 37 – 66, Springer.

Homepage of Dato' Wan Ahmad Rasdi Secondary School, the URL is http://welcome.to/smkb8

Hrycej, T. (1992), *Modular Learning in Neural Networks: A Modularized Approach To Neural Network Classification*, John Wiley & Sons, Inc.

Huang, Chia-Sheng (1999), *An Information Technology Approach to Production and Operations Decision Making in Dairy Processing*, Dissertation (PhD), The University of Wisconsin.

Hudson, P. T. W. and Postma, E. O. (1995), "Choosing and Using a Neural Net", in Braspenning, P. J., Thuijsman, F. and Weijters, A. J. M. M. (Eds.), *Artificial Neural Networks: An Introduction to ANN Theory and Practice*, pp. 273 – 287, Springer.

Indro, D. C., Jiang, C. X., Patuwo, B. E., and Zhang, G. P. (1999), "Predicting Mutual Performance Using Artificial Neural Networks", *Omega,* Jun, Vol. 27, Issue 3, pp. 373 – 380.

Kim, S. H. (1994), *Learning and Coordination: Enhancing Agent Performance Through Distributed Decision Making*, Kluwer Academic Publishers.

Kohonen, T. (1988), "An Introduction to Neural Computing", *Neural Networks*, Vol. 1, pp. 3 – 16.

Law, R. (1998), "Room Occupancy Rate Forecasting: a Neural Network Approach", *International Journal of Contemporary Hospitality Management*, Vol. 10, Issue 6, pp. 234 – 239.

Lipmann, R. P. (1987), "An Introduction to Computing with Neural Nets", *IEEE ASSP (Acoustics, Speech, and Signal Processing) Magazine*, April, Vol. 4, No. 2, pp. 4 – 22.

Lisboa, P. G. J. (1992), (Ed.), *Neural Networks Current Applications,* Chapmann & Hall.

Luger, G. F. and Stubblefield, W. A. (1989), *Artificial Intelligence and the Design of Expert Systems,* The Benjamin/Cummings Publishing Company, Inc.

Mahmood, M. A., Sullivan, G. L., and Tung, R. (1999), "A New Approach To Evaluating Business Ethics: An Artificial Neural Networks Application", *Journal of End User Computing,* Jul – Sep, Vol. 11, Issue 3, pp. 11 – 19.

McMenamin, J. S. (1997), "A Primer On Neural Networks For Forecasting", *The Journal of Business Forecasting Methods & Systems,* Fall, Vol. 16, Issue 3, pp. 17 – 22.

Mendelsohn, L. (1993), Preprocessing Data For Neural Networks, URL for this article is http://www.wetradefutures.com/preprocessing_data.htm

Moriera, M. and Fiesler, E. (1995), Neural Networks with Adaptive Learning Rate and Momentum Terms, *IDIAP Technical Report.*

Muzaffer Uysal and M Sherif El Roubi (1999), "Artificial Neural Networks Versus Multiple Regression In Tourism Demand Analysis", *Journal of Travel Research,* Nov, Vol. 38, Issue 2, pp. 111 – 118.

Q & A (1999), Neural Connection: Questions and Answers, URL for this article is http://www.spss.com/software/Neuro/ncqa.html

Raggett, J. and Bains, W. (1992), *Artificial Intelligence From A to Z*, Chapman & Hall.

Ramlan Goddos (1997), "Pendidikan Hari Ini: Realiti, Isu Dan Cabaran", *Berita IDS Development Review*, Vol. 12, No. 2, April-June.

Renders, J. M., de Vlaminde, M. and De Viron, F. (1998), "ANN in Process Monitoring a practical experience with a prototype to Perform Early Warning in Nuclear Power Plant", in Fogelman-Soulié, F. and Gallinari, P. (Eds.), *Industrial Applications of Neural Network*, World Scientific.

Ritter, H., Martinetz, T. and Schulten, K. (1992), *Neural Computation and Self-Organizing Maps*, Addison-Wesley Publishing Company.

Roberts, S. J. and Penny, W. (1997), "Neural Networks: Friends or Foes?", *Sensor Review*, Vol. 17, No. 1, pp. 64 – 70.

