

QUANTITATIVE PRECIPITATION ANALYSIS AND OFFLINE GUI USING  
NEURAL NETWORK SYSTEM

SITI NURSYUHADA MAHSAHIRUN

This thesis is submitted as partial fulfillment of the requirements for the award of the  
Bachelor of Electrical Engineering (Electronics)

Faculty of Electrical & Electronics Engineering  
University Malaysia Pahang

November, 2009

## UNIVERSITI MALAYSIA PAHANG

### BORANG PENGESAHAN STATUS TESIS♦

**JUDUL:** QUANTITATIVE PRECIPITATION ANALYSIS AND OFFLINE GUI USING NEURAL NETWORK SYSTEM

**SESI PENGAJIAN:** 2009/2010

Saya SITI NURSYUHADA MAHSAHIRUN (861110-12-5328)  
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)\* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. \*\*Sila tandakan ( √ )

**SULIT**

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

**TERHAD**

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

**TIDAK TERHAD**

Disahkan oleh:

\_\_\_\_\_  
(TANDATANGAN PENULIS)

\_\_\_\_\_  
(TANDATANGAN PENYELIA)

Alamat Tetap:

**LOT-20, PERINGKAT 1**  
**RANCANGAN SUNGAI MANILA,**  
**90009 SANDAKAN,**  
**SABAH NEGERI DI BAWAH BAYU.**

**EN. MOHD ZAMRI BIN IBRAHIM**  
( Nama Penyelia )

Tarikh: **23 NOVEMBER 2009**

Tarikh: **23 NOVEMBER 2009**

- CATATAN:
- \* Potong yang tidak berkenaan.
  - \*\* Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
  - ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

“I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)”

Signature : \_\_\_\_\_

Name : MR. MOHD ZAMRI BIN IBRAHIM

Date : 23/11/2009

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : \_\_\_\_\_

Author : SITI NURSYUHADA BINTI MAHSAHIRUN

Date : 23/11/2009

**DEDICATION**

*Specially dedicated to  
My lovely parent, sisters, brothers  
and all of my best friends.*

## ACKNOWLEDGEMENT

I would like to seize this opportunity with deep humble to extend my deepest gratitude to the following persons who have working with me upon finishing this project.

First and foremost, a special thank to my supervisor, Mr Mohd Zamri bin Ibrahim who helped me a lot during the progress of the research and project, for all significant thought, supporting and continuous patience in supervising during these two semesters project. He never refuses to give hands when we called a help. Without his sacrifice sparing the precious time to guide us, this project would never be released successfully.

My greatest thankfull also to all the lecturers and laboratory instructors for their friendly helps and guidance. Besides, special thank to my entire course mates and siblings who have accompanied me though the whole semesters.

Lastly but the most important, I would like to spread my words of appreciation to the most important persons in my life, Mr Mahsahirun Harun and Mdm Fatimah Madyasin who never tired in guiding, inspiring and supporting me in my life. They are the people who the most responsible enable me carving name in this thesis.

God bless you!

Thank you very much.

Siti Nursyuhada Mahsahirun

## ABSTRACT

This project discovers the implementation of Artificial Neural Network (ANN) for forecasting weather based on past relevant data. Neural network is constructed using empirical network architecture and (17) training types. They are such as BFGS quasi-Newton backpropagation, Cyclical order incremental training w/learning functions, Levenberg-Marquardt backpropagation, Resilient backpropagation and others. The ANN has been trained using 2008 weather data and tested with data year 2009. As result, the system has successfully generating accuracy up to 78.69% for quantitative precipitation (QP) prediction. Analysis on time consumption of all those training types is made and shows that Resilient backpropagation with 1.92s training time consumption is the fastest and Cyclical order incremental training w/learning functions with 463.215s is the slowest. This project concluded that ANN is an alternative method in controlling and understanding the way of non-linear set of data and variables to become mutually correlated with each other. It is a powerful yet significant method in embedding intelligent system into application for meteorological tools.

## ABSTRAK

Projek ini menyingkap pelaksanaan Artificial Neural Network (ANN) untuk ramalan cuaca berdasarkan data-data masa lalu yang berkaitan. Jaringan saraf dibina dengan menggunakan binaan rangkaian yang dijana secara empirik dan (17) jenis latihan. Ianya adalah seperti BFGS quasi-Newton backpropagation, agar Cyclical order incremental training w/learning functions, Levenberg-Marquardt backpropagation, Resilient backpropagation dan lain-lain. ANN telah dilatih dengan menggunakan data cuaca 2008 dan diuji dengan data tahun 2009. Hasilnya, sistem ini telah Berjaya menghasilkan ketepatan sehingga 78.69% untuk ramalan kuantitatif curahan hujan (QP). Analisis tempoh latihan semua jenis fungsi latihan dibuat dan menunjukkan bahawa Resilient backpropagation dengan 1.92s tempoh latihan adalah yang terpantas sementara Cyclical order incremental training w/learning functions dengan 463.215s adalah yang paling lambat. Projek ini menyimpulkan bahawa ANN merupakan salah satu alternatif dalam mengendalikan dan memahami cara non-linear set data dan pembolehubah untuk menjadi saling berkorelasi satu sama lain. Ini adalah kaedah yang signifikan dan mantap dalam memasukan sistem cerdas, AI dalam aplikasi peralatan kajicuaca.



**TABLE OF CONTENTS**

<b>CHAPTER</b>	<b>TITLE</b>	<b>PAGE</b>
	<b>TITLE PAGE</b>	i
	<b>DECLARATION</b>	iii
	<b>DEDICATION</b>	iv
	<b>ACKNOWLEDGEMENT</b>	v
	<b>ABSTRACT</b>	vi
	<b>ABSTRAK</b>	viii
	<b>TABLE OF CONTENT</b>	xi
	<b>LIST OF FIGURES</b>	xii
	<b>LIST OF TABLES</b>	xiv
	<b>LIST OF ABBREVIATIONS</b>	xv
	<b>LIST OF APPENDICES</b>	xvi
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Project Background	1
	1.2 Project Objective	3
	1.3 Project Scope	4
	1.4 Problem Statement	5
	1.5 Thesis Outline	7

**2****LITERATURE REVIEW**

2.1	Introduction	8
2.2	Biological Neural Network	9
2.3	Artificial Neuron Network	10
2.3.1	Structure	11
2.3.2	Training and Learning Types	16
2.3.3	Algorithm	16
2.4	Creating Graphical User Interface	17
2.5	Past Relevant Project/Research	18

**3****METHODOLOGY**

3.1.	Introduction	20
3.2.	System Block Diagram	21
3.3	Analysis	23
3.3.1	Loading Data	25
3.3.2	Training Network	26
3.3.3	Testing Network	29
3.4	Constructing GUI	30

<b>4</b>	<b>RESULT AND ANALYSIS</b>	
4.1	Introduction	31
4.2	Neural Network Configuration	
4.2.1	Programming structure	31
4.2.2	Simulation result	33
4.3	Graphical User Interface	
4.3.1	Layout	44
<b>5</b>	<b>CONCLUSION AND RECOMMENDATION</b>	
5.1	Conclusion	47
5.2	Future Recommendation	48
5.3	Costing and Commercialization	49
	<b>REFERENCES</b>	50
	<b>APPENDICES</b>	52

## LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Biological Neural Network	2
1.2	Biological Neural Network schematic diagram	2
1.3	People affected by natural disaster, global region, 1991-2000 and 2001-2008 (Source: EM-DAT)	5
1.4	Floods at Kangkar Tebrau Village, Johor Bahru on Dicember 2006 (Source: The Star)	6
2.1	Schematic Biological Neuron	9
2.2	Artificial Neural Network	11
2.3	(a) Multi input, single output, no hidden layer, Feed-Forward Topology, (b) multi-input, single output, one hidden layer, (c) multi-input, multi output, one hidden layer	12
2.4	Recurrent Topology	12
2.5	A Layers of Neuron	14
2.6	Schematic diagram for multiple layers of neurons	15
2.7	Artificial Neural Network diagram in MATLAB	17
3.1	ANN Training flow diagram	21
3.2	Tan-Sigmoid Activation Function	22
3.3	Testing ANN Flow Diagram	22
3.4	Project Flow Chart	24
3.5	Copying data from Microsoft Excel in tabling order	25
3.6	Constructing FFNN	27

3.7	GUI ANN input and Output Block Diagram	30
4.1	Training Programming Structure	32
4.2	BFGS quasi-Newton backpropagation Training Function	39
4.3	Powell-Beale conjugate gradient backpropagation Training Function	40
4.4	Fletcher-Powell conjugate gradient backpropagation Training Function	40
4.5	Levenberg Marquardt Training Function	41
4.6	One step secant backpropagation Training Function	42
4.7	Resilient backpropagation Training Function	42
4.8	Scaled conjugate gradient backpropagation Training Function	43
4.9	Graphical User Interface representation	44
4.10	About window	45
4.11	Read me text	45

**LIST OF TABLE**

<b>TABLE NO.</b>	<b>TITLE</b>	<b>PAGE</b>
3.1	Activation Function Types	26
3.2	Neural Network Training Function	28
4.1	Description of programming structure	32
4.2	Time Consumption corresponding difference training type	33
4.3	Time Consumption corresponding to difference training types	34
4.4	The best network choose to be used at second stage	35
4.5	Summary of the Analysis	47

