FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA

INTERACTIVE LEARNING PACKAGE FOR ARTIFICIAL NEURAL NETWORK (DEMONSTRATION MODULE)

CAMELLIA BTE MOHD KAMAL
WEK010026

This technical report is submitted to fulfill part of the requirement for the awarding of the Bachelor of Computer Science degree

JANUARY 2004

TO

My Beloved father, mom, sisters, loving Mal and best friends…

ABSTRACT

ABSTRACT

Interactive Learning Package for Artificial Neural Network (Demonstration Module) is a learning package that allows the users to learn and enhance the understanding of the ANN subject in more detailed way and apply in the demonstration parts. Every demonstration will be followed by a brief and compact explanation therefore the user can understand more clearly about the demonstration and know how to do it very well. The demonstration module provides the users only with the important notes that the user must know because the explanation already described in learning modules. For the tutorial questions, for the demonstration modules most of the questions are the calculation format based on the demonstration example. Every question is related because from there we can know the level of the user. Five main topics will be discussed in this learning package, which are the ***Introduction of Artificial Neural Network, Perceptron, Feed Forward Network, Recurrent Network and Self Organizing Maps (SOM).*** The introduction topic will discuss about the biological neuron with one demo, directed graph description also with one demo, the neuron model that included three subtopic: single input neuron with one demo, transfer function and multiple input neuron with one demo, architecture of all the network layers with two subtopics Layer of neuron and multiplayer neuron. For Perceptron there will be the Description Neuron Model, Perceptron Basic Architecture and Perceptron Algorithm with one example of solved problem. For the Feed Forward, Recurrent and Self Organizing Map Networks there are the Neuron Model, Basic Architecture and Training Alogorithm.The demonstrations are included each of every topic. It also has glossary that gives sense to certain new words in the text.

In providing convenience to users, this system is developed as a stand-alone application. The system is built-up in convergence of artificial neural network concepts and the latest available Macromedia Flash MX and it runs on the requirement of Windows 98 and above operating system.

This system will describe the demonstration mechanism that is controlled by the hypermedia elements and that attempts to assist the student to learn the material and concepts presented with their participation.

ACKNOWLEDGEMENT

ACKNOWLEDGEMENTS

With The Name of Almighty, I praise for His favour to me in completing this thesis. My gratefulness to the Almighty for good health and compelling strength through out the system development.

First and foremost I would like to thank my supervisor, **Ms Manggalam Sankupellay** for her invaluable spiritual, academic advice and for her subtle direction of my efforts throughout the preparation of this thesis. I am deeply grateful to my moderator, **Dr Rukaini Hj Abdullah** for whose stimulating suggestions and motivating comments encouraged me to produce a better system.

I extend my sincere gratitude and appreciation to **Mr Nagulu** and **Mr Haza** who has been patiently provided me the technical guidance in this report compilation. Also special thanks to my friend, **A Deenalatha** for her invaluable assistance and undivided support throughout the project.

Last but not least, I wish to express my appreciation to my beloved parents, **Mr.Mohd Kamal** and **Mrs. Norani**, my sisters, **Nurul Ain, Nur Adiba,Umi Nadia,Nur Nabila** and **Nurul Hana,** and my best friends, **Nora Aini Sulong,Deepa Chandren, Kumutha, Jiznie Arnie** and **Ellyda**.For my lovely boyfriend **Mal** for the showering love and endless support given to me. Many more persons participated in various ways to ensure this thesis succeeded and I am thankful to them all. May God Bless each and every one of you.

TABLE OF CONTENT

CONTENTS

CHAPTER PAGE

LIST OF FIGURES

LIST OF FIGURES

Figure No. **Page No.**

LIST OF TABLES

LIST OF TABLE

LIST OF TABLES

Table No. **Page No.**

CHAPTER 1: INTRODUCTION

CHAPTER 1

1.0 Project Introduction

1.1 Problem Definitions

An Interactive Learning Package for Artificial Neural Network (ANN) is the one of the learning packages that aid students especially in ANN. Nowadays, hard for them to find the suitable ANN learning software or package that compatible with their syllabuses in the class. Mostly it only gives the basic description and demonstration with a little explanation .It will make the students feasible to have more understanding in ANN. Besides, ANN is also one of the new subjects in computer science field. Thus hard for them to find the related reference books from beginner, intermediate and expert level in the library. And the other connection is taking the articles from other international institutes through the web. The possibilities are maybe the in formations are too detail and complex, the syllabuses are not suitable, no illustration and graphical figures that can attract student to learn. And in the class the student can become bored if they cannot clearly understand the lecturer's explanation since it too fast, the unpleasant class situation and disturbed their attention to learn until they cannot paying attention while in the lecture hall and the incomplete lecture notes etc.

1.2 Project Overview

This is multimedia learning package designed for the students who is taking ANN subjects in their course. It assists them in learning process and designed to encourage them in the interactive approaches.Therefore,multimedia technologies that had been choose in this package consist all multimedia element likes animation,text,graphic and audio to generate more successful and exciting learning situations.

The important factors to develop this learning package in the education system are because currently many students are having the conventional way to learn ANN with lecturer supervised. The weakness is sometimes they cannot ensure all the students able to acquire what they want to explain so it is convenient and less impressive. It also becomes one of the government initiatives to uprising the education system in the method to set up the usage of the information technologies components in this field.Basiclly, the student already have the fundamental knowledge in information technology, they have been learn in the school and home previously so easier for them to apply this package as one of their way to learn ANN. Besides, the lecturer will also get the advantage when they have the aided software to make their explanation besides demonstration more understandable in the lecture hall. With the existing of this package, difficulty in searching the ANN learning package will decrease and the student becomes more understand and concerned to learn in more effective way.

1.3 Project Aims

1. To create an interactive and effective teaching in demonstration and learning tool to understand and promote ANN as an interesting subject.
2 The package will also help create positive impacts on the learning of subject often considered difficult among the students.
3. To interrelate the enormous amount of facts and reactions in this subject with the concept of mechanism to ease them to learn with doing.
4. The resulting outcome is important to encourage the students, lecturer and others to get in touch with ANN thus enhancing more understanding among them.
5. This project is also fully equipped with sufficient knowledge to guide and educate the students and others with the implementation that showed in this system in order to practice and implement what they have been learn in their daily techniques.
6. This package will be the one of the variety learning package in the market and the user will able to do the best selection.

1.4 Project Scopes

The two modules discussed have same goal and scopes. The several considerations will be making during the development to ensure the learning package specifically developed are suitable and necessary for them.

i. **Student**

Especially who are taking ANN as their major course in Artificial Intelligent. It will be designed mainly to make them more understanding from the basic level so in the demonstration modules they should not have a problem. With interesting way learning they can identify the functions and calculation for the demonstrations more effective and faster.

ii. **Lecturer**

Most of them sometimes cannot present well in the lecture hall thus with the aided package like this they will be able to give as an extra guidance preparation for them to make the student clearly obtain what they try to explain with a lot of formulas and examples of demonstrations.

1.5 Project Limitations

The Interactive Learning Package for Artificial Neural Network is divided into two learning modules: learning modules and demonstration modules. As explained in the learning module before, only five topics will be discussed and it also same in the demonstration module which includes Introduction of Artificial Neural Network, Perceptron Network, Feed Forward Network, Recurrent Network and Self Organizing Maps (SOM) .ANN is a wide topic and these limitations will provide particular concentration in the package so it will limit the understanding of the subject in detail.

But, the specific demonstration and information will be given according to the sub topic above so they can understand details all particular topics.

1.6 Project Development Schedule

A Bar chart or the other name is Gantt chart is an easy way to schedule tasks. It shows who is responsible for each activity and when the activity is scheduled to begin and end. It is essentially a chart on which bars represent each task or activity also the length of each bar represents the relative lengths of the tasks. It can be generated automatically from the database of project information using a project management tool.

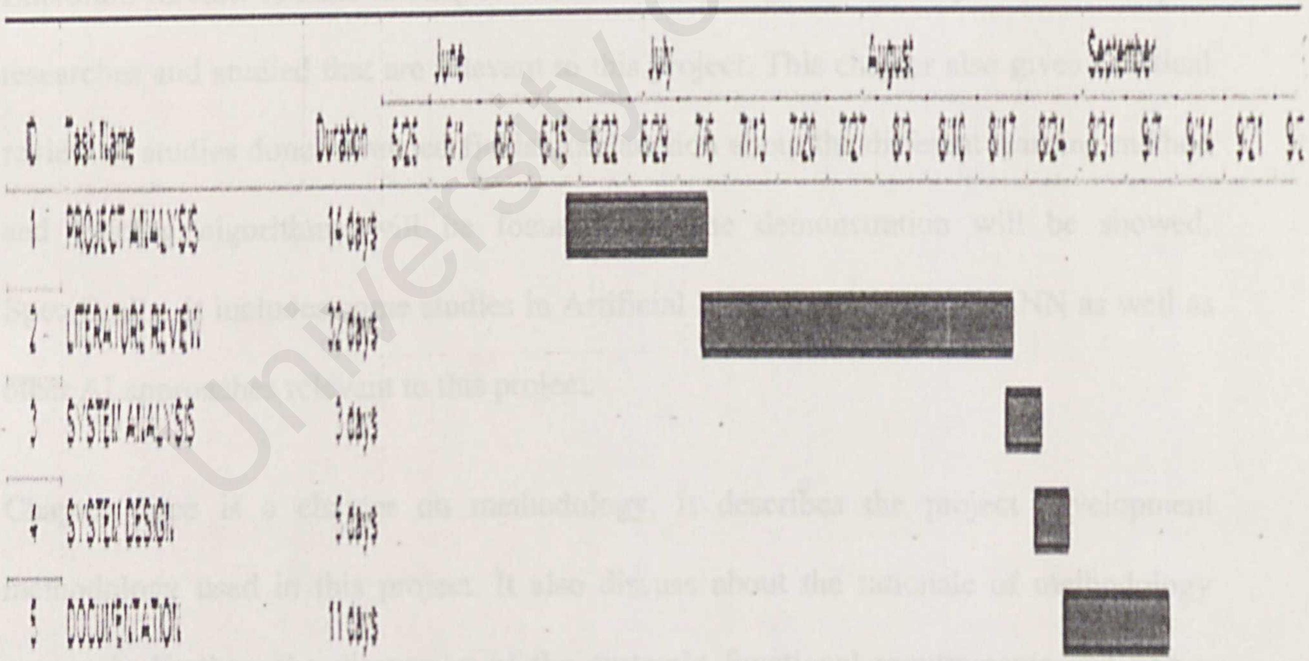


Figure 1: Bar Chart

1.5 Report Layout

The main rationale of this project is to report in full the processes and major contents, which will be included and involved in the development of the "Interactive Learning Package for Artificial Neural Network". The report consists of several chapters.

The first chapter is all about the project definition, which gives an overview of the project. The problem definition, project aims, scope and limitations, project development schedule and report layout are provided.

Literature Review is done in chapter two. This section gives brief explanation on topics researches and studied that are relevant to this project. This chapter also gives a critical review of studies done in related fields. Explanation about the different learning method and training algorithms will be focused and the demonstration will be showed. Specifically, it includes some studies in Artificial Intelligence (AI) and ANN as well as other AI approaches relevant to this project.

Chapter three is a chapter on methodology. It describes the project development methodology used in this project. It also discuss about the rationale of methodology approach. Further, the discussion of the system's functional requirements and non – functional requirements will be included. Explanation about the selection of development tool and the hardware requirements are also described.

Chapter four is on system analysis and design. It describes the steps taken to accomplish this project. The waterfall model with prototyping is used as the development model. Further, the discussion of the preliminary design of the system, which is the flow chart, context diagram, data flow system and the user interface design is included. The statement of expected outcome during the process of testing is discussed.

The last chapter is the reference section. In this section, the books and journals that is used for the research are listed.Futher, there are lists of websites that I have used to search the information that can be used for the further researches.

CHAPTER TWO: LITERATUTE REVIEW

CHAPTER 2

2.0 Literature Review

2.1 Introduction

Research and literature review were the critical tasks to be done in this part. Effort to dig deeper for information about the system is very important. From that, the overview of the system can be seen. The materials below are concluded following some effort such as studying and looking for the previous thesis on artificial neural network, artificial neural network references related to learning package development, journals regarding artificial neural network and the collection of the articles from the international higher institution that get it through the internet. The most important point here is to have an understanding about what is Artificial Intelligence (AI) carried out. In this project, ANN as the one of the AI approaches had been used and applied.

2.2 The Interactive Learning Package

As the world moves increasingly towards a computer capable society it is more important than ever that students learn how to enrich their working practices using the latest technology. It will be allowing the students to learn at their own speed about the subject. The interactive learning system actually is the teaching basic for any subject concepts with the multimedia elements that will include all the hypermedia elements to encourage the students learn in the more effective and interactive ways. The system will include interactive activities which will provide immediate feedback to the students on their progress. It is a test bed to show that how the iearning system and the hypermedia elements may be combined successfully to form a framework useful for any educational material.

2.3 The Hypermedia elements

The hypermedia systems are now de facto standard on personal computers and increasing number of interactive learning package .The hypermedia system provides a graphical environment in which the student can practice and interact direct with subjects without the lecturer supervised. The hypermedia is used enhance the pedagogy and for incorporating different strategy into the courseware design. The hypermedia components are accessed through the graphical user interface, allowing them to be developed independently. Incorporating multimedia systems into learning package allows the user to easily navigate through the instructional units for every demonstration. Additional use of color, auditory and visual clues is used to highlight important aspects. The hypermedia also consists of the following: graphical objects with texts and audio, digital video images with text and audio, hypertext, to explain any learning with successfully. This "modality" effect improves working memory and enhances performance of the system.The delivery platforms for any learning package are currently on a personal computer, this supports limited input from multiple channels. The current channels are keyboard, mouse, microphone and video. The future implementation would be over a network and the Internet.

Advantages of using multimedia objects for demonstrations in learning package are including:

1. The students can be used to gain more information especially in all the demonstrations that needed student participation in the more effective ways.
2. The hypermedia elements will help the students like to learn more because the environments will give the encouragement for them.
3. The use of multimedia objects during learning session will enhance all aspects and can significantly increase the effectiveness of using the learning package as an instructional tool in the class or at home.

2.4 Comparison of existing system

2.4.1 System Titles: Kimia SPM Multimedia by MindSpark Interactive

[Online] http://www.mindspark.com.my/

2.4.1.1 System Descriptions

This is a multimedia learning package for students who are taking Kimia in SPM [1]. It provides a graphical environment in which the student can practice conceptual laboratory experiment and interact direct with e-tutorial and e-notes. It combines intelligent tutoring with multimedia systems, for facilitating student to 'learn-by-doing' the experiment provided and it also allows the user to easily navigate through the instructional units. It also features:

-attractive and easy to use interface

-virtual lab let the student conduct chemistry experiment using computer

-67 interactive experiments will help the student to understand the experiment steps effectively.

-291 interesting graphics and 27 animations to describe complex chemical reactions

-17 video records on the real world chemical reactions

- human voices guide the student to navigate the whole system.

-tutorials are provided with text, graphics, and animation, and video, 3D model, interactive experiment and human voices narrations.

-little content has been planned and written carefully by experience teacher

-chemistry dictionary helps the student to understand the key words.

2.4.1.2 Advantages

- It has the ability of incorporating multimedia objects into learning package components. It also emphasis the functional contents of chemistry experiment in terms of simulation processes, functionality and multimedia objects.
- It supports classroom teaching and learning from textbooks. This could help to extend the knowledge by revision of tutorials and application new information.
- The use of multimedia objects during tutorial presentation will enhance all aspects of the tutorial system and the pedagogy contents of the syllabus. It also combining multimedia objects excellently with more constructive tools can significantly increase the effectiveness of using this web site as an instructional tool.

2.4.2.1 System Titles: Episodic Learner Model the Adaptive Remote Tutor By Elm Research

[Online]

http://www.psychologie.unitrier.de:8000/projects/ELM/elmart

2.4.2.1 System Description

This is a multimedia model for the episodic learner [2]. With ELM-ART are trying to make the benefits of our learning environment ELM-PE available via World Wide Web. This reduce a potential users requirement in hardware and software and lets them choose the time and place of their training more freely. It also features the Lisp course and the user can see the system work by looking at this course. All they need is a HTML browser. Capable of dealing with forms, frames and Java scripts. They will complete course of the six lessons with all the text book material, experiments, exercise and mini evaluator at their hand. The session can be contributed later with the system remembering their former interactions.

2.5 Artificial Neural Network

Artificial Neural Network is a system loosely modeled on the human brain. Artificial Neural Networks or Neural Networks can also be loosely defined as large sets of interconnected simple units, which execute in parallel to perform a common global task. These units usually undergo a learning process, which automatically updates network parameters in response to a possibly evolving input environment. The field goes by many names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons. Each neuron is linked to

certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output appropriate results.

2.6 Artificial Neural Networks

Artificial Neural Network is a system loosely modeled on the human brain [3]. Artificial Neural Networks or Neural Networks can also be loosely defined as large sets of interconnected simple units, which execute in parallel to perform a common global task. These units usually undergo a learning process, which automatically updates network parameters in response to a possibly evolving input environment. The field goes by many names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons. Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output appropriate results.

2.7 Introduction of ANN

2.7.1 Learning and Connections

A biological neural network is a collection of interconnected neurons, brain cells. Although biological neural networks exist in many places in the human body and in the bodies of other animals, the most interesting biological neural network is the human brain. The brain consists of billions of highly interconnected brain cells known as neurons. Individually these neurons are very simple and can accomplish very little. However, when connected together, they form a powerful reasoning system. Each neuron has three basic parts: dendrites, a soma, and an axon. A neuron receives many input signals through its many dendrites. The signals then enter the soma of the neuron and are all summed together. If the summed signal is strong enough, the neuron fires; that is, it sends a signal out through its axon. The many neurons in the brain are connected together, axon to dendrite, to form a network of neurons or a biological neural network. One important thing to remember is that the signal on the axon of a neuron is not going to be exactly the same as the signal on a dendrite of the neuron it is connected to. The signal is changed, generally strengthened or weakened, by the chemical process that transmits the signal across the synaptic gap. The figure below shows a simplified biological neuron and the relationship of its four components.

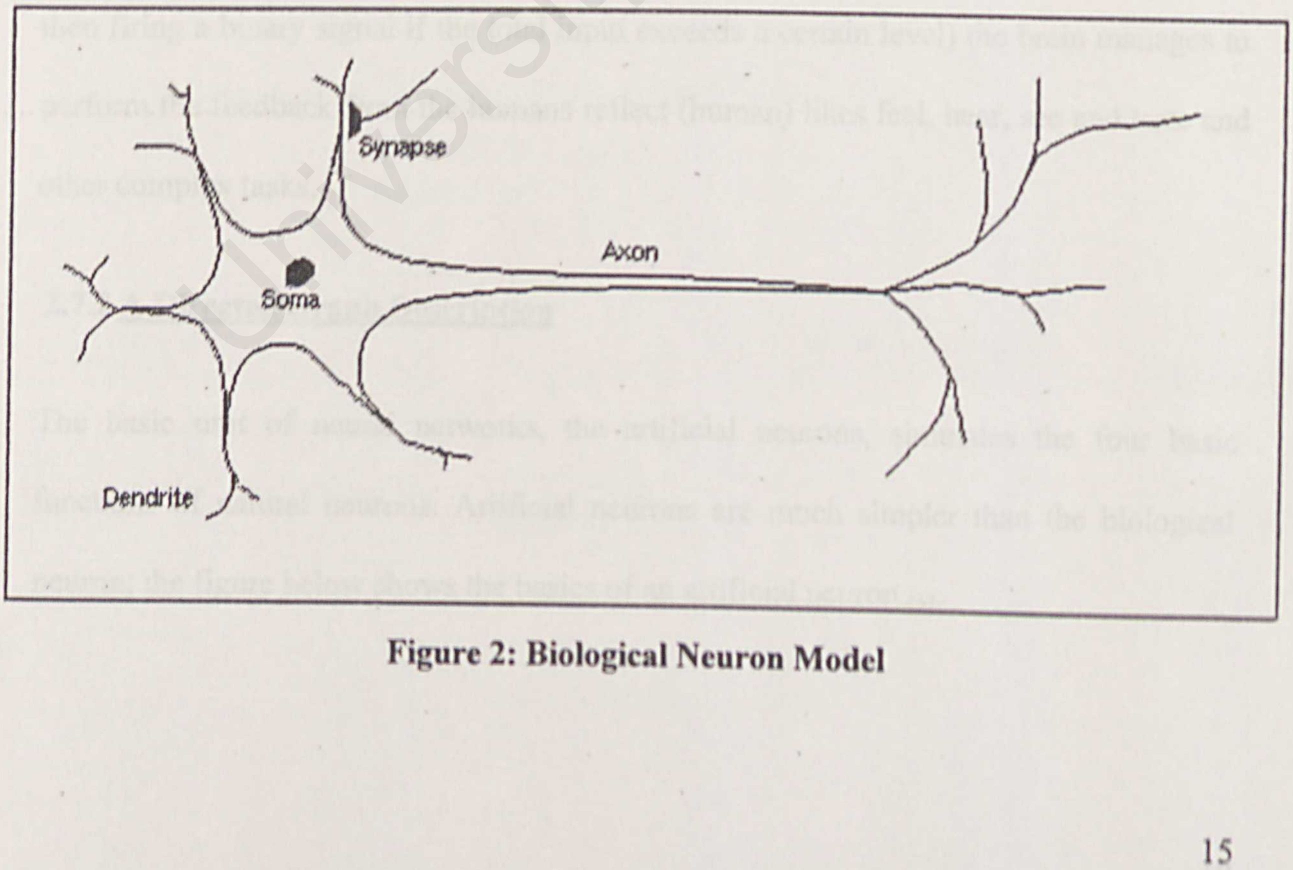


Figure 2: Biological Neuron Model

2.7.1.1 Demonstration 1: Simultaneous Process between Neuron

This demo shows how two neurons receive electrochemical inputs (raw inputs from the brain are transfer into the information for the human feedback) from other neurons at the dendrites. Each important part: axon, dendrites and cell body will state. The process transmission begins, if the sum of the electrical inputs is sufficiently powerful to activate the neuron, it will transmit an electrochemical signal along the axon, and passes this signal to the other neurons whose dendrites are attached at any of the axon terminals. These attached neurons may then fire. It is important to note that a neuron fires only if the total signal received at the cell body exceeds a certain level. The neuron either fires or it doesn't, there aren't different grades of firing. So, the entire brain is composed of these interconnected electro-chemical transmitting neurons. From a very large number of extremely simple processing units (each performing a weighted sum of its inputs, and then firing a binary signal if the total input exceeds a certain level) the brain manages to perform the feedback from the humans reflect (human) likes feel, hear, see and taste and other complex tasks.

2.7.2 A Directed Graph Description

The basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Artificial neurons are much simpler than the biological neuron; the figure below shows the basics of an artificial neuron [5].

1. The nodes of the graph are called processing elements.

2. The links of the graph are called connections. Each connection functions as an instantaneous unidirectional signal-conduction path.

3. Each processing element can receive any number of incoming connections

4. Each processing element can have any number of outgoing connections but the signals in all of these must be the same.

5. Processing elements can have local memory

6. Each processing element processes a transfer function which can use (and modify) local memory, can use input signals, and which produces the processing element's output signal.

7. The only inputs allowed to the transfer function are the values stored in the processing element's local memory and the current values of the input signals in the connections received by the processing element.

2.7.2.1 Demonstration 2: Directed Graph Description

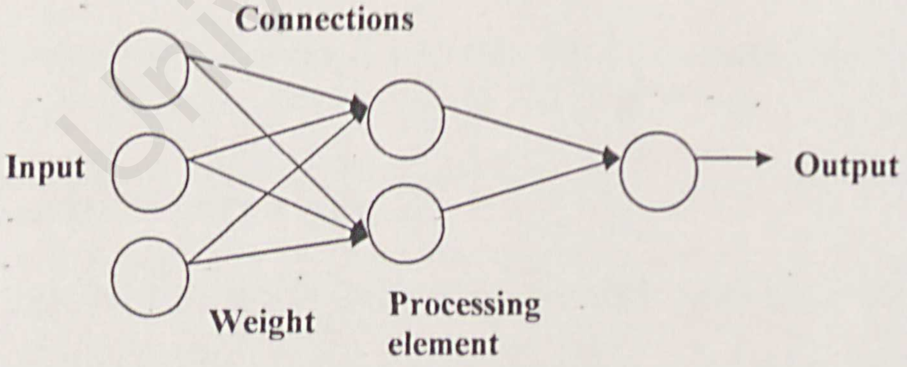


Figure3: Directed Graph

This demo shows the pictures on how output is get through the graph. Each of the inputs are multiplied by a weight at the connection link, then the multiplied weight with the input will be process in the processing element. In the simplest case, these products are simply summed, fed through a transfer function to generate a desired output, and then come out the result.

2.8 Single Input Neuron

A single-input neuron is shown in Figure 2.6.1.5.The scalar input, p is multiplied by the scalar weight, w to form one of the terms that is sent to the summer. The other input, is multiplied by a bias, b and then passed to he summer. The summer output,n often referred to as the net input, goes into a transfer function ,f which produces the scalar neuron output, a. Some designer use the term activation function rather than transfer function and offset rather than bias. If this simple model back to the biological neuron that discussed in before, the weight corresponds to the strength of a synapse, the cell body is represented by the summation and the transfer function, and the neuron output represents the signal on the axon.

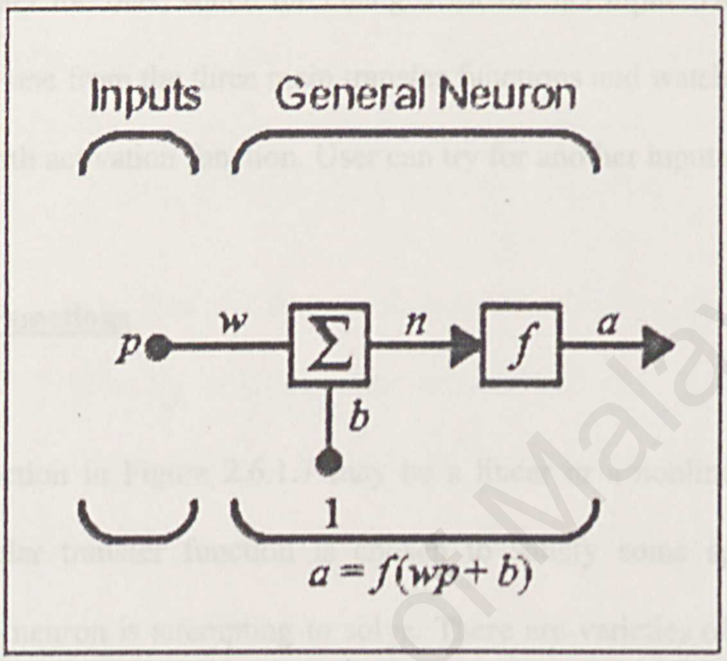


Figure 4: Single Input Neuron

Formula = f (Wp +b)

The neuron output is calculated as **a = f (Wp + b)** if for instances **W= 3, p = 2 and b = -1.5.**

Then **a = f (3(2) + (-1.5)) = f (4.5)**

The actual output depends on the particular transfer function that is chosen. The bias is much like a weight, except that it has a constant input of 1.If the bias unneeded in a particular neuron, it can be omitted.

2.8.1 Demonstration 3: Single Input Neuron Calculation

This demo shows how the calculation for the single neuron with selected input and transfer function. The user must follow the instruction: enter the input and then the weight. After enter the bias, watch the changes for the net input. The transfer function must be selected one from the three main transfer functions and watch the changes of the desired output with activation function. User can try for another input randomly.

2.8.2 Transfer Functions

The transfer function in Figure 2.6.1.7 may be a linear or a nonlinear function of net input. A particular transfer function is chosen to satisfy some specification of the problem that the neuron is attempting to solve. There are varieties of transfer functions but here only three of the most commonly used functions are discussed then.

- **Hard Limit Transfer Function**

The ***hard limit transfer function***, hardlim shown on the left side of Figure 2.6.1.7, sets the output of the neuron to 0 if the function argument is less than 0, or 1 if its argument is greater than or equal to 0.

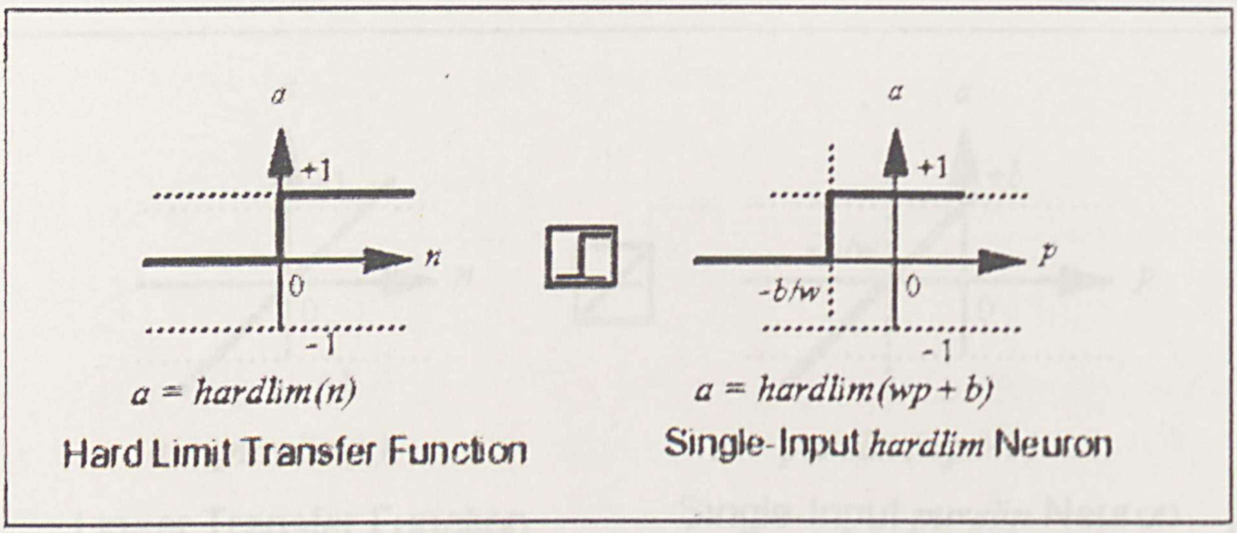


Figure 5: Hard Limit Transfer Function

The graph on the right side of hard limit transfer function figure, illustrates the input/output characteristic of a single-input neuron that uses a hard limit transfer function. The effect of the weight and the bias can be seen. Note that an icon for the hard limit transfer function is shown between the two figures. Such icons will replace the general in network diagrams to show the particular transfer function that is being used.

- **Linear Transfer Function**

The output of a ***linear transfer function*** is equal to its input as illustrated in next figure. The output (a) versus input (p) characteristic of a single-input linear neuron with a bias is shown on the right of Figure 5.

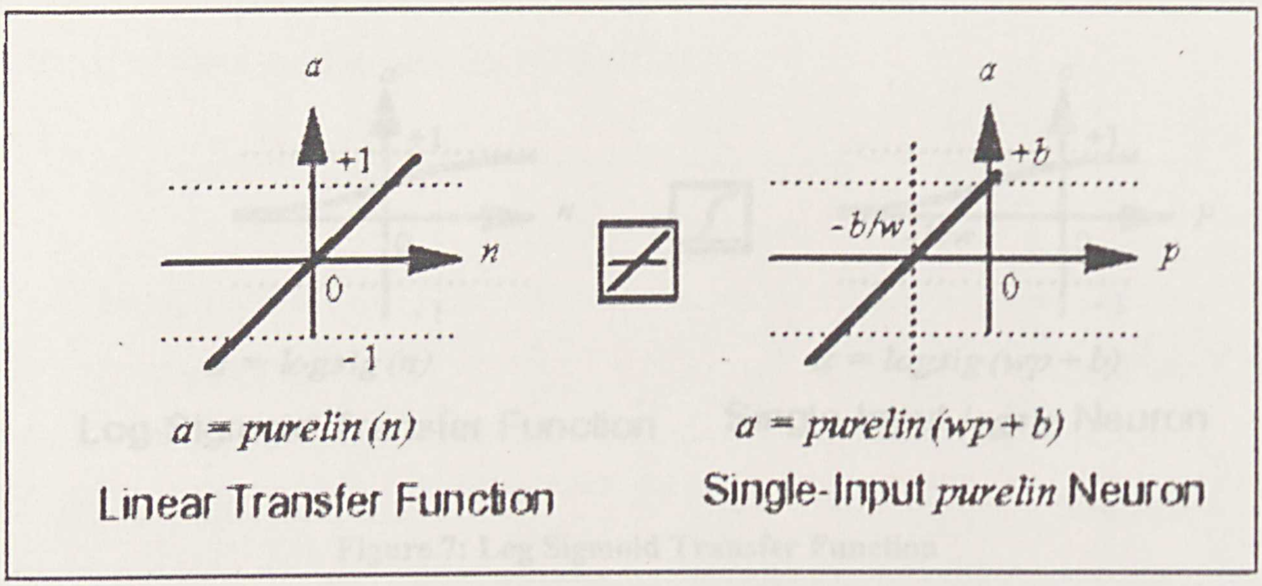


Figure 6: Purelin Transfer Function

- **Log Sigmoid Transfer Function**

The *log-sigmoid transfer function* is shown in Figure 2.1.6.9.