Rohwer, R., Wynne-Jones, M. and Wysotzki, F. (1994), Neural Networks, in Michie, D, Spiegelhalter, D. J., and Taylor, C. C. (Eds.), *Machine Learning, Neural and Statistical Classification*, pp. 84 – 106, Ellis Horwood.

Roselina S., Mohd Salihin N. and Siti Mariyam S. (1999), "Penentuan Saiz dan Bilangan Nod Tersembunyi Rangkaian Neural Bagi Ramalan", *Jurnal Teknologi Maklumat*, Vol. 11, No. 1, Jun, pp. 67 – 80.

Roshidi Din (1999), *Teknik Tapisan Dalam Pemodelan Perceptron Multi Aras*, Thesis (M. Sc), Universiti Utara Malaysia, Sintok Kedah.

Schalkoff, R. J. (1997), *Artificial Neural Networks*, The McGraw-Hill.

Schürmann, J. (1996), *Pattern Classification: A Unified View of Statistical and Neural Approaches*, John Wiley & Sons, Inc.

Silvert, W and Baptist, M (1998), Can Neural Networks Be Used In Data-Poor Situations? URL for this article is http://www.ipimar.pt/~wsilvert/output/toulouse/

Sutton, R. S. (1988), "Learning to Predict by the Methods of Temporal Differences", *Machine Learning*, Aug, Vol. 3, No. 1, pp. 9 – 44.

Tsoukalas, L. H., and Uhrig, R. E. (1997), *Fuzzy and Neural Approaches in Engineering*, John Wiley & Sons, Inc.

Turban, E. (1992), *Expert Systems and Applied Artificial Intelligence*, Macmillan Publishing Company.

Villegas, L. and Eberts, R. (1992), "Implementing a Neural Network for a Cognitive Text-editing Task", in Beale, R. and Finlay, J. (Eds.), *Neural Networks and Pattern Recognition in Human-Computer Interaction*, pp. 267 – 282, Ellis Horwood.

Weijters, A. J. M. M. and Hoppenbrouwers, G. A. J. (1995), "Backpropagation Networks for Grapheme-Phoneme Conversion: a Non-Technical Introduction", in Braspenning, P. J., Thuijsman, F. and Weijters, A. J. M. M. (Eds.), *Artificial Neural Networks: An Introduction to ANN Theory and Practice*, pp. 11 – 36, Springer.

Appendix A
Detailed results of the experiments

Data Set A where,	Data Set B, where,
▪ 80%: Training ▪ 10%: Validation ▪ 10%: Testing	▪ 70%: Training ▪ 20%: Validation ▪ 10%: Testing

Experiment 1:
The Data Set A was trained and tested sequentially

Results From Training Phase

Hidden neuron	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
1	54.72	48.89	41.39	41.67	91.67	55.67
2	86.94	49.17	48.33	51.94	71.67	61.61
3	84.44	66.94	60.56	53.61	73.89	67.89
4	86.94	71.94	68.33	66.39	81.39	75.00
5	86.11	70.83	66.67	65.28	81.94	74.17
6	86.11	71.94	66.39	65.00	81.11	74.11
7	86.39	72.50	66.94	65.28	81.94	74.61
8	87.22	73.61	71.11	68.89	83.33	76.83
9	88.06	75.28	71.39	69.72	83.61	77.61
10	86.11	65.44	65.28	63.06	81.39	72.26

Results From Testing Phase

Hidden neuron	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
1	40.00	35.56	46.67	46.67	100.00	53.78
2	86.67	28.89	11.11	60.00	86.67	54.67
3	88.89	55.56	42.22	60.00	86.67	66.67
4	91.11	57.78	48.89	68.89	88.89	71.11
5	88.89	55.56	48.89	68.89	88.89	70.22
6	91.11	57.78	46.67	68.89	88.89	70.67
7	88.89	60.00	44.44	68.89	88.89	70.22
8	93.33	51.11	42.22	55.56	93.33	67.11