## LIST OF ABBREVIATIONS

NN	-	Neural Network
ANN	-	Artificial Neural Network
QP	-	Qualitative Precipitation
QPF	-	Qualitative Precipitation Forecasting
PoP	-	Probability of Precipitation
FF	-	Feed Forward
BP	-	Backpropagation
FFBP	-	Feed Forward Backpropagation
GUI	-	Graphical User Interface
IW	-	Input weight
LW	-	Layer weight
b	-	Bias

**LIST OF APPENDICES**

<b>APPENDIX</b>	<b>TITLE</b>	<b>PAGE</b>
1	Transfer Function Types	53
2	Stage 1: ANN Training Codes	55
3	Stage 2: GUI Parent Window Codes	56
4	Stage 2: About Window Codes	68
5	Stage 2: Close Dialog Box Codes	71
6	Function Created Codes	69
7	Weather Data	70
8	Data Distribution Conditions of Malaysian Meteorology Department	80



## CHAPTER 1

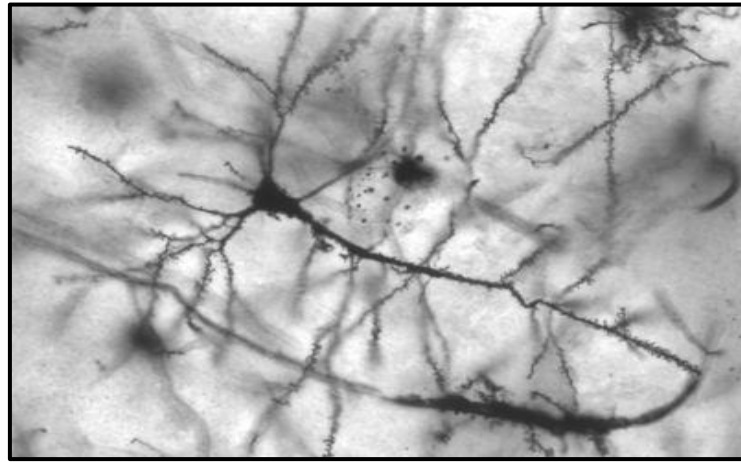
### INTRODUCTION

#### 1.1 Project Background

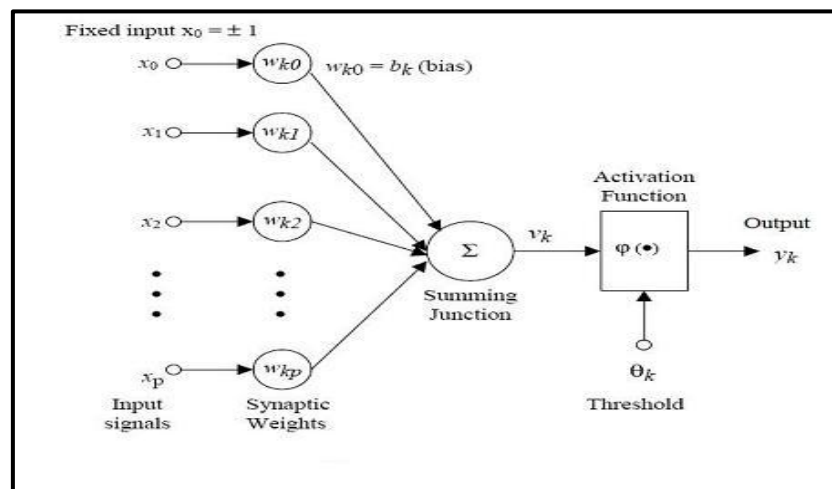
Over decades, the development of intelligence system demand highly discovery in many prospect of human life. All of it has the purpose to facilitate with accurate, fastest, and reliable services for the maximum convenient for the user. Machines evolved from hard to the soft computation, from black-and-white language of Boolean logic to fuzzy sets and artificial neural network. The very question asked in the early 1940s ‘whether machines can think’ brings to the merge of Artificial Intelligence (AI). Until today, branches of AI have widely grown in many aspect of human life.

It was since at least the nineteenth century that human start attempted to predict the weather. Probability of precipitation (PoP) is waited as news everyday. It is extremely important for many decision makers due all sensitive activities to precipitate occurrences. Quantitative precipitation forecast (QPF) is the amount or quantity of prediction at the particular area. Precipitation can be in many forms from rain to snow to hail to sleet [12]. They each fall from the clouds in the sky to the ground. In this work, the data of rainfall at Kuantan area is considered as the parameter. It can identify the potential of heavy precipitation which favouring to hydrologic interest such as for flash flood prediction and other alike. Precipitation is measured using rain gauge. This natur phenomenon is always seems to be fluctuating

in event and not easily predicted. Thus, the need of intelligent system is crucial in this field.



**Figure 1.1:** Biological Neural Network



**Figure 1.2:** Biological Neural Network schematic diagram

This project is about to focus on Artificial Neural Network (ANN) that is a model of reasoning basic information-processing units, called neurons. This network was inspired by the actual human neuron interconnection system which exhibit the most powerful information network that work much faster than the fastest computers

in existence today. It is applying a mathematical model alike that neuroscientist believes the actual brain cell might work. Generally, ANN consists of multiple artificial neuron layers which can be interconnected in different ways for different types of data processing.

ANN neuron model i.e. architecture and training type are modified to find the best trained network for the forecasting. ANN has the ability to learn from the environment (input and output pairs), self organise its structure, and adapt to it in an interactive manner. Specifically, in this project the daily rainfall, mean temperature, relative humidity and mean wind for Kuantan area is train using several types of neural network. Finally, a QP for 24 hour mean rainfall (mm) of Kuantan area's GUI is developed. Data of 2008 weather, provided by Malaysian Meteorology Department is utilised during the investigation.

For those purpose, massive calculation involved. Thus, MATLAB R2007a software which offers tools to capture the best ANN configuration for weather forecasting purpose is used.

## **1.2 Project Objectives**

The purpose of this project is to apply Artificial Neural Network system for forecasting activity. Data provided by Malaysia Meteorology Department is as suitable for rainfall occurrence prediction. Thus, this project is carried out with the following objectives:-

- i. To examine the theory of neural network in weather forecasting field
- ii. To make analysis and identify the best configuration of neural network for QP occurrence
- iii. To set up Graphical User Interface, GUI of QP using ANN system

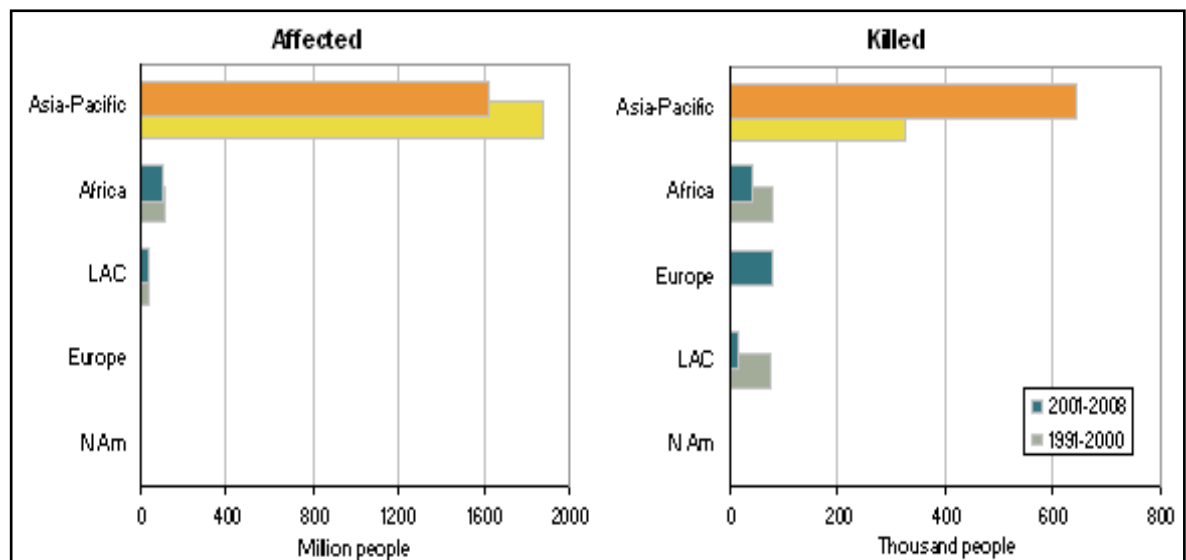
### **1.3 Project Scope**

There are three scopes set up to guide this project toward its objectives. The scopes are summarized as follow:-

- i. Generate neural network based FeedForward Backpropagation (FFBP) algorithm for weather data of Kuantan Area using data of year 2008. Hidden layer and training types are the manipulated variables. The fixed variable is the data types that are, mean temperature, relative humidity and mean wind while the resulting variable is the QP accuracy.
- ii. Search and analysis performance network for given data by manipulating the network architecture that are the nodes, number hidden layer besides training type. The regression plots are plot as result. All this activities is done using MATLAB R2007a.
- iii. Generate offline graphical user interface using MATLAB R2007a for rainfall forecasting against Kuantan, Pahang area that would allow user to test data using the best trained networks by difference training types.

### **1.4 Problem Statement**

Malaysia is located in Tropical Zone. The characteristic features of the climate of Malaysia are uniform temperature, high humidity and copious rainfall and they arise mainly from the maritime exposure of the country. Winds are generally light. It is also spared from natural disasters such as earthquakes and typhoons but still not immune to climatic problems. Tragedies such flash floods, heavy thunderstorms, drought, erosion and haze occurred every year.