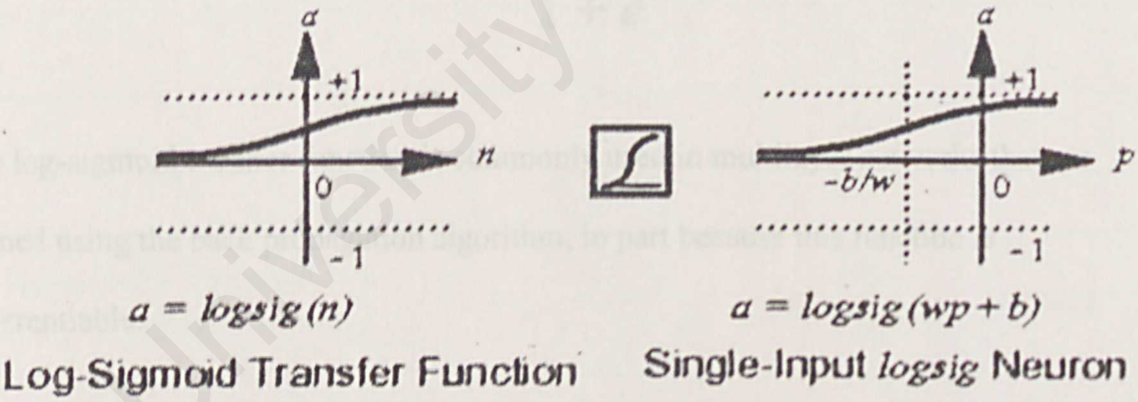
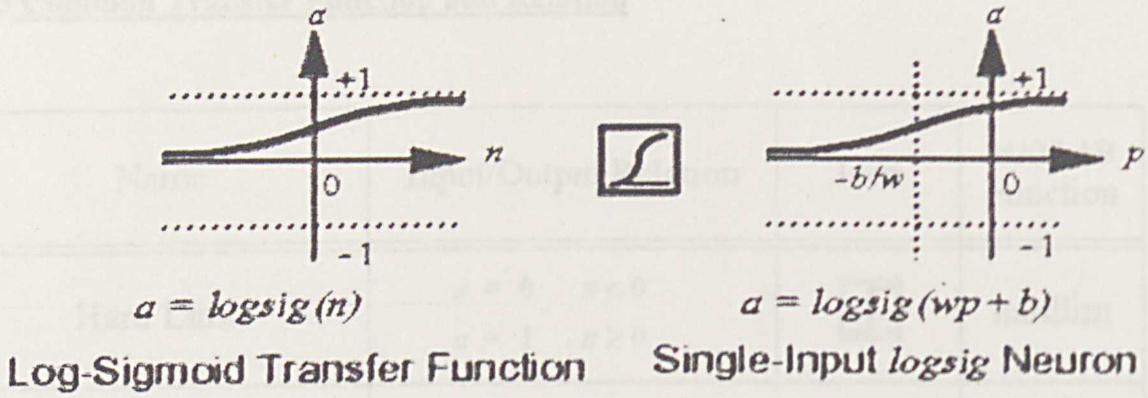


Figure 7: Log Sigmoid Transfer Function

This transfer function takes the input (which may have any value between plus and minus infinity) and squashes the output into the range 0 to 1, according to the expression:



Log-Sigmoid Transfer Function

Single-Input *logsig* Neuron

Figure 7: Log Sigmoid Transfer Function

This transfer function takes the input (which may have any value between plus and minus infinity) and squashes the output into the range 0 to 1, according to the expression:

$$a = \frac{1}{1+e^{-n}}.$$

The log-sigmoid transfer function is commonly used in multilayer networks that are trained using the back propagation algorithm, in part because this function is differentiable.

2.8.3 Common Transfer Function and Relation

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons	C	compet

Figure 8: Transfer Functions

2.8.4 Multiple Input Neurons

Typically, a neuron has more than one input. A neuron with p1, p2 ... pR...an input is shown. The individual inputs,p are each weighted by corresponding elements w1,1, w1,2,...w1,R of the weight matrix,R.The neuron has a bias,b which is summed with the weighted inputs to form the net input, n :

n = w1, 1 * p1 + w1, 2 * p2..... + w1, R * pR

This expression can be written in matrix form: **n = Wp + b**

Where the matrix, W for the single neuron case has only one row. Now the neuron output can be written as:

$$a = f(\mathbf{W}\mathbf{p} + b) .$$

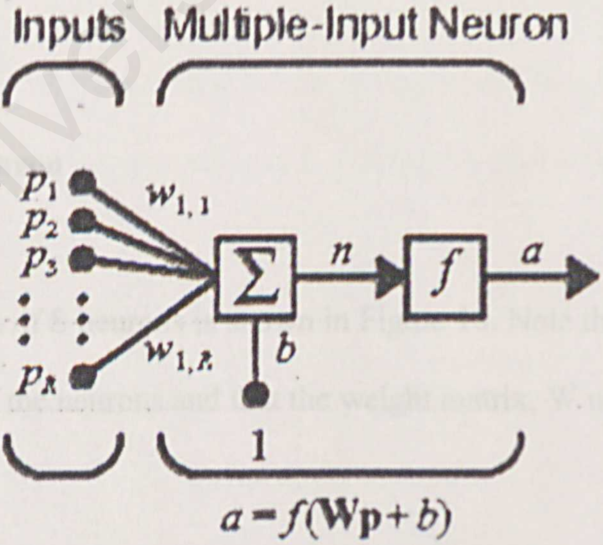


Figure 9: Multiple Input Neurons

2.8.5 Demonstration 4

Multiple Input Neurons

This demo shows how the calculation for the single neuron with selected input and transfer function. The user must follow the instruction: enter the input and then the weight. After enter the bias, watch the changes for the net input. The transfer function must be selected one from the three main transfer functions and watch the changes of the desired output with activation function. User can try for another input randomly.

2.9 Network Architectures

Commonly one neuron, even with many inputs, may not be sufficient. It might need five or ten, operating in parallel, in what it will call a layer. This concept of a layer is discussed below.

- **A Layer of Neuron**

A single-*layer* network of S neurons is shown in Figure 10. Note that each of the inputs is connected to each of the neurons and that the weight matrix, W now has S rows.

Multiple Input Neurons

This demo shows how the calculation for the single neuron with selected input and transfer function. The user must follow the instruction: enter the input and then the weight. After enter the bias, watch the changes for the net input. The transfer function must be selected one from the three main transfer functions and watch the changes of the desired output with activation function. User can try for another input randomly.

2.9 Network Architectures

Commonly one neuron, even with many inputs, may not be sufficient. It might need five or ten, operating in parallel, in what it will call a layer. This concept of a layer is discussed below.

- **A Layer of Neuron**

A single-*layer* network of S neurons is shown in Figure 10. Note that each of the inputs is connected to each of the neurons and that the weight matrix, W now has S rows.

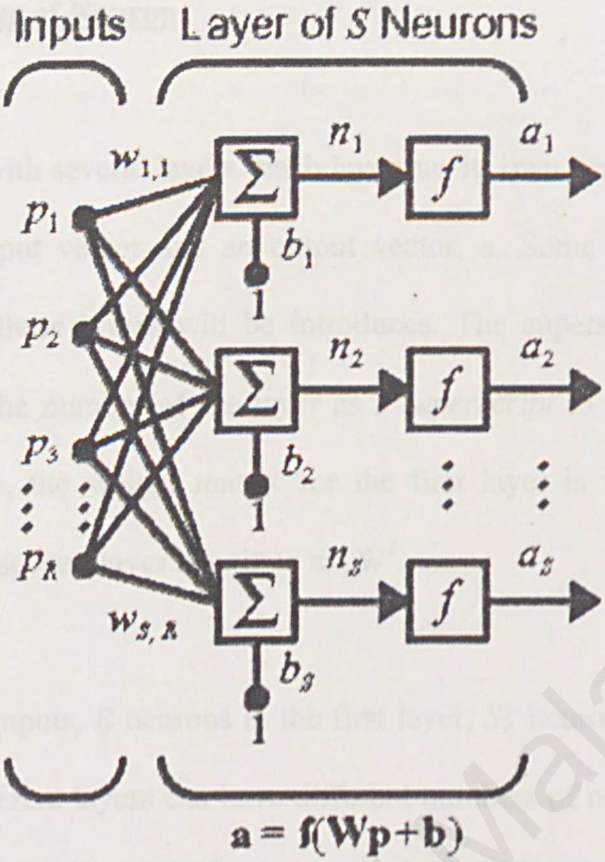


Figure 10: Layer of S Neurons

The layer includes the weight matrix, the summers, the bias vector, b, the transfer function, boxes and the output vector, a .Each element of the input vector, p is connected to each neuron through the weight matrix, W. Each neuron has a bias, bi, a summer, a transfer function and an output,a_i , Taken together, the outputs form the output vector, a. It is common for the number of inputs to a layer to be different from the number of neurons (i.e.R ≠S).

- **Multiple Layer of Neuron**

Consider a network with several layers. Each layer has its own weight matrix, W its own bias vector, b net input vector and an output vector, a. Some additional notation to distinguish between these layers will be introduces. The superscripts will be used to identify the layers. The number of the layer as a *superscript* to the names for each of these variables. Thus, the weight matrix for the first layer is written,W^1 as and the weight matrix for the second layer is written as ,W^2.

As shown, there are inputs, *R* neurons in the first layer, *S*1 neurons in the second layer, *S*2 etc. As noted, different layers can have different numbers of neurons. The outputs of layers one and two are the inputs for layers two and three. Thus layer 2 can be viewed as a one-layer network with *R* = *S*1, inputs, *S*1 = S2 neurons, and a weight matrix. The input to layer 2 is, **a**1 and the output is **a**2.

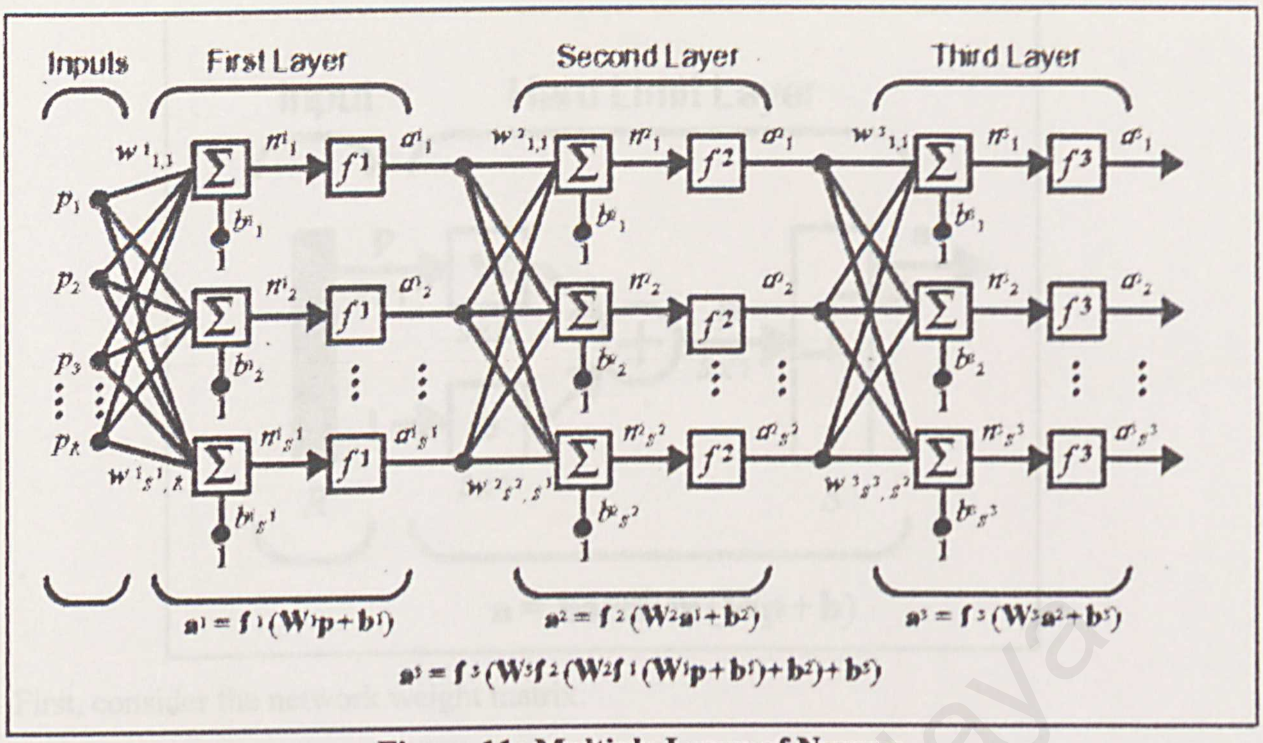
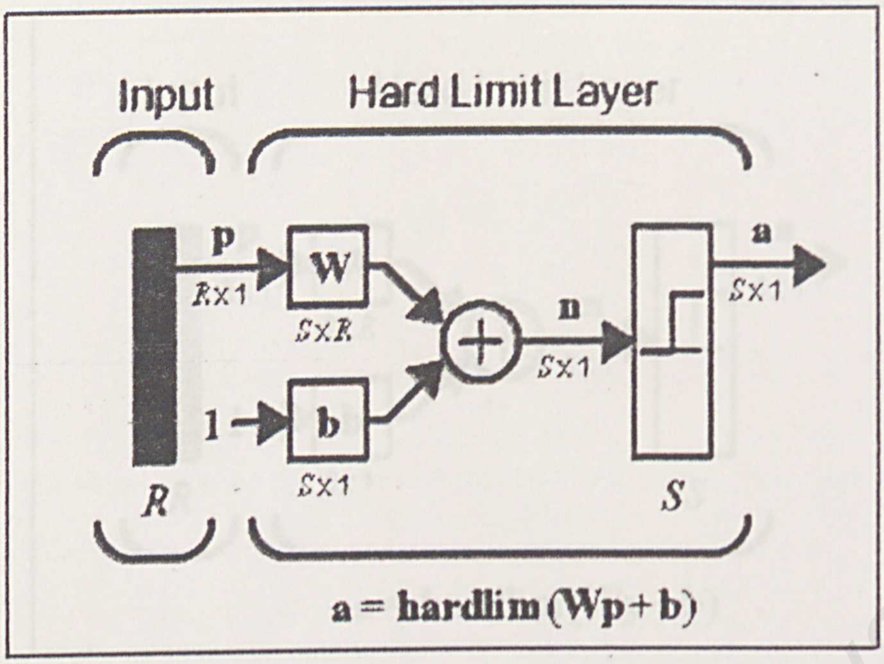


Figure 11: Multiple Layer of Neuron

The output of layer 1 and 2 are the inputs for layer 2 and 3.The input to layer 2 is a a1 and the output is a2.The multilayer network are more powerful than the single layer .For instance, a second layer network having a sigmoid first layer and a linear second layer can be trained to approximate most functions arbitrarily well and the single layer network cannot do this. A layer whose output is the network output called output layer and the other layer are called hidden layers. The network had shown an output layer (layer R) and two hidden layer (layer 1 and 2).



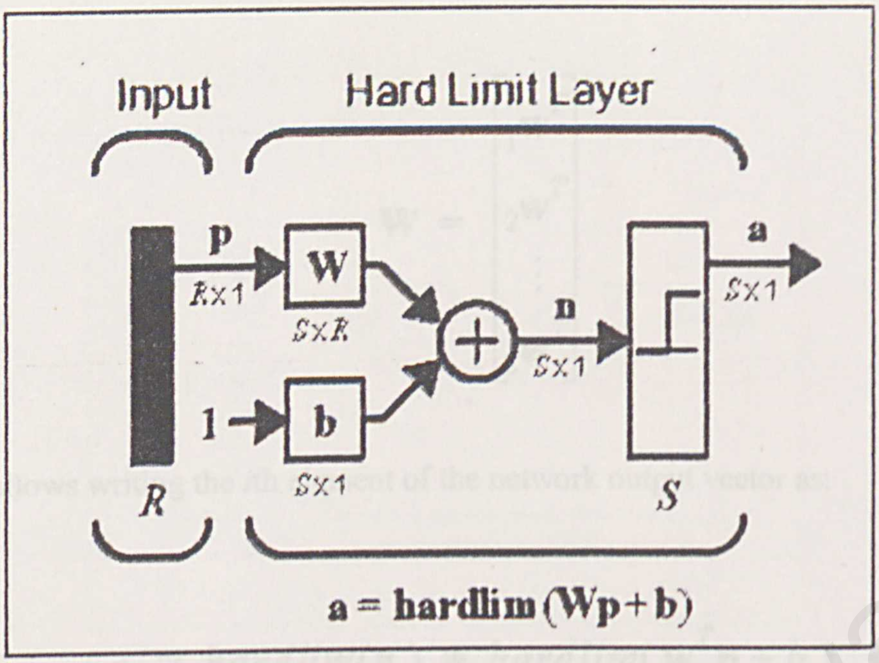
First, consider the network weight matrix:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

The vector composed of the elements will be defined of the *i*th row of vector **W.**

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

So now the weight matrix can do the partition:



First, consider the network weight matrix:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

The vector composed of the elements will be defined of the *i*th row of vector **W**.

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

So now the weight matrix can do the partition:

$$\mathbf{W} = \begin{bmatrix} {}_1\mathbf{w}^T \\ {}_2\mathbf{w}^T \\ \vdots \\ {}_S\mathbf{w}^T \end{bmatrix}$$

And this allows writing the *i*th element of the network output vector as:

$$a_i = hardlim(n_i) = hardlim({}_i\mathbf{w}^T\mathbf{p} + b_i)$$

Recall that the hardlim transfer function (shown at left) is defined as:

$$a = hardlim(n) = \begin{cases} 1 & if \;\; n \geq 0 \\ 0 & otherwise \end{cases}$$

Therefore, if the inner product of the *i*th row of the weight matrix with the input vector is greater than or equal to - *bi*, the output will be 1, otherwise the output will be 0. Thus each neuron in the network divides the input space into two regions. It is useful to investigate the boundaries between these regions.

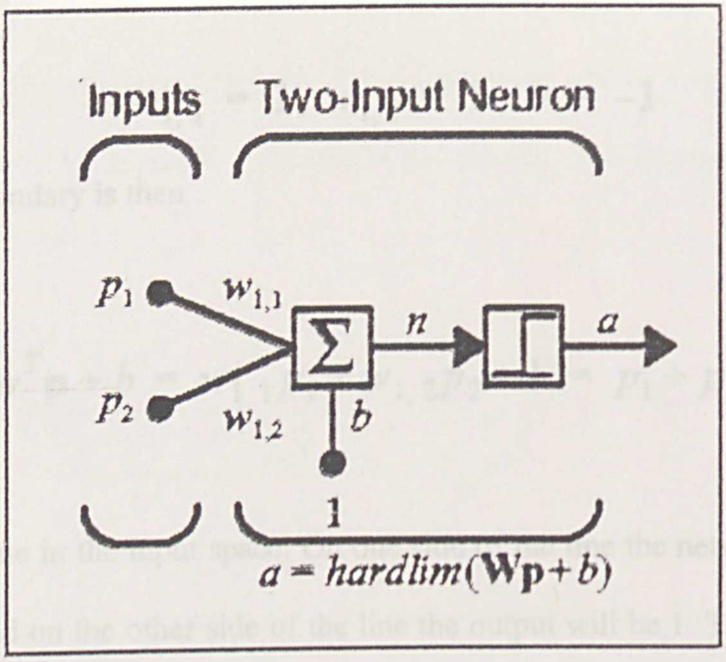


Figure 12: Perceptron Single Input Neuron

The output of this network is determined by:

$$a = hardlim(n) = hardlim(\mathbf{W}\mathbf{p} + b)$$

$$= hardlim({}_1\mathbf{w}^T\mathbf{p} + b) = hardlim(w_{1,1}p_1 + w_{1,2}p_2 + b)$$

2.10.3 Decision Boundary

The *decision boundary* is determined by the input vectors for which the net input, n is zero:

$$n = {}_1\mathbf{w}^T\mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + b = 0$$

To make the example more concrete, let's assign the following values for the weights and bias:

$$w_{1,1} = 1,\ w_{1,2} = 1,\ b = -1$$

The decision boundary is then

$$n = {}_1\mathbf{w}^T\mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + b = p_1 + p_2 - 1 = 0$$

This defines a line in the input space. On one side of the line the network output will be 0; on the line and on the other side of the line the output will be 1. To draw the line, we can find the points where it intersects the p1 and p2 axes. To find p2 the intercept set p1 = 0.

$$p_2 = -\frac{b}{w_{1,2}} = -\frac{-1}{1} = 1 \quad \text{if } p_1 = 0$$

To find the p1 intercept, set p2 = 0:

$$p_1 = -\frac{b}{w_{1,1}} = -\frac{-1}{1} = 1 \quad \text{if } p_2 = 0$$

To find out which side of the boundary corresponds to an output of 1, need to test one point. For the input p = $\begin{bmatrix} 2 & 0 \end{bmatrix}^T$ the network output will be :

$$a = hardlim({}_1\mathbf{w}^T\mathbf{p} + b) = hardlim\left(\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \end{bmatrix} - 1\right) = 1$$

Therefore, the network output will be 1 for the region above and to the right of the decision boundary. This region is indicated by the shaded area above.

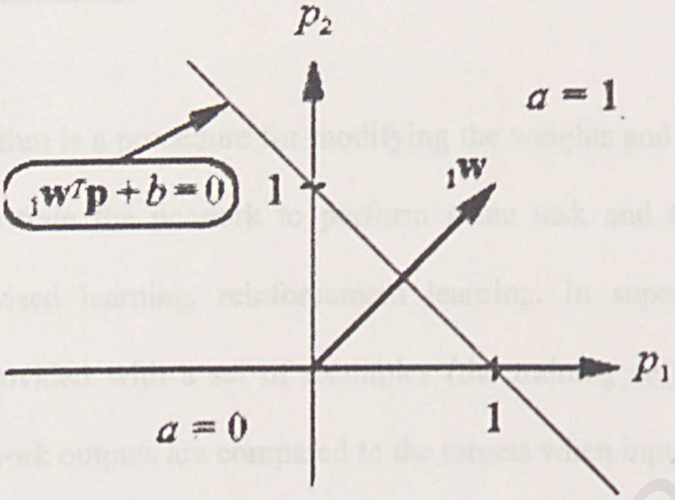


Figure 13: Decision Boundary for Two-Input Perceptron

2.10.4 Multiple-Neuron Perceptron

Note that for perceptrons with multiple neurons, as in Figure 11, there will be one decision boundary for each neuron. The decision boundary for neuron i will be defined by :

$$ {}_i\mathbf{w}^T\mathbf{p} + b_i = 0 $$

A single-neuron perceptron can classify input vectors into two categories, since its output can be either 0 or 1. A multiple-neuron perceptron can classify inputs into many categories. Each category is represented by a different output vector. Since each element of the output vector can be either 0 or 1, there are a total of 2^S possible categories, where S is the number of neurons.

2.11 Perceptron Learning Rules

2.11.1 Learning rule

The training algorithm is a procedure for modifying the weights and biases of a network. The purpose is to train the network to perform some task and fall into: supervised learning, unsupervised learning, reinforcement learning. In supervised learning, the learning rule is provided with a set of examples (the training set) of proper network behavior. The network outputs are compared to the targets when input applied.

The reinforcement learning similar to supervised learning except instead of being provided with the correct output for each network input, the algorithm is only given a grade. In unsupervised learning, the weights and biases are modified in response to network inputs only. There are no target outputs available.

The learning rule is provided with a set of examples of proper network behavior:

{p1, t1}, {p2, t2},......., {pq,tq}

The pq is an input and tq is target output. The network output is compared to the target when input applied. The learning rule then adjusts the weights and biases of the network in order to move the network output closer to the target.

The initial weight vector results in a decision boundary that incorrectly classifies the vector .P1. It needs to alter the weight vector so that it points more toward, P1 so that in the future it has a better chance of classifying it correctly.

One approach would be to set ${}_1W$ equal to ${}_1P$. This is simple and would ensure that P_1 was classified properly in the future. Unfortunately, it is easy to construct a problem for which this rule cannot find a solution. The diagram to the lower left shows a problem that cannot be solved with the weight vector pointing directly at either of the two classes 1 vector. If we apply the rule ${}_1W = P$ every time one of these vectors is misclassified, the network weights will simply oscillate back and forth and will never find a solution.

Another possibility would be to add p1 to ${}_1W$. Adding p1 to ${}_1W$ would make ${}_1W$ point more in the direction of p1. Repeated presentations of p1 would cause the direction ${}_1W$ of to asymptotically approach the direction of p1 .

This rule can be stated:

$$\textbf{If } t = 1 \textbf{ and } a = 0, \textbf{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}$$

Applying this rule to our test problem results in new values for :**w** 1

$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$

The initial weight vector results in a decision boundary that incorrectly classifies the vector .P1. It needs to alter the weight vector so that it points more toward, P1 so that in the future it has a better chance of classifying it correctly.

One approach would be to set $_1W$ equal to $_1P$. This is simple and would ensure that P_1 was classified properly in the future. Unfortunately, it is easy to construct a problem for which this rule cannot find a solution. The diagram to the lower left shows a problem that cannot be solved with the weight vector pointing directly at either of the two classes 1 vector. If we apply the rule $_1W = P$ every time one of these vectors is misclassified, the network weights will simply oscillate back and forth and will never find a solution.

Another possibility would be to add p1 to $_1W$. Adding p1 to $_1W$ would make $_1W$ point more in the direction of p1. Repeated presentations of p1 would cause the direction $_1W$ of to asymptotically approach the direction of p1 .

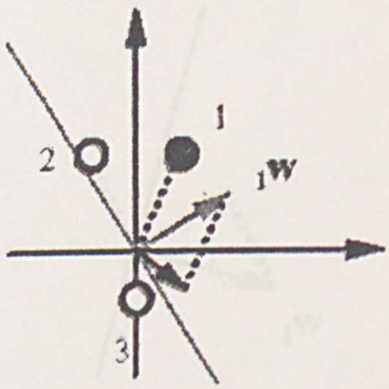
This rule can be stated:

$$\text{If } t = 1 \text{ and } a = 0, \text{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}$$

Applying this rule to our test problem results in new values for :w 1

$$ {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix} $$

This operation is illustrated in the adjacent figure.



Now move on to the next input vector and will continue making changes to the weights and cycling through the inputs until they are all classified correctly. The next input vector is P2. When it is presented to the network found:

$$a = hardlim({}_1\mathbf{w}^T\mathbf{p}_2) = hardlim\left(\begin{bmatrix}2.0 & 1.2\end{bmatrix}\begin{bmatrix}-1\\2\end{bmatrix}\right)$$

$$= hardlim(0.4) = 1 \; .$$

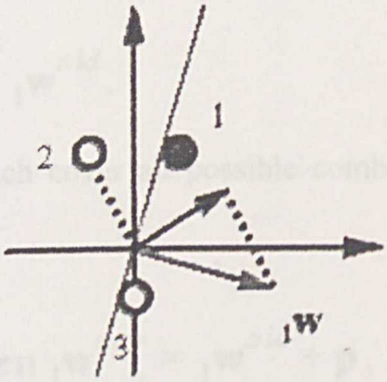
The target t2 associated with p2 is 0 and the output a is 1. A class 0 vector was misclassified as a 1.Since like to move the weight vector away from the input, it can simply change the addition in to subtraction:

$$\text{If } t = 0 \text{ and } a = 1\text{, then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}$$

If it applied to the test problem we find:

$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}_2 = \begin{bmatrix}2.0\\1.2\end{bmatrix} - \begin{bmatrix}-1\\2\end{bmatrix} = \begin{bmatrix}3.0\\-0.8\end{bmatrix}$$

which is illustrated in the adjacent figure



Now third vector p3 presented:

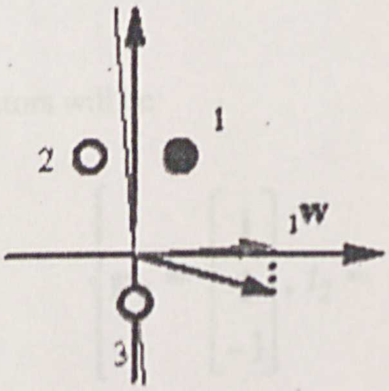
$$a = hardlim({}_1\mathbf{w}^T\mathbf{p}_3) = hardlim\left(\begin{bmatrix}3.0 & -0.8\end{bmatrix}\begin{bmatrix}0\\-1\end{bmatrix}\right)$$

$$= hardlim(0.8) = 1 .$$

The current $_1$W results in a decision boundary that misclassifies p_3. This is a situation for which we already have a rule, so $_1$W will be updated again.

$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}_3 = \begin{bmatrix}3.0\\-0.8\end{bmatrix} - \begin{bmatrix}0\\-1\end{bmatrix} = \begin{bmatrix}3.0\\0.2\end{bmatrix}$$

The diagram shows that the perceptron has finally learned to classify the three vectors properly.



If any of the input vectors presented to the neuron, it will output the correct class for that input vector. This brings to the third and final rule: if it works, do not fix it.

$$\text{If } t = a, \text{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old}.$$

Here are the three rules, which cover all possible combinations of output and target values:

$$\text{If } t = 1 \text{ and } a = 0, \text{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}.$$

$$\text{If } t = 0 \text{ and } a = 1, \text{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}.$$

$$\text{If } t = a, \text{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old}.$$

Example of the solved problem: Perceptron Rule

The *perceptron rule* can be written conveniently in matrix notation:

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{e}\mathbf{p}^T$$

and

$$\mathbf{b}^{new} = \mathbf{b}^{old} + \mathbf{e}$$

The input/output prototype vectors will be:

$$\left\{\mathbf{p}_1 = \begin{bmatrix}1\\-1\\-1\end{bmatrix}, t_1 = \begin{bmatrix}0\end{bmatrix}\right\} \quad \left\{\mathbf{p}_2 = \begin{bmatrix}1\\1\\-1\end{bmatrix}, t_2 = \begin{bmatrix}1\end{bmatrix}\right\}$$

Typically the weights and biases are initialized to small random numbers.

Suppose that here start with the initial weight matrix and bias:

$$\mathbf{W} = \begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix},\ b = 0.5$$

The first step is to apply the first input vector p1, to the network:

$$a = hardlim(\mathbf{W}\mathbf{p}_1 + b) = hardlim\left(\begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + 0.5\right)$$
$$= hardlim(2.5) = 1$$

Then we calculate the error:

$$e = t_1 - a = 0 - 1 = -1$$

The weight update is:

$$\mathbf{W}^{new} = \mathbf{W}^{old} + e\mathbf{p}^T = \begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix} + (-1)\begin{bmatrix} 1 & -1 & -1 \end{bmatrix}$$
$$= \begin{bmatrix} -0.5 & 0 & 0.5 \end{bmatrix}.$$

The bias update is

$$b^{new} = b^{old} + e = 0.5 + (-1) = -0.5$$

This completes the first iteration.

The second iteration of the perceptron rule is:

$$a = hardlim\,(\mathbf{W}\mathbf{p}_2 + b) = hardlim\left(\begin{bmatrix}-0.5 & 0 & 0.5\end{bmatrix}\begin{bmatrix}1\\1\\-1\end{bmatrix} + (-0.5)\right)$$

$$= hardlim\,(-0.5) = 0$$

$$e = t_2 - a = 1 - 0 = 1$$

$$\mathbf{W}^{new} = \mathbf{W}^{old} + e\mathbf{p}^T = \begin{bmatrix}-0.5 & 0 & 0.5\end{bmatrix} + 1\begin{bmatrix}1 & 1 & -1\end{bmatrix} = \begin{bmatrix}0.5 & 1 & -0.5\end{bmatrix}$$

$$b^{new} = b^{old} + e = -0.5 + 1 = 0.5$$

The third iteration begins again with the first input vector:

$$a = hardlim\,(\mathbf{W}\mathbf{p}_1 + b) = hardlim\left(\begin{bmatrix}0.5 & 1 & -0.5\end{bmatrix}\begin{bmatrix}1\\-1\\-1\end{bmatrix} + 0.5\right)$$

$$= hardlim\,(0.5) = 1$$

$$e = t_1 - a = 0 - 1 = -1$$

$$\mathbf{W}^{new} = \mathbf{W}^{old} + e\mathbf{p}^T = \begin{bmatrix}0.5 & 1 & -0.5\end{bmatrix} + (-1)\begin{bmatrix}1 & -1 & -1\end{bmatrix}$$
$$= \begin{bmatrix}-0.5 & 2 & 0.5\end{bmatrix}$$

$$b^{new} = b^{old} + e = 0.5 + (-1) = -0.5$$

If the iterations continued the both input vectors that found will now be correctly classified. The algorithm has converged to a solution.

2.12 Feed Forward Network

2.12.1 Introduction of Feed Forward

Feed forward neural networks (FNNs) are composed of layers of neurons, in which the input layer of neurons are connected to the output layer of neurons. The training process of FNN is undertaken by changing the weights such that a desired input-output relationship is realized.

Advantages

-the fastest models to execute

-the universal function approximators

Disadvantages:

-that no fast and reliable algorithm

-can be extremely slow to train.

-if rapid execution rates are required, multilayer

-Feed-forward networks should be chosen

2.12.2 Feed Forward Neuron Model

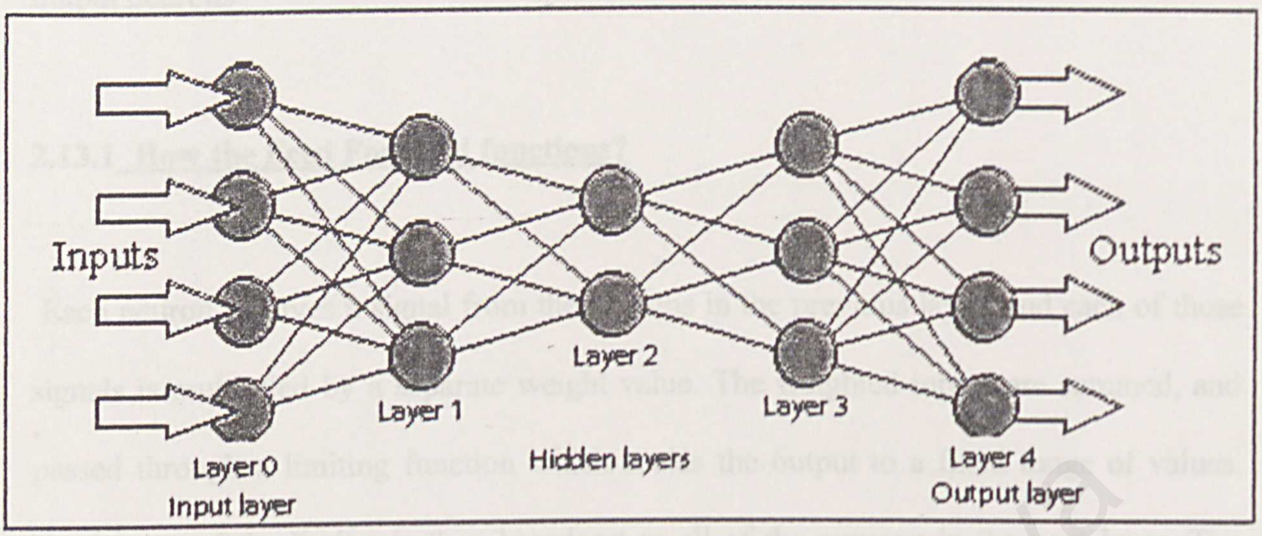


Figure 14: Feed Forward Neuron Model

2.13 Feed Forward Basic Architecture

It usually consists of 3 to 4 layers .The first and last layers are the input and output layers, and there are usually one or more hidden layers in between the other layers. The minimum of three layers (one hidden layers) is required to solve complex problems. The term feed-forward means that the information is only allowed to "travel" in one direction and means that the output of one layer becomes the input of the next layer, and so forward. In order for this to occur, each layer is fully connected to next layer (each neuron is connected by a weight to a neuron in the next layer).A multilayer feed-forward network is also often called a multi-layer perceptron.The input layer does have a part in calculation, rather than just receiving the inputs. The raw data is computed, and activation functions are applied in all layers and process occurs until the data reaches the output neurons

2.13.1 How the Feed Forward functions?

Each neuron receives a signal from the neurons in the previous layer, and each of those signals is multiplied by a separate weight value. The weighted inputs are summed, and passed through a limiting function which scales the output to a fixed range of values. The output of the limiter is then broadcast to all of the neurons in the next layer. The network solve a problem, when the user apply the input values to the inputs of the first layer, The signal allow to propagate through the network, and read the output values The network will be at a loss if is asked to associate the same input pattern with different output patterns depending on which other pattern(s) preceded the current pattern.