9	93.33	62.22	57.78	77.78	84.44	75.11
10	88.89	53.33	46.67	66.67	88.89	68.89

<u>Experiment 2:</u>
The Data Set A was trained and tested randomly

Results From Training Phase

Hidden neuron	Output	Output	Output	Output	Output	Average Training Result %
1	55.28	43.89	37.50	41.67	93.61	54.39
2	88.33	46.67	35.83	53.33	87.22	62.28
3	86.67	64.17	57.78	61.11	80.56	70.06
4	88.61	69.17	64.44	70.00	86.94	75.83
5	88.89	70.28	64.44	67.17	86.39	75.43
6	88.33	70.28	65.00	70.83	86.39	76.17
7	85.56	68.61	60.28	67.50	86.39	73.67
8	89.44	70.56	65.83	70.00	86.39	76.44
9	85.83	65.28	58.33	65.28	86.11	72.17
10	84.44	59.17	51.67	57.22	83.61	67.22

Results From Testing Phase

Hidden neuron	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
1	55.56	51.11	40.00	42.22	86.67	55.11
2	84.44	48.89	35.56	46.67	80.00	59.11
3	84.44	73.33	66.67	57.78	75.56	71.56
4	86.67	77.78	77.78	64.44	75.56	76.45
5	84.44	77.78	77.78	64.44	75.56	76.00
6	84.44	75.56	73.33	64.44	75.56	74.67
7	80.00	77.78	75.56	62.22	75.56	74.22
8	88.89	73.33	75.56	66.67	75.56	76.00
9	82.22	75.56	64.44	57.78	75.56	71.11
10	77.78	71.11	60.00	48.89	73.33	66.22

<u>Experiment 3:</u>
The Data Set B was trained and tested sequentially

Results From Training Phase

Hidden neuron	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
1	52.06	45.08	42.86	46.67	92.70	55.87
2	84.13	47.30	24.13	54.60	77.14	57.46
3	84.13	65.71	59.68	58.73	77.14	69.08
4	84.13	66.67	60.32	59.37	75.56	69.21
5	85.08	69.52	67.62	67.94	83.81	74.79
6	84.76	68.57	66.98	65.40	80.32	73.21
7	84.76	69.21	67.94	66.35	81.27	73.91
8	85.08	69.52	70.48	67.30	81.59	74.79
9	83.81	67.30	65.08	63.17	80.32	71.94
10	85.71	68.57	66.67	67.94	82.86	74.35

Results From Testing Phase

Hidden neuron	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
1	40.00	35.56	42.22	48.89	100.00	53.33
2	84.44	31.11	6.67	62.22	86.67	54.22
3	84.44	51.11	42.22	60.00	86.67	64.89
4	91.11	57.78	46.67	66.67	91.11	70.67
5	84.44	51.11	48.89	66.67	91.11	68.44
6	86.67	53.33	46.67	68.89	91.11	69.33
7	88.89	55.56	44.44	66.67	91.11	69.33
8	93.33	55.56	44.44	53.33	91.11	67.55
9	84.44	51.11	46.67	66.67	91.11	68.00
10	88.89	46.67	48.89	71.11	88.89	68.89

Experiment 4
The Data Set B was trained and tested randomly

Results From Training Phase

Hidden neuron	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
1	50.48	43.17	40.32	46.67	93.97	54.92
2	86.35	47.62	21.59	53.33	93.97	60.57
3	88.89	60.32	33.65	49.52	86.03	63.68
4	88.25	69.84	55.24	66.35	87.30	73.40
5	86.67	69.84	65.40	68.57	86.67	75.43
6	87.30	70.16	65.08	66.67	86.67	75.18
7	88.25	71.43	63.17	67.30	86.67	75.36
8	89.21	71.11	68.25	69.84	86.35	76.95
9	86.98	69.84	63.17	68.25	87.30	75.11
10	85.71	69.21	62.22	66.98	86.03	74.03

Results From Testing Phase

Hidden neuron	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
1	57.58	57.78	37.78	42.22	86.67	56.41
2	88.89	53.33	15.56	42.22	86.67	57.33
3	88.89	66.67	33.33	42.22	71.11	60.44
4	88.89	82.22	60.00	55.56	75.56	72.45
5	88.89	77.78	75.56	60.00	73.33	75.11
6	86.67	82.22	73.33	55.56	73.33	74.22
7	88.89	84.44	68.89	55.56	73.33	74.22
8	88.89	82.22	73.33	62.22	73.33	76.00
9	86.67	82.22	62.22	57.78	75.56	72.89
10	86.67	82.22	64.44	53.33	73.33	72.00