**Figure 1.3:** People affected by natural disaster, global region, 1991-2000 and 2001-2008 (Source: EM-DAT)

According to statistic, for Asia and the Pacific, 2008 From January to September 2008, a total of 28 disasters caused by natural hazards affected more than 101 million people, killed more than 223,000 and caused more than \$103 billion in economic damage. Efforts have already being made to adopt the multi model numerical weather forecasting system. Obviously, there are multiple alternative to predict weather. Typical instrument and equipment with various sensor detecting climate changing demand high costing and would not always suitable to implement in all application. The derivation of soft computing, mathematical expression especially in intelligent system provide other more cost effective relatively to detecting natural phenomenon pattern.

With weather forecast system damages can be minimize. Hence, this project is a good approach in experimenting non-linear data to provide simple summary and general information on neural network implementation in weather forecasting and the best to expose newbie in ANN forecasting field. It thus, would give another perspective of forecasting alternative to in real life.



**Figure 1.4:** Floods at Kangkar Tebrau Village, Johor Bahru on December 2006  
(Source: The Star)

## 1.5 Thesis Outline

Overall, this thesis is divided into (5) chapters. Each chapter provide texts elaboration with visual aid graphical representation. In introduction chapter, the general information about this project is discussed which include introduction, project objective, project scope and also problem statement.

Next is the literature review chapter. Here ANN and its application in forecasting are elaborate. Besides several outcomes of recent ANN research or project are also includes.

Methodology chapter discuss about the how this project done. With the aid of graphical representation system in block diagram and flow chart, the methodology was explained. Programming codes and how the project done stage by stage using MATLAB has also been touch in these pages.

For result and analysis chapter, the result of each stage is shown. Here the discussions cover comparison of training types towards accuracy and also the technique used in developing GUI.

Lastly is the conclusion and recommendation. In this chapter conclusion responding the objectives of this project are stated. It is also included with (3) recommendation for improvement if possible may apply for the next research or project.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

Intelligence is the ability to learn and understand, to solve problems and to make decisions [13]. AI is first discovered 1940s as machines do things that would require human intelligent. In the modern era of neural network research at 1943, Warren McCulloch and Walter Pitts proposed *McCulloch-Pitts Theory of Formal Neural Networks*. It was a model of artificial neural networks which each neuron was postulated as being in binary state or either on or off. This discovery was inspired from the central nervous system research done by McCulloch and from the young mathematical prodigy, Pitts [5].

From that, in 1949 the new development in neural network technology arrived in with a book, "The Organization of Behavior" written by Donald Hebb. This book validating and make further reinforced McCulloch-Pitts's theory about neurons and how they work. In 1951, Marvin Minsky from New York is the first person who invent first neural machine [5].

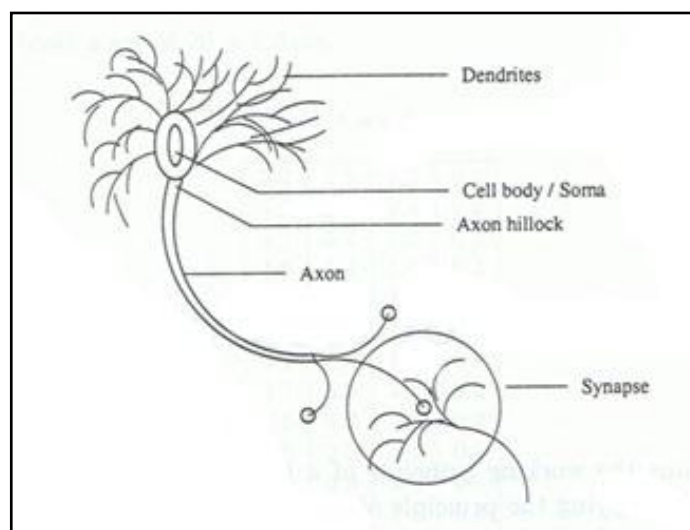
The evolution of this technology brought to the development of ideas in such field as knowledge representation, learning algorithms, neural computing and computing



with words. A professor from University of California at Berkeley, Lotfi Zadeh first discover about fuzzy logic in 1960s [2]. Turn out of that, Alan Turing in 1970s introduced the neural computing and artificial neural network (ANN). After that, the evolutionary of computing with words begin at late 1980s until today.

This chapter are going to discuss the literature review about element that been use in this project. The purpose of it is to give an overview or state of knowledge on neural network.

## 2.2 Biological Neuron Network



**Figure 2.1:** Schematic Biological Neuron

The *human* brain incorporates nearly 10 billion neurons and 60 trillion connections, synapses, between them. By using multiple neurons simultaneously, the brain can perform its functions much faster than the fastest computers in existing today.

In biological neuron, the main components are the dendrites (bush of thin fibers), cell body (soma cell), axon and synapse. Interconnection between neuron to its neighborhood is done by dendrites. Signals are propagated by complex electrochemical reactions. Chemical substances released from the synapses cause a change in the electrical potential of the cell body. As the potential reaches its threshold, an electrical pulse, action potential, is sent down through the axon. The pulse spreads out and eventually reaches synapses, causing them to increase or decrease their potential. Neural network posed plasticity characteristic. Neuron can also form new connections with other neuron and the entire neuron can also migrate from one place to another. Neurons leading to the 'right answer' are strengthened while those leading to the 'wrong answer' weaken. As result, neural network have the ability to learn through experience.

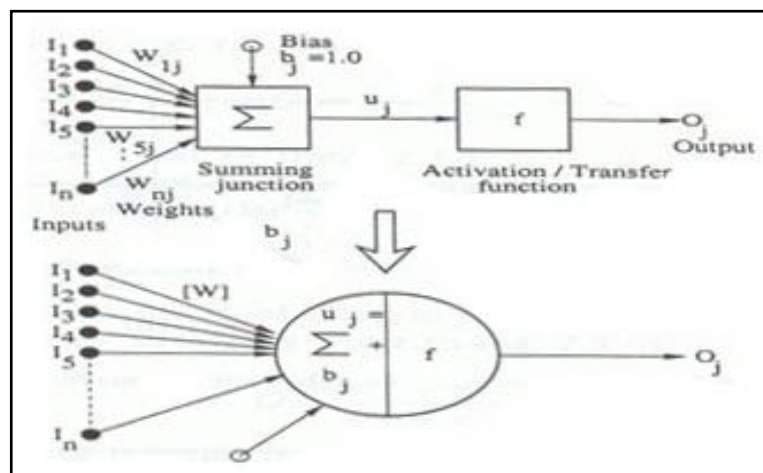
### 2.3 Artificial Neural Network

*Artificial neural network (ANN)* consists of number of very simple and highly interconnected processors, also called *neurons*, which are analogous to the biological neurons in the brain. The neurons are connected by weighted links passing signals from one neuron to another. Each neuron receives number of inputs but only provide single output.

There were (5) main factors that influencing the speed and memory consumption in Neural Network toolbox. There are listed below: [6]

- i. Complexity of the problem,
- ii. Number of data points in the training set,
- iii. Number of weights and biases in the network,
- iv. Error goal, and
- v. Whether the network is being used for pattern recognition (discriminant analysis) or function approximation (regression)

### 2.3.1. Structure

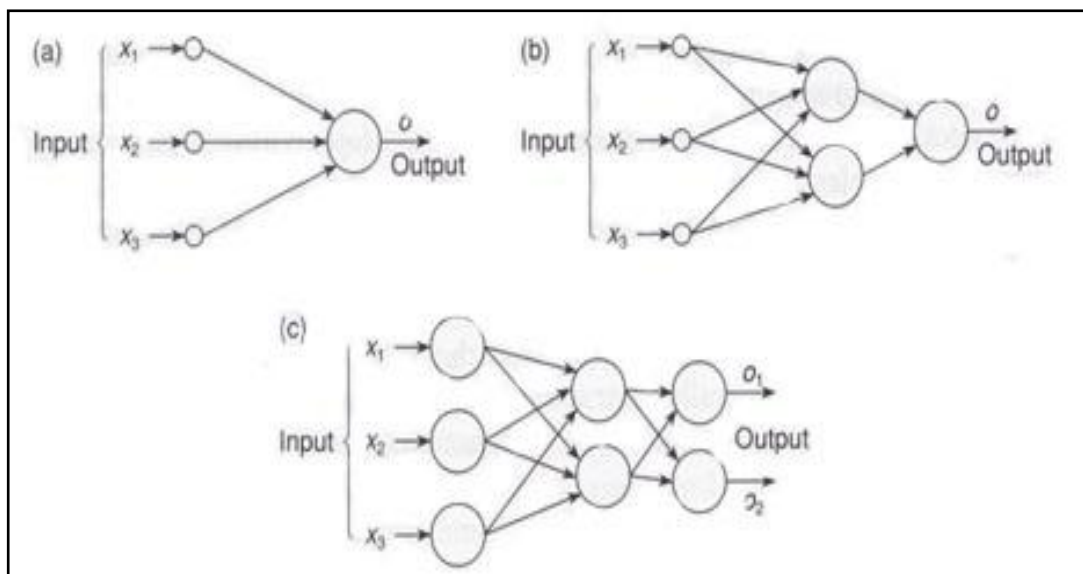


**Figure 2.2:** Artificial Neural Network

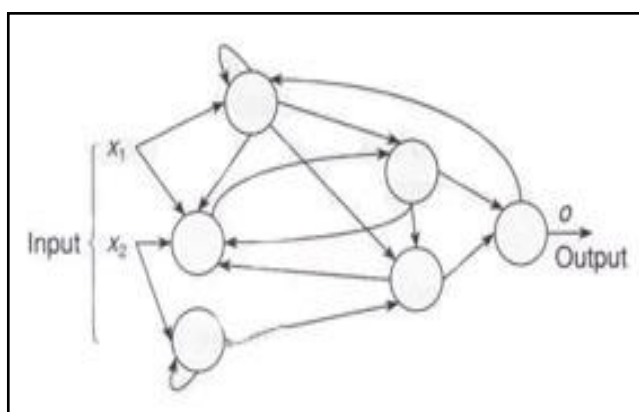
Three important features generally characterized ANNs that includes as follow:-

#### i. Network topology

Determine the order and organization of the nodes from input to output layer. Which include the interconnections is arranged of the nodes within the layers. There are several topology of ANN available today. However the obvious are feed-forward and recurrent NNs. **Feed-Forward Neural Networks, FFNNs** simply passed the information into one direction. Otherwise, the **Recurrent Neural Networks, RNNs** forms loop, namely involving feed-forward and feedback [1].