2.13.2 Steps to calculate activation of hidden nodes

The inner workings of a feed-forward network are very simple. This example is through the calculations needed to do a feed-forward pass through the net (that is to present data to the network).

Step by step:

-add up the products of the input node outputs and their respective weights

-this value (-0.5) is then fed into the activation function of the node.

-select the transfer function

-the same calculation is done for every hidden node, and finally get output node.

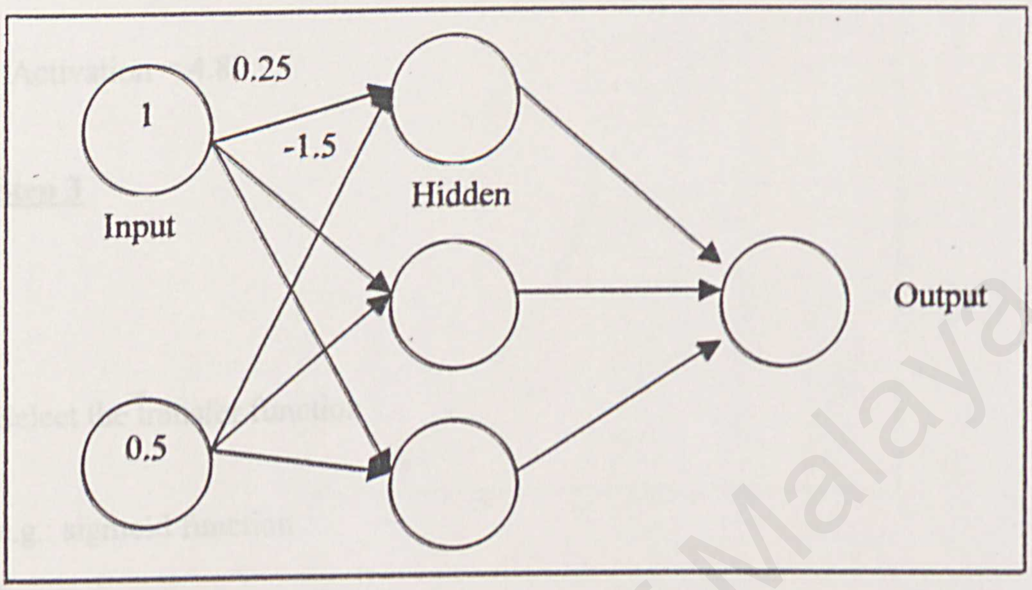


Figure 15: Activation Function for Hidden Nodes

<u>**Example of the calculation activation of the hidden nodes**</u>

Step 1

h1 = (2(0.4) + (0.3(-1.9)) = 0.23

h2 = (2(-0.6) + (0.3(1.5)) = 0.75

h3 = (2(2.4) + (0.3(-3.2)) = 3.84

Step 2

h1 + h2 +h3 = 0.23 +0.75 +3.84

Activation = 4.82

Step 3

Select the transfer function

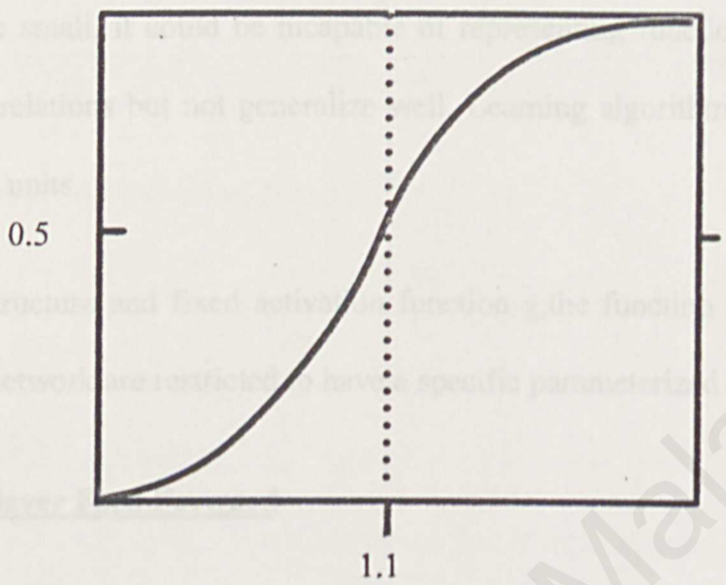
e.g.: sigmoid function

=1 / 1+ e-activation

=1.01

2.13.3 Demonstration 5: Calculation on activation of the hidden nodes

The user must follow the instruction given before they start entering the inputs. The user must pick how many inputs they want to try. For each value of input they enter, it's the same things with the value of weight and hidden layer input. There are three main steps. The first step is input is multiply with the weight. The second step is adding all the value of the multiplied input. The total is also known as value of activation. Then the users neuron in response to a stimulus is about 300 Hz.It turns out that a non-linear activation function is necessary to enable feed-forward networks to approximate any arbitrary function.



2.14 Single Layer Feed Forward

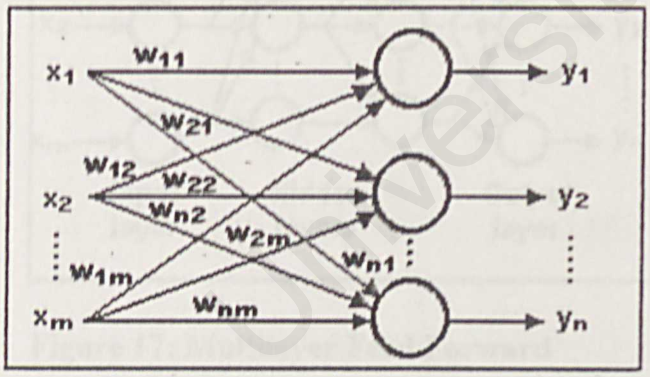


Figure 16: Single Layer Feed Forward

-Links are unidirectional, no cycles

-In layered feed forward, each unit is linked only to units in next layer. No links within layer, no links to previous layer, no links that skip a layer.

-choice of network structure therefore determines what functions can be presented. If network is too small, it could be incapable of representing function. If too large, may memorize all relations but not generalize well. Learning algorithms have been used to add or remove units.

-With fixed structure and fixed activation function g,the function represent able by a feed forward network are restricted to have a specific parameterized structure.

2.14.1 Multiplayer Feed Forward

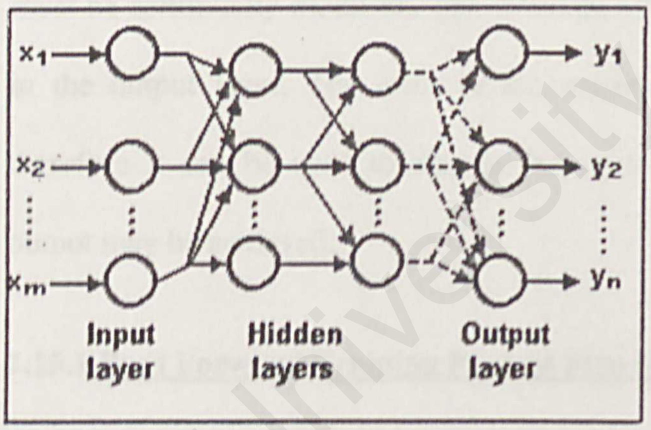


Figure 17: Multilayer Feed Forward

-same idea for learning present input, compare output to desired, adjust weights if there is an error. Trick is to asses blame and divides it among contributing weights.

-in multilayer many weights connecting input to output. Each weight connects to more than one output.

-training curves shows how the error decreases through the learning trials.

2.15 Feed Forward Training Algorithm

The training algorithm is to adjust the network's weight to minimize the delta error. Also to train the neural network, sets of known input-output data points must be assembled. The Back-propagation algorithm is widely used for multi-layer feed-forward. It based on the Delta Rule that states that if the difference (delta) between the user's desired output and the network's actual output is to be minimized, the weights must be continually modified. The result of the transfer function changes the delta error in the output layer. The error in the output layer has been adjusted or fixed, and therefore it can be used to change the input connection weights so that the desired output may be achieved.

2.15.1 Feed Forward Training Process Step by Step

-Converting all input weights to small non-zero values.

- A subset of training samples is presented to the network.

-Each example is fed into the network separately and the obtained outputs are compared to the desired outputs and the size of the error is measured.

- The input connection weights are adjusted and error will be minimized.

-This process is repeated (many epochs) several times until satisfactory results are obtained.

-A threshold of .001 mean squared errors is good.

-The mean squared error is computed by "summing the squared differences between what was a predicted variable should be versus what it actually is, then dividing by the number of components that went into the sum.

2.16 Recurrent Network

2.16.1 <u>Introduction of Recurrent Network</u>

The Recurrent network is similar to a feed forward network. The variation of multilayer perceptron, called 'Elman Network'. A three layer network is used, with the addition of a set 'context units' in the inputs layer. The network has an output connection that travel back to the hidden units. These context units form cyclic flow in the network. There are connections from the hidden layer to these contexts units fixed with the weight 1 and are not included in updating during back propagation of error. When applying the back propagation procedure, the idea of backwards do not violated flow of error, since only the weight of the connections can be update from the context unit to the hidden unit (which flow is the same direction of the network).

- The input connection weights are adjusted and error will be minimized.

-This process is repeated (many epochs) several times until satisfactory results are obtained.

-A threshold of .001 mean squared errors is good.

-The mean squared error is computed by "summing the squared differences between what was a predicted variable should be versus what it actually is, then dividing by the number of components that went into the sum.

2.16 Recurrent Network

2.16.1 Introduction of Recurrent Network

The Recurrent network is similar to a feed forward network. The variation of multilayer perceptron, called 'Elman Network'. A three layer network is used, with the addition of a set 'context units' in the inputs layer. The network has an output connection that travel back to the hidden units. These context units form cyclic flow in the network. There are connections from the hidden layer to these contexts units fixed with the weight 1 and are not included in updating during back propagation of error. When applying the back propagation procedure, the idea of backwards do not violated flow of error, since only the weight of the connections can be update from the context unit to the hidden unit (which flow is the same direction of the network).

2.16.2 Examples of Recurrent Networks

Hop field Network

The network is probably best understood class. It uses bi-directional connections with symmetric weights. The associative memory-after training on a set of examples, new stimulus will cause network to settle into activation pattern that most closely resembles the new stimulus. The pattern are not stored seperately, each weight is a partial encoding of all patterns

Boltzmann Machine

The network is using the symmetric weights. It use stochastic activation function, probability of output being 1 is function of total weighted input. It also resemble simulated annealing

2.17 Recurrent Neuron Model

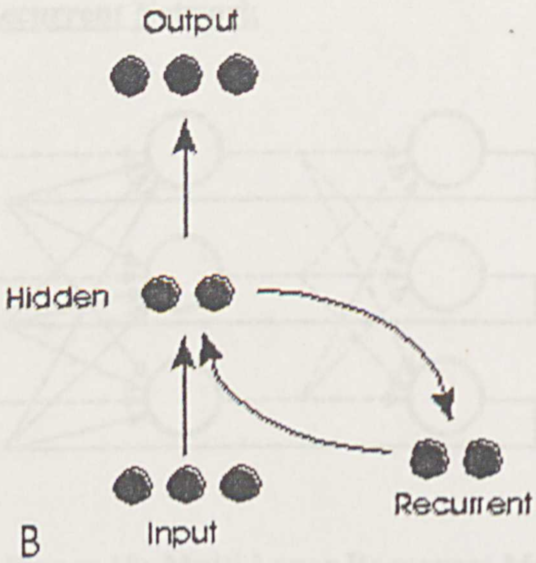


Figure 18: Recurrent Neuron Network

2.18 Recurrent Basic Architecture

The context layer is to create a sense of memory for the network. At the beginning of the network, nothing has been fed through the network yet. At time t0 the first passed input to the input units. In turn the input unit calculates their activation functions and feed the results along to the hidden layer. Since the context layer has not received input, the context layer either does nothing or supplies a default value to be passed to the input as well. Then the hidden units apply their activation function to the sum of all

incoming connections and pass the result along all output connections, which includes context units as well as input units. If error propagation should take place it will happen all this point based on the output to the output units. For the arbitrary time t > t0 when input taken from the input stream. Again apply the input to the input units and the result

2.18.2 Multi Layer Recurrent Network

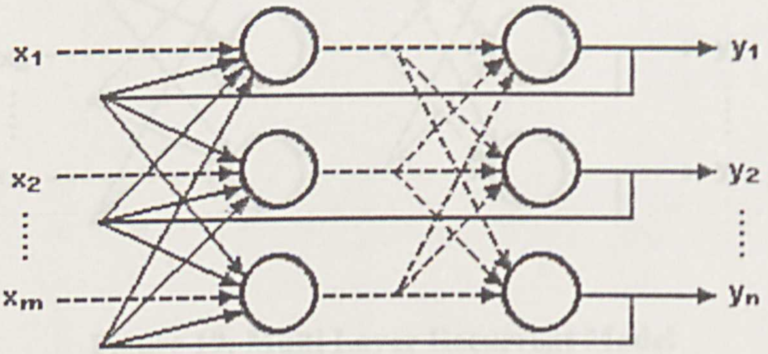


Figure 19: Multi Layer Recurrent Model

2.19Self Organizing Maps (SOM)

2 .19.1 Introduction of SOM

SOM is known as 'Kohonen map', is an unsupervised learning technique that reduces the dimensionality of data through the use of SOM network. It based on a grid artificial neuron whose weights are adapted to match input vectors in a training set. The SOM algorithm is fed with feature vectors, which can be of any dimension will be high. The Output maps can also be made in different dimensions -1 **dimensional,** 2 dimensional, 3 dimensional (most popular) and mainly used for dimensionality reduction rather than expansion.

2.19.2 Model output maps in 1-dimensions

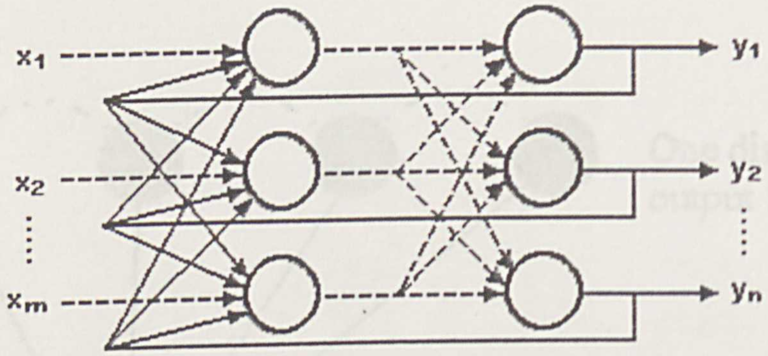


Figure 19: Multi Layer Recurrent Model

2.19 Self Organizing Maps (SOM)

2 .19.1 Introduction of SOM

SOM is known as 'Kohonen map', is an unsupervised learning technique that reduces the dimensionality of data through the use of SOM network. It based on a grid artificial neuron whose weights are adapted to match input vectors in a training set. The SOM algorithm is fed with feature vectors, which can be of any dimension will be high. The Output maps can also be made in different dimensions -**1 dimensional**, 2 dimensional, 3 dimensional (most popular) and mainly used for dimensionality reduction rather than expansion.

2.19.2 Model output maps in 1-dimensions

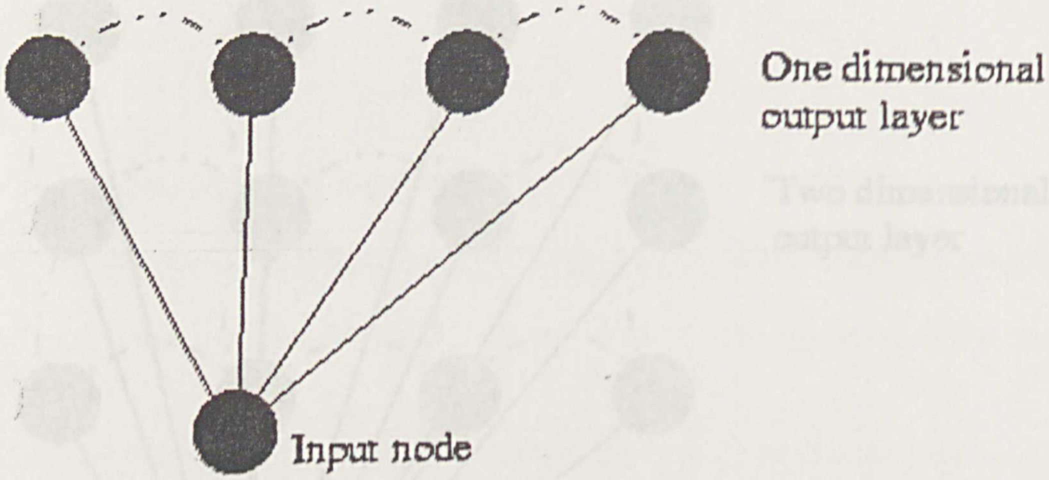


Figure 20: 1 dimensional output layer

SOM is essentially a two layer neural network with full connections, only now with connections between neurons in the neuron layer.

2.19.3 Model in 2 dimensional neuron output

SOM represent a different topology because of the connections between neurons in the output layer.

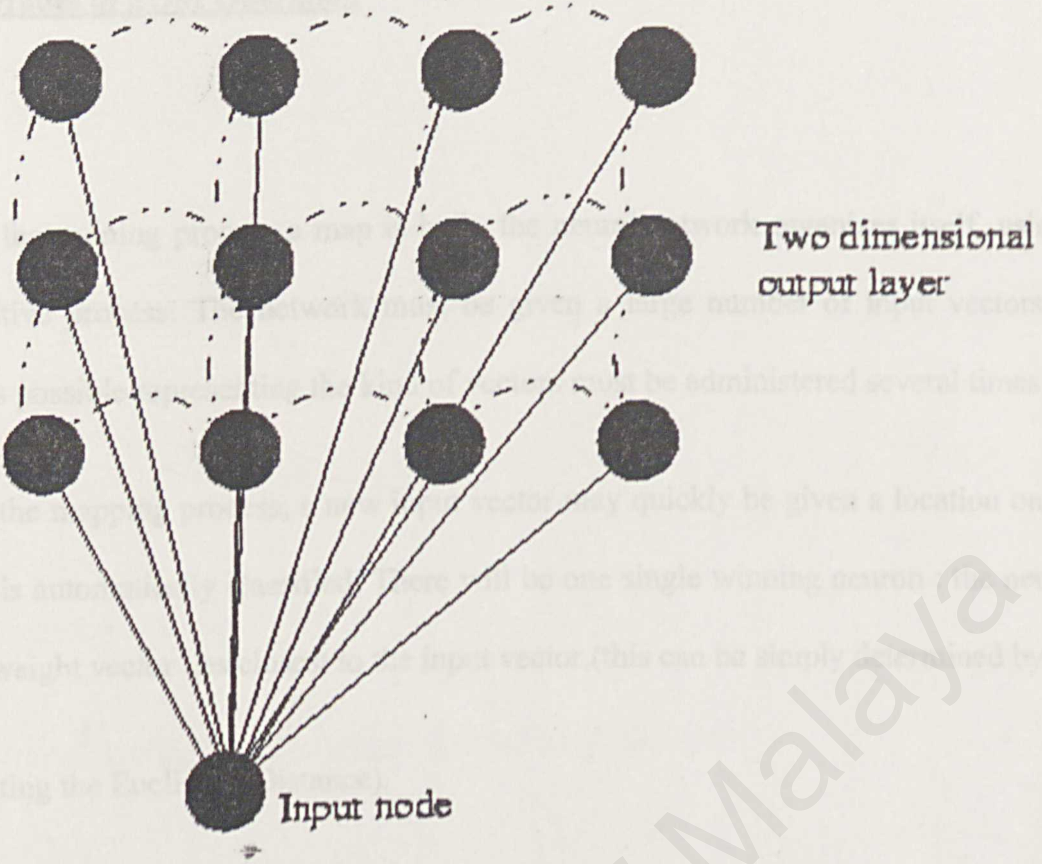


Figure 21: 2 dimensional neuron output

2.20 SOM Training

2.20.1 SOM Algorithm Step by Step

The algorithm is explained a set artificial neuron that each having its own physical location on the output map, which take part in a winner-take-all process (competitive network).It is the situation where a node with its weight vector closest to the vector of inputs is declared the winner and its weights are adjusted making them closer to the input vectors. When this node wins the competition, the neighbor's weight are also

2.20.3 Modes of SOM Operation

During the training process a map is built, the neural network organizes itself, using a competitive process. The network must be given a large number of input vectors, as much as possible representing the kind of vectors must be administered several times.

During the mapping process, a new input vector may quickly be given a location on the map, it is automatically classified. There will be one single winning neuron : the neuron whose weight vector lies closest to the input vector.(this can be simply determined by calculating the Euclidean Distance).

2.20.4 How to calculate Euclidean distance

The **Euclidean distance** of two points $x = (x_1,\ldots,x_n)$ and $y = (y_1,\ldots,y_n)$ in Euclidean n-space is computed as

$$\sqrt{}(x1-y1)2 + (x2 - y2)2 +\ldots+ (xn - yn)2$$

It is the "ordinary" distance between the two points that one would measure with a ruler, which can be proven by repeated application of the Pythagorean theorem. By using this formula as distance, Euclidean space becomes a metric space (even a Hilbert space).The Euclidean distance of two points input vector an weight vector x = (x1,....xn) and (y1,....yn) in Euclidean n-space.

Example of Euclidean Distance calculation

√(x1-y1)2 + (x2 - y2)2 +.....+ (xn - yn)2

1) for 3 inputs, the input vectors are x1 = 2.1, x2 = -0.3, x3 = 3.0
2) the weight vectors are y1 = 0.1, y2 = 2.4, y3 = 0.1
3) √(2.1 -0.1)2 +(-0.3 -2.4)2+(3.0-0.1)2

   = √19.7

4) Euclidean Distance = 3.718

2.20.5 Demonstration 7: Calculation on Euclidean Distance

The user must follow the instruction given before they start entering the inputs. The user must pick how many inputs they want to try. For each value of input they enter, it's the same things with the value of weight and input vector. The total is also known as value of Euclidean Distance. After that they can see the output.

2.20.6 Important keys for the Kohonen SOM

- Competition

for each input pattern, the neuron compete to see which one is the closest to the input.

- Cooperation

the winning neuron determines a neighborhood-a group of networks close by providing the basis for the neighboring neurons to cooperate that is to increase the weights together.

- Adaption

the excited neuron steadily adapt their values to be closer to that of the input pattern through adjustments to their weight.

CHAPTER THREE: METHODOLOGY

CHAPTER 3

3.0 Methodology

A methodology predefines the set of steps followed to build the system (together with a collection of tools to design a system).It also the collection of procedures, techniques, tools and documentation, which helps system developers in their development. It consists of a set of phases, which includes a set of sub phases. The methodology will guide the developers to select the techniques at various stages in the project and helps them to plan, handle, organize and evaluate information systems of the project. The major objective in the methodology is to make the development cycle become well-organized, to minimizes development within the lowest cost, maxsimizes the system quality and finished on time. Also for the maintenance later, it will not be a big problem.

3.1 Project Development Life Cycle

In order to develop a system in an organized and effective way, it is necessary to follow a sequence of steps to accomplish a complete set of task, which is generally called a process. A process is referred to a series of steps involving activities;

constraints and resources that help us produce our intended output. Thus, it is usually involves a set of tools and techniques. When the process involves the building of a certain kind of output it is referred as a life cycle. Therefore the development process of this system is defined as the Project Development Life Cycle, because it describes the life cycle of the system from its conception to its implementation, use and maintenance.

3.2 Waterfall Model with Prototyping

For this project, the waterfall model with prototyping will be used. This model is actually a classic model without the prototyping in its carry stages. Waterfall model is defined as a model process where stages are depicted as cascading from one to another. Thus when the entire requirement are identified, analyzed for completeness and consistency and documented the development can proceed to system design and so on. Prototyping is a sub process that can help in enhancing understanding of the developing system. Prototyping will be involved in the early stages of the model where there is a need for experimentation and learning before commitment of any resources to develop the full-scale system [16]

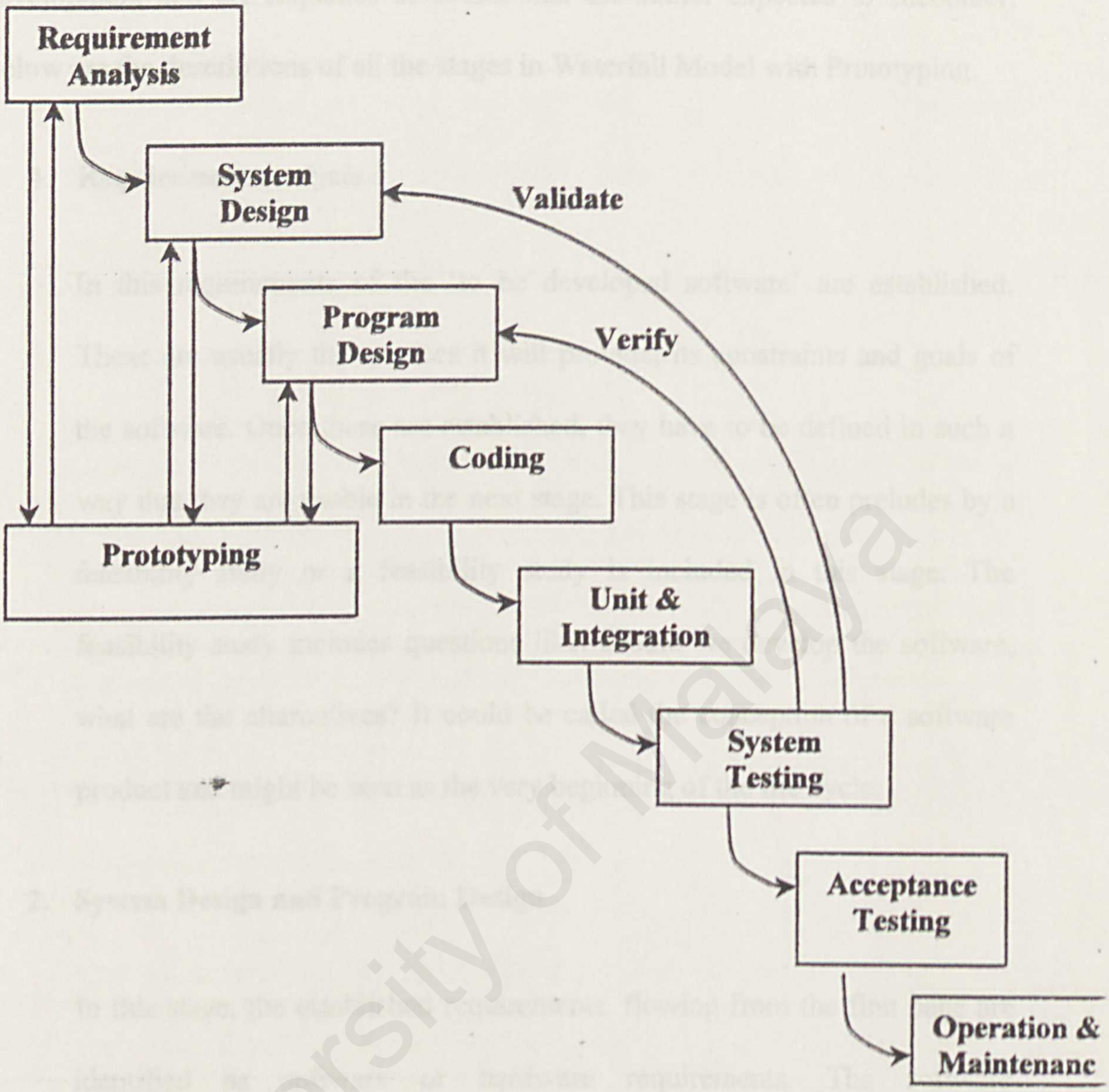


Figure 22: Waterfall Model with Prototyping

Waterfall Model with Prototyping is chosen based on the nature of this project, which demands a systematic and sequential approach to software development that begins at the requirement analysis stage and progress through system design, coding, testing and lastly system operation and maintenances. In overall, the proposed methodology present a high level view of what goes on during development and the sequence of events that the author expected to encounter. Below are the descriptions of all the stages in Waterfall Model with Prototyping.

1. **Requirement analysis**

   In this requirements of the 'to be developed software' are established. These are usually the services it will provide, its constraints and goals of the software. Once these are established, they have to be defined in such a way that they are usable in the next stage. This stage is often preludes by a feasibility study or a feasibility study is included in this stage. The feasibility study includes questions like: should we develop the software, what are the alternatives? It could be called the conception of a software product and might be seen as the very beginning of the life cycle.

2. **System Design and Program Design**

   In this stage, the established requirements, flowing from the first page are identified as software or hardware requirements. The software requirements are then translated in such an away that they can be readily transformed into computer programs.

3. **Coding and Unit and Integration**

   This is the stage where the computer programs are created. Each program is called a unit and unit testing is the verification that every unit meets its specification.

4. **System Testing**

   All the units are combined and now the whole is tested. When the combined programs are successfully tested the software product is finished.

5. **Acceptance Testing and Operation and Maintenance**

   Most software products include this stage of the development involves correcting errors that have gone undetected before, improvement and other forms of support. This stage is part of the life cycle of a software product and not of the strict development although improvements and fixes can still be considered as 'development'.

6. **Requirements prototyping**

   It is to ensure that the requirements are feasible and practical, if not revisions are made at the requirements stage.

7. **Design Prototyping**

   It helps developers assess alternative design strategies and decide which is best for a particular project.

8. **Validation**

   It is to ensure that the system has implemented all the requirements, so that each system function can be traced back to particular requirements in the specification. It is also to make sure that the developer is building the right product according to the specification.

9. **Verification**

   It will ensure that each function works correctly and as needed. It will also will checks the quality of the implementations.

10. **Prototyping**

    Product that has been developed half way to let the user and the developer to check and evaluate the suggested system.

3.3 Rationale of Methodology Approach

The Waterfall Model with Prototyping is much more easier compares to other models, such as waterfall model which only ask the user's requirements at the last phase and a lot of phase and a lot of bugs will be found at the testing and maintenance stage, which will cost a lot of time and money in order to eliminate this bugs and meet the users requirements.

Therefore Waterfall Model with Prototyping allows us to ask the user requirements at each stage in order to produce a good system to the customer. It allows us to do changes while the development phase was going on. For instance, if there are any new requirements appears while designing the system, prototype model allows us to modify the data structure immediately to suite the new requirements. In long term, it will reduce the total cost and time of the project. This is because it is better for us to retrieve the problem at the early stage, rather than finding out during the testing phase. It was hoped that this model would meet up all the users requirements.

<u>Advantage of Waterfall Model with Prototyping</u>

- End – user participation encouraged or required.
- Suited naturally to iteration and change.
- Clarifies for those users who need to see something before they know what they want or for evaluating complex designs.
- Prototypes are experiential, active models.
- Allows early detection of errors and misunderstandings.
- Quicker user feed back for better solutions
- Useful when several stakeholders involved
- Can accelerate life cycle phase by:
    - Allowing experimentation with requirements
    - Showing screen and report layouts
    - Giving code base for implementation when totals are available.

3.4 Requirements Specifications

The requirement of this system can be divided into two categories, functional and non-functional requirement. Functional requirement describe how the system interact with its operating environment. On the other hand, non-functional requirement is a description of feature that completes the system. It can be viewed as extra accessories that complement the functional requirement.

3.4.1 Functional Requirement

A functional requirement describes an interaction between the system and its environment. It also describes how the system should behave gives certain stimuli. Functional requirement for this system is described as follows: -

- **Demonstrations and Calculations**

  That will be provided for the only four particular subtopic: *Introduction of Artificial Neural Network, Feed Forward, Recurrent* and *Self Organizing Maps*. Before the student start doing the demo, there have an example to guide them to do themselves. The example is exactly same with the demo and the explanation and the instruction will be described in details.

- **Tutorials**

  The tutorials will be included in every topic. But the tutorial is different page from the main topic. The format is in the multichoices questions. The student only has to select the correct answer and when they choose it the answer will come out immediately to tell either correct or incorrect. The questions are designed to test a student's grasp for each topic's fundamental concept. Some of the questions are in calculation format but emphasis is given to problems based on real life examples.

- **Concise Information**

Every topic will have the brief explanation before they can understand the demonstration. The essential tips also provided on how to do the correct calculations based on the examples and the formula.

- **Glossary**

It provided to help students to search for the meaning of the certain word. The glossary is arranged alphabetical order to ease the searching. They only needs to take the front alphabet with click on top it. All the explanation given is taken from the proven articles.

- **Help**

This part is like an assistant to the student if they have any problem while using this application. From the main menu until the tutorial and glossary if they have problem they can referred here. Only a short and compact description provided and it will guide them to continue their learning better.

3.4.2 Non – Functional Requirement

Non – Functional Requirement is a description of features that complete the system. These requirements are very subjective but are as important. Non – Functional Requirement for this system is describe as follows: -

- **User friendly**

  This system is easy to use because it the web based format and everyone was familiar with it. This package can be used by anyone although without any computer literate. From the main menu page, the user will feel easy to navigate because all the button are provided clearly like quit,tutorial,glossary and learning module. Thus the user will not become confuse.