Summary Results From Experiments 1, 2, 3, and 4

Experiment 1

Hidden neuron	Average Training Result %	Average Testing Result %
1	55.67	53.78
2	61.61	54.67
3	67.89	66.67
4	75.00	71.11
5	74.17	70.22
6	74.11	70.67
7	74.61	70.22
8	76.83	67.11
9	77.61	75.11
10	72.26	68.89

Experiment 3

Hidden neuron	Average Training Result %	Average Testing Result %
1	55.87	53.33
2	57.46	54.22
3	69.08	64.89
4	69.21	70.67
5	74.79	68.44
6	73.21	69.33
7	73.91	69.33
8	74.79	67.55
9	71.94	68.00
10	74.35	68.89

Experiment 2

Hidden neuron	Average Training Result %	Average Testing Result %
1	54.39	55.11
2	62.28	59.11
3	70.06	71.56
4	75.83	76.45
5	75.43	76.00
6	76.17	74.67
7	73.67	74.22
8	76.44	76.00
9	72.17	71.11
10	67.22	66.22

Experiment 4

Hidden neuron	Average Training Result %	Average Testing Result %
1	54.92	56.41
2	60.57	57.33
3	63.68	60.44
4	73.40	72.45
5	75.43	75.11
6	75.18	74.22
7	75.36	74.22
8	76.95	76.00
9	75.11	72.89
10	74.03	72.00

From all four experiments that were performed previously, experiment 2 (Data Set A, random training and testing) provided the best result. Experiment 5 then was performed to choose the best number of hidden neurons (between 4 and 5) using the Data Set A. In this experiment, the weight seed was first assigned to 1 and then the value was increased one at a time up to 10.

Experiment 5

(i) Changing the Weight Seed with Hidden Neurons = 4

Results From Training Phase

Weight Seed	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
1	88.61	69.17	64.44	70.00	86.94	75.83
2	88.33	68.89	63.61	68.89	86.39	75.22
3	88.33	69.72	64.72	70.83	86.67	76.05
4	87.78	53.06	32.78	48.33	85.83	61.56
5	88.61	70.00	65.00	70.83	86.39	76.17
6	87.78	66.11	59.17	64.17	85.28	72.50
7	86.67	64.44	57.78	65.56	86.11	72.11
8	85.56	60.56	53.06	60.83	86.11	69.22
9	85.83	61.11	55.28	57.78	85.28	69.06
10	86.67	66.11	59.44	66.11	86.11	72.89
Total						720.61
Average						72.06

Results From Testing Phase

Weight Seed	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
1	86.67	77.78	77.78	64.44	75.56	76.45
2	84.44	77.78	77.78	64.44	75.56	76.00
3	86.67	77.78	80.00	64.44	75.56	76.89
4	84.44	60.00	40.00	44.44	73.33	60.44
5	86.67	82.22	77.78	64.44	75.56	77.33
6	84.44	75.56	73.33	57.78	75.56	73.33
7	84.44	73.33	64.44	57.78	75.56	71.11

8	82.22	66.67	62.22	55.56	75.56	68.45
9	82.22	68.89	64.44	51.11	73.33	68.00
10	84.44	75.56	71.11	60.00	75.56	73.33
Total						721.33
Average						72.13

(ii) Changing the Weight Seed with Hidden neurons = 5

Results From Training Phase

Weight Seed	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
1	88.89	70.28	64.44	69.17	86.39	75.83
2	88.89	70.28	63.33	71.11	86.39	76.00
3	88.61	69.72	65.00	71.67	86.67	76.33
4	88.33	70.83	63.61	69.44	85.83	75.61
5	87.78	66.11	60.28	67.22	86.39	73.56
6	86.39	64.72	56.94	65.00	86.39	71.89
7	89.44	70.00	65.28	70.00	86.39	76.22
8	88.33	70.56	65.00	70.56	86.11	76.11
9	87.22	68.06	61.11	66.11	85.00	73.50
10	87.50	68.61	63.61	69.72	86.39	75.17
Total						750.22
Average						75.02

Results From Testing Phase

Weight Seed	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
1	84.44	77.78	77.78	64.44	75.56	76.00
2	86.67	82.22	77.78	64.44	75.56	77.33
3	86.67	77.78	80.00	64.44	75.56	76.89
4	84.44	80.00	77.78	64.44	75.56	76.44
5	84.44	75.56	64.44	60.00	77.78	72.44
6	84.44	75.56	68.89	57.78	75.56	72.45
7	88.89	77.78	77.78	64.44	75.56	76.89
8	84.44	77.78	80.00	66.67	75.56	76.89

9	84.44	75.56	75.56	60.00	75.56	74.22
10	84.44	75.56	77.78	64.44	75.56	75.56
Total						755.12
Average						75.51

After performing experiment 5, the best value of hidden neuron was identified by selecting the one that gave the highest percentage of correct testing. Experiment 6 was then performed to determine the best value of learning coefficient.