**Figure 2.3:** (a) Multi input, single output, no hidden layer, Feed-Forward Topology, (b) multi-input, single output, one hidden layer, (c) multi-input, multi output, one hidden layer



**Figure 2.4:** Recurrent Topology

## ii. Network transfer functions

There are several types of transfer function in ANN. Such as Linear, sigmoid, log sigmoid and other. Refer appendix 1 which provide some of obvious used in ANN.

### iii. Network learning algorithm

Training algorithm is use to update the parameters at the interconnection level o of the neurons. There are three most often used, those are the supervised, unsupervised (selves organized) and the reinforced. [1]

Figure 2.2 above shows the ANN or neuron  $j$ .

Where,

$n$  inputs ( $I_1, I_2, I_3 \dots I_n$ )

The weights of inputs represented by,

$$[W] = [W_{1j}, W_{2j}, \dots, W_{nj}]$$

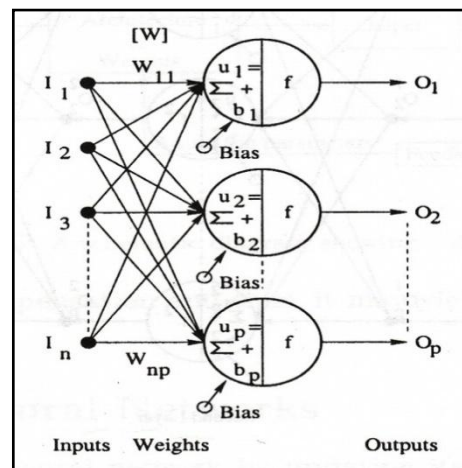
Summing junction is to collect the weighted inputs and sum then up.

$B_j$  is the bias fixed value

Then,

$$u_j = \sum_{k=1}^n I_k W_{kj} + b_j \quad (1)$$

$$O_j = f(u_j) = f\left(\sum_{k=1}^n I_k W_{kj} + b_j\right) \quad (2)$$



**Figure 2.5:** A Layers of Neuron

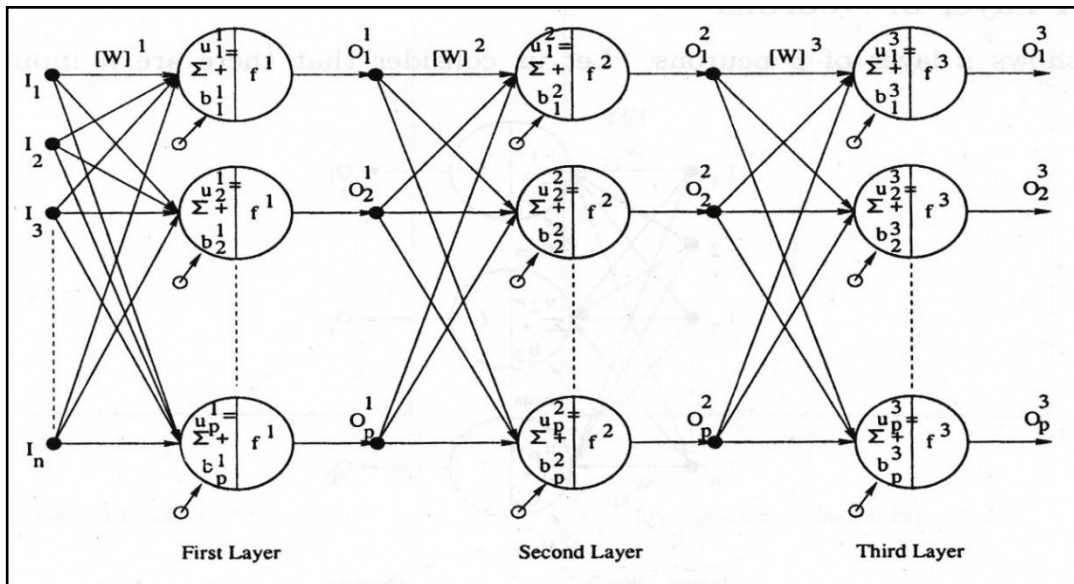
The weights represented by,

$$[W] = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1p} \\ W_{21} & W_{22} & \dots & W_{2p} \\ \vdots & \vdots & \dots & \vdots \\ W_{n1} & W_{n2} & \dots & W_{np} \end{bmatrix}$$

Then, the output of k-th neuron represented by,

$$O_k = f^1 \left( \sum_{j=1}^n I_j W_{jk}^1 + b_k \right) \quad (3)$$

$k = 1, 2, 3, \dots, p$  and  $b_k$  is the bias value of k-th neuron



**Figure 2.6:** Schematic diagram for multiple layers of neurons

Where the output are respectively,

$$O_k^1 = f^1 \left( \sum_{j=1}^n I_j W_{jk}^1 + b_k^1 \right) \quad (4)$$

$$O_l^2 = f^2 \left( \sum_{j=1}^p O_k^1 W_{jl}^2 + b_l^2 \right) \quad (5)$$

$$O_m^3 = f^3\left(\sum_{j=1}^p O_l^2 W_{jm}^3 + b_k^3\right) \quad (6)$$

ANN can be static or dynamic. For *static* NN, the output is not compared with its target value to determine the error. In *dynamic* NN, error is determined and then feedback to the network to modify its architecture and update the weights. [1]

### 2.3.2. Training and Learning Types

The *training* style is typically to determine when the weights and biases should be updated. At this stage, the networks parameter is optimized. The network undergoes through an internal process of curve fitting. Today there are many types of NN training available.

Particularly, *training scenario* is where the error corresponding to target and output value is propagated backward to update the connecting weights of the neurons and biases. Two obvious training types are the incremental training and batch training [1].

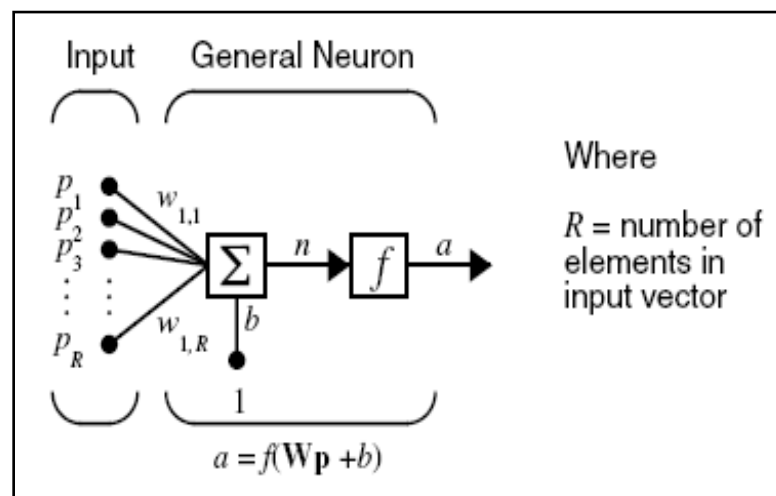
*Incremental training* is an approach where the network is modified utilizing the training scenarios sent one after another in a sequence. Here, weights and biases if the networks are updated each time an input are presented to the network. *Batch training* else, the whole training set consisting of a large number of scenarios, is passed through the network and an average error in predictions is determined [1].

On the other hand, *learning* rule is defined as a procedure for modifying the *weights and biases*. The *supervised* learning is only available to implement if only the relationships of the training scenarios are available. While that, the *unsupervised* learning has the opposite trait. This cause the network to passes through a self-organizing process.

### 2.3.3 Algorithm

Generally, there are many training algorithm available nowadays. However, in this project, the BP is the only algorithm used and discussed.

**BP** refers to the manner in which the gradient is computed for non-linear multilayer network [6]. It is actually the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer function.



**Figure 2.7:** Artificial Neural Network diagram in MATLAB

## 2.4. Creating Graphical User Interface

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components, which enable a user to perform interactive tasks. By using MATLAB software, user would also be able to generate own GUI that works as their desired. Inserting and rearranging GUI element are done by drag-and-drop.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders. Most GUI waits for their user to manipulate a control, and



then respond to each action in turn. Every each control or GUI itself may contain one or more user-written routines known as callbacks, named for the fact that they “call back” to MATLAB to ask it to do things. [14] The action may be triggered by pressing a screen button, clicking a mouse button, selecting a menu item, typing a string or a numeric value, or passing the cursor over a component.