- **Effectiveness**

  The system is easy to understand, clear, relevant, compact and prioritize data accuracy, and achieve the predetermined goals and objectives said that out in the planning session.

- **Learn ability**

  The guide and instructions to navigate the system are easy to follow. Therefore anyone who is new also can learn how to use it without any difficulties with the guide form Help. It also easy to remember how to use after a long term of not being used for the new user.

- **Portability**

  The system can be used in any platform or operating system Window 95 and up.

- **Scalability**

  With the future in mind, the system should be able to expand or shrink, according to the user's needs.

- **Modularity**

  The system should be built in the form of modules to ensure easy maintenance and debugging.

- **Respond Time**

  The respond time for data access should be within a reasonable allocation. This means that the data inside the system should be ready at any time and presented with minimal delay.

- **User Interface**

  The user interface for this system is attractive and comfortable to use. Icons enter for example at the front page is clear and direct to the point, entering the main content without puzzling the user. The interface provided to make the system' user is feel like welcoming.

3.5 Selection of Development Tool

A study of the available software tools for the development of a system was conducted to get right ones for the system.

3.5.1 Operating system

- **Microsoft Windows 2000**

This operating system was developed by Microsoft and remains the most popular choice to date. With the Windows 2000 version, the file system has been changed from FAT 32 to NTES allowing a more secure framework for business online. It is added security features and overall ease of handling, in addition to the compatibility with a majority of other software has made Windows 2000 the operating system of choice for this project.

3.5.2 Software

❖ **Macromedia Flash MX**

Macromedia Flash MX allows us to create a web experiences that is more attractive, more compelling and compatible with more browsers that with any other web solution. Dynamic HTML (DHTML), Java and other advanced web design formats are not reliable alternatives, since they are either not compatible or inconsistent with different browsers. With Macromedia Flash Player's pervasive penetration, wide availability, scalable vectors, animation, sound and more. Macromedia Flash MX sites provide the high production quality that attracts visitors and brings tem back to your site. Since Macromedia Flash MX is free of the limiting design capabilities of more traditional web display options. Macromedia Flash MX is also the key to designing and delivering low bandwidth animations presentations and websites. It offers scripting capabilities and server side connectivity for creating engaging applications, web interfaces and training courses.

Macromedia Flash MX Features

- **Enhanced color control**

The color panels within Flash are an improvement in Flash new lists the hexadecimal codes for colors when user run the cursor over the color swatch. And it's easier to charge colors by simply selecting an objects and then selecting a different color with either the color selector or the swatch panel.

- **Shared Libraries**

User can share library element and save fonts to the library in this Flash version. The user can click the library items in other Flash Player movies.

- **Expanded Action Script**

Action Script is a robust object oriented programming language used to enter the actions to Flash movies. Users can have a control over the behavior of objects on the stage with the new predefined classes.Besides,the user have the ability to export and import Action Scripts (.as) file which makes sharing Action Scripts between files a simple task. The development of Demonstration, modular scripts that are parameterized into movie clips which can be shared and reuse between the system individuals, is an excellent time saving feature.XML objects and the debugger are tow useful advanced Action Scripts function.

- **The Movie Explorer**

The movie explorer enable user to navigate the timeline of a movie, search for elements within a movie and perform site widely editing. It's extremely hard to have the capability to display the complete contents of the movie. The movie explorers letting the sites search their Flash sites for any symbols, picture or string or text.

- **Support for importing Mp3 sound files**

User can import mp3 files in Flash MX.They do not have to compress the sound during export a movie with sounds, thus it can reduces the time requires for publishing and exporting. Using compressed sounds also reduces the file size of completed movies and reduces memory requirement during authoring.

Advantages of Macromedia Flash MX

a. Vector graphics are flexible where they are very small in size and are therefore quick to download.

b. Little or no programming knowledge is necessary to use Flash compared to Java Applets, it is easy for non – technical user's to create Flash movies.

c. Flash's SWF file format is fairly open standard.

d. Widespread acceptance and adoption of Flash player.

e. Cross – platform development environment.

f. The Flash development environment promotes designing movies as a collection of reusable objects. This results in smaller download sizes and reduced development.

Disadvantages of Macromedia Flash MX

Flash suffers from some usability problems. The usability problems are as follows: -

a. Long download times.

Despite lightweight vector animations, Flash movies can take longer to download than most users want to wait.

b. **Flash can destroy normal user navigation.**

Sites that are done entirely on Flash are a usability nightmare. Web users are accustomed to certain navigation controls such as back and forward button. Unfortunately, in multi page Flash sites, the back button doesn't take us to the previous page, it will take us to the previous site.

c. **Stale content since creating content with Flash is typically more abor –** Intensive than creating similar content with HTML, Flash content tends to be changed less frequently.

d. **Lack of user control.**

The users cannot use the browser's 'Find' feature to find specific text on a page.

e. **Accessibility Problems.**

It also appears that Flash content cannot be accessed by screen readers and therefore is unavailable to users with poor or no eyesight.

Why chose Macromedia Flash MX and not Microsoft Front Page?

a. Front Page has a tendency to overwrite your code with proprietary code, including some that is very Internet Explorer specific. That never happens in Flash MX. It produces clean code and it can always be edited the way a user wants to.

b. In addition, while Front Page includes an extension that lets a user add functionality to their website, it must be installed on your web server, which creates a security issue. Doing this creates a 'hole', through which the Front Page extensions on the developer's computer communicate. Unfortunately, hackers often exploit that hole. Flash MX also lets user to add features to interfaces, but none of them rely on a server to work.

c. The Flash MX interface also makes designing sites easy. There's a split screen View, which lets users see their design and code at the same time and the program's flexible, customizable templates.

❖ **Adobe Photoshop 6.0**

Adobe Photoshop is professional graphic editing software. Disputably the best there is, Photoshop has been the primary choice of digital artists worldwide. Its ability to edit and change image properties, digital effects, lighting and color as well as a graphic text editing makes it a very powerful tool indeed. Adobe Photoshop also supports PNG (Portable Network Graphics), Progressive JPEG and PDF formats as well as the regular host of file formats.

3.5.3 Hardware Requirements

The hardware listed below is considered sufficient to develop, test and run the Interactive Learning Package: -

- **An Intel Pentium IV Desktop computer**

  The Pentium IV Processor is chosen as the minimum requirements for its multimedia processing capability.

- **64 MB Main Memory (128 MB recommended)**

  64 MB of main memory is required to run Windows 2000, but to smooth things along the way, 128 MB would be sufficient, so as not to encounter hiccups while developing or running the system.

- **1.0 GB hard disk space**

  The system deals with graphics as well as a substantial amount of text and coding. Therefore, the 1.0 GB estimate is considered safe to accommodate the data needed to develop and run the system.

3.6 Summary

This chapter discusses the methods and techniques used in information gathering and construction of the system. The use of the waterfall model with prototyping was explained so as to get the best results for a risk based model, ensuring the quality of the end product. It also included the requirements analysis, detailing them in choice of hardware and giving the traits in which they were preferred for substantiated software.

CHAPTER FOUR: SYSTEM ANALYSIS AND DESIGN

CHAPTER FOUR: SYSTEM ANALYSIS AND DESIGN

CHAPTER 4

4.0 System Analysis and Design

4.1 System Design

In this chapter the specification of each chapter will be explained. After doing a through analysis, the system can be design and that will meet the requirement. The objectives of these phases are to design an effective system that meets the requirements and are within the constraint bounds. This is just a basic logical design that is going to be used in designing Interactive Learning Package for Artificial Neural Network and it is subjected to changes. System design is important for planning the overall structure of the system. Through this chapter, the system layout and structural blueprint are documented for future use when the author would actually develop the system.

4.2 System Modeling

A system is representation of an in-place or proposed system that described the data flow through out the structure. The module described the point where data or information enters a system and the places where it will be processed, as well as the actions taken and the points where data will be output. A system model is documented through a variety of design diagrams. A design diagram is a graphical or visual representation of a structure. Design diagrams include DFD, structured charts, decision tree and other items.

4.3 System Hierarchy

Representing structural design for system hierarchy for the system.

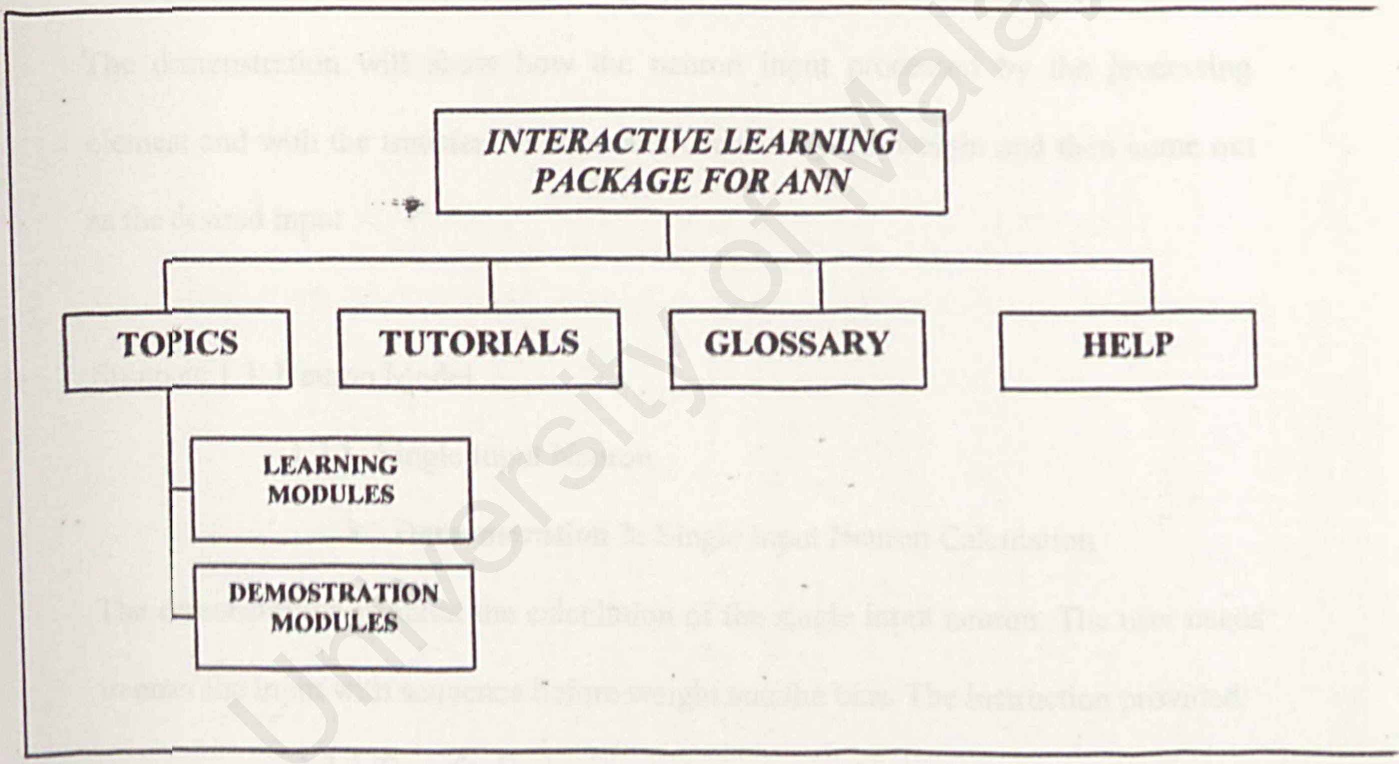


Figure 23: The chart of system hierarchy

4.3.1 Main topics

<u>Topic 1: Introduction</u>

Subtopic 1.1-The Biological Neuron Model

- **Demonstration 1:** The simultaneous process between the neuron.

The demonstration will show how the human input carried by the neuron form the brain and produce the human feedback like see, hear, taste and touch.

Subtopic 1.2-Directed Graph Description of ANN

- **Demonstration 2:** The directed graph of the neuron

The demonstration will show how the neuron input processed by the processing element and with the transfer function and connect by the weight and then come out as the desired input

Subtopic 1.3: Neuron Model

1.3.1- Single Input Neuron

- **Demonstration 3:** Single Input Neuron Calculation

The demonstration is about the calculation of the single input neuron. The user needs to enter the input with sequence before weight and the bias. The instruction provided.

1.3.2-Transfer Function

1.3.3-Multiple Input Neuron

- **Demonstration 4:** Multiple Input Neuron Calculation

The demonstration is about the calculation of the multiple input neurons. The user needs to enter the input with sequence before weight and the bias. Because it the multiple neuron so the user can change the amount of 2 and 3 input. The instruction provided.

Subtopic 1.4: Network Architecture

1.4.1: Layer of Neuron

1.4.2 – Multilayer Neuron

This subtopic will discuss about the layer of the neuron and how it work with the calculation.

Topic 2: PerceptronNetworks

Subtopic 2.1: Neuron Model

This subtopic will discuss about the history of the Perceptron and the basic neuron model.

Subtopic 2.2: Perceptron Learning Rules

This topic will discuss about the transfer function of the Perceptron, descriptions about the Perceptron learning rule, how to constructing the Perceptron Learning Rule with the example of solved problem. All the minor subtopic are explained clearly.

Subtopic 2.3: Perceptron Network Architecture

This topic will discuss about the Perceptron architecture form the single input to the multiple input with the picture explanation and the formula to search the decision boundary with graph.

Topic 3: Feed forward Network

Subtopic 3.1: Feed Forward Neuron Model.

This subtopic will discuss about the basic of Feed forward network, the advantages and disadvantages. Also the explanation of neuron model.

Subtopic 3.2: Feed Forward Basic Architecture

This topic will discuss the architecture, the description on how feed forward function, step on how to calculate the activation function for the hidden nodes, the squashing function, the picture of the single layer and multiplayer Feed Forward.

➢ **Demonstration 4: Calculation of Activation Function On Hidden Nodes.**

The demonstration shows the calculation on how to get the activation function for the hidden nodes. Before the user start the demo, the example will be given in detail so they can understand clearly what to do. The user needs to enter the input with sequence before weight and the value of the hidden nodes .Because it the multiple neuron so the user can change the amount of 2 and 3 input. The instruction provided.

Subtopic 3.3: Feed Forward Training

This topic will discuss the Feed Forward training algorithm, training process step by step in details.

Topic 4: Recurrent

Subtopic 4.1: Recurrent Neuron Model

This subtopic will discuss about the basic recurrent neuron model and the example of the Recurrent network. : Hopfield and Elman in details.

- **Demonstration 5:** Recurrent Neuron Model

The demonstration will show the movement of the neuron input goes to contex layer before processed in the hidden layer and come out as the desired output.

Topic 5: Self Organizing Maps (SOM)

Subtopic 5.1: SOM Neuron Model

This subtopic will discuss about the basic design of the SOM neuron model, the 1 dimensional and 2 dimensional model along the attractive picture also the understandable explanations.

Subtopic 5.2: SOM Training

This topic will discuss the SOM algorithm step by step, modes of SOM operation, how to get the self organizing feature maps through the algorithm and how to find the Euclidean Distance.

- **Demonstration 6: Calculation on how to find the Euclidean Distance.**

The demonstration shows the calculation of the Euclidean Distance on how to get the winning neuron from the minimum distance of input and weight vector. Before the user start the demo, the example will be given in detail so they can understand clearly what to do. The user needs to enter the input with sequence before weight. Because it the multi input neuron so the user can change the amount of input. The instruction provided.

4.3.2 Tutorial

At the main menu also got the link to the Tutorial. In this page provided the tutorial for every topics.Therefore, every time they finished certain topic so they can go to do self - evaluation through the exercises. The tutorial are included the mathematical calculations or comprehension in application within five question per topics. The student can assess themselves when they try the tutorial repeatly.From there, they will know the level of the understanding in how certain subject topic. The tutorial provided are in the multichoices question with A, B, C, and D answer. The student must read through the question before they answer because when they selected the command will come out either correct (right symbol) or incorrect (wrong symbol).

4.3.3 Glossary

This module will help user to find the certain important word to know the meaning in more detail. The word can be finding through the sequences of alphabetical and the user only need to pick one word and select used the front alphabet and the system will list down all the meaning for the particular word.

4.3.4 Help

The module will help the user on how to use the whole package especially when they have a problem in demonstration modules. Also if the users do not know how to use the certain function in this module they can go to the Help link and search out. The system are provided the solution how to solve it through the user's guide.

4.4 Flow Chart of System

The flow chart presents the flow of the series for every each module in the Interactive Learning Package for Artificial Neural Network. Every figure is pictured with more detail and specifically representing the flow chart of the system.

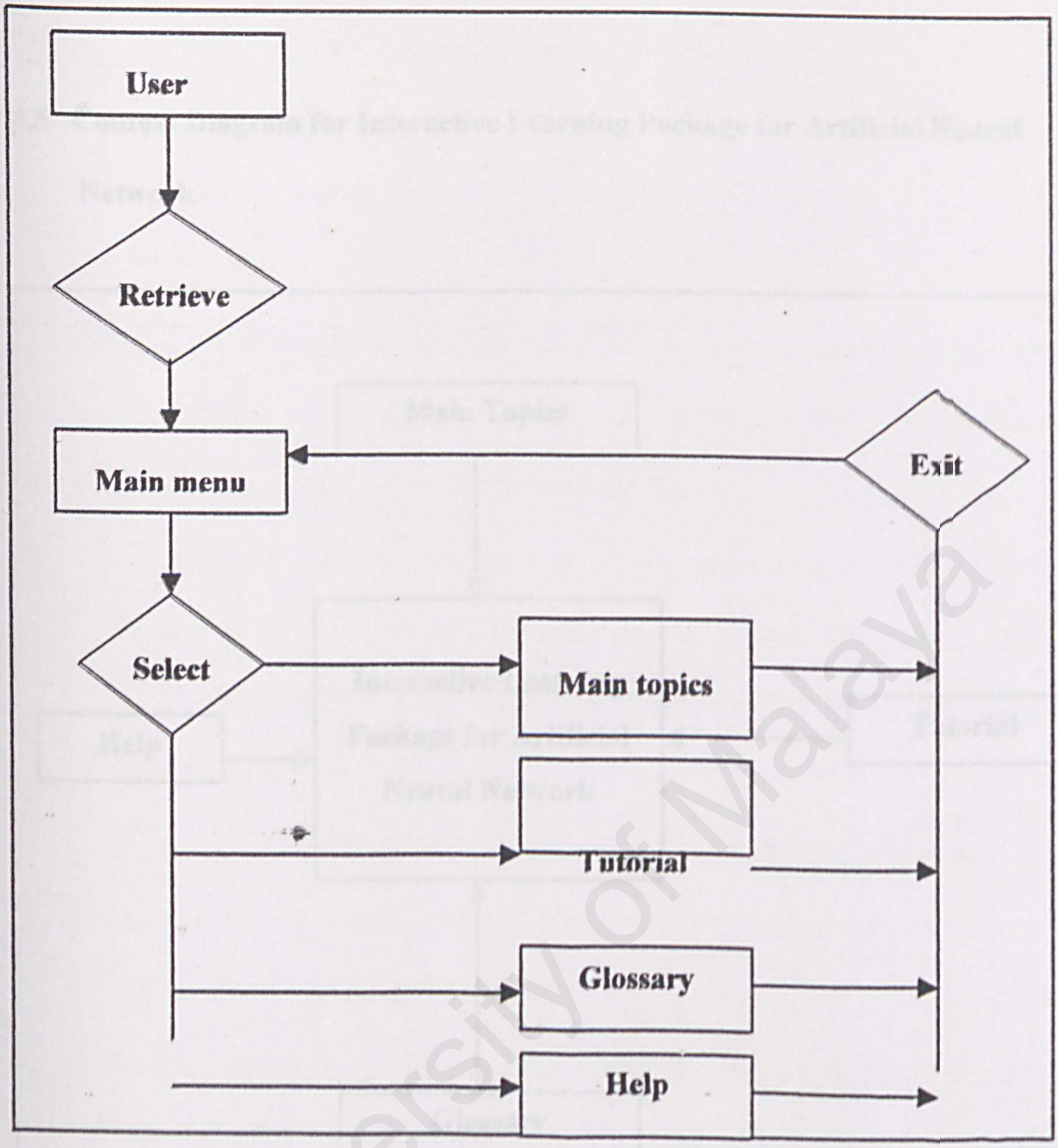


Figure 25: Flow Chart of System

4.5 Context Diagram for Interactive Learning Package for Artificial Neural Network

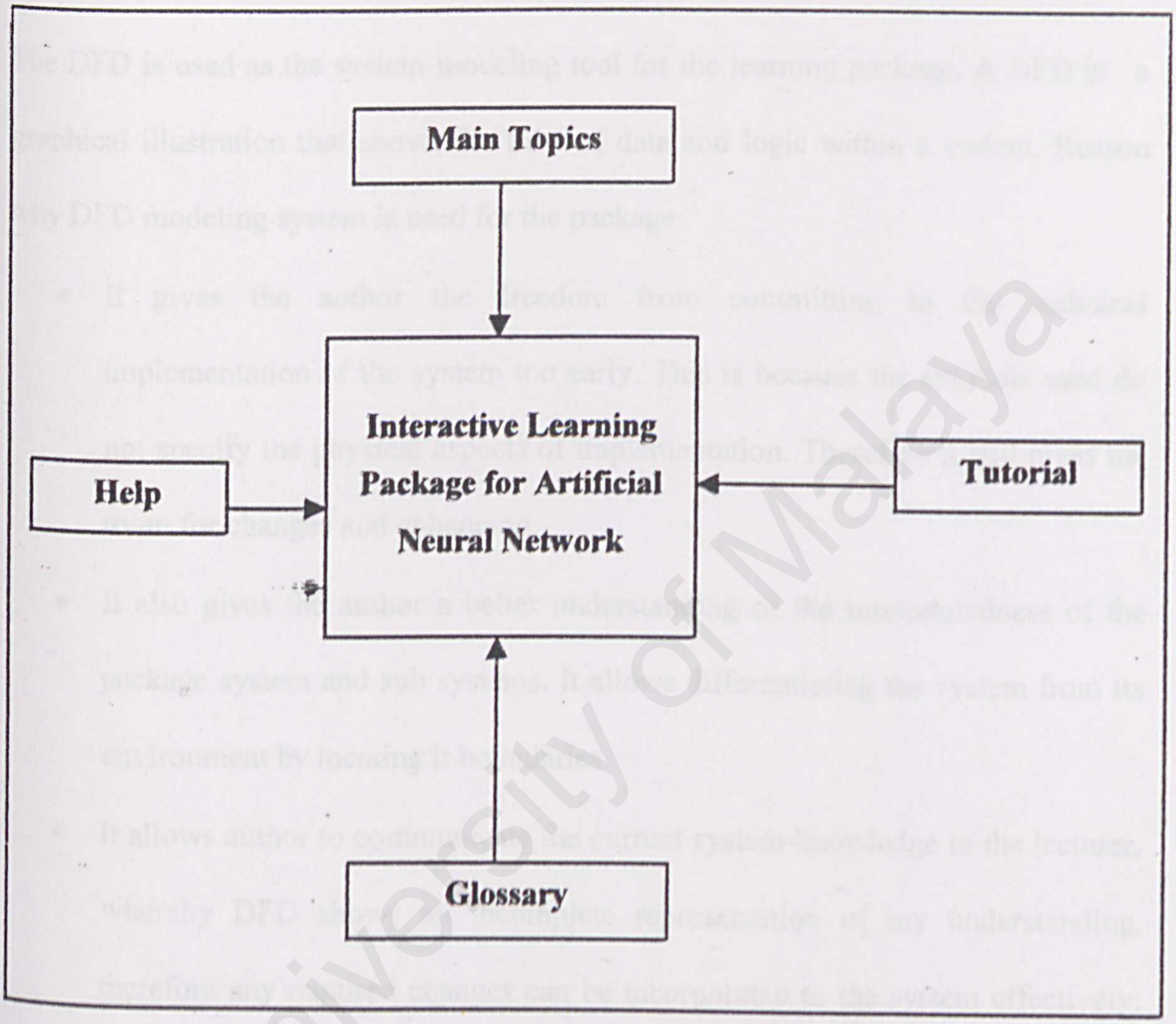


Figure 26: System's Context Diagram

4.6 Data Flow Diagram (DFD)

The DFD is used as the system-modeling tool for the learning package. A DFD is a graphical illustration that shows the flow of data and logic within a system. Reason why DFD modeling system is used for the package:

- It gives the author the freedom from committing to the technical implementation of the system too early. This is because the symbols used do not specify the physical aspects of implementation. Therefore it still gives me room for changes and enhancing
- It also gives the author a better understanding of the interrelatedness of the package system and sub systems. It allows differentiating the system from its environment by locating it boundaries.
- It allows author to communicate the current system knowledge to the lecturer, whereby DFD shows an incomplete representation of my understanding, therefore any required changes can be incorporated to the system effectively. Necessary background is provided so that the data flow diagram will be meaningful rather than confusing.
- It also allows the author to describe each component used in the diagram whereby to ensure all necessary output can be obtained from the input data and that processing logic is reflected in the diagram. Moreover, it also allows early detection and correcting error and design flows in the early stage.

4.6.1 DFD for Interactive Learning Package for Artificial Neural Network

The DFD for the system are divided into two levels where level one is for the main menu and the second level is for the main topics. The figure below showed the DFD for the system.

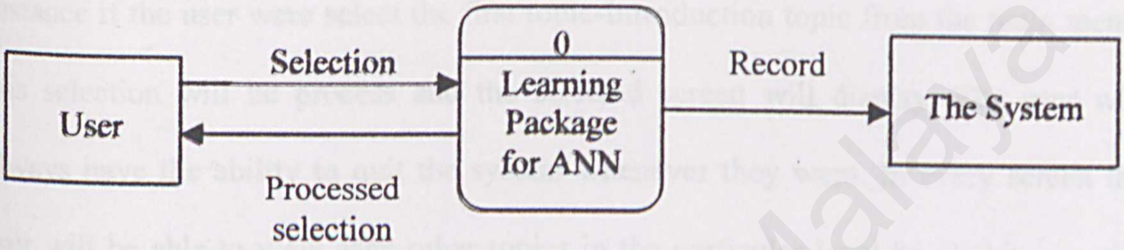


Figure 27: Data Flow Diagram Level 0

4.6.2 DFD of the First Level for the Main Menu

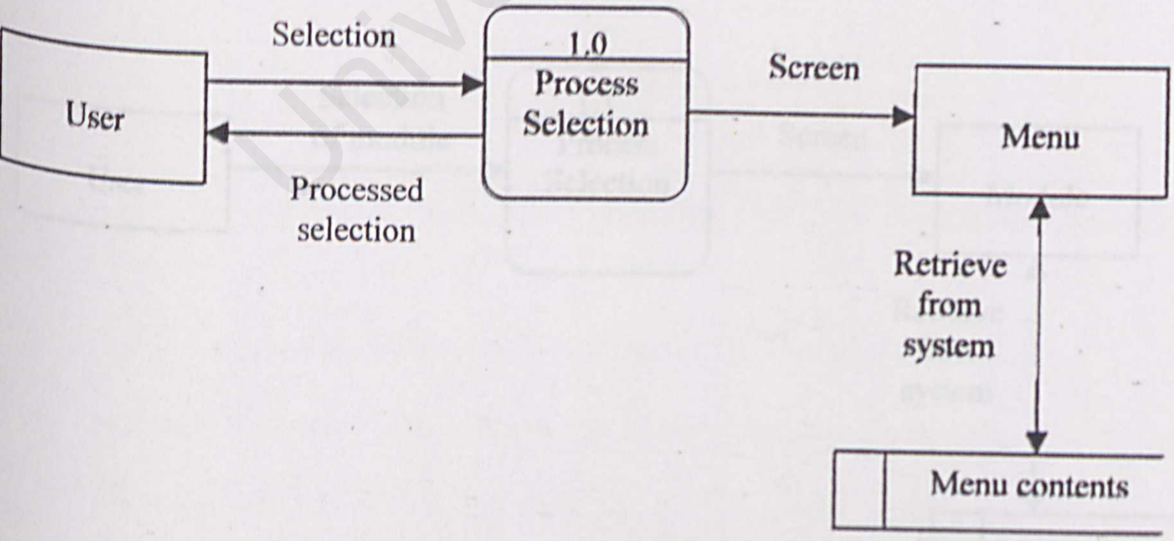


Figure 28: Data Flow Diagram First Level

The figure above is the first level of DFD in learning package where the user can choose which level the user wants to view. From the main menu the user can select any of the five main topics, certain subtopics, tutorials, glossary and help to view. For instance if the user were select the first topic-Introduction topic from the main menu, this selection will be process and the selected screen will display. The user will always have the ability to quit the system whenever they want. In every screen the user will be able to view each other topics in the particular level by clicking on the link button. A back link also provided in the screen to ease navigation and to allow the user the ability to traverse back and forth between the pages.

4.6.3 DFD of the Second Level for the Main Modules.

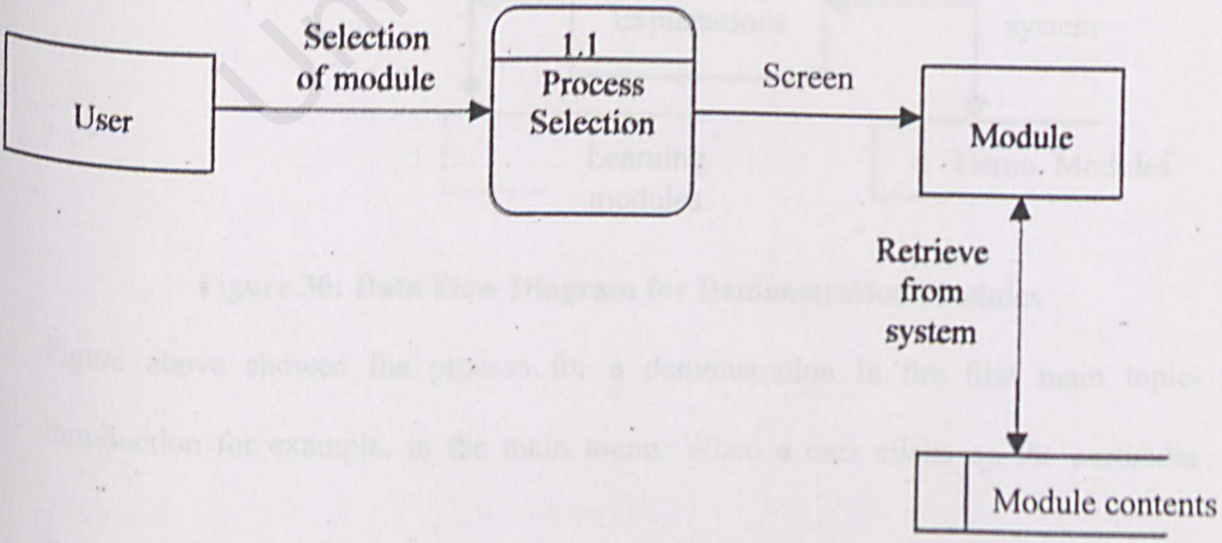


Figure 29: Data Flow Diagram Second Level

In the second level of DFD that included all the five main modules of the learning package included Introduction of Artificial Neural Network, Recurrent, Perceptron, Feed Forward and SOM. Above are the processes that happen when the user clicks on the particular link button at the main menu. When the user click on the link of the last topic-SOM in the menu page, where it is will be divided into several other subtopic. The user is allowed to select any subtopics they want to view. The particular subtopics will have their own demonstration on calculations or others thus if the user want to view and the link button are provided.

4.6.4 DFD for Demonstration modules

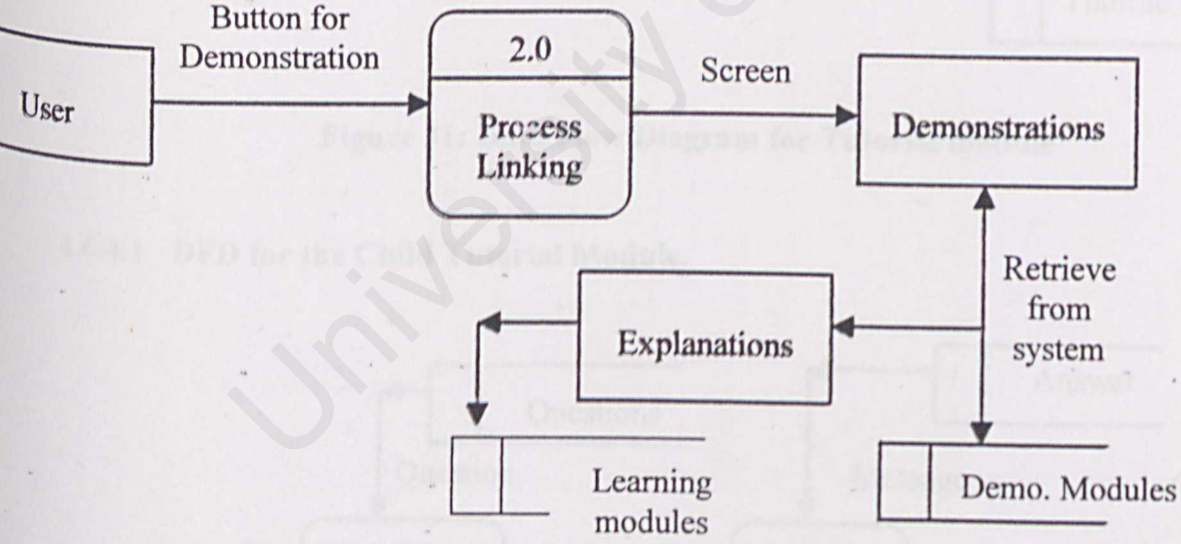


Figure 30: Data Flow Diagram for Demonstration Modules

Figure above showed the process for a demonstration in the first main topic-Introduction for example, in the main menu. When a user clicks on the particular interface content, for instance second subtopic-Directed Graph of ANN, the selection will be process and the demonstration interface will then displayed on the screen to the user. The user can choose to go back to the explanation page with the button provided.