Experiment 6

To Determine The Best Learning Coefficient With Hidden Neurons = 5

Results From Training Phase

Learning Coefficient	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
0.1	88.89	70.28	64.17	68.61	86.11	75.61
0.2	89.17	70.56	64.17	68.61	86.39	75.78
0.3	88.89	70.28	63.89	68.61	86.39	75.61
0.4	87.78	69.72	64.72	69.44	86.67	75.67
0.5	87.78	67.78	64.44	70.56	86.67	75.45
0.6	88.61	69.72	65.56	71.67	86.94	76.50
0.7	88.89	69.44	65.28	69.72	86.94	76.05
0.8	88.61	69.72	65.28	69.72	86.67	76.00
0.9	88.89	70.28	64.44	69.17	86.39	75.83
1.0	86.39	64.17	59.72	66.39	86.11	72.56

Results From Testing Phase

Learning Coefficient	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
0.1	84.44	75.56	77.78	64.44	75.56	75.56
0.2	84.44	75.56	77.78	64.44	75.56	75.56
0.3	84.44	77.78	77.78	64.44	75.56	76.00
0.4	84.44	75.56	77.78	64.44	75.56	75.56
0.5	84.44	75.56	77.78	64.44	75.56	75.56
0.6	84.44	75.56	77.78	64.44	75.56	75.56
0.7	84.44	77.78	77.78	64.44	75.56	76.00
0.8	84.44	75.56	77.78	64.44	75.56	75.56

0.9	84.44	77.78	77.78	64.44	75.56	76.00
1.0	84.44	73.33	68.89	60.00	75.56	72.44

After getting the best value of learning coefficient, experiment 7 was performed to determine the best value of momentum coefficient

Experiment 7

To Determine The Best Momentum Coefficient With Hidden Neurons = 5 And Learning Coefficient = 0.3

Results From Training Phase

Momentum Coefficient	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
0.1	88.89	70.28	63.89	68.61	86.39	75.61
0.2	88.61	70.28	64.17	68.61	86.39	75.61
0.3	88.33	70.00	64.44	69.17	86.39	75.67
0.4	87.50	68.89	64.44	70.00	86.67	75.50
0.5	87.50	67.22	63.61	70.28	86.67	75.06
0.6	88.33	69.44	65.83	72.22	86.67	76.50
0.7	88.06	68.06	64.44	70.83	86.67	75.61
0.8	88.06	70.00	64.44	70.00	86.11	75.72
0.9	89.44	70.28	65.56	72.78	86.94	77.00
1.0	85.83	65.83	58.06	55.56	81.11	69.28

Results From Testing Phase

Momentum Coefficient	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
0.1	84.44	77.78	77.78	64.44	75.56	76.00
0.2	84.44	77.78	77.78	64.44	75.56	76.00
0.3	84.44	75.56	77.78	64.44	75.56	75.56
0.4	84.44	75.56	77.78	64.44	75.56	75.56
0.5	84.44	75.56	77.78	64.44	75.56	75.56
0.6	86.67	75.56	77.78	64.44	75.56	76.00
0.7	84.44	75.56	77.78	64.44	75.56	75.56
0.8	84.44	75.56	77.78	64.44	75.56	75.56
0.9	88.89	80.00	75.56	64.44	77.78	77.33
1.0	80.00	73.33	71.11	55.56	75.56	71.11

Experiment 8 was performed to determine the best stopping criteria for the overall neural network model.