By using `guide` command execute GUI development environment. It provides a set of tools for creating GUIs. All information about MATLAB creating GUI and its codes can be found in help documentation. It contains theory and also provides examples for the user to use as tutorial. Future details about creating GUI in this project can be view in Chapter 3 and 4.

## **2.5 Past Relevant Project / Research**

ANN has been widely use in industry such as in aerospace, automotive, financial, defense, electronics, entertainment, insurance, manufacturing, medical, securities, transportation and etc. It was apply to determine or predict.

ANN been implemented to many innovations involving non linear set of data such as in meteorological parameter [9]. The prediction of global solar radiation (GSR) research [4] is one of them, in future time domain based inputs such as mean air temperature, relative humidity and other the minimum mean square error in figure of 0.00405400, 0.00439553 and 0.008879 systems has successfully generated using ANN.

There are several methods in weather forecasting. [3] The graphical observation from radar view of particular area is one of them. It involves direct analysis of current situation is one of the way. As ANN has the ability to learn pattern of set of inputs and targets data, it is also become one of them which may really suitable for forecasting field. In this project, ANN is applied to predict of qualitative of precipitation that means in weather forecasting.

[7] Use ANN ensemble quantitative rainfall forecasting model based on K-nearest neighbor nonparametric estimation of regression. This work shows that prediction by using non parametric assemble model is generally better than those using other model.

ANN weather forecasting have implemented to recognizing pattern of natural phenomena. [8] Recognizing pattern of monsoon rainfall. By this works 8-Parameter Probabilistic ANN model developed using (11) neurons in input layer and (1) output neuron are ready to be used at districts of Chhattisgarh.

[11] Provide a significant engine for systematic design of a multistage ANN based short-term load forecaster (ANNSTLF). It provides good impact towards cost of all power utilities in the planning of energy supply.

In weather forecasting, scientist as been successfully generated weather prediction using mathematical soft computation up to 85% accuracy to the actual value of both PoP and QPF [15].

## CHAPTER 3

### METHODOLOGY

#### 3.1 Introduction

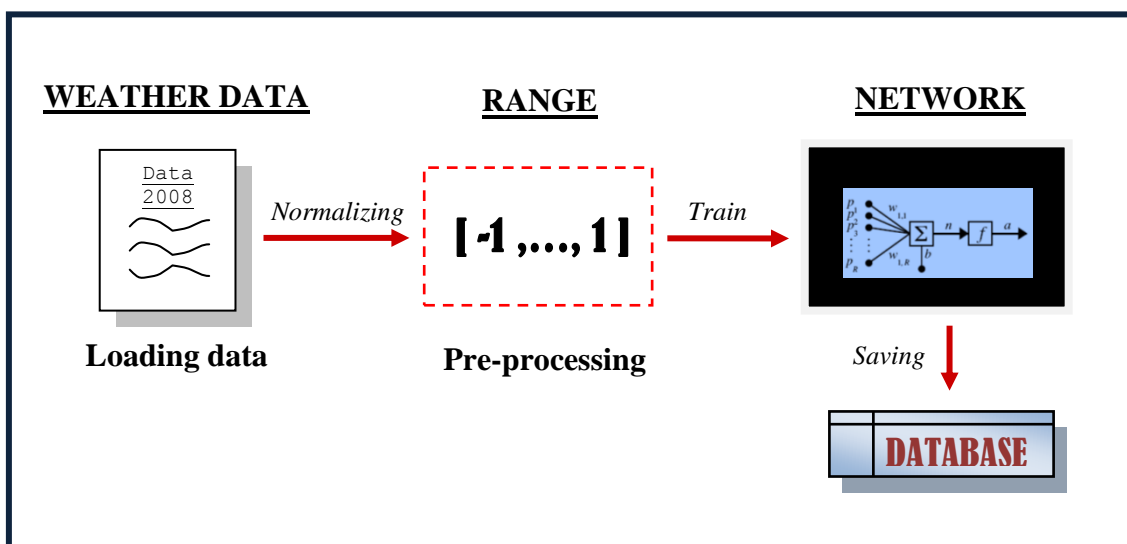
NNs provide a methodology for extracting patterns from noisy data. The standard procedure for use of NN involves training network with a large sample of representative data. In between the input and output layer i.e. hidden layer/s, were all determined empirically. Thus, to make this part be in a better order, periodic increment of hidden layering and number of nodes was applied for the first stage. All the training types available in MATLAB R2007a were tested.

The *first stage* of this project is to find the best NNs configuration for weather forecasting data available. In this stage, analysis was made involving loading data, training and testing.

Trained networks were then be embedded into GUI at the *second stage* for the user to generate QP corresponding inputs with the NN develop in previous stage. This chapter describes the procedures used in developing the best network and also it's GUI.

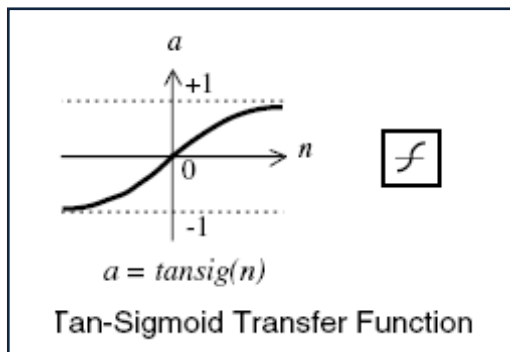
### 3.2 System Block Diagram

Basically, the system has (3) inputs and an output. Simple understanding on the inputs and output mapping and the ANN developed can be view in Figure 3.8 as used in stage (2).



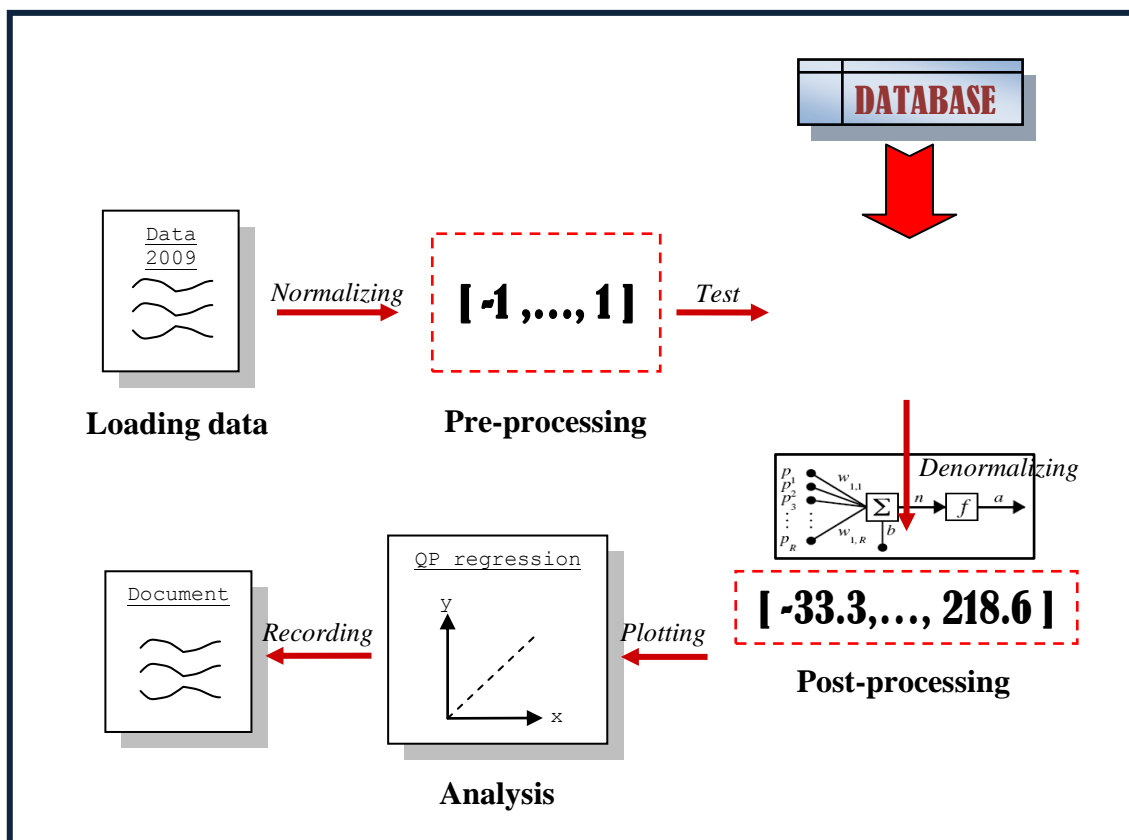
**Figure 3.8:** ANN Training Flow Diagram

As discussed previously, stage (1) involves training and testing. The training process is illustrated at Figure 3.1. As data loaded into the workspace, pre-processing is essential to feed the requirement of activation function chosen. For tan-sig transfer function, the input domain should be in -1 to 1 ranging. Thus, `mapminmax` function was used to accomplish the desired data status. The normalized data was then readily to be trained in variety of ANNs architectures and algorithms. Figure 3.2 is the Tan-sig activation function graph.



**Figure 3.9:** Tan-Sigmoid activation function

Besides saving the weights and biases of trained ANN, time consumption and mean time error of each training type was also documented for analysis. Trained ANN was stored in a database and been used for testing purpose as the diagram continued in Figure 3.3.