DFD for Tutorial Module

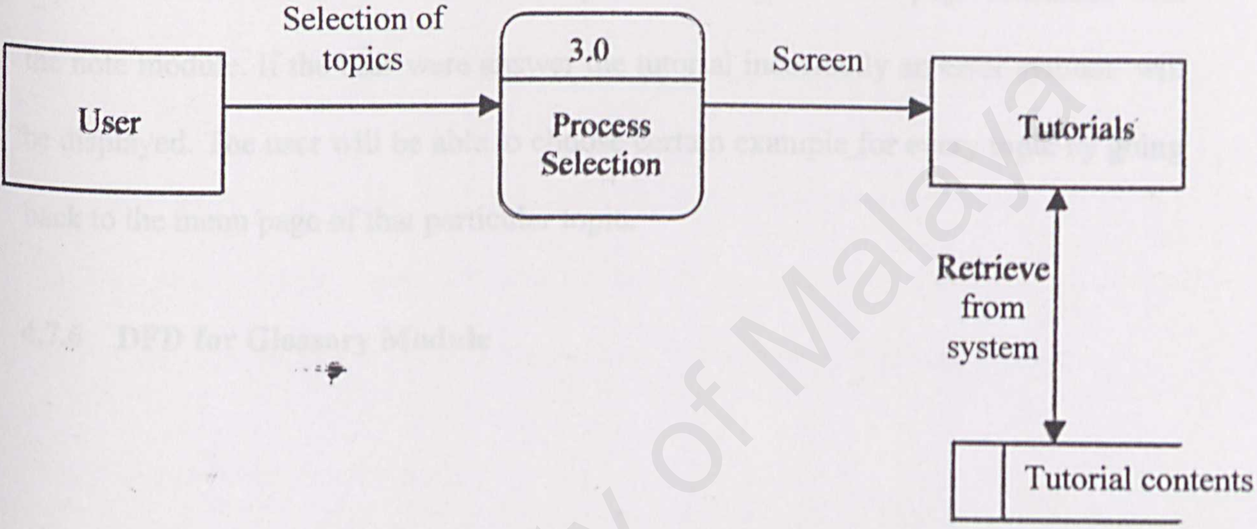
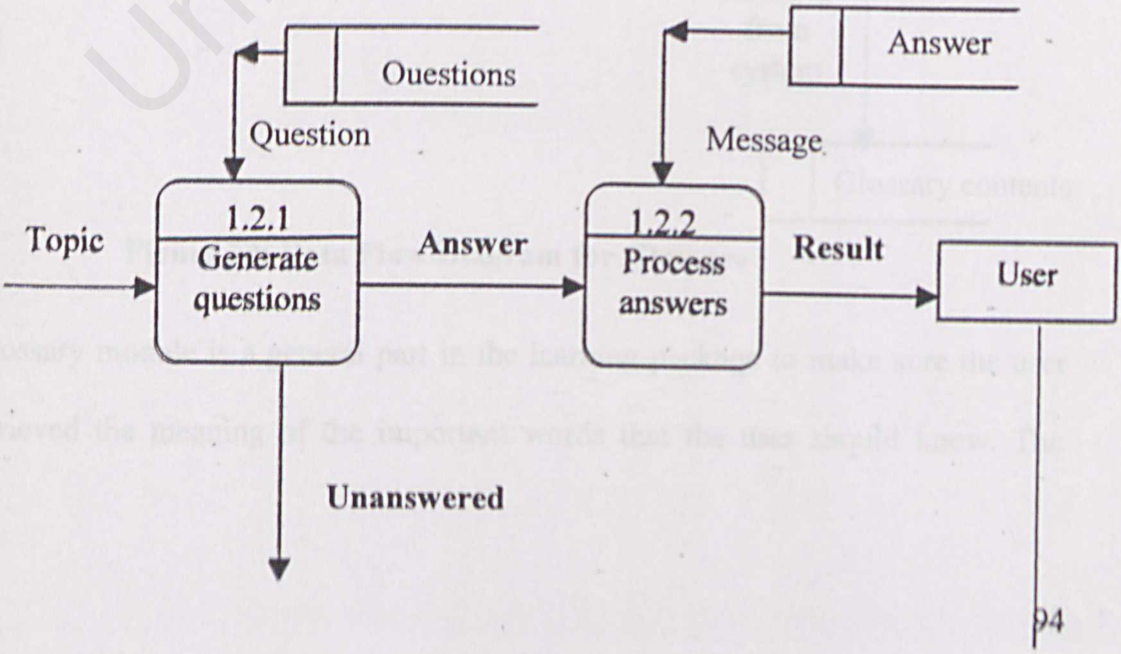


Figure 31: Data Flow Diagram for Tutorial module

4.6.4.1 DFD for the Child Tutorial Module.



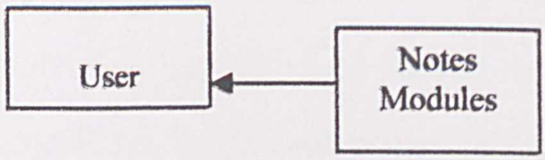


Figure 32: Data Flow Diagram for child diagram of Tutorial module

When the user enters the selection for tutorial than the above process will take place. The tutorial will be retrieved from the system and displayed to the user. The user can choose to either answer or return to the particular menu's main page continues with the note module. If the user were answer the tutorial incorrectly an error symbol will be displayed. The user will be able to choose certain example for every topic by going back to the menu page of that particular topic.

4.7.6 DFD for Glossary Module

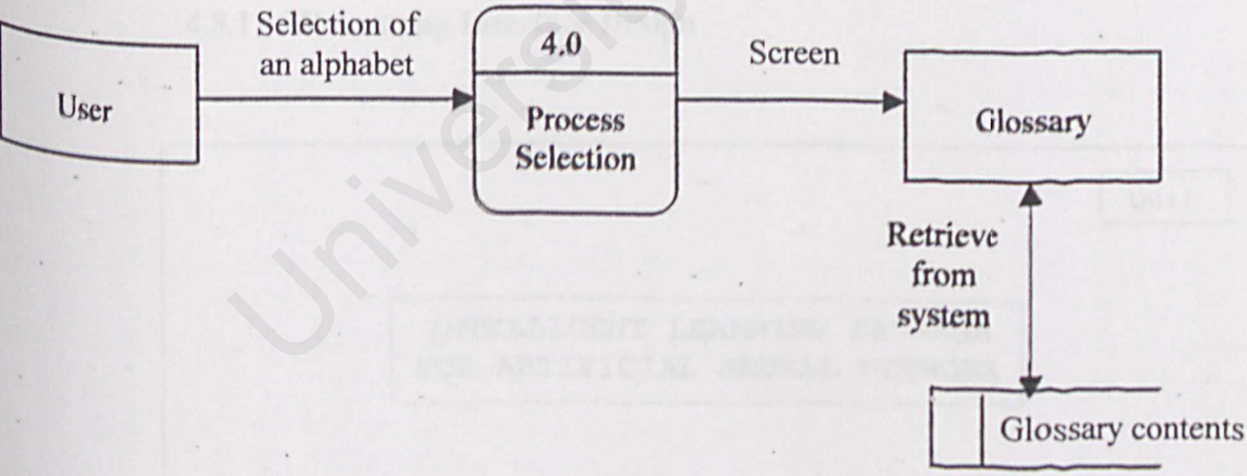


Figure 33: Data Flow Diagram for Glossary

The Glossary module is a general part in the learning package to make sure the user can retrieved the meaning of the important words that the user should know. The

4.7.7 DFD for Help

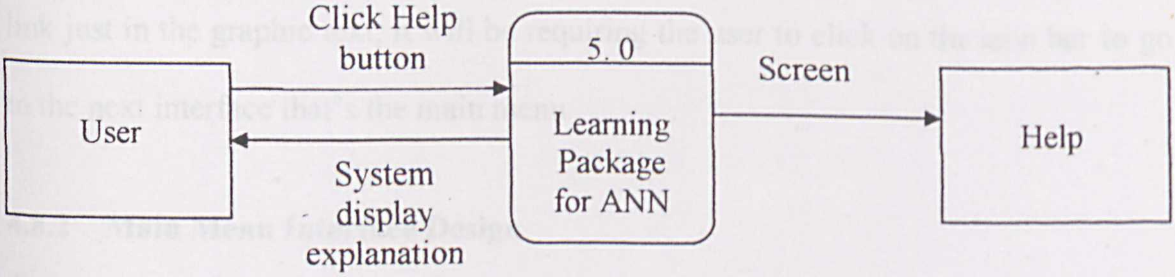


Figure 34: Data Flow Diagram for Help Modules

The help function gives the user guidance on how to navigate the system correctly in the way to achieve the optimum performance level. They can type the key word of those questions, submit to the system and answers will be displayed directly.

4.8 User Interface Design

4.8.1 Welcoming Interface Design

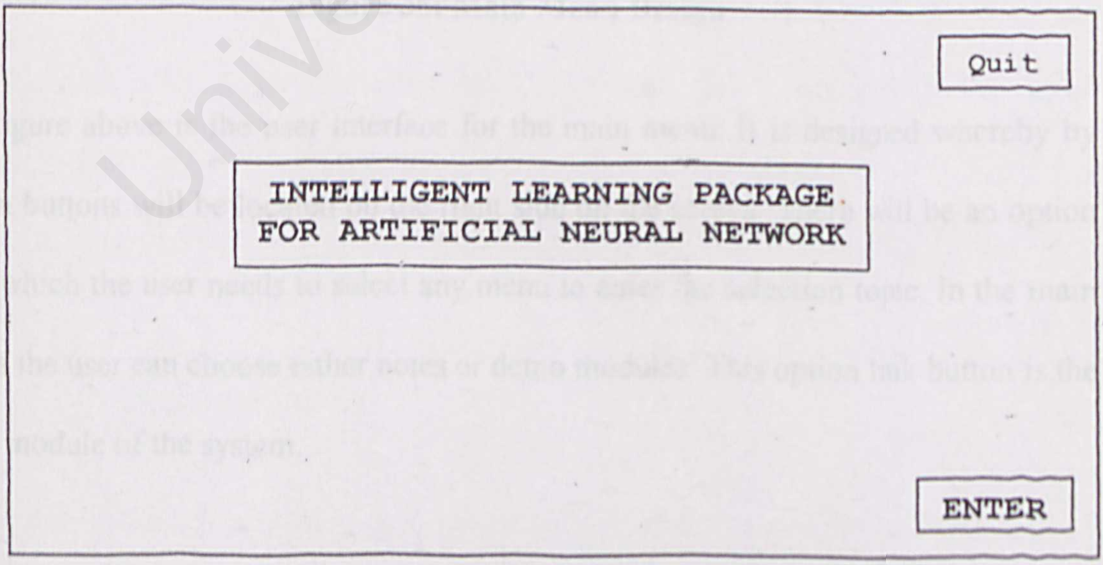


Figure 35: Interface Design

The above user interface is for welcoming all users when they visit this web site. The design is meant where it given and expression that they will be going to enjoy this site exploring every interface. The top text will be the themes for the system. As for the link just in the graphic text, it will be requiring the user to click on the icon bar to go to the next interface that's the main menu.

4.8.2 Main Menu Interface Design

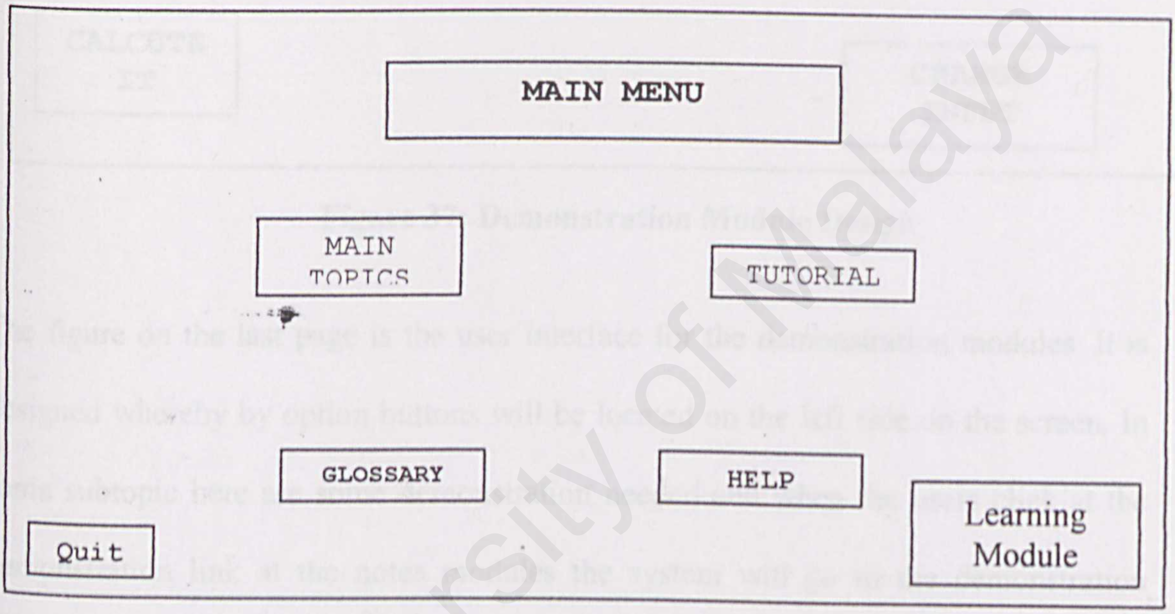


Figure 36: Main Menu Design

The figure above is the user interface for the main menu. It is designed whereby by option buttons will be located on the right side on the screen. There will be an option link, which the user needs to select any menu to enter the selection topic. In the main topics the user can choose either notes or demo modules. This option link button is the main module of the system.

4.8.3 Demonstration Modules Interface Design

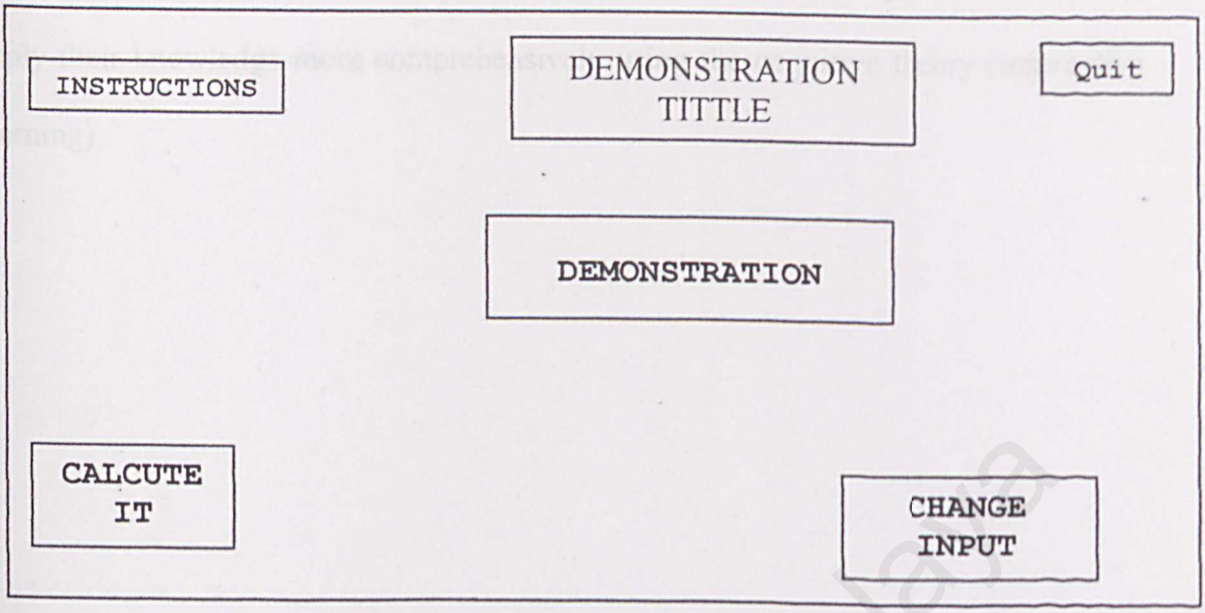


Figure 37: Demonstration Module Design

The figure on the last page is the user interface for the demonstration modules. It is designed whereby by option buttons will be located on the left side on the screen. In some subtopic here are some demonstration needed and when the users click at the demonstration link at the notes modules the system will go to the demonstration screen that it will show the demonstration. There will be an option link, which the user needs to select to calculate the calculation: calculate it and change the input: change. Button quit will bring the user back to the explanation before.

4.9 Excepted Output

It is expected that after utilizing the learning package, the student and the lecturer would have a better understanding of ANN. It is also expected that they will no longer suffer from computer anxiety but have total confidence in approaching the Information Computer Technology (ICT) nowadays. And also they would be able to apply their knowledge more comprehensively using the cognitive theory (interactive learning)

CHAPTER FIVE: SYSTEM IMPLEMENTATION

CHAPTER 5

5.0 SYSTEM IMPLEMENTATION

5.1 Introduction

System implementation is the following phase after system design. It is a process that converts the system requirements and design into 'production'. Its also includes building and testing the system, which it also called the construction phase. Construction phase of the system involves the conversion of the system requirements and design into program codes.

5.2 System Coding

Coding is the process that translates a detail design representation of software into a programming language realization. Notes that, the coding section just involving the build of interactive application part and for creating a web page, which is implemented using the programming language. The following coding principles were applied during the implementation of learning package.

- Coding conventions

Coding conventions such as program labeling, naming connections such as naming conventions and indentation contribute to program readability.

- Readability

Codes should be very easily revised or corrected. To facilitate maintenance, code should be readable, modular and as general as possible.

- Robustness

The codes should be able to handle cases of user error by responding appropriately, perhaps with a diagnostic error message and system failure should not result.

5.3 Coding Methodology

The design of the system must be translated into a form that can be understood or run by the machine. The code generation steps perform this task. These learning package modules were written using programming language. Method used for coding this learning package is top-down method where high-level modules will be coded first and the lower modules which to be filled later. The lower module is only a shell with an entry and an exit. This approach in used is allowed testing to begin on some of the modules while others are still being coded.

5.4 Coding Style

Below are descriptions of the code scraps for the interactive learning package for the overall system module and the demonstration modules.

5.4.1 Coding for connecting the Welcoming coding

After combining all the movie files into one folder namely 'Tesis',a projector file is created in order to change all this movie file (.swf) to Tesis.exe file. A simple coding written to integrate this file with the Welcoming and Main Menu page. The coding for the is shown below:

```
fscommand("fullscreen", "true");
fscommand("showmenu", "false");
fscommand("allowscale", "false");
fscommand("trapallkeys", "false");
gotoAndPlay(1);
```

Welcoming Coding

5.4.2 Coding for connecting all movie files into 1 movie.

All movies files must be placed in the same folder a simple coding is written to connect all the movie files. The code is shown below:

```
on (release) {
loadMovie("Demo_1.swf", 1);
}
```

5.4.3 Coding to loading the Learning Modules to the Demonstration Module

```
on (release) {
   loadMovieNum("content_layout.swf", 0);
}
```

5.4.4 Coding for connecting the Main Topic to the Contents Topic

All main topics are connecting to the contents with the same simple coding. The coding is shown below:

```
on (release) {
   gotoAndPlay("content", 1);
}
```

It's the same coding like tutorials just change the scene name.

```
on (release) {
   gotoAndPlay("tutorial 1", 1);
}
```

5.4.5 Coding for the Demonstration 3

There are four main coding:

Coding for Calculate button

```
on (press) {

 var n, w, p, ;{

 n = (w1*p1+int(b))+int(w2*p2+int(b));

   gotoAndPlay(2);
```

Coding for transfer function button

Hardlim

```
on (release) {

            if (n>0) {

                  a = 0;

             } else if (n == 0) {

                  a = 1;

             }

          }

 on (press) {

              if (n>0) {

                        a = 0;

                        } else if (n == 0) {
```

```
                a = 1;

                }

gotoAndPlay(25);        }
```

Purelin

```
on (release) {

        a = int(n);

}

on (press) {

 a = int(n);

 gotoAndPlay(20);

}
```

Saturing Line

```
        on (release) {

 if (n<0) {

     a = 0;

 } else if ((n>=0) && (n<=1)) {

     a = int(n);

 }

}
```

```
on (press) {
  if (n<0) {
    a = 0;
  } else if ((n>=0) && (n<=1)) {
    a = int(n);
  }
  gotoAndPlay(30);
}
```

5.4.6 Coding for the Demonstration 4

Coding for Calculate button for two inputs

```
on (press) {
          var n, w, p, b;
// if (((p<=-2)||(p>=2)) && ((w<=-2)||(w>=2)) && ((w<=-2)||(w>=2))) {
  n = (w1*p1+int(b))+int(w2*p2+int(b));
// }
  gotoAndPlay(2);
}
```

Coding for Calculate button for three inputs

```
on (press) {
    var n, w, p, b;
 // if (((p<=-2)||(p>=2)) && ((w<=-2)||(w>=2)) && ((w<=-2)||(w>=2)))
 {
    n = (w1*p1+int(b))+int(w2*p2+int(b))+int(w3*p3+int(b));
    // }
    // gotoAndPlay(22);
}
```

Coding for transfer function for two and three inputs

Same with the transfer function coding in Demonstration 3

Coding for connecting the User Input to the selected demo

```
    on (press) {
 if (nin == 2) {
gotoAndPlay("input_2");
 } else if (nin == 3) {
 gotoAndPlay("input_3");
  } }
```

5.4.7 Coding for the Demonstration 6

Coding for Calculate Button for One Input

```
on (press) {
    var h;
    j = p*w;
    h1 = (p*w1);
    h2 = (p*w2);
    h = h1+h2;
    o = 1/(1+(Math.pow(Math.E, h))); }
```

Coding for Calculate Button for Two Input

```
on (press) {
    var j;
    h1 = (p1*w11)+(p2*w21)+(p3*w31);
    h2 = (p1*w12)+(p2*w22)+(p3*w32);
    h3 = (p1*w13)+(p2*w23)+(p3*w33);
    h4 = (p1*w14)+(p2*w24)+(p3*w34);
    j = h1+h2+h3+h4;
    o = 1/(1+(Math.pow(Math.E, Math.abs(j))));
```

Coding for Calculate Button for Three Input

```
on (press) {
  var j;
   h1 = (p1*w11)+(p2*w21)+(p3*w31);
   h2 = (p1*w12)+(p2*w22)+(p3*w32);
   h3 = (p1*w13)+(p2*w23)+(p3*w33);
   h4 = (p1*w14)+(p2*w24)+(p3*w34);
   j = h1+h2+h3+h4;
o = 1/(1+(Math.pow(Math.E, Math.abs(j))));
```

5.4.7 Coding For Demonstration 7

Coding for Calculate button for one inputs

```
on (press) {
   ed = Math.sqrt(Math.pow((x-y), 2));
}
```

Coding for Calculate button for three inputs

```
on (press) {
  var j;
  j = Math.pow((x1-y1), 2)+Math.pow((x2-y2), 2)+Math.pow((x3-y3), 2);
  ed = Math.sqrt(j);
}
```

Coding for Calculate button for two inputs

```
on (press) {
 var j;
  j = Math.pow((x1-y1), 2)+Math.pow((x2-y2), 2);
 ed = Math.sqrt(j); }
```

Coding for Calculate button for four inputs

```
        on (press) {
                var j;
                j = Math.pow((x1-y1), 2)+Math.pow((x2-y2), 2)+Math.pow((x3-
        y3), 2)+Math.pow((x4-y4), 2);
         ed = Math.sqrt(j);
}
```

5.5 System Development

The development environment influence part of a system production. The usage of appropriate tools, software and hardware determines whether or not the system is successful formed. This phase must be carefully constructed, as any mistake would affect the entire project. The three sections that must be considered to intensify the system quality are:

i). Development strategy

For the certain system, each module is sovereign to each other. Therefore, the bottom-up development strategy used and every module is constructed separately. At the last part, the modules are combined to produce the complete system.

ii). **Development platform configuration**

This development has a certain impact on the development of a system. Using the suitable hardware and software will not only help to speed up the system development but also determine the success of the project.

iii). **Development techniques**

a) **Prototype production**

During the prototype stage, a lot of things must be considered likes user interface, user input, and the output design. Frequently, while prototyping all the module were initially fix. After that, Macromedia Flash MX coding will take place to make some function worked. The prototype was created in module by module before they were combined become one complete system.

b) **Testing**

Testing is performed to ensure that the program are executed correctly and conform to the specified requirements.

5.6 System Integration

This aspect is one of the important aspects that ensuring the system meets all requirements. Integration will make all the separate modules were combined to make one complete system .Omissions that were not recognized during the module testing were corrected during the integration process and the errors also can be easily corrected.

5.7 Summary

The System Implementation discussed a few important aspects of the Interactive Learning package for Artificial Neural Network for Demonstration Module. This chapter also reviews all the important coding for the system and the demonstration module. All the coding explained are fully programmed by Flash MX and the language is similar like the Java Scripts. The system development were discussed about three important aspects which are the development strategy, development platform configurations and development techniques

CHAPTER SIX: SYSTEM TESTING

CHAPTER 6 .

6.0 System Testing

6.1 Introduction

System Testing is a probably the least understood part of the software development project. Bug is any unexpected, questionable or undesired aspect or behavior displayed, facilitated or caused by the software being tested. Testing can uncover different classes of error in a minimum of time and with a minimum amount of effort.

6.2 Testing Phase

There are five major types of testing in the system testing and there are shown below:

- Unit Testing

The Unit Testing is the process of testing each program component units own and the component is totally isolated from the rest of the component in the system. Each sub modules can be referred as a single unit. In order to perform unit testing, the module interface is tested to ensure that information flows properly into and out from the program unit during testing.

- Module testing

A module is a collection of dependent components and encapsulates where related components. Module testing enables each module to be tested independently.

- Integration testing

When collection of a components have been module tested. The next step is to ensure that the interface among the components are delivered and handled properly. Integration testing is the process of verifying that the system computer work together as desired in the system and program design specifically.

Condition to test	Data / steps to perform	Expected results	Changes made after testing
Tutorials	-Select the certain tutorial. -Answer the selected question -Pick only one answer	-The answer will come out immediately. -There will be a command to tell whether the user's	-The correct answer is appearing in right symbols with added sound. -The wrong

		answer is correct or not.	answer is appearing in wrong symbol also with added sound. answer it will give the correct or the wrong answer
Glossary	-Select the wanted meaning of the word. -Select at the alphabet of the word.	-The pages of the certain meaning will displayed and listed.	-Every page are having the same contents of alphabet in order form Ato Z. -Make the user feel user friendly and they can navigate easier.
Help	-Select the Help part from the 'Main Menu' if users have any doubts.	-the Help front page will be displayed and there will have the linking button.	-Every button in Help will allow user to go to other linking in Help.
Demonstrations	-Select the topic that provided demo: Feed Forward	-the topic will be displayed and search for the demo -click the demo button	-the demo will ask the user input. -the button on the page will function when the user enter the input.

Figure 38: Integration testing for certain modules

- System testing

The last testing procedure completed is system testing. System testing tests all implementation aspects of the design. Once the entire system is validated it must be combined with other system elements such as hardware, end user and database.Sytem testing verifies that elements are functioning properly and the overall system performed and objectives are achieved.

In this case Interactive Learning Package for ANN must achieve reliability, robustness, accuracy and modularity. Next is the performance testing where function performance is evaluated. The performance testing consists of a variety of testing. These tests depend in the non functional system specified.

Configuration tests are conducted on various types of hardware and the result must meet minimum requirement. The best results will appear when the hardware configuration is above the minimum requirements.The tables above is the system testing for the whole system.

Testing Part	Test procedures	Expected Output	Results of Testing
Welcoming Page	-Select the enter button to come in the 'Main Menu' -Select the 'Quit' button to exit the system.	-Go to the main menu page.	-The welcoming voice inserted at the welcoming page ask user to click the enter button to go to the

			main menu.
Main menu	-Select topics from the 'Main Menu' -Click at the 'Learning Module' button to get an explanation details about demo.	-the menus for the 'Main Topic; will be displayed. -the next and previous button also displayed.	-the selected topic page will be displayed and user can get the contents or the demo included. -user can go to the next and previous page.
Topic	-Select the one topic from the 'Main Menu' through the button:Perceptron	-The Perceptron front page will displayed	-User can click at the subtopic and then the subtopic page will be displayed.

Figure 39: System testing for whole system

- Acceptance testing

This testing is defined as the process of formal testing conducted to determine whether or not the system satisfies with acceptance criteria and to enable the user to determine whether or not they accept the system. The purpose of this testing is to ensure that user's requirement objectives are met and all the components are correctly included in a customer package.

6.3 Testing Strategies

The objectives are to design and organize testing activities. Strategies will make testing processes simply performed. There are a lot of testing strategies such as:

i). Develop test objectives

ii). Design test cases

iii). Compose test cases

iv). Testing test cases

v). Implementing test cases

vi). Validating testing results

6.4 Maintenance Testing

After software system has been verified, tested and implementation, it must continue to be maintained.Maintance routines will very depending on the type and the complexity of the technology. System will need to be maintained to ensure that they continue to perform to the level demonstrated during the system testing stage, whëre modifications to software are made as a result of system maintance and upgrades. It may be necessary to investigate furthermore for the system verification and testing to ensure that standards are still met by the modified system.

6.5 Summary

The System Testing discussed clearly about the testing processes which were performed on the system. Testing is the requirement to test the constructed programs to search out any program faults. Many techniques such as reviews, analysis, inspections and walkthroughs were performed for the testing system to make sure that the system is flowing smooth. Debugging activity which also supports the testing processes is a method of analyzing and locating bugs when the system does not work properly. The levels of testing that correspond to the of software specification, were the module testing, integration testing, system testing user acceptance testing and maintance.

CHAPTER SEVEN: SYSTEM EVALUATION

CHAPTER 7

7.0 System Evaluation

System evaluation is the stage where the system was fully tested and no error, to have the system evaluation we are conducted a survey to evaluate the effectiveness and any difficulties that faced by the user. The objectives are to define all the weakness or the advantages of the learning package in the ways to meets the users' requirement and expectation.

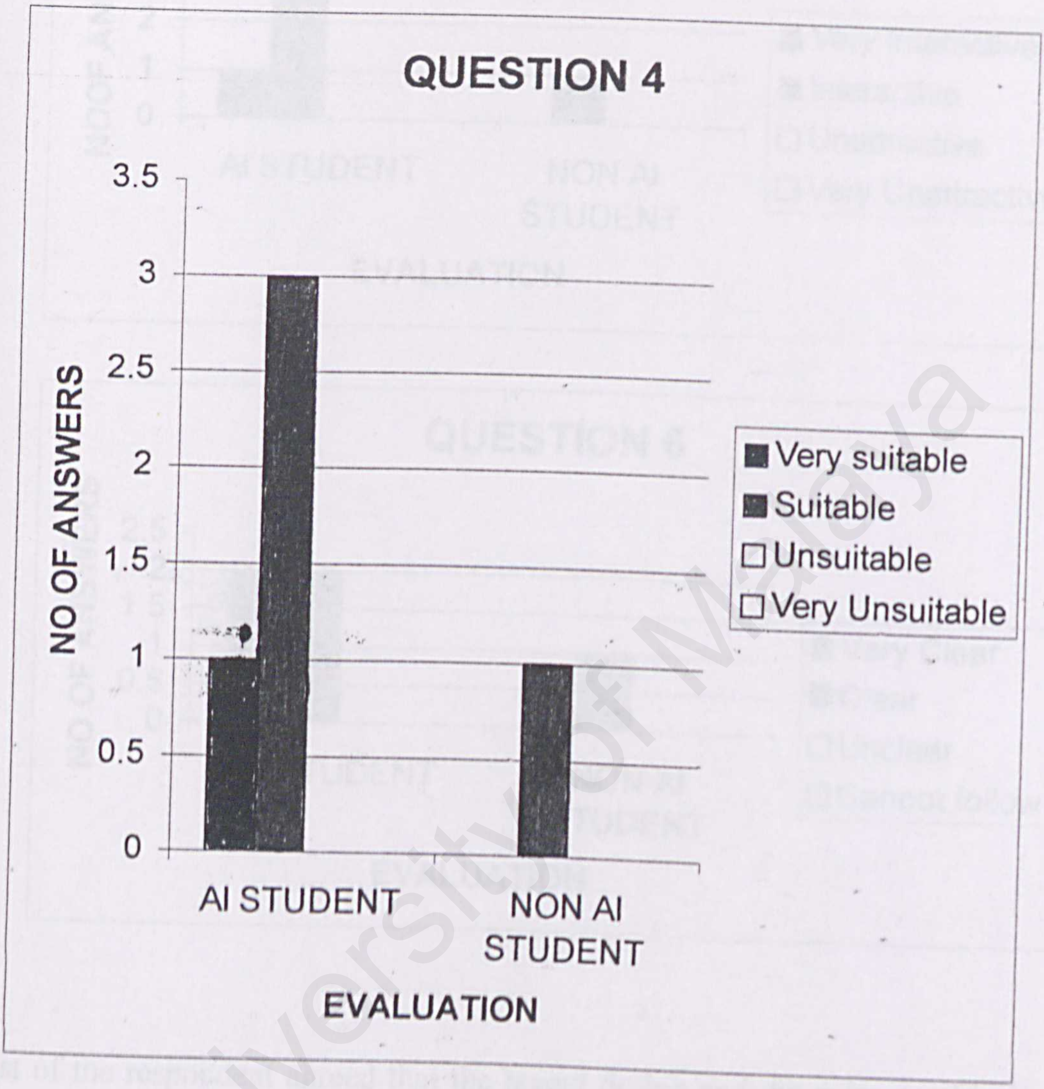
7.1 Evaluation Manual

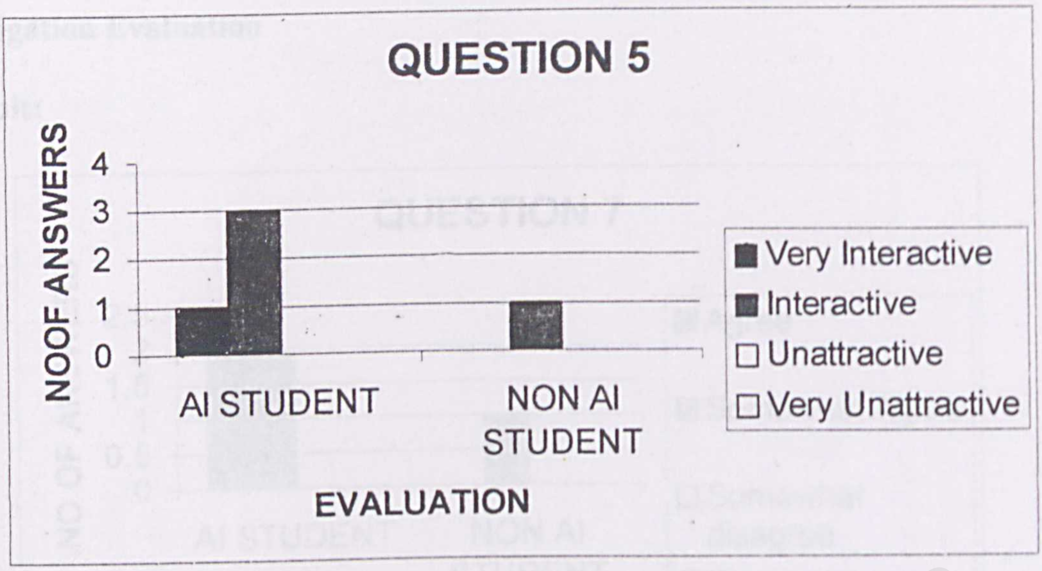
After the system was fully developed, user evaluation was conducted to asses the effectiveness, the general of this Interactive Learning Package modules. Five samples were taken from my course mate in Artificial Intelligence Department and five samples were taken from the other non Artificial Intelligence student.