Experiment 8
To Determine The Best Stopping Criteria

Results From Training Phase

Stopping Criteria	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
80%	89.44	70.28	65.56	72.78	86.94	77.00
85%	89.44	70.28	65.56	72.78	86.94	77.00
90%	89.44	70.28	65.56	72.78	86.94	77.00
95%	89.44	70.28	65.56	72.78	86.94	77.00

Results From Testing Phase

Stopping Criteria	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
80%	84.44	77.78	77.78	64.44	75.56	76.00
85%	88.89	80.00	75.56	64.44	77.78	77.33
90%	88.89	80.00	75.56	64.44	77.78	77.33
95%	88.89	80.00	75.56	64.44	77.78	77.33

Experiment 9
To Determine The Best Activation Function

Results From Training Phase

Activation Function	Output 1	Output 2	Output 3	Output 4	Output 5	Average Training Result %
Linear	69.72	32.50	24.17	35.56	89.72	50.33
Tanh	88.06	71.39	67.78	73.89	87.50	77.72
Sigmoid	89.44	70.28	65.56	72.78	86.94	77.00

Results From Testing Phase

Activation Function	Output 1	Output 2	Output 3	Output 4	Output 5	Average Testing Result %
Linear	60.00	40.00	28.89	37.78	82.22	49.78
Tanh	82.22	77.78	77.78	64.44	80.00	76.44
Sigmoid	88.89	80.00	75.56	64.44	77.78	77.33

Appendix B
Actual and Predicted Grades

The actual grades obtained by the students in 1999 trial SPM

	Input Variables								Actual Grade				
No.	Yr	Gend	Subj1	Subj2	Subj3	Subj4	Subj5	Subj 6	Targ 1	Targ 2	Targ 3	Targ 4	Targ 5
1	0.90	0.00	0.56	1.00	0.78	1.00	1.00	1.00	0	0	0	1	0
2	0.90	0.00	0.33	1.00	0.67	0.89	1.00	1.00	0	0	0	1	0
3	0.90	1.00	0.67	1.00	1.00	1.00	1.00	1.00	0	0	0	1	0
4	0.90	1.00	0.89	1.00	1.00	1.00	1.00	1.00	0	0	0	0	1
5	0.90	0.00	0.89	1.00	0.89	1.00	1.00	1.00	0	0	0	1	0
6	0.90	1.00	1.00	1.00	1.00	0.89	1.00	1.00	0	0	0	0	1
7	0.90	1.00	0.44	1.00	0.22	0.89	0.89	0.89	0	1	0	0	0
8	0.90	1.00	0.89	1.00	0.78	0.89	1.00	1.00	0	0	0	0	1
9	0.90	1.00	1.00	1.00	0.89	1.00	1.00	1.00	0	0	0	1	0
10	0.90	1.00	0.78	1.00	0.78	0.78	1.00	1.00	0	0	0	0	1
11	0.90	0.00	0.56	0.89	0.44	0.78	1.00	1.00	0	0	1	0	0
12	0.90	1.00	0.33	0.89	0.44	0.78	1.00	0.89	0	1	0	0	0
13	0.90	0.00	0.22	0.67	0.11	0.67	0.11	0.67	1	0	0	0	0
14	0.90	0.00	0.44	0.89	0.56	0.78	1.00	0.78	0	1	0	0	0
15	0.90	1.00	0.33	0.67	0.44	0.89	1.00	0.89	0	1	0	0	0
16	0.90	0.00	0.22	0.78	0.56	0.56	0.56	0.89	1	0	0	0	0
17	0.90	0.00	0.67	1.00	1.00	1.00	1.00	1.00	0	0	0	1	0
18	0.90	0.00	0.56	0.89	0.78	0.89	1.00	1.00	0	0	1	0	0
19	0.90	0.00	0.33	0.89	0.44	0.89	1.00	0.89	0	0	1	0	0
20	0.90	1.00	0.78	1.00	0.89	1.00	1.00	1.00	0	0	0	0	1
21	0.90	1.00	0.33	0.56	0.22	0.67	0.89	0.78	1	0	0	0	0
22	0.90	0.00	0.33	0.89	0.22	0.67	0.67	0.78	1	0	0	0	0
23	0.90	0.00	0.78	1.00	0.78	0.78	1.00	0.89	0	0	0	0	1
24	0.90	1.00	0.44	1.00	0.67	0.89	1.00	0.89	0	0	1	0	0
25	0.90	0.00	0.22	0.89	0.11	0.67	0.44	0.56	1	0	0	0	0
26	0.90	0.00	0.11	0.78	0.22	0.67	0.44	0.78	1	0	0	0	0
27	0.90	0.00	0.33	0.89	0.56	1.00	1.00	0.89	0	0	1	0	0
28	0.90	0.00	0.33	0.89	0.44	0.78	1.00	1.00	0	1	0	0	0
29	0.90	0.00	0.44	0.89	0.67	0.78	1.00	0.89	0	0	1	0	0
30	0.90	0.00	0.33	1.00	0.22	0.67	1.00	0.78	0	1	0	0	0