**Figure 3.10:** Testing ANN Flow Diagram

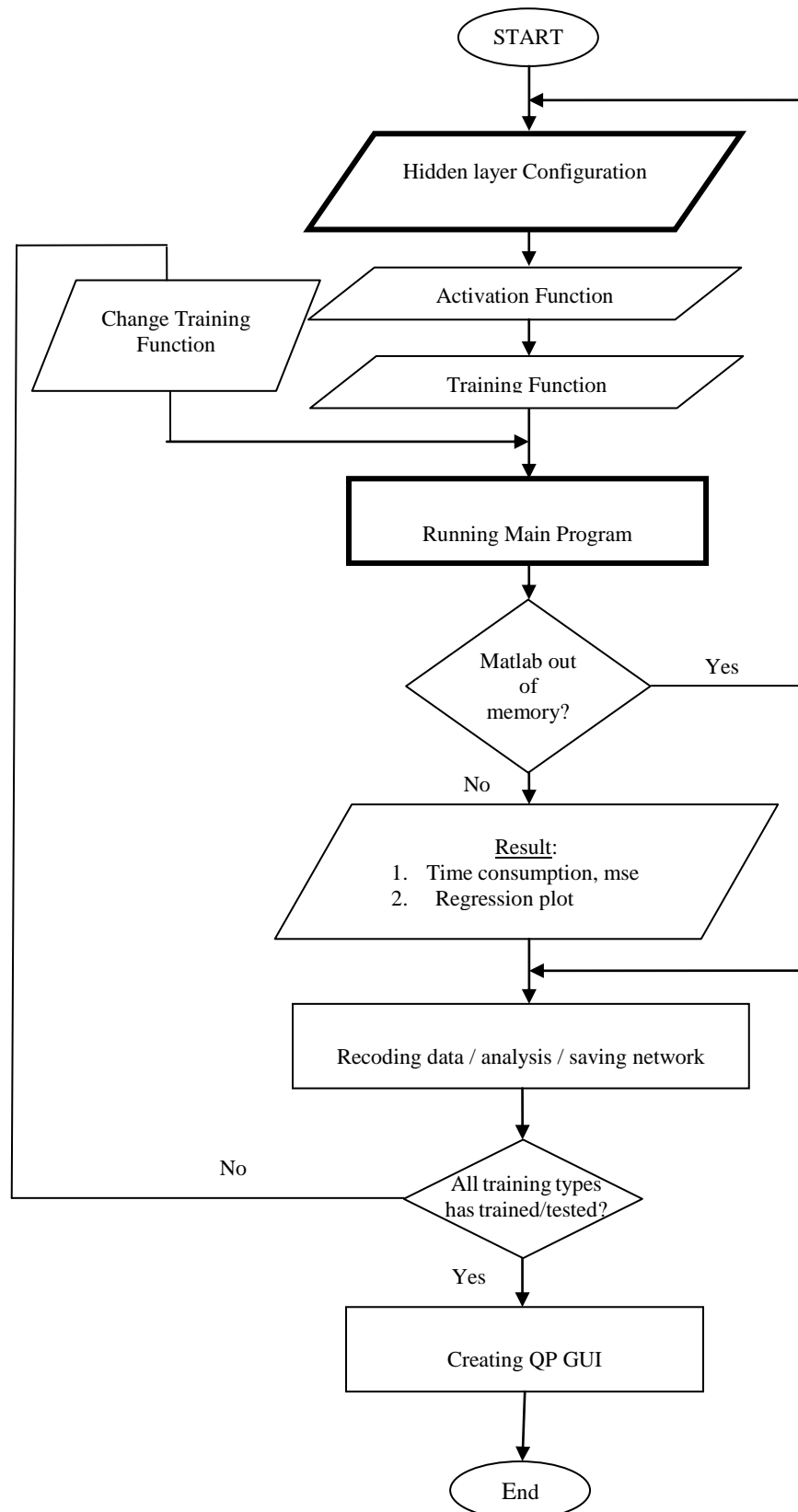
The testing part involves both pre-processing and post-processing of data. As mentioned, pre-processing is done to fulfill the requirement of the activation function, where at this case tan-sig transfer function was used. The post-processing was required to convert the output that is QP to actual ranging value. Ranging -33.3 to 218.6 are the minimum and maximum values for QP of 2008. It was used because data 2008 has been utilized in developing the trained ANN.

Denormalized QP from the ANN prediction versus the actual data values was then used to construct regression plot. Analysis on accuracy was available to be made using those plotting graph. Figure 3.4 summarized how this project is done.

### **3.3 Analysis**

The purpose of doing the analysis stage is to find the suitable configuration of the network. As software is used in this stage, coding of the programming was arranged according to MATLAB R2007a instruction.

Seems the entire calculation is done by MATLAB R2007a, the main programming command was set up. Those include loading data into workspace, normalizing data, into -1 to 1 scale, constructing FFNN, training and result in graphical representation for analysis command. Thus, for training network process, the programming was simply changed on the training type and network configuration while the rest are used throughout the entire training processes.



**Figure 3.5:** Project Flow Chart

### 3.1 Loading Data

Loading Data to workspace is the first step in the programming. It was first be prepared in .DAT format. By using Notepad, data provided in Microsoft Excel 2007 was copied and paste in Notepad in tabling order as shown in Figure 3.5. It was then been save as .DAT format. The data .DAT is now available for loaded in MATLAB workspace.

					Rainfall (08-08 MS1 (mm)	24 Hour Mean Temp. (° C)	Relative Humidity (%)	24 Hour Mean Wind (m/s)
8								
9								
10								
11	Stnno	Year	Month	Day				
12	48657	2008	1	1	6.6	23.9	92.0	2.0
13	48657	2008	1	2	28.6	24.0	91.7	2.0
14	48657	2008	1	3	9.2	24.1	94.6	1.6
15	48657	2008	1	4	-33.3	25.1	89.5	1.7
16	48657	2008	1	5	1.0	25.2	90.3	1.7
17	48657	2008	1	6	3.6	25.0	91.7	1.6
18	48657	2008	1	7	6.8	25.4	90.0	1.8
19	48657	2008	1	8	3.2	25.5	92.4	1.8
20	48657	2008	1	9	1.4	25.9	90.6	1.6
21	48657	2008	1	10	0.0	26.4	87.6	1.6
22	48657	2008	1	11	0.0	27.0	85.3	1.5
23	48657	2008	1	12	0.0	26.9	85.5	1.5
24	48657	2008	1	13	0.0	26.6	86.3	1.6
25	48657	2008	1	14	0.0	26.3	86.8	1.5

Figure 3.12: Copying data from Microsoft Excel in tabling order

On the other hand, still ranging of the variables was not as desired. NN training actually can be made more efficient if certain preprocessing step on the network inputs and targets is performed. Thus, in this project preprocess data command 'premnmx' was used. This command normalizes variables in -1.0 to 1.0 ranging.

Next, the data that read by in matrix form as save in the order in Figure 3.5 need to be transpose. This can be entertain by simply putting apostrophe (') after the variable tagging as shown below.

```
p=Year08(1:366,4:6);
```

```
p';
```

At this time the variables are ready to be used for the next process. Which are training and testing.



### 3.3.2 Training Network

Supervised learning is like learning with teacher. Training Network is like teaching trainee. There are many method or teaching and learning. The network is like being train by giving several exercises. At this part, weather data of the whole year 2008 is been used. Generally, the more frequent training activity undergoes the better understanding.

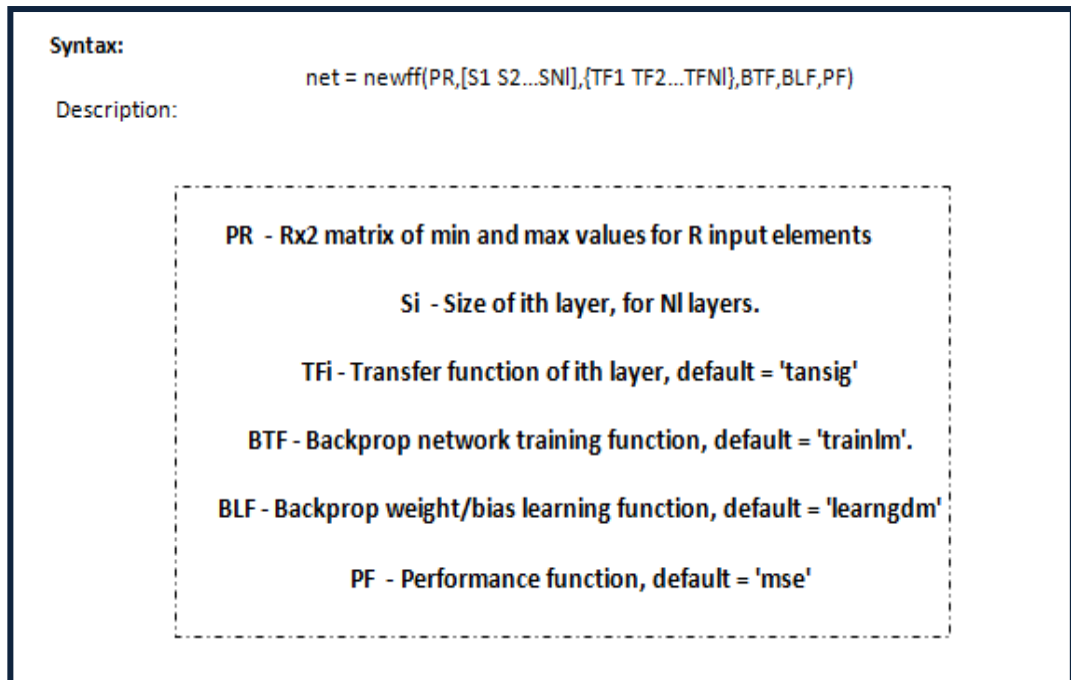
At this point, the network firstly needed to be set up. This project is focus on examining FFBP NN. To create FF network, ‘newff’ was used. Figure 3.6 is the syntax and description of ‘newff’.