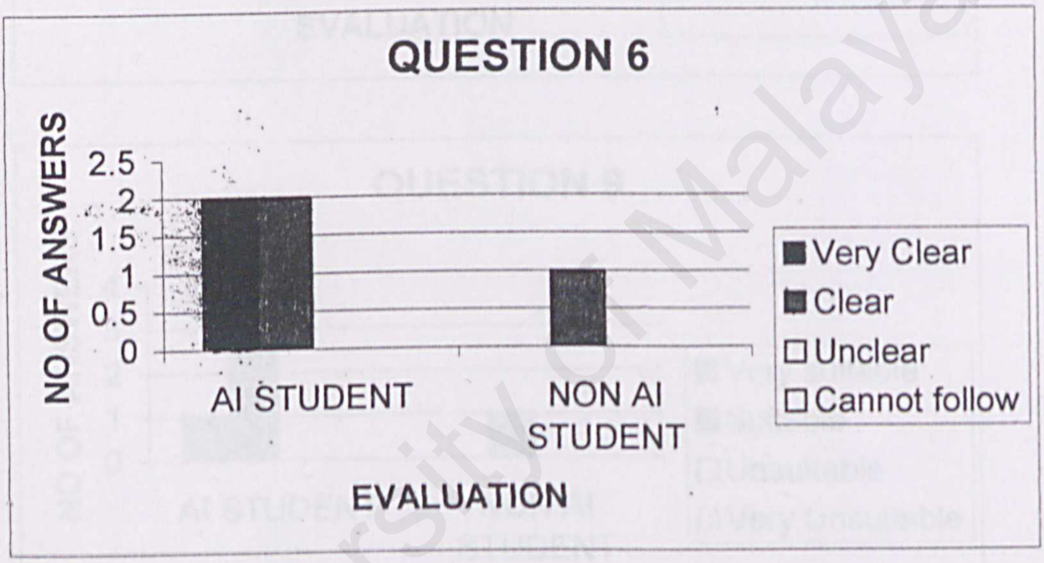
The user evaluation's questionnaire can be obtained from the appendix at the end of the report.

7.1.1 Interface Evaluation

Result:



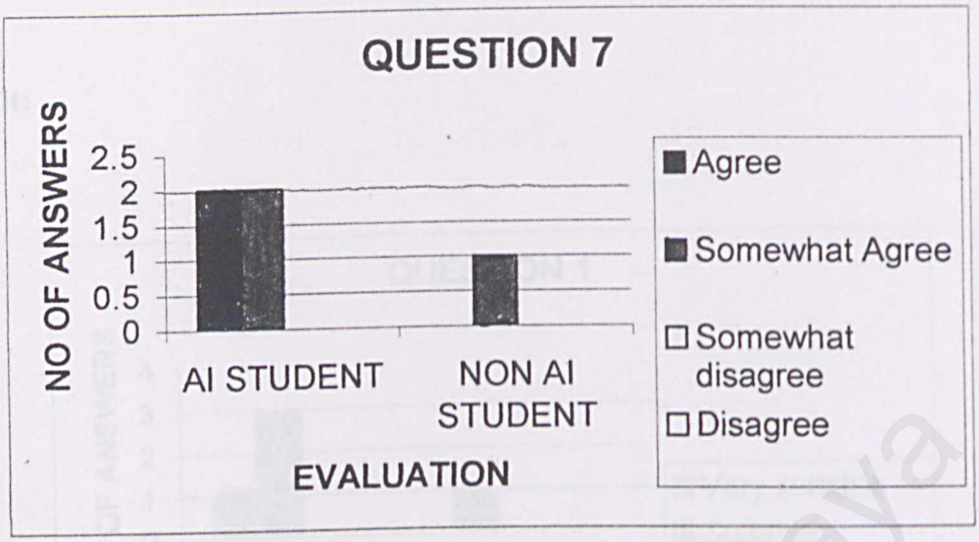


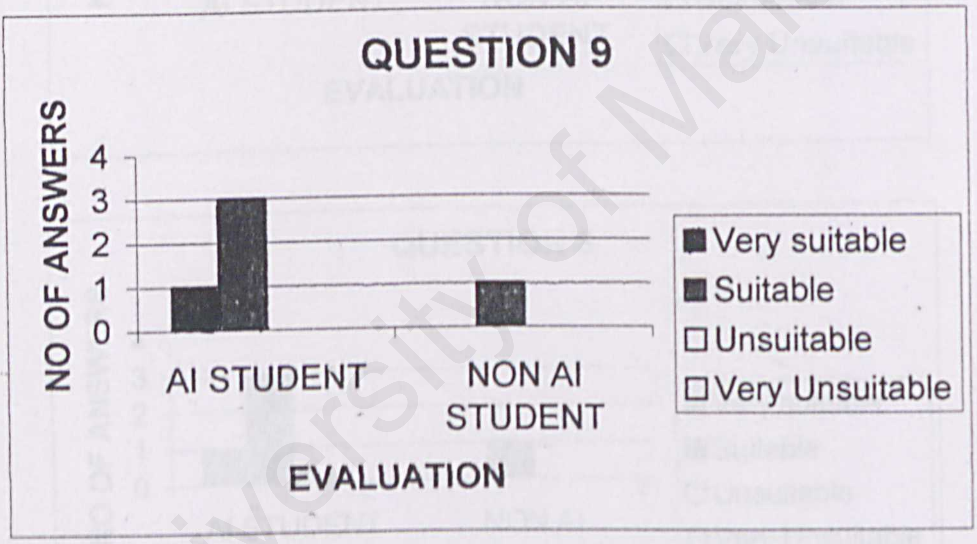


Most of the respondent agreed that the layout design and the color combination for the learning package are *very interactive*. It's the same thing with the instruction given in the both modules are *very clear* and can be followed by the students.

7.1.2 Navigation Evaluation

Result:

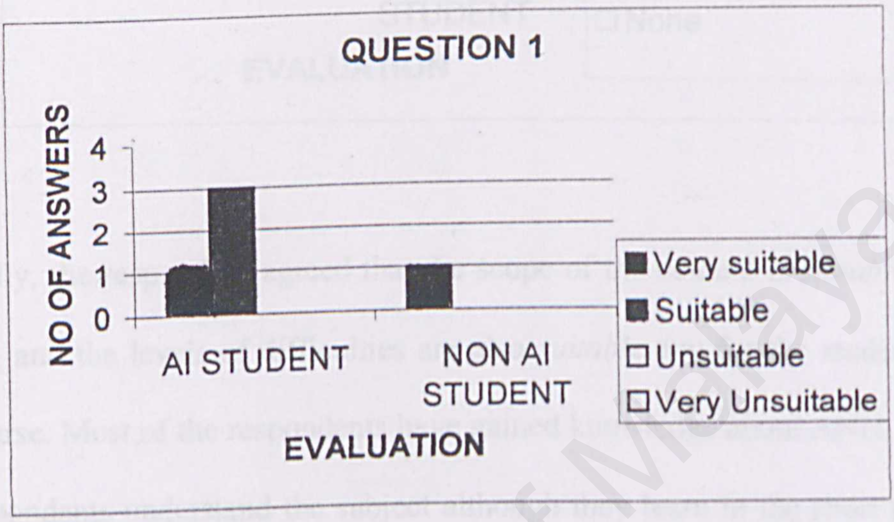


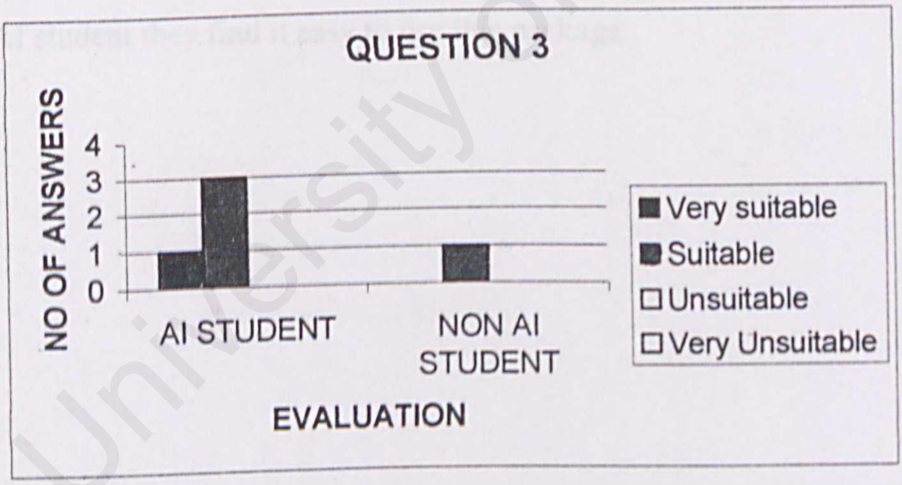


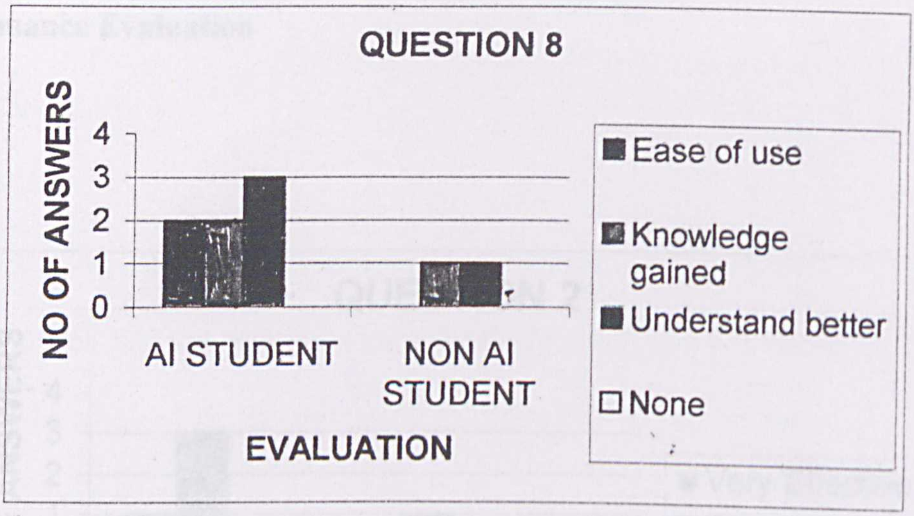
Some of the respondent agreed that for the navigation the user can navigate through the pages without any assistance. The relationship between the two modules are also *suitable* and they commented that the linking between two modules not in the proper ways.

7.1.3 Help Evaluation

Result:



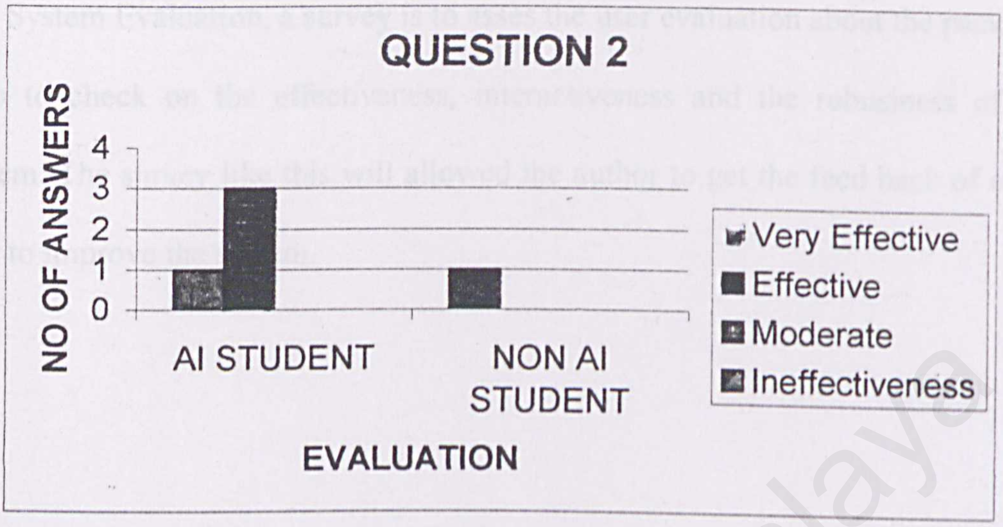


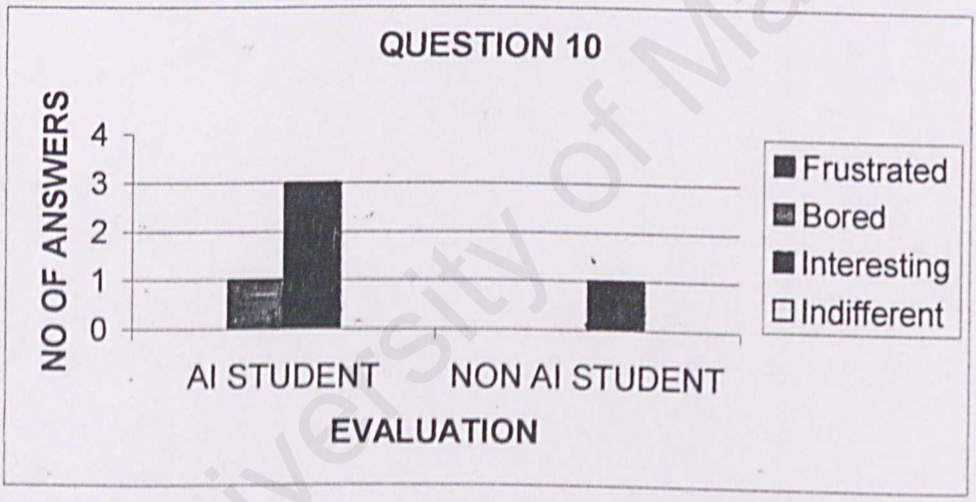


Generally, the respondent agreed that the scope of the content is *suitable* for the student, and the levels of difficulties are also *suitable* any for the students level and course. Most of the respondents have gained knowledge about ANN, some of the respondents understand the subject although they learn in the short time and for the AI student they find it easy to use this package.

7.1.4 Performance Evaluation

Result:





It is clear that the AI students and non-AI student find it effective to use this system after they have tested it for the very first time. For the non AI student, they find it the topic is very interesting and easier to learn in this way. While AI students concluded that they would like to use this package for learning purposes as it provides the information and the demonstration they wanted to learn.

7.1.5 Summary

The System Evaluation, a survey is to asses the user evaluation about the package. Also to check on the effectiveness, interactiveness and the robustness of the system. The survey like this will allowed the author to get the feed back of other user to improve the system.

CHAPTER EIGHT : DISCUSSION AND CONCLUSIONS

CHAPTER 8

8.0 DISCUSSION AND CONCLUSION

8.1 Problems Encountered and Recommended Solutions

Some problem was encountered in the development of the Interactive Learning Package for ANN. These problems, together with the approaches taken to solve them are documented into two separate sections:

- Project studies and analysis
- Project implementation and testing

8.1.1 Problem and Solution

Project studies and analysis in respect of window based learning system are needed in order to have a basic knowledge for the development of this system. The problems encountered during these phases are explained below.

- **Difficulties in choosing a development technology, programming and tools**

Our technology is leading toward a new era. There are many software tools available to develop the system as stated in a chapter before. Choosing the suitable technology and tools was a critical process as all tools have their strength and weakness .In addition the availability of the required tool for the development is also being considered.

In order to solve this problem, seeking advices and views from senior previous project, ask for supervisor advises also the course mates.

- **Lack of ANN knowledge**

Development of the powerful system requires a deep understanding in the ANN. During the selection of the suitable stuff and notes for the demo, many problems encountered because of the lack of the ANN knowledge. To overcome this problem, other ANN books besides recommended books by Mr. Woo had been studied.

- **Determines the scope of the system**

It is possible to build the full scale complete system within the time frame given. This system consists five main topics and it is huge system. Inexperience with the learning package, the development caused the difficulty in determining the scope of the system. Many discussions were held with project supervisor to continue the scope that had been defined. An analysis of the current system was done.

8.2 System Strengths

- **Easy to use interface**

The Learning Package is designed with two different interfaces that are categorized to Learning Module and Demonstration Module. The colors of the background are interactive and the user can select the main menu easier because the titles are stated in the nice and big font.

- **Tutorial answers**

In the tutorial session, when the user answer the question they will be shown the correct answer automatically. The users will know their level of selecting the answer. Sounds were added to make user feel interesting while answering the question.

- **Interactive Demonstrations**

The user is provided with the interactive demo because the user can enter the input and see the changes when they put their random input meaning .The demo only come out when the user click on the Demo button and it will appear without wait a long time.

8.4 System Limitations

Every system should have their limitations. For the Interactive Learning Package for ANN the limitations are:

- **Linking to Demonstration Module**

  The Learning Modules are only have linking at the main menu. It should have in every demonstration pages.

- **Limited tutorial question**

  The tutorial provided are very less because only five questions. The student cannot evaluate their performances very well. Also with the format of the questions. It must be variety.

- **Limited demonstration**

Now the system only got six demos and for the topic Perceptron and Feed Forward doesn’t have the demo but only the solved problem. For the next time the two topic must have their own demos.

8.3 Future Enhancements

The tutorials must be added from five or maybe to 20 questions. All the questions have divided into more than one design such as like subjective, and false or wrong and etc.So that the student can try and tested widely. Also about the linking from learning module to the demonstration module is should have done subtopic by subtopic. The interfaces also must be the same so that the user cannot get confused. This will allow the user to see the demonstration from the learning module without delay of browsing through all the pages in the demonstration module from the starting. Although it is the interactive learning package but the animations in this system are less than user expected and for the next time the designer must have to add in order to create more interactive and interesting learning.

8.4 Summary

This is the last system to encounter. The entire problems are ending with the solution. Its discussed all about the problem faced such as identifying problems, system strengths and limitations and future enhancements for the system. Mostly the system will having the problem at beginning. The discussion and conclusion chapter concludes with few suggestions for future enhancements that can be made on the learning package.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1]. http://www.mindspark.com.my/

[2].http://www.psychologist.uni.triei.de:8000/projects/ELM/elmart.html

[3].Marilyn McCord Nelson,W.T Illington,(1991)*A Practical Guide to Neural Net.* p.43

[4].Marilyn McCord Nelson,W.T Illington,(1991)*A Practical Guide to Neural Net.* p.37

[5]. http://www.dacs.dtic.mil/techs/neural/neural_ToC.html

[6]. Marilyn McCord Nelson,W.T Illington,(1991)*A Practical Guide to Neural Net.* p.43

[7].Marilyn McCord Nelson,W.T Illington,(1991)*A Practical Guide to Neural Net.* p.44

[8] Marilyn McCord Nelson,W.T Illington,(1991)*A Practical Guide to Neural Net.* p.47

[9].Marilyn McCord Nelson,W.T Illington,(1991)*A Practical Guide to Neural Net* p.48

[10]. http://www.dacs.dtic.mil/techs/neural/neural_ToC.html

[11]. Simon Haykin. (1999). *Neural Networks: A Comprehensive Foundation.* 2nd ed. Prentice Hall.

[12]. www-cse.stanford.edu/class/sophomore-college/projects-00/neuralnet

[13]. Fredric M.Ham,Ivica Kostanic(2001).*Principles of Neurocomputing for Science and Engineering*.McGraw Hills.p.165

[14]. Fredric M.Ham,Ivica Kostanic(2001).*Principles of Neurocomputing for Science and Engineering*.McGraw Hills.p.198

[15]. Fredric M.Ham,Ivica Kostanic(2001).*Principles of Neurocomputing for Science and Engineering*.McGraw Hills.p.120

[16]. Ian Sommerville(2000).Software Engineering.6th ed.Addison-Wesley.p.45

[17]. http://www.mathworks.com/products/matlab/

USER MANUAL

APPENDIX

USER MANUAL

Getting started

Part 1: Installation Guide

Part 2: Main Menu

Part 3: Main Topics

Part 4: Tutorial

Part 5: Glossary

Part 6: Help

Part 7: Demonstartions

Part 1: Installation Guide

There is an application to install Interactive Learning Package for Artificial Neural Network. The hardware and software requirements are the important terms for the installation.

Hardware and software requirement

- 32 MB RAM or more
- At least Pentium MMX 166 MB processor
- Minimum of 100 MB free hard disk space
- Other standard peripherals including keyboard, mouse, speaker.
- Operating system Windows 95/98/200/ME/NT/XP

The installation steps involved are listed below.

Installation of Interactive Learning Package (Demonstration Modules)

- Insert the CD ROM into CD ROM driver.
- Go to My Computer
  - \D:
  - Search for file named Demo.

- Click on the DEMO.swf (flash movie)
- Main menu is then displayed.

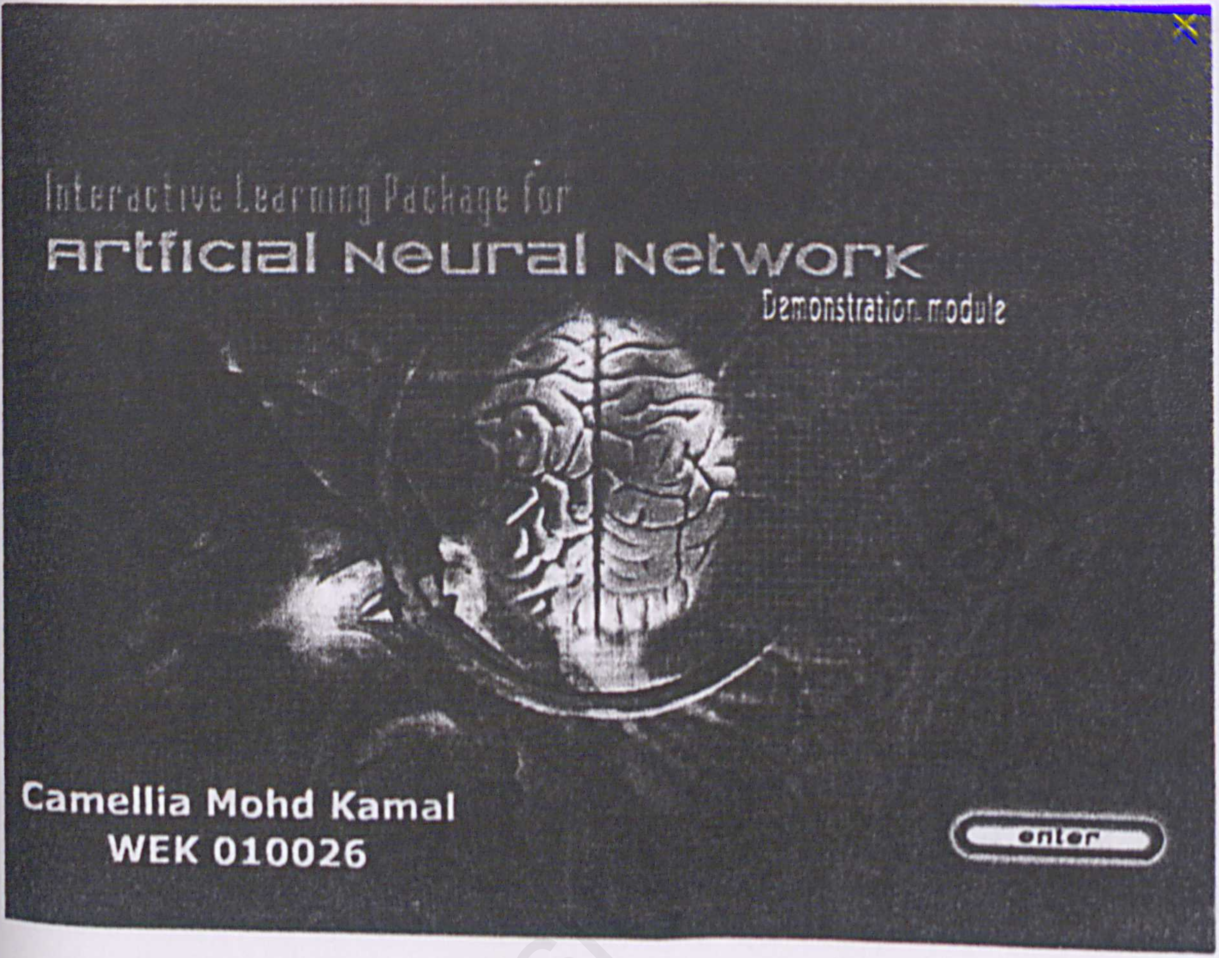


Figure 1: Main Menu

- User have to click the enter button to get into the main menu
- If the user want to quit the system go to the top page , click the ☒

Part 2: Main Menu

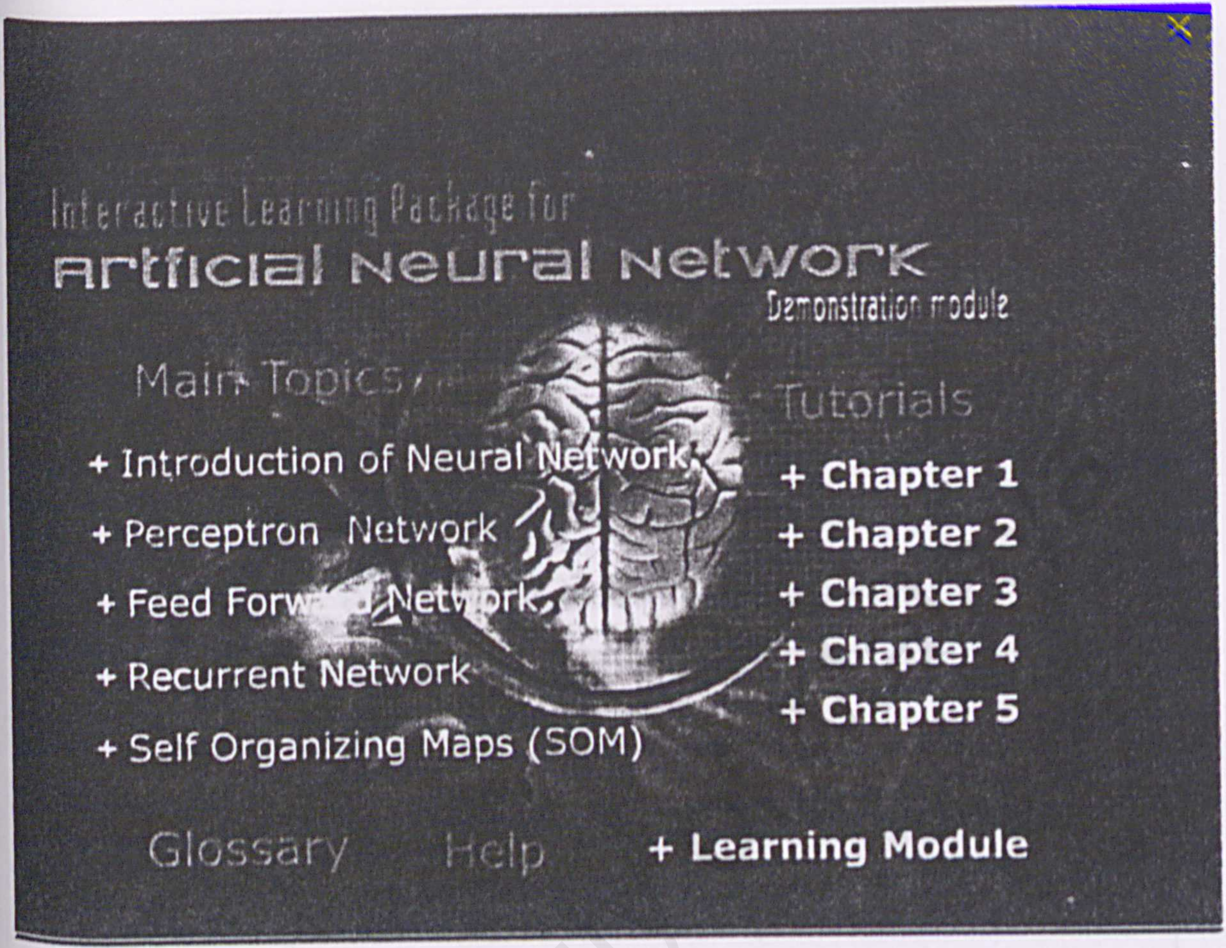
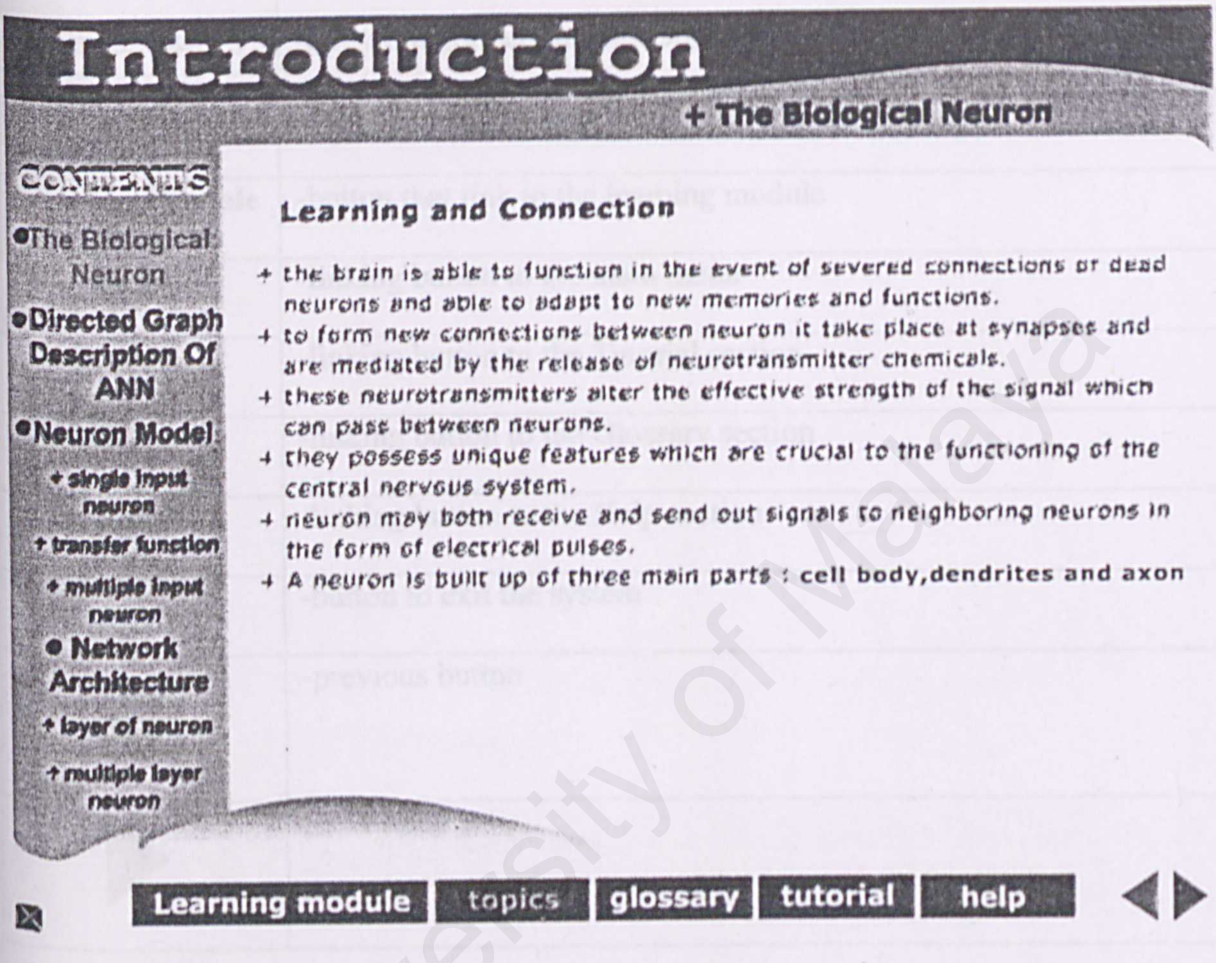


Figure 2: Main Menu Interface

- On the main menu, select the section required by clicking on the topic.
- The entire topics there are the button.
- Click on the quit on the top of the page button exit the module
- The four main section linked in the main menu are listed below:

Section	***Descriptions***
Main Topics	-Contains four titles: *Introduction of ANN, Perceptron, Feed Forward, Recurrent, and SOM.* -Each title is connected to the main contents.
Tutorials	-Contains four chapters of tutorial for each topic. -Each title provide links to the selected tutorial
Glossary	-Contains all the meaning of the selected word. -The title is connected to the glossary contents.
Help	-Contains all help manual :*Main menu,Tutorial,Glossary.* -The title is connected to the glossary contents
Learning Module	-The title is connected to the learning module part. - Contains four titles: *Introduction of ANN, Perceptron, Feed Forward, Recurrent, and SOM.*

Part 3: Main Topics



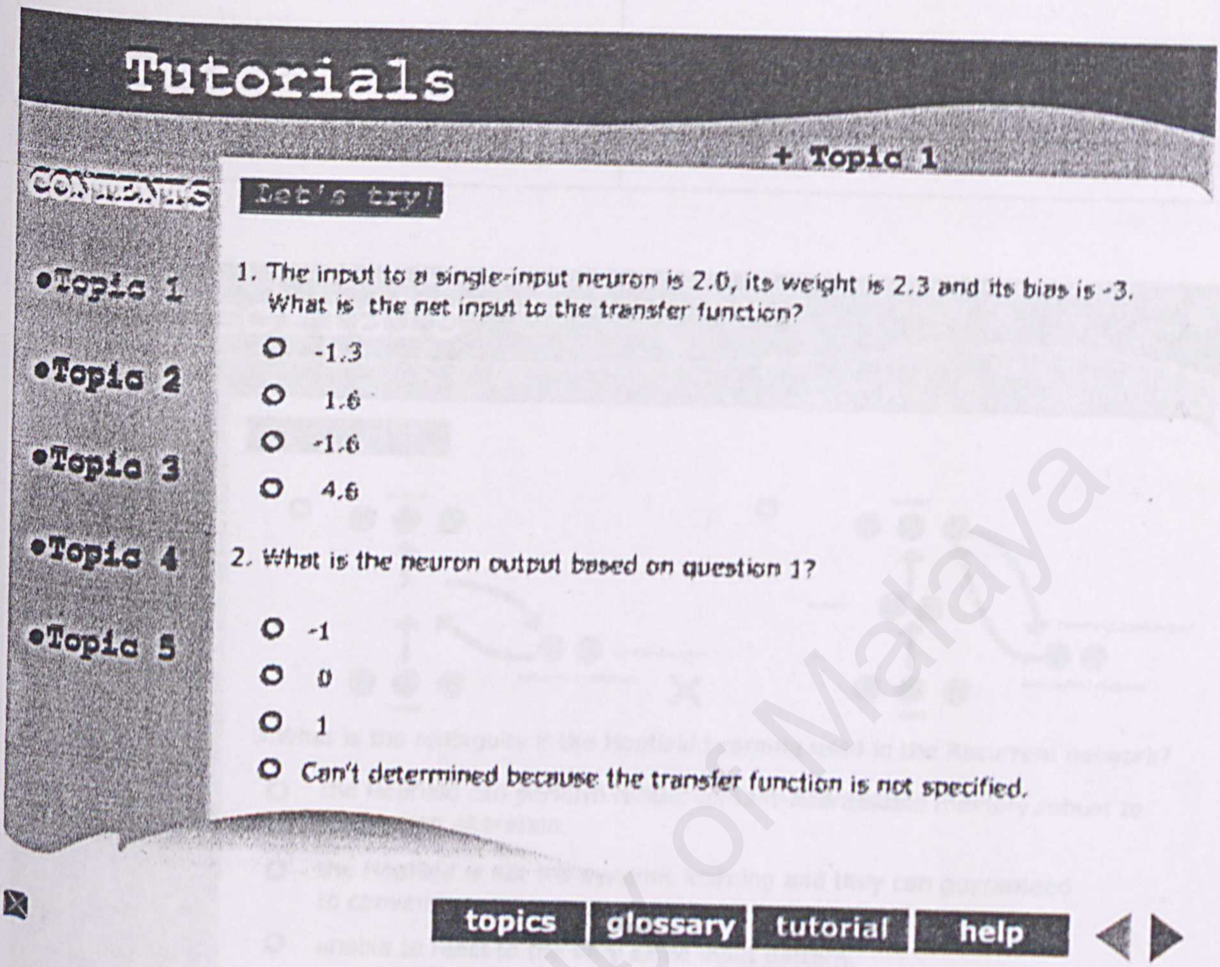
Learning and Connection

+ the brain is able to function in the event of severed connections or dead neurons and able to adapt to new memories and functions.
+ to form new connections between neuron it take place at synapses and are mediated by the release of neurotransmitter chemicals.
+ these neurotransmitters alter the effective strength of the signal which can pass between neurons.
+ they possess unique features which are crucial to the functioning of the central nervous system.
+ neuron may both receive and send out signals to neighboring neurons in the form of electrical pulses.
+ A neuron is built up of three main parts : cell body,dendrites and axon

- This is the example of Topic 1: Introduction of ANN.
- The table below is the description all the function provided in Topic 1.