The predicted grades of the 1999 trial SPM suggested by the finished neural network model

	Input Variables								Predicted Grade				
No.	Yr	Gend	Subj 1	Subj 2	Subj 3	Subj 4	Subj 5	Subj 6	Targ 1	Targ 2	Targ 3	Targ 4	Targ 5
1	0.90	0.00	0.56	1.00	0.78	1.00	1.00	1.00	0	0	1	0	0
2	0.90	0.00	0.33	1.00	0.67	0.89	1.00	1.00	0	0	1	0	0
3	0.90	1.00	0.67	1.00	1.00	1.00	1.00	1.00	0	0	0	0	1
4	0.90	1.00	0.89	1.00	1.00	1.00	1.00	1.00	0	0	0	0	1
5	0.90	0.00	0.89	1.00	0.89	1.00	1.00	1.00	0	0	0	0	0
6	0.90	1.00	1.00	1.00	1.00	0.89	1.00	1.00	0	0	0	0	1
7	0.90	1.00	0.44	1.00	0.22	0.89	0.89	0.89	0	1	0	0	0
8	0.90	1.00	0.89	1.00	0.78	0.89	1.00	1.00	0	0	0	1	0
9	0.90	1.00	1.00	1.00	0.89	1.00	1.00	1.00	0	0	0	0	1
10	0.90	1.00	0.78	1.00	0.78	0.78	1.00	1.00	0	0	0	1	0
11	0.90	0.00	0.56	0.89	0.44	0.78	1.00	1.00	0	1	0	0	0
12	0.90	1.00	0.33	0.89	0.44	0.78	1.00	0.89	0	1	0	0	0
13	0.90	0.00	0.22	0.67	0.11	0.67	0.11	0.67	1	0	0	0	0
14	0.90	0.00	0.44	0.89	0.56	0.78	1.00	0.78	0	1	0	0	0
15	0.90	1.00	0.33	0.67	0.44	0.89	1.00	0.89	0	1	0	0	0
16	0.90	0.00	0.22	0.78	0.56	0.56	0.56	0.89	1	0	0	0	0
17	0.90	0.00	0.67	1.00	1.00	1.00	1.00	1.00	0	0	0	1	0
18	0.90	0.00	0.56	0.89	0.78	0.89	1.00	1.00	0	0	1	0	0
19	0.90	0.00	0.33	0.89	0.44	0.89	1.00	0.89	0	1	0	0	0
20	0.90	1.00	0.78	1.00	0.89	1.00	1.00	1.00	0	0	0	0	1
21	0.90	1.00	0.33	0.56	0.22	0.67	0.89	0.78	1	0	0	0	0
22	0.90	0.00	0.33	0.89	0.22	0.67	0.67	0.78	1	0	0	0	0
23	0.90	0.00	0.78	1.00	0.78	0.78	1.00	0.89	0	0	0	1	0
24	0.90	1.00	0.44	1.00	0.67	0.89	1.00	0.89	0	0	1	0	0
25	0.90	0.00	0.22	0.89	0.11	0.67	0.44	0.56	1	0	0	0	0
26	0.90	0.00	0.11	0.78	0.22	0.67	0.44	0.78	1	0	0	0	0
27	0.90	0.00	0.33	0.89	0.56	1.00	1.00	0.89	0	1	0	0	0
28	0.90	0.00	0.33	0.89	0.44	0.78	1.00	1.00	0	1	0	0	0
29	0.90	0.00	0.44	0.89	0.67	0.78	1.00	0.89	0	1	0	0	0
30	0.90	0.00	0.33	1.00	0.22	0.67	1.00	0.78	0	1	0	0	0