**Table 3.1:** Activation Function Types

No.	Transfer Function	Syntax
1.	Compet	compet
2.	Hard-Limit	hardlim
3.	Symmetric Hard-Limit	satlins
4.	Log-Sigmoid	logsig
5.	Positive Linear	poslin
6.	Linear	purelin
7.	Radial Basis	radbas
8.	Satlin	satlin
9.	Satlins	satlins
10.	Softmax	softmax
11.	Tan-sigmoid	tansig
12.	Triangular Basis	tribas
13.	Netinv	netinv

**Hidden layer** and node configuration is determined empirically. It was incremented with (25) until (100) nodes for each layer. The precision of the better

performance is then found by decrement nodes with 5, then with 1. Moreover, the hidden layer was also limited until (5) considering the previous literature review [10].



**Figure 3.13:** Constructing FFNN

Transfer function is the activation function for the network. There were several available in MATLAB. Table 3.1 shows the list of those transfer function. However in this project, tan-sigmoid activation function was the only used in training the network. This is because it has been frequently used in BP NN which provides better resulting and yield output laying in the range of -1.0 to 1 which same as the variable ranging. Figure 3.2 shows the graphical representation of tan-sig and the command in MATLAB.

Now, the training function also is the manipulating variable. There are (17) types of it available in MATLAB R2007a. Table 3.2 listed out all of them.

**Table 3.2:** Neural Network Training Function

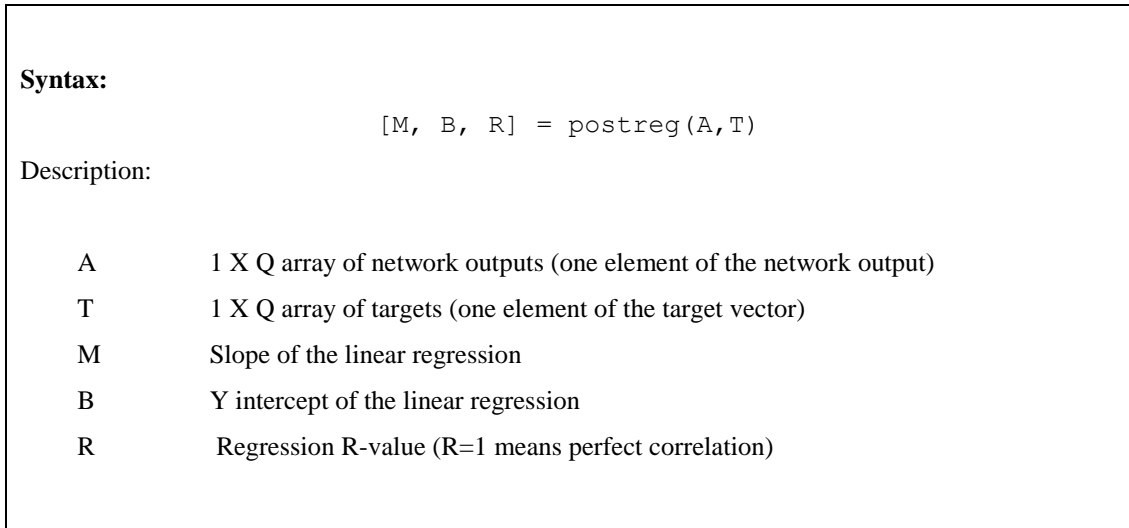
No.	Types	Syntax
1.	Batch training with weight & bias learning rules	trainb
2.	BFGS quasi-Newton backpropagation	trainbfg
3.	Bayesian regularization	trainbr
4.	Cyclical order incremental training w/learning functions	trainc
5.	Powell-Beale conjugate gradient backpropagation	traincgb
6.	Fletcher-Powell conjugate gradient backpropagation	traincgf
7.	Polak-Ribiere conjugate gradient backpropagation	traincgp
8.	Gradient descent backpropagation	traingd
9.	Gradient descent with momentum backpropagation	traingdm
10.	Gradient descent with adaptive lr backpropagation	traingda
11.	Gradient descent w/momentum & adaptive lr backpropagation	traingdx
12.	Levenberg-Marquardt backpropagation.	trainlm
13.	One step secant backpropagation	trainoss
14.	Random order incremental training w/learning functions	trainr
15.	Resilient backpropagation (Rprop).	trainrp
16.	Sequential order incremental training w/learning functions	trains
17.	Scaled conjugate gradient backpropagation	trainscg

### 3.3.3 Testing Network

Test or exam is done to verify a trainee understanding regard what he has been taught and learn. This mechanism is same goes with ANN. The testing determined whether the network configuration need to be modified or more training is necessary. After been trained network was then tested with data 2009.

Network testing result was illustrated with **regression plot** analysis. Post-process trained network response with linear regression. The regression performance

is view using R value. Time consumption can be documented using profiler tool. The profiler tool was instantly started right before the program is run. It is documenting all activity during the program running.

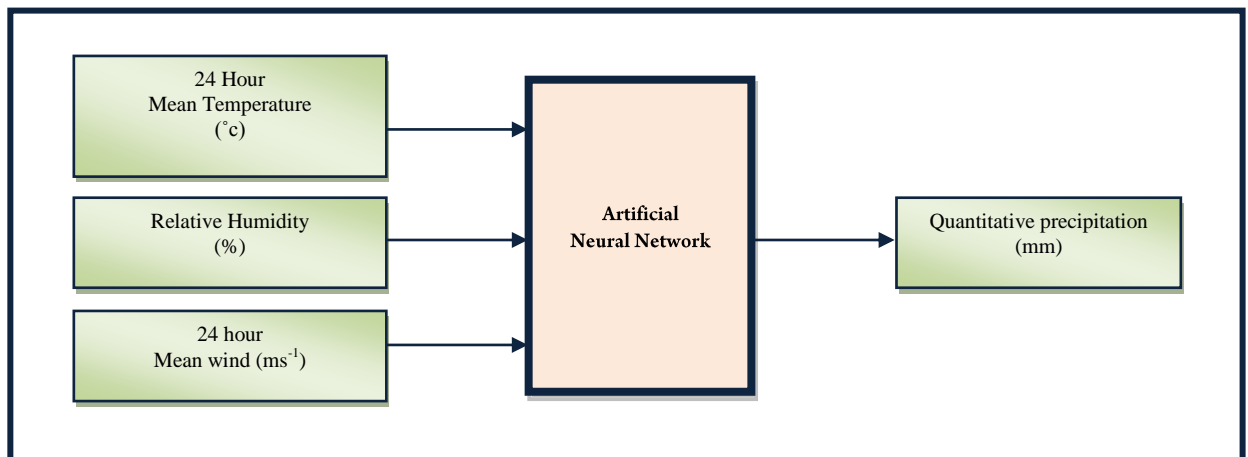


**Figure 3.14:** Regression plot

### 3.4 Constructing GUI

GUI was done for the purpose of assisting user of using the system developed in graphical representation. Here, `guide` was used to open template, open new or saved GUI for editing.

The GUI function is to generate quantization of precipitation with input variables are mean temperature, relative humidity and mean wind. To execute output program, push button was used. Pop-out menu used to list out types of training. The result was then displayed using edit text uicontrol same as the input key in uicontrol. Obvious additional menu also provide such as Read me, about and close using push button.



**Figure 3.15:** GUI ANN Input and Output Block Diagram

Besides, the technique on retrieving number from text box, calling dialog window, inserting image on axes, switch controlling code flow, executing other file using push button and developing function were also used on stage (2).

## **CHAPTER 4**

### **RESULT AND ANALYSIS**

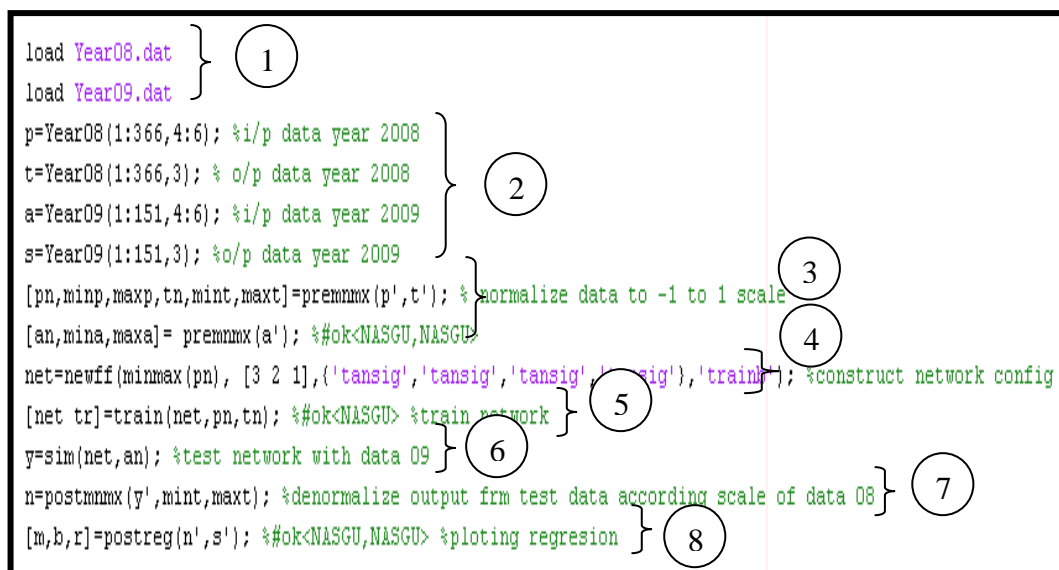
#### **4.1 Introduction**

As mention before, this project can be divided into (2) stages which are firstly analysis stage and secondly GUI development stages. The analysis was performed to search the best configuration of ANN representing weather data. GUI development else was about to prepared user friendly environment allowing user to view qualitative precipitation in Kuantan area.