Section	*Descriptions*
Topic Title	-title of major certain main topic
Content	-contains all the subtopic provided for each topic
Learning Module	-button that link to the learning module
Topics	-linking button to the main menu
Tutorial	-linking button to the Tutorial section
Glossary	-linking button to the Glossary section
Help	-linking button to the Help section
Quit	-button to exit the system
◀	-previous button
▶	-next button

Part 4: Tutorial



1. The input to a single-input neuron is 2.0, its weight is 2.3 and its bias is -3. What is the net input to the transfer function?

○ -1.3

○ 1.6

○ -1.6

○ 4.6

2. What is the neuron output based on question 1?

○ -1

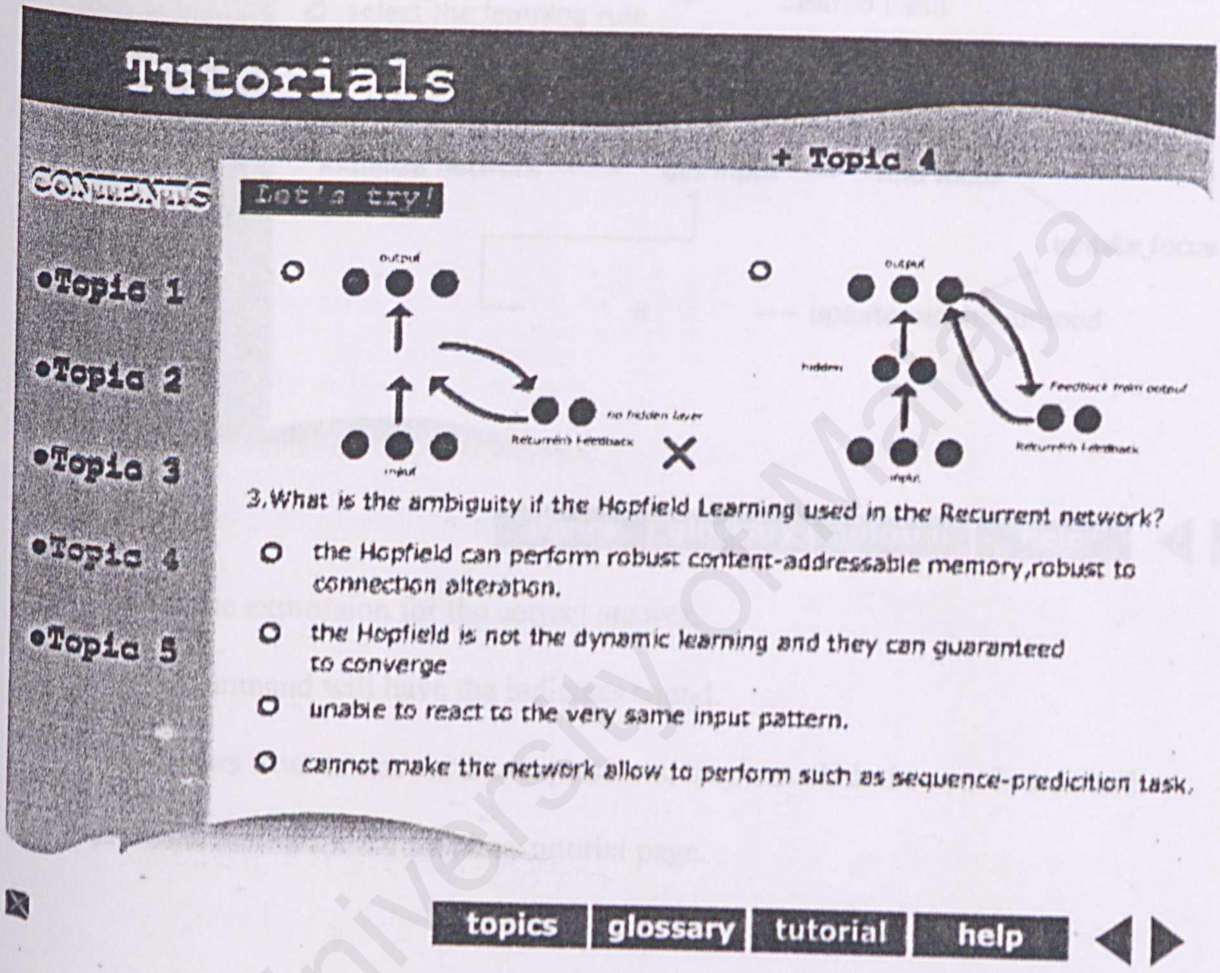
○ 0

○ 1

○ Can't determined because the transfer function is not specified.

- This is the example of tutorial in Topic 5:Self Organizing Maps
- Each topic will have only five questions.
- User must select only one correct answer because the answer will be given immediately when the user pick the answer.
- The table below described the important function on the Tutorial page.

Section	*Descriptions*
Contents	-Contain all five Tutorial topics: *Topic 1,Topic 2,Topic 3,Topic 4,Topic 5*



○ ○ ✕

3.What is the ambiguity if the Hopfield Learning used in the Recurrent network?

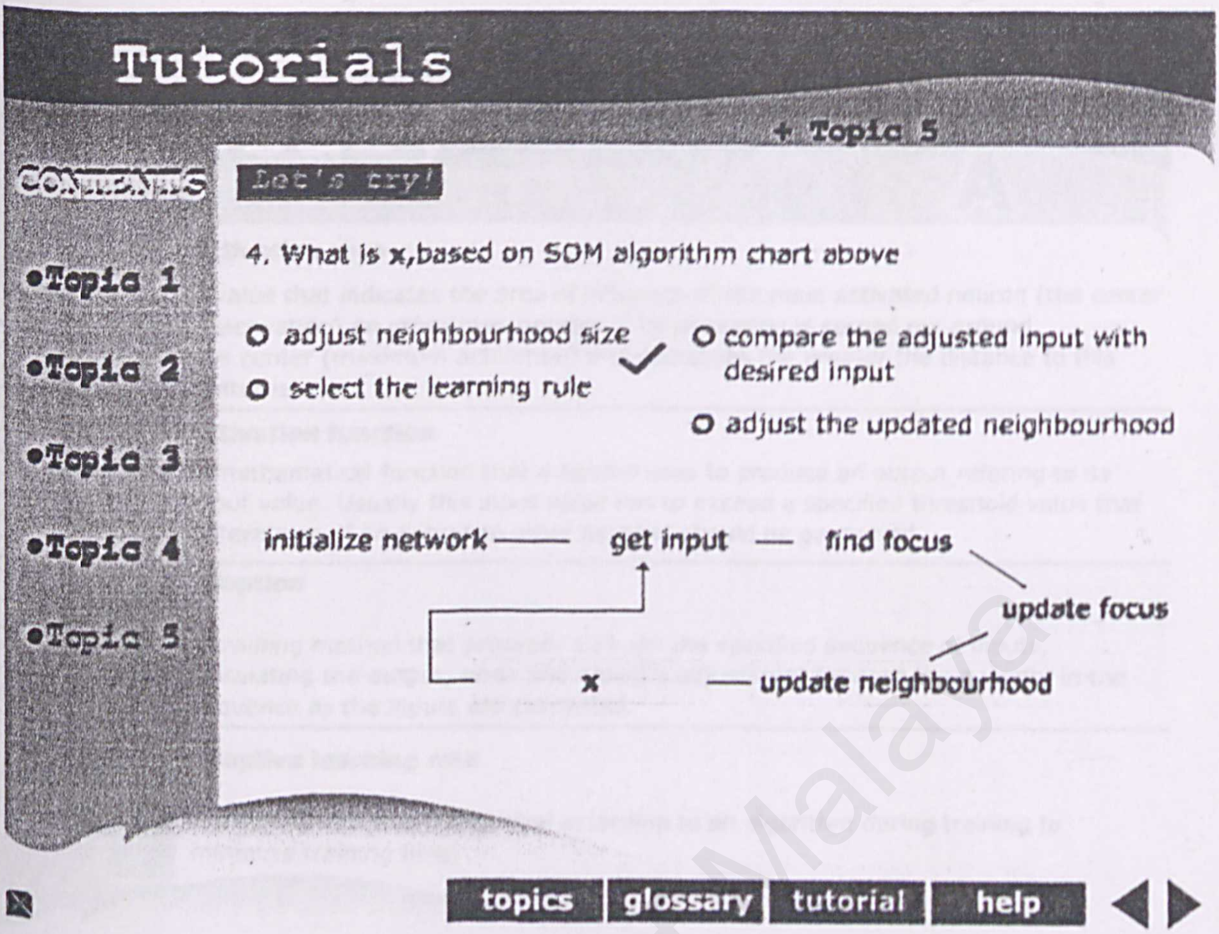
○ the Hopfield can perform robust content-addressable memory,robust to connection alteration.

○ the Hopfield is not the dynamic learning and they can guaranteed to converge

○ unable to react to the very same input pattern.

○ cannot make the network allow to perform such as sequence-prediciton task.

- This is the expression for the incorrect answer



Tutorials

+ Topic 5

CONTENTS

Let's try!

4. What is x,based on SOM algorithm chart above

○ adjust neighbourhood size ✓

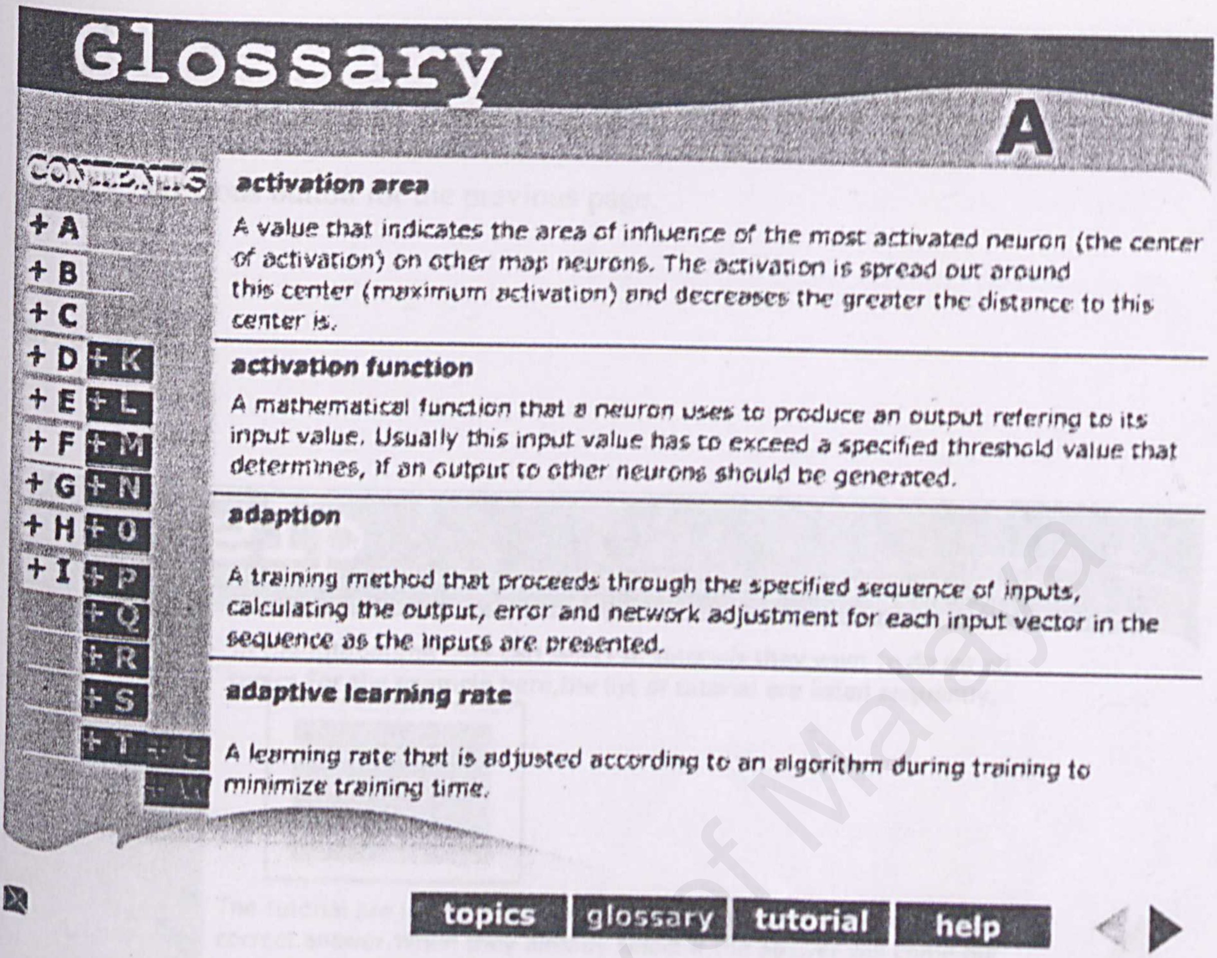
○ select the learning rule

○ compare the adjusted input with desired input

○ adjust the updated neighbourhood

- This is the expression for the correct answer.
- Every command will have the indicates sound.
- If the users want to answer more questions, they can click the next button and previous button for the previous tutorial page.

Part 5: Glossary



activation area

A value that indicates the area of influence of the most activated neuron (the center of activation) on other map neurons. The activation is spread out around this center (maximum activation) and decreases the greater the distance to this center is.

activation function

A mathematical function that a neuron uses to produce an output refering to its input value. Usually this input value has to exceed a specified threshold value that determines, if an output to other neurons should be generated.

adaption

A training method that proceeds through the specified sequence of inputs, calculating the output, error and network adjustment for each input vector in the sequence as the inputs are presented.

adaptive learning rate

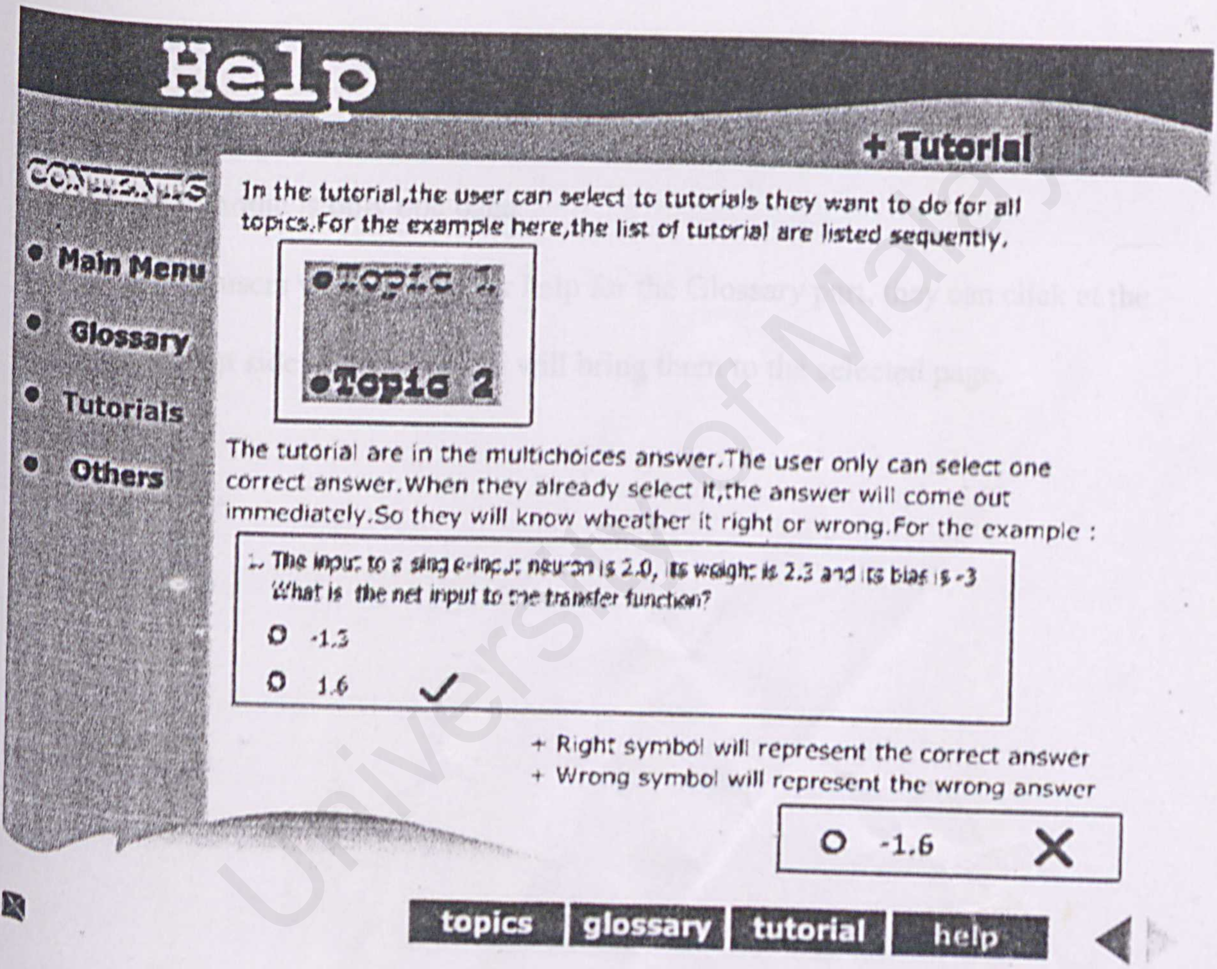
A learning rate that is adjusted according to an algorithm during training to minimize training time.

- This is the example of Glossary interface for alphabet A
- Every alphabet for every page will have a link to certain pages.
- User must select which page they want to go with clicking the chosen word.
- The table above is the description about the important function in the Glossary.

Section	*Descriptions*
Content	-Contains all 25 alphabets starting from A to Z. -The same design for every Glossary page.

- This is the example if the users want to know the meaning of Learning.
- They selected the L alphabet and then the system will bring them to L pages.
- If the users want to go another L pages, they can click the next button and previous button for the previous page.

Part 6: Help



In the tutorial,the user can select to tutorials they want to do for all topics.For the example here,the list of tutorial are listed sequently.

The tutorial are in the multichoices answer.The user only can select one correct answer.When they already select it,the answer will come out immediately.So they will know wheather it right or wrong.For the example :

1. The input to a single-input neuron is 2.0, its weight is 2.3 and its bias is -3
What is the net input to the transfer function?

○ -1.3

○ 1.6 ✓

+ Right symbol will represent the correct answer
+ Wrong symbol will represent the wrong answer

○ -1.6 ✕

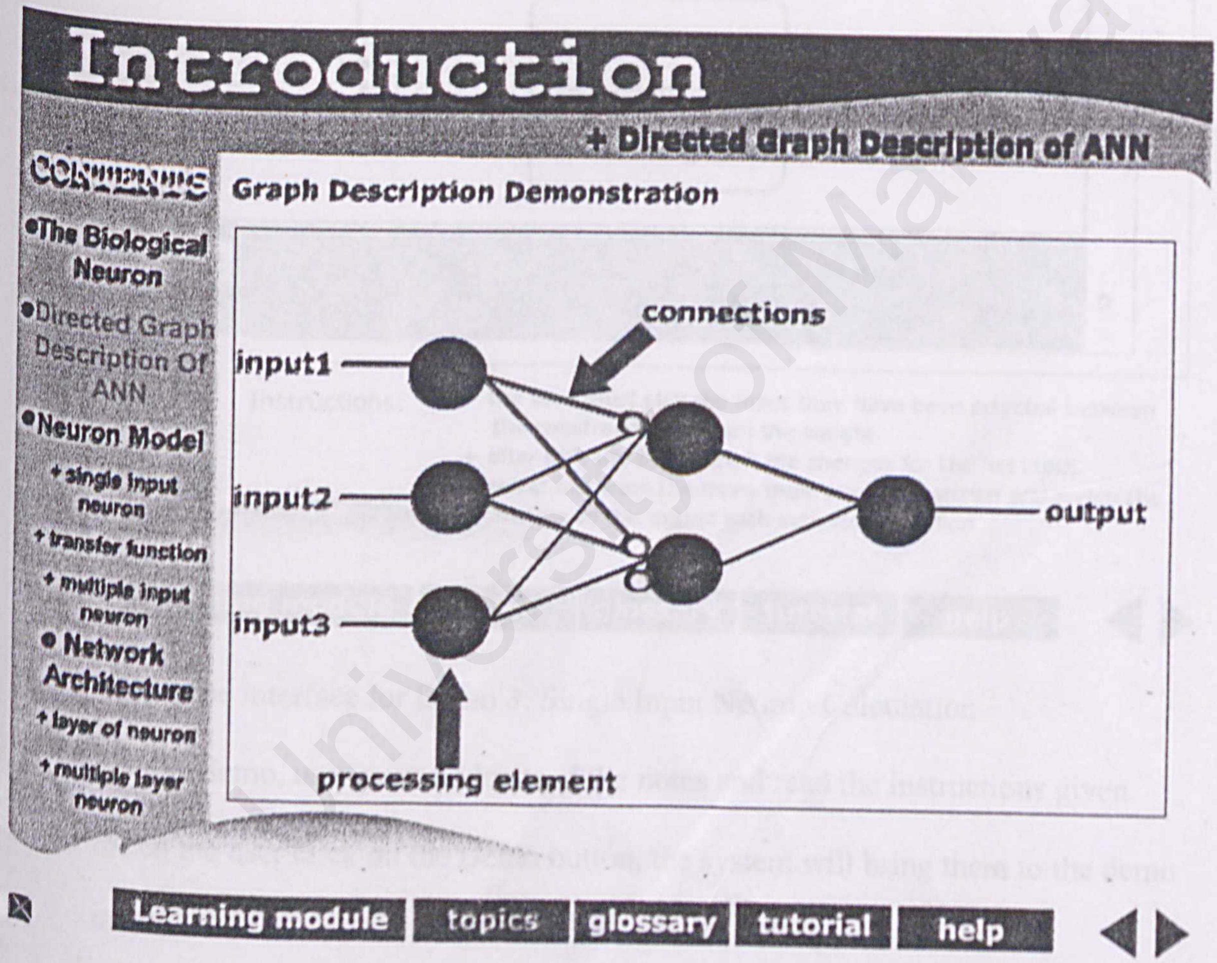
- This is the interface of Help part.
- Help will assist the user who lost while using this system
- The table below described the important function on the Help page.

Section	***Descriptions***
Contents	-Contains only four main topics: Main Menu, Tutorial, Glossary and Help itself. -The same design for every topic Help page.

- For Help, the next and previous buttons are not function because the page for every topic is only one page.
- If the users want to seek for help for the Glossary part, they can click at the content side and the system will bring them to the selected page.

- This is the interface for Demo 1: Simultenous Process Between Two Neuron
- To experience it, user can go to the blue bullet there and every time they took the mouse roll over, the name of that part will appear.
- User must try one by one for every bullet.
- User can leave the demo and go to the next page because the demo will continually playing.

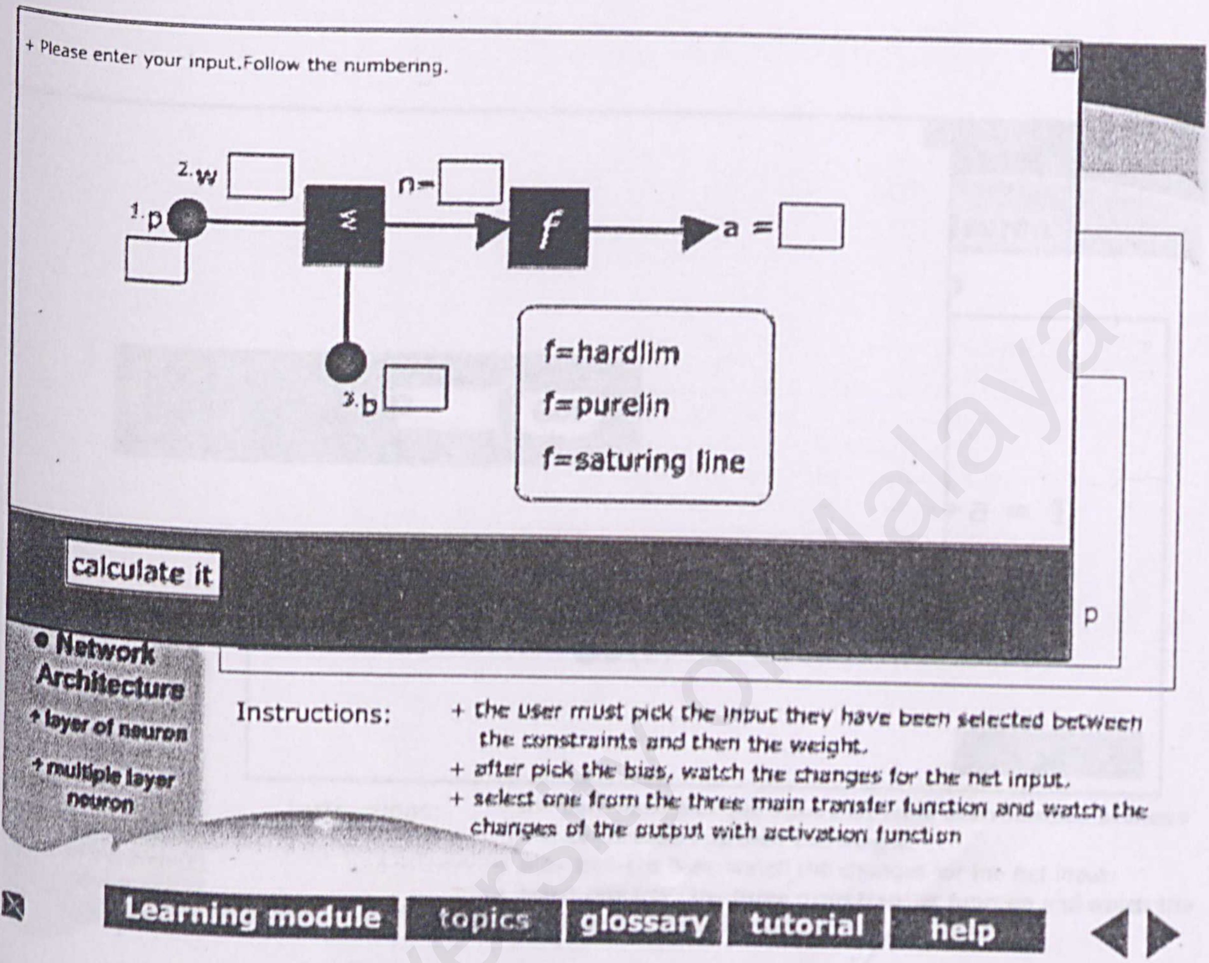
Demonstration 2: Directed Graph



- This is the interface for Demo 2: Directed Graph
- The user only can look for the demo and try to relate with the previous information.

- User can leave the demo and go to the next page because the demo wills continually playing.

Demonstration 3: Single Input Neuron Calculation



Instructions:

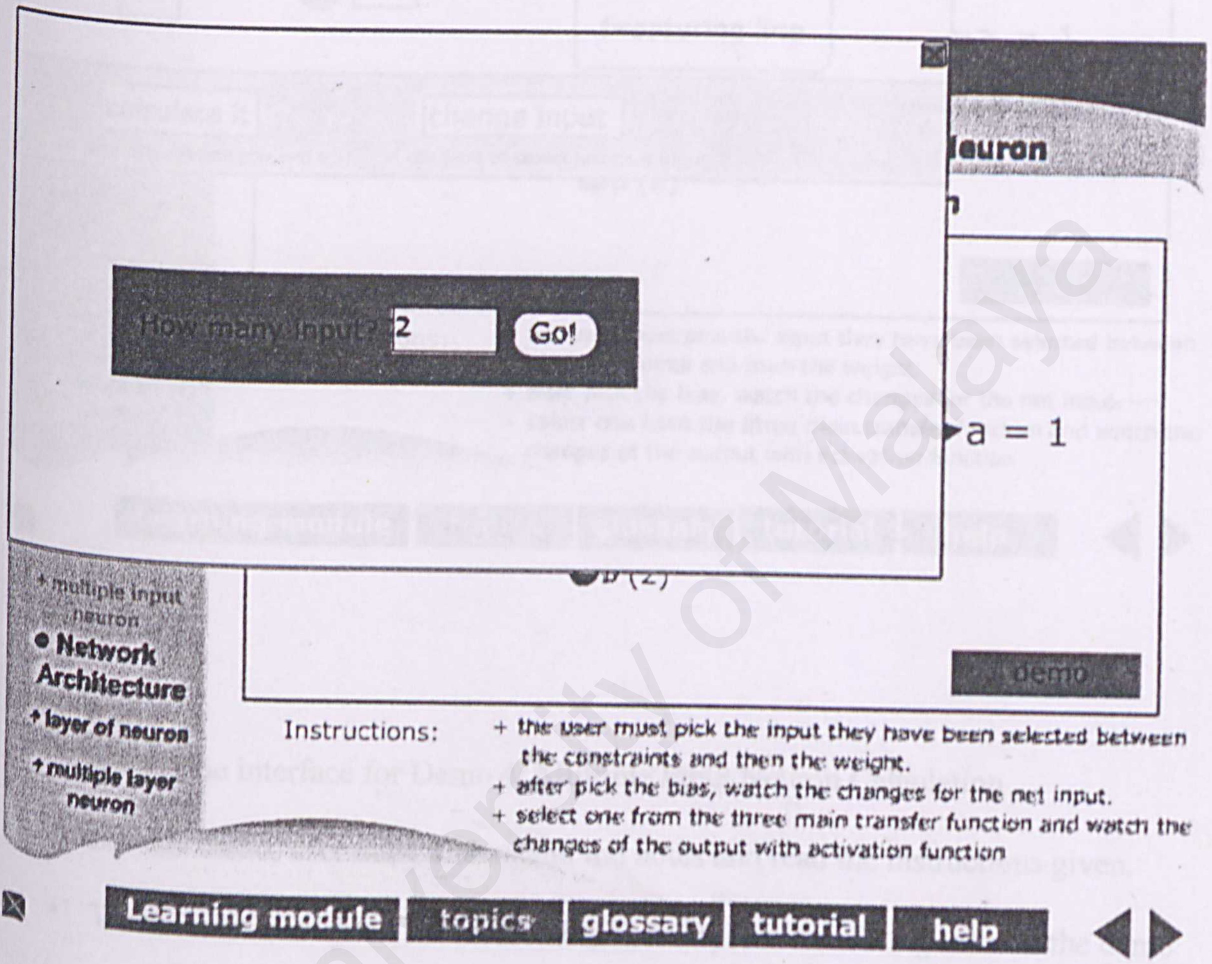
+ the user must pick the input they have been selected between the constraints and then the weight.
+ after pick the bias, watch the changes for the net input.
+ select one from the three main transfer function and watch the changes of the output with activation function

- This is the interface for Demo 3: Single Input Neuron Calculation
- For this demo, user must understand the notes and read the instructions given.
- When the user click on the Demo button, the system will bring them to the demo movie.
- If they want to quit, click quit button at the top.
- At the demo movie will have the command button ***calculate it***, after user enter the input follow the numbering they must click at this button to calculate the

calculation.

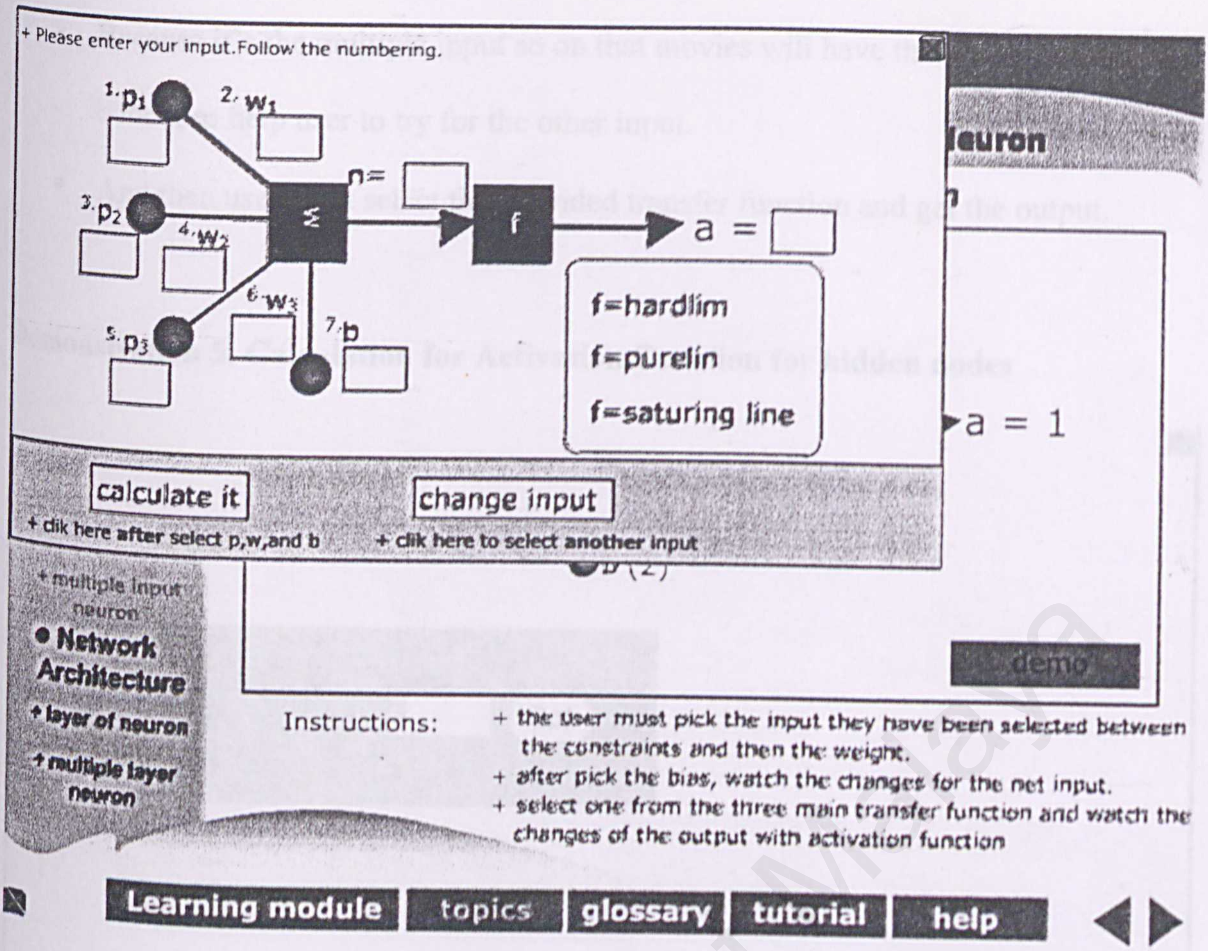
- And then user must select the provided transfer function and get the output.

Demonstration 4: Multiple Input Neuron Calculation



Instructions:

+ the user must pick the input they have been selected between the constraints and then the weight.
+ after pick the bias, watch the changes for the net input.
+ select one from the three main transfer function and watch the changes of the output with activation function



+ Please enter your input.Follow the numbering.

+ clik here after select p,w,and b

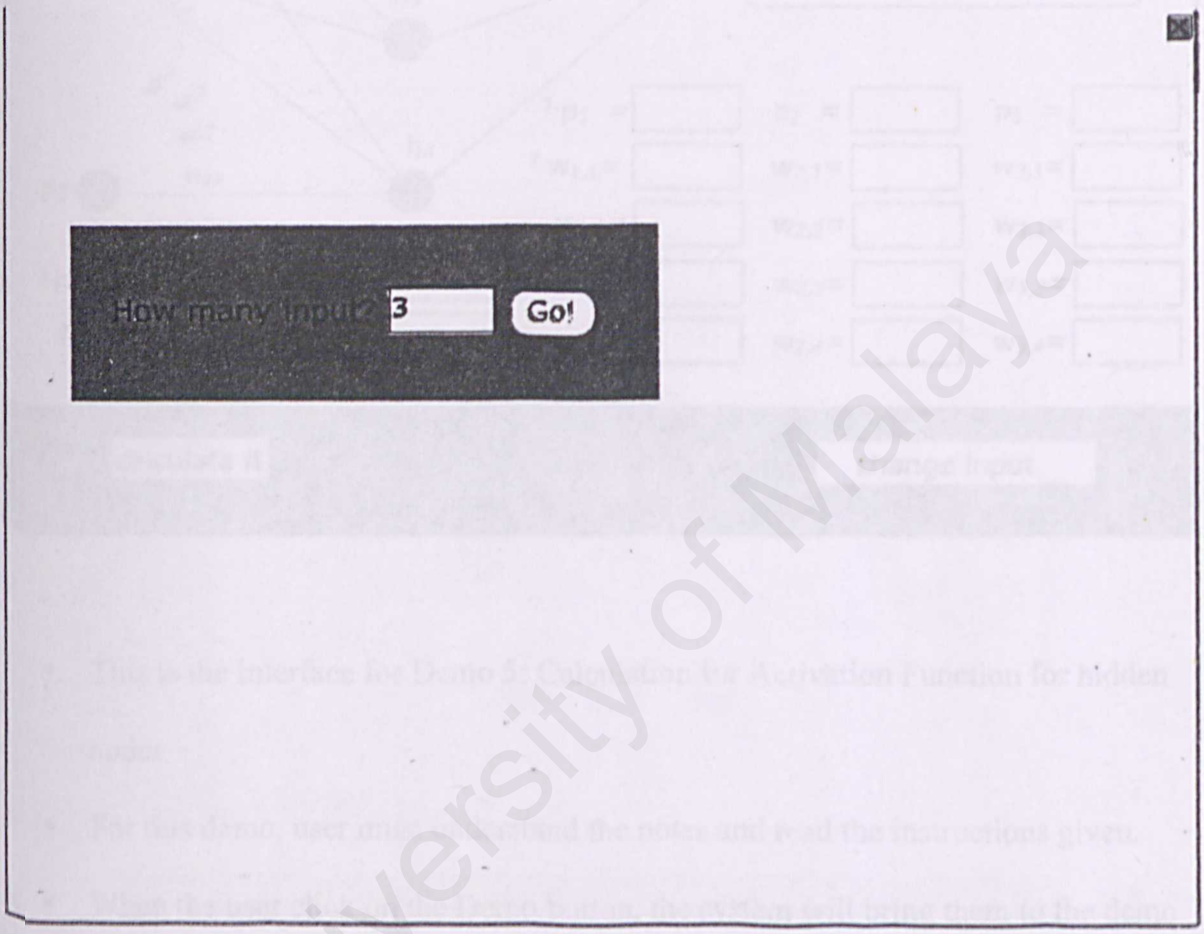
+ clik here to select another input

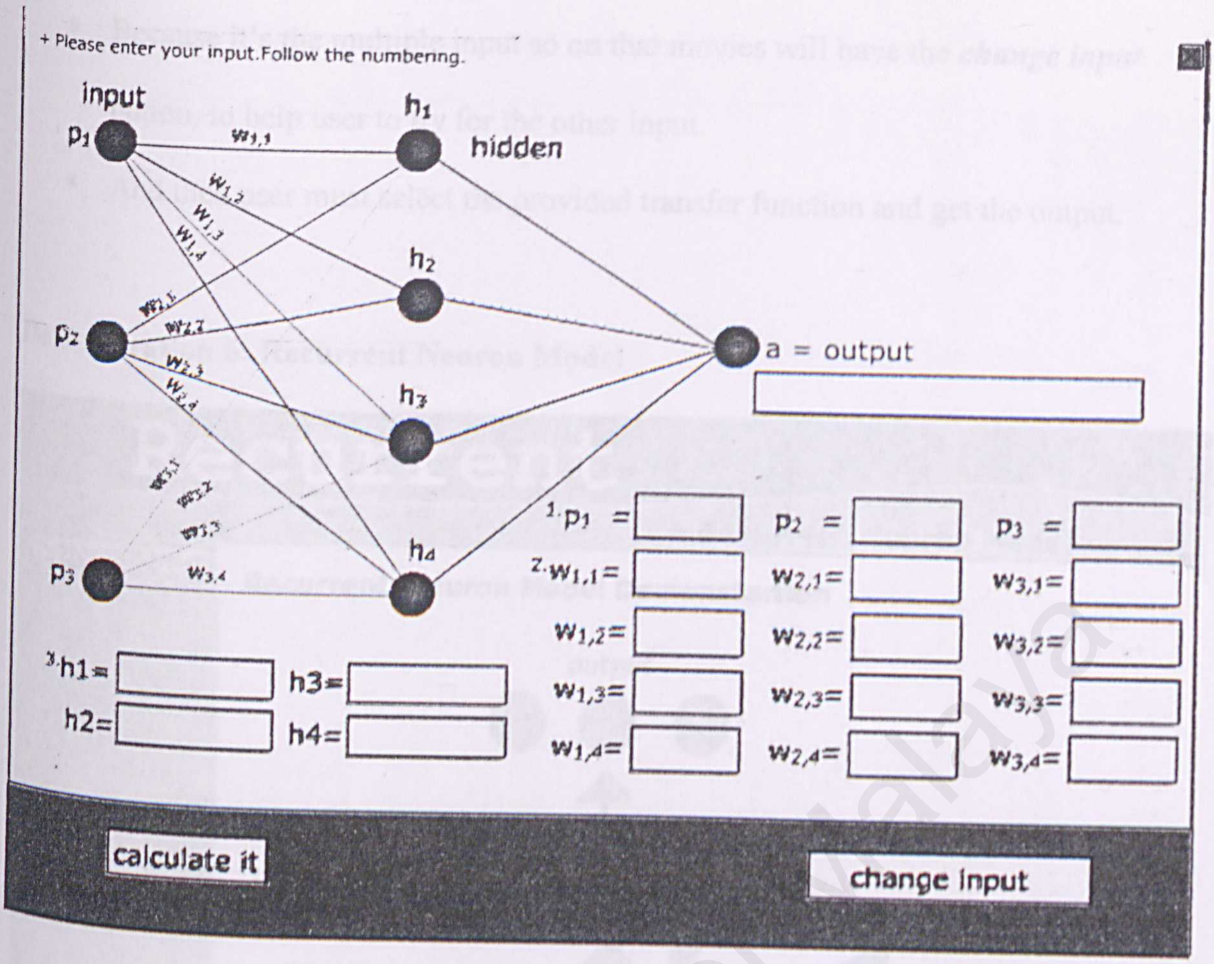
Instructions:
+ the user must pick the input they have been selected between the constraints and then the weight.
+ after pick the bias, watch the changes for the net input.
+ select one from the three main transfer function and watch the changes of the output with activation function

- This is the interface for Demo 4: Multiple Input Neuron Calculation
- For this demo, user must understand the notes and read the instructions given.
- When the user click on the Demo button, the system will bring them to the demo movie.
- If they want to quit, click quit button at the top.
- At the demo movie will have the command button ***calculate it***, after user enter the input follow the numbering they must click at this button to calculate the calculation.

- Because it's the multiple input so on that movies will have the ***change input*** button, to help user to try for the other input.
- And then user must select the provided transfer function and get the output.

Demonstration 5: Calculation for Activation Function for hidden nodes

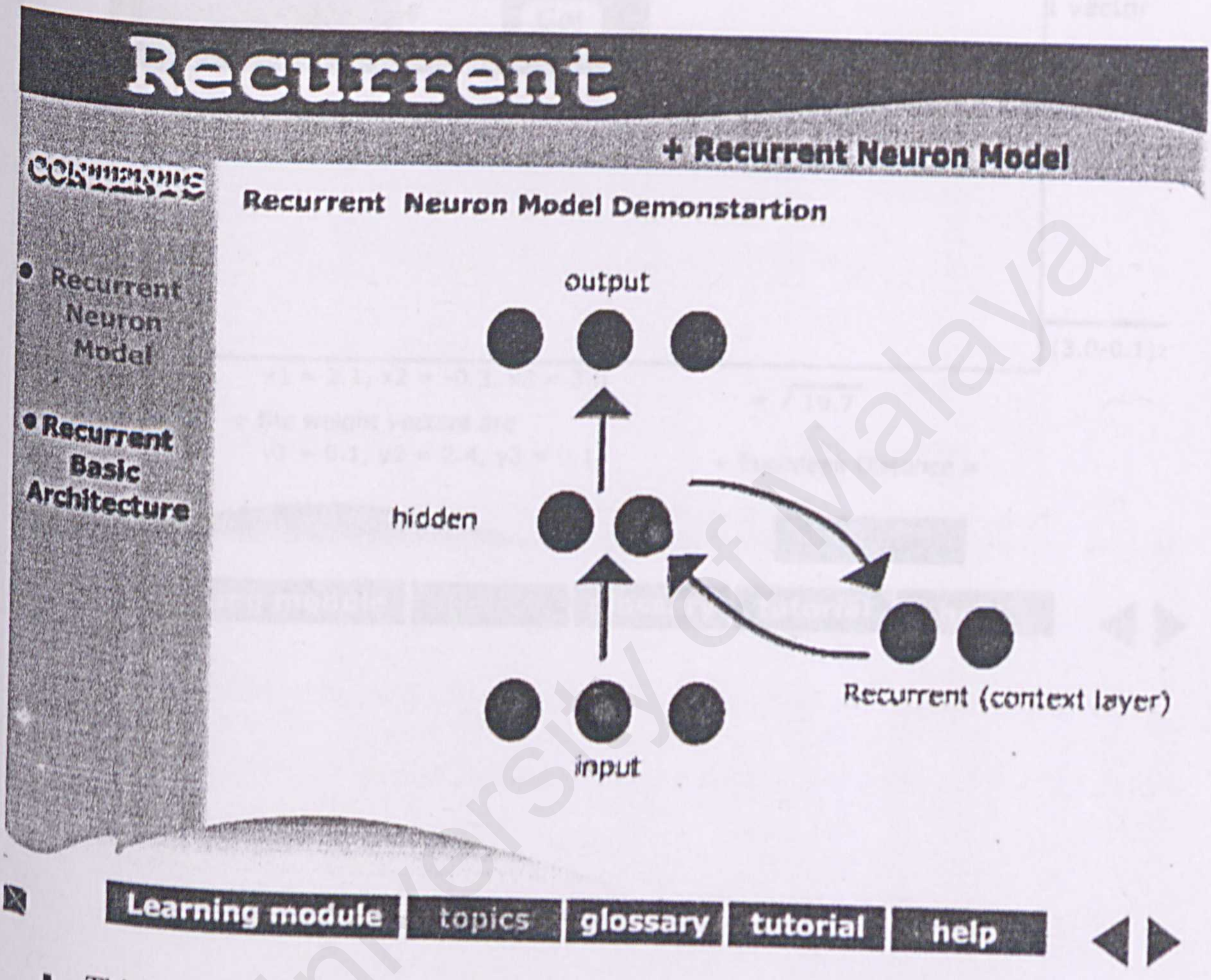



+ Please enter your input.Follow the numbering.

- This is the interface for Demo 5: Calculation for Activation Function for hidden nodes
- For this demo, user must understand the notes and read the instructions given.
- When the user click on the Demo button, the system will bring them to the demo movie.
- If they want to quit, click quit button at the top.
- At the demo movie will have the command button ***calculate it***, after user enter the input follow the numbering they must click at this button to calculate the calculation.

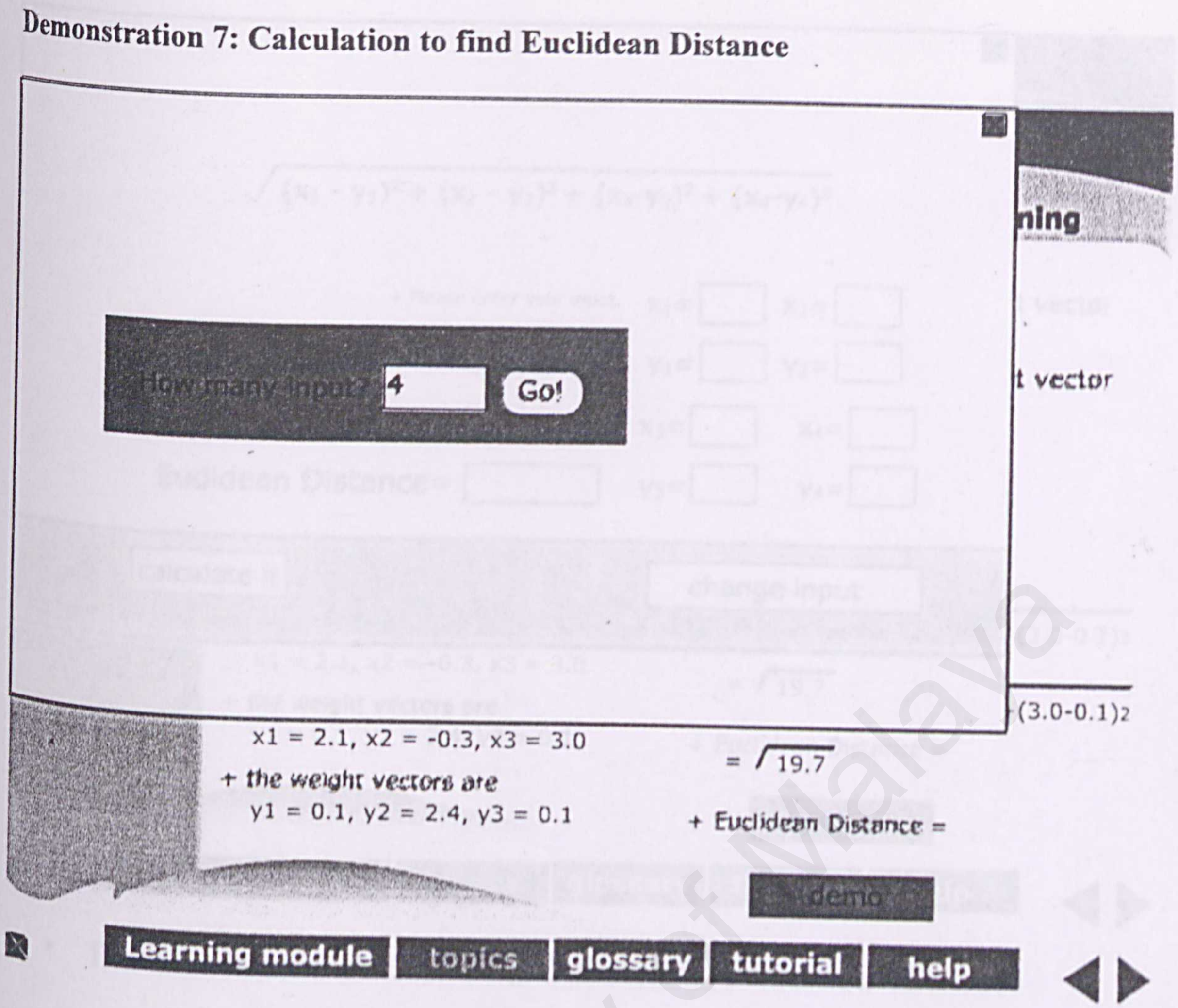
- Because it's the multiple input so on that movies will have the ***change input*** button, to help user to try for the other input.
- And then user must select the provided transfer function and get the output.

Demonstration 6: Recurrent Neuron Model



- This is the interface for Demo 6: Recurrent Neuron Model
- The user only can look for the demo and try to relate with the previous information.
- User can leave the demo and go to the next page because the demo wills continually playing.

Demonstration 7: Calculation to find Euclidean Distance



How many input? 4 Go!

x1 = 2.1, x2 = -0.3, x3 = 3.0

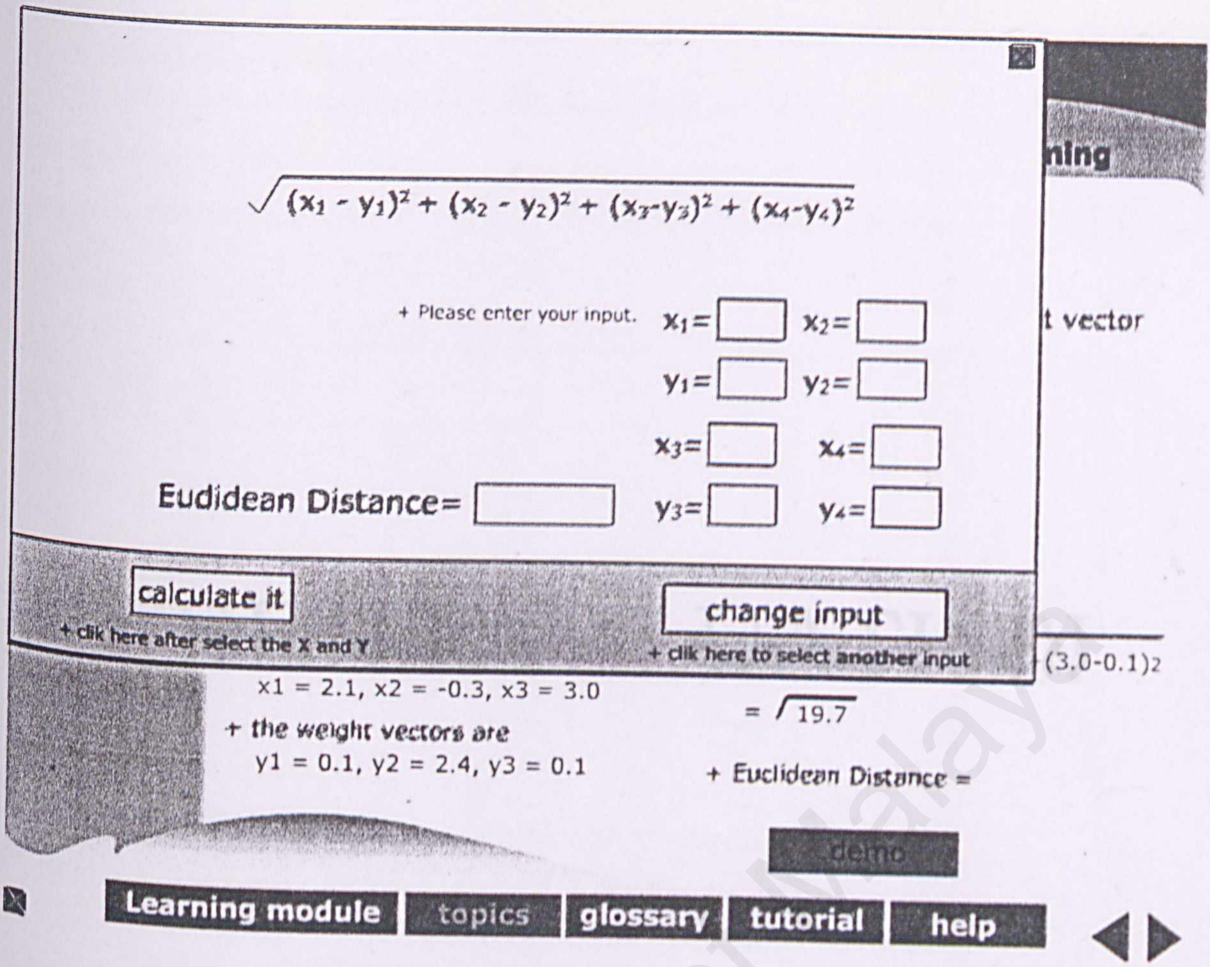
+ the weight vectors are

y1 = 0.1, y2 = 2.4, y3 = 0.1

$= \sqrt{19.7}$

+ Euclidean Distance =

- For this demo, user must understand the notes and read the instructions given.
- When the user click on the Demo button, the system will bring them to the demo movie.
- If they want to quit, click quit button at the top.
- At the demo movie, will have the command button *calculate it*, after user enter the input follow the numbering they must click at this button to calculate the calculation.
- Because it's the multiple input so on that movies will have the *change input* button, to help user to try for the other input.
- And then user must select the provided transfer function and get the output.



$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3-y_3)^2 + (x_4-y_4)^2}$$

+ Please enter your input. $x_1=$ □ $x_2=$ □

$y_1=$ □ $y_2=$ □

$x_3=$ □ $x_4=$ □

Eudidean Distance= □ $y_3=$ □ $y_4=$ □

x1 = 2.1, x2 = -0.3, x3 = 3.0

+ the weight vectors are

y1 = 0.1, y2 = 2.4, y3 = 0.1

$+(3.0-0.1)2$

$= \sqrt{19.7}$

+ Euclidean Distance =

- This is the interface for Demo 7: Calculation to find Euclidean Distance
- For this demo, user must understand the notes and read the instructions given.
- When the user click on the Demo button, the system will bring them to the demo movie.
- If they want to quit, click quit button at the top.
- At the demo movie will have the command button ***calculate it***, after user enter the input follow the numbering they must click at this button to calculate the calculation.
- Because it's the multiple input so on that movies will have the ***change input*** button, to help user to try for the other input.
- And then user must select the provided transfer function and get the output.

USER EVALUATION

User evaluation

The purposes of this survey are:

- To access the effectiveness and the general outlook of the **Interactive Learning Package for Artificial Neural Network**.
- To determines if this system meet the users need and expectations.

Please kindly take a moment off to complete this survey. Your cooperation is very much appreciated.

User's Detail

Course: SCIENCE BIOINFORMATICS
Semester: 02
Faculty: SAINS

1. The scope of the contents used in the package

☑ Very suitable
□ Suitable
□ Unsuitable
□ Very unsuitable

2. The effectiveness of the system (knowledge gained) after used the system.

☑ Very effectiveness
□ Effectiveness
□ Moderate
□ Ineffectiveness

3. Level of difficulty is appropriated for the stated study level.

□ Very suitable
☑ Suitable
□ Unsuitable
□ Very unsuitable

4. The color combination of the user interface design

□ Very suitable
☑ Suitable
□ Unsuitable
□ Very unsuitable

5. The Interactive Learning Package layout design

☑ Very interactive

□ Interactive

□ Unattractive

□ Very unattractive

6. The instruction of the demonstration

□ Very clear

☑ Clear

□ Unclear

□ Cannot follow

7. User will be able to use the demo independently

☑ Agree

□ Somewhat agree

□ Somewhat disagree

□ Disagree

8. Things you get from the all the demonstration

□ Ease of use

☑ Knowledge gained

☑ Understand better

□ None

If others, please specify: ______________________________

9. The relation between learning modules and the demonstration module.

□ Very suitable

☑ Suitable

□ Unsuitable

□ Very unsuitable

10. General attitudes

□ Frustrated

□ Bored

☑ Interesting

□ Indifferent

Suggestion (s) to improve this Interactive Learning Package

The layout should be same.

Thank you for your cooperation.

Prepared by:
Camellia Mohd Kamal WEK010026
A Deenalatha WEK010001

Faculty of Computer Science and Information Technology
University Malaya

5. The Interactive Learning Package layout design

□ Very interactive

☑ Interactive

□ Unattractive

□ Very unattractive

6. The instruction of the demonstration

☑ Very clear

□ Clear

□ Unclear

□ Cannot follow

7. User will be able to use the demo independently

□ Agree

☑ Somewhat agree

□ Somewhat disagree

□ Disagree

8. Things you get from the all the demonstration

□ Ease of use

□ Knowledge gained

☑ Understand better

□ None

If others, please specify: ______________________

9. The relation between learning modules and the demonstration module.

☑ Very suitable

□ Suitable

□ Unsuitable

□ Very unsuitable

10. General attitudes

□ Frustrated

☑ Bored

□ Interesting

□ Indifferent

Suggestion (s) to improve this Interactive Learning Package

/ Studied this subject too much!

Thank you for your cooperation.

Prepared by:
Camellia Mohd Kamal WEK010026
A Deenalatha WEK010001

Faculty of Computer Science and Information Technology
University Malaya

User evaluation

The purposes of this survey are:

- To access the effectiveness and the general outlook of the **Interactive Learning Package for Artificial Neural Network**.
- *To determines if this system meet the users need and expectations.*

Please kindly take a moment off to complete this survey. Your cooperation is very much appreciated.

User's Detail

Course: Artificial Intelligence

Semester: 2

Faculty: FSKTM

1. The scope of the contents used in the package

□ *Very suitable*

☑ Suitable

□ Unsuitable

□ Very unsuitable

2. The effectiveness of the system (knowledge gained) after used the system.

□ Very effectiveness

☑ Effectiveness

□ Moderate

□ Ineffectiveness

3. Level of difficulty is appropriated for the stated study level.

□ Very suitable

☑ Suitable

□ Unsuitable

□ Very unsuitable

4. The color combination of the user interface design

□ Very suitable

☑ Suitable

□ Unsuitable

□ Very unsuitable

5. The Interactive Learning Package layout design

- □ Very interactive
- ☑ Interactive
- □ Unattractive
- □ Very unattractive

6. The instruction of the demonstration

- □ Very clear
- ☑ Clear
- □ Unclear
- □ Cannot follow

7. User will be able to use the demo independently

- ☑ Agree
- □ Somewhat agree
- □ Somewhat disagree
- □ Disagree

8. Things you get from the all the demonstration

- □ Ease of use
- ☑ Knowledge gained
- □ Understand better
- □ None

If others, please specify: ______________________________

9. The relation between learning modules and the demonstration module.

- □ Very suitable
- ☑ Suitable
- □ Unsuitable
- □ Very unsuitable

10. General attitudes

□ Frustrated

□ Bored

☑ Interesting

□ Indifferent

Suggestion (s) to improve this Interactive Learning Package

- Add some animated stuff.

Thank you for your cooperation.

Prepared by:
Camellia Mohd Kamal WEK010026
A Deenalatha WEK010001

Faculty of Computer Science and Information Technology
University Malaya

User evaluation

The purposes of this survey are:

- To access the effectiveness and the general outlook of the **Interactive Learning Package for Artificial Neural Network**.
- *To determines if this system meet the users need and expectations.*

Please kindly take a moment off to complete this survey. Your cooperation is very much appreciated.

User's Detail

Course: Artificial Intelligence

Semester: 2

Faculty: FSKTM

1. The scope of the contents used in the package

□ Very suitable

☑ Suitable

□ Unsuitable

□ Very unsuitable

2. The effectiveness of the system (knowledge gained) after used the system.

☑ Very effectiveness

□ Effectiveness

□ Moderate

□ Ineffectiveness

3. Level of difficulty is appropriated for the stated study level.

□ Very suitable

☑ Suitable

□ Unsuitable

□ Very unsuitable

4. The color combination of the user interface design

☑ Very suitable

□ Suitable

□ Unsuitable

□ Very unsuitable

5. The Interactive Learning Package layout design

☑ Very interactive

□ Interactive

□ Unattractive

□ Very unattractive

6. The instruction of the demonstration

☑ Very clear

□ Clear

□ Unclear

□ Cannot follow

7. User will be able to use the demo independently

☑ Agree

□ Somewhat agree

□ Somewhat disagree

□ Disagree

8. Things you get from the all the demonstration

☑ Ease of use

☑ Knowledge gained

☑ Understand better

□ None

If others, please specify: ______________________

9. The relation between learning modules and the demonstration module.

☒ Very suitable

☑ Suitable

□ Unsuitable

□ Very unsuitable

10. General attitudes

□ Frustrated

□ Bored

☑ Interesting

□ Indifferent

Suggestion (s) to improve this Interactive Learning Package

Linking from the first module to second module should be consistent.

Thank you for your cooperation.

Prepared by:
Camellia Mohd Kamal WEK010026
A Deenalatha WEK010001

Faculty of Computer Science and Information Technology
University Malaya

User evaluation

The purposes of this survey are:

- To access the effectiveness and the general outlook of the **Interactive Learning Package for Artificial Neural Network**.
- *To determines if this system meet the users need and expectations.*

Please kindly take a moment off to complete this survey. Your cooperation is very much appreciated.

User's Detail
Course: AI
Semester: 2
Faculty: FSKTM .

1. The scope of the contents used in the package

   □ Very suitable
   ☑ Suitable
   □ Unsuitable
   □ Very unsuitable

2. The effectiveness of the system (knowledge gained) after used the system.

   □ Very effectiveness
   ☑ Effectiveness
   □ Moderate
   □ Ineffectiveness

3. Level of difficulty is appropriated for the stated study level.

   □ Very suitable
   ☑ Suitable
   □ Unsuitable
   □ Very unsuitable

4. The color combination of the user interface design

   □ Very suitable
   ☑ Suitable
   □ Unsuitable
   □ Very unsuitable

5. The Interactive Learning Package layout design

□ Very interactive

☑ Interactive

□ Unattractive

□ Very unattractive

6. The instruction of the demonstration

□ Very clear

☑ Clear

□ Unclear

□ Cannot follow

7. User will be able to use the demo independently

□ Agree

☑ Somewhat agree

□ Somewhat disagree

□ Disagree

8. Things you get from the all the demonstration

☑ Ease of use

□ Knowledge gained

☑ Understand better

□ None

If others, please specify: ____________________

9. The relation between learning modules and the demonstration module.

□ Very suitable

☑ Suitable

□ Unsuitable

□ Very unsuitable

10. General attitudes

□ Frustrated

□ Bored

☑ Interesting

□ Indifferent

Suggestion (s) to improve this Interactive Learning Package

none.

Thank you for your cooperation.

Prepared by:

Camellia Mohd Kamal	WEK010026
A Deenalatha	WEK010001

Faculty of Computer Science and Information Technology

University Malaya