#### **4.2 Neural Network Configuration**

##### **4.2.1 Programming Structure**

As mention at the previous chapter, the programming is simulated in MATLAB R2007a. The structure as shown in Figure 4.1 is used for stage 1. The modification is only made on the configuration of hidden layer and training type. Table 4.1 indicates details about the programming structure.



**Figure 4.1:** Training Programming Structure

**Table 4.1:** Description of programming structure

No.	Explanation
1.	Data in .dat format is loaded to workspace
2.	Assigning input/output variables
3.	Normalising data ranging [-1, 1]
4.	Constructing ANN: <ul style="list-style-type: none"> <li>▪ Newff means creating feedforward ANN</li> <li>▪ Minmax(pn) means the max and minimum input value is referred by pn value as assign at (3)</li> <li>▪ {'*','*',...} is the activation function corresponding to each layer. At this example it was Tansig.</li> <li>▪ '*' is the training type. At this example trainb</li> </ul>
5.	Training network / Network is learning
6.	Simulating the trained network (Default, epoch:100) with data 2009
7.	Denormalizing data into range of target data on 2008
8.	Plotting regression graphical of actual value with ANN developed value.

#### 4.2.2 Simulation Result

Table 4.2 below constructed to shows result on time consumption of difference training type. The (7) training types below are among the best training types that could provide good regression result with regard data 2009. While that, Table 4.3 shows all other training types available in MATLAB which is not suitable to be implement at the next stage due to high training time consumption and low accuracy. For time consumption analysis, all training types are train to network with 3 2 2 1 hidden layer configuration.

**Table 4.2:** Time Consumption corresponding difference training type (training type that suitable to implement at second stage)

Training Function	Time/s	Regression / R	GRADIENT/ MSE
Trainrp	1.92	0.73267	Epoch 0/100, MSE 1.26553/0, Gradient 3.60336/1e-006 Epoch 25/100, MSE 0.0359902/0, Gradient 0.00807687/1e-006 Epoch 50/100, MSE 0.0332362/0, Gradient 0.00227138/1e-006 Epoch 75/100, MSE 0.0321797/0, Gradient 0.00151153/1e-006 Epoch 100/100, MSE 0.0319482/0, Gradient 0.00132709/1e-006
Trainscg	2.24	0.71358	Epoch 0/100, MSE 0.428015/0, Gradient 2.57825/1e-006 Epoch 25/100, MSE 0.0452894/0, Gradient 0.000892809/1e-006 Epoch 50/100, MSE 0.0452801/0, Gradient 2.2854e-005/1e-006 Epoch 75/100, MSE 0.0452206/0, Gradient 0.00474354/1e-006 Epoch 100/100, MSE 0.0372313/0, Gradient 0.012722/1e-006
Trainoss	2.65	0.74096	Epoch 0/100, MSE 0.0547899/0, Gradient 0.0868433/1e-006 Epoch 25/100, MSE 0.0420652/0, Gradient 0.015537/1e-006 Epoch 50/100, MSE 0.038683/0, Gradient 0.00844786/1e-006 Epoch 75/100, MSE 0.0372757/0, Gradient 0.00539151/1e-006 Epoch 100/100, MSE 0.0358356/0, Gradient 0.0225047/1e-006
Traincgf	3.01	0.72121	Epoch 0/100, MSE 1.86637/0, Gradient 2.76429/1e-006 Epoch 25/100, MSE 0.0419963/0, Gradient 0.0829975/1e-006 Epoch 50/100, MSE 0.0362228/0, Gradient 0.0424302/1e-006 Epoch 75/100, MSE 0.0337451/0, Gradient 0.0141439/1e-006 Epoch 100/100, MSE 0.0332817/0, Gradient 0.00473269/1e-006
Trainbfg	3.18	0.25431	Epoch 0/100, MSE 0.174251/0, Gradient 1.09481/1e-006 Epoch 25/100, MSE 0.0380791/0, Gradient 0.0199351/1e-006 Epoch 50/100, MSE 0.0327267/0, Gradient 0.00498777/1e-006 Epoch 75/100, MSE 0.0325937/0, Gradient 0.00504845/1e-006 Epoch 100/100, MSE 0.032172/0, Gradient 0.00221845/1e-006



Trainlm	3.68	0.35588	Epoch 0/100, MSE 1.72376/0, Gradient 2.85442/1e-010 Epoch 25/100, MSE 0.0297398/0, Gradient 0.00602685/1e-010 Epoch 50/100, MSE 0.0289771/0, Gradient 0.00140866/1e-010 Epoch 75/100, MSE 0.02884/0, Gradient 0.00331317/1e-010 Epoch 100/100, MSE 0.0287225/0, Gradient 0.00177592/1e-010
Traincgb	4.02	0.73734	Epoch 0/100, MSE 0.0918388/0, Gradient 0.135081/1e-006 Epoch 25/100, MSE 0.0361597/0, Gradient 0.000498729/1e-006 Epoch 50/100, MSE 0.0347369/0, Gradient 0.00249653/1e-006 Epoch 75/100, MSE 0.0341495/0, Gradient 0.00211446/1e-006 Epoch 100/100, MSE 0.033975/0, Gradient 0.00326533/1e-006

Table 4.2 above is arranged by ascending time consumption. It shows that trainrp is the fastest training types. It followed by Trainscg which 0.32 s more time consumption. Trainoss with 0.41 s higher than trainscg and 0.36 s lesser than traincgb. Next is Trainbfg which more 0.17 s than the previous training but lower than 0.5 s than trainlm which 0.32 s faster than traincgb. However according to the mse error between every training type is lower than  $\pm 1$  s.

**Table 4.3:** Time Consumption corresponding to difference training types (training types that not suitable to implement at second stage)

Training Function	Time/s	Regression / R	GRADIENT / MSE / SSE
Trains	0.04	0.076499	-
Traingd	1.762	0.1676	Epoch 0/100, MSE 0.0975705/0, Gradient 0.103163/1e-010 Epoch 25/100, MSE 0.0947799/0, Gradient 0.10823/1e-010 Epoch 50/100, MSE 0.0917153/0, Gradient 0.113271/1e-010 Epoch 75/100, MSE 0.08837/0, Gradient 0.118112/1e-010 Epoch 100/100, MSE 0.0847504/0, Gradient 0.122523/1e-010
Traingdm	1.82	0.21866	Epoch 0/100, MSE 1.34939/0, Gradient 3.59528/1e-010 Epoch 25/100, MSE 0.0835198/0, Gradient 0.455007/1e-010 Epoch 50/100, MSE 0.0491997/0, Gradient 0.081022/1e-010 Epoch 75/100, MSE 0.0475397/0, Gradient 0.0530401/1e-010 Epoch 100/100, MSE 0.0468733/0, Gradient 0.0459604/1e-010
Traingda	1.891	0.3408	Epoch 0/100, MSE 2.5249/0, Gradient 1.28468/1e-006 Epoch 25/100, MSE 0.140548/0, Gradient 0.898372/1e-006 Epoch 50/100, MSE 0.0455178/0, Gradient 0.0142567/1e-006 Epoch 75/100, MSE 0.045424/0, Gradient 0.00239216/1e-006 Epoch 100/100, MSE 0.0453256/0, Gradient 0.00223135/1e-006

Trainidx	2.20	0.39132	Epoch 0/100, MSE 0.127511/0, Gradient 1.03044/1e-006 Epoch 25/100, MSE 0.0527983/0, Gradient 0.0281962/1e-006 Epoch 50/100, MSE 0.0522992/0, Gradient 0.0270245/1e-006 Epoch 75/100, MSE 0.0510739/0, Gradient 0.0181013/1e-006 Epoch 100/100, MSE 0.0487861/0, Gradient 0.0118828/1e-006
Trainb	2.933	0.19312	Epoch 0/100, MSE 2.8051/0. Epoch 25/100, MSE 2.73857/0. Epoch 50/100, MSE 2.58159/0. Epoch 75/100, MSE 2.00345/0. Epoch 100/100, MSE 0.280573/0.
Traincgp	3.405	0.55649	Epoch 0/100, MSE 0.663751/0, Gradient 2.67571/1e-006 Epoch 25/100, MSE 0.0420875/0, Gradient 0.00440629/1e-006 Epoch 50/100, MSE 0.0389939/0, Gradient 0.0108948/1e-006 Epoch 75/100, MSE 0.0381219/0, Gradient 0.00162758/1e-006 Epoch 100/100, MSE 0.0374121/0, Gradient 0.00699941/1e-006
Trainbr	15.052	0.48255	Epoch 0/100, SSE 780.169/0, SSW 50.47, Grad 9.46e+002/1.00e-010, #Par 2.90e+001/29 Epoch 25/100, SSE 12.7005/0, SSW 18.3016, Grad 6.33e-001/1.00e-010, #Par 8.05e+000/29 Epoch 50/100, SSE 11.4928/0, SSW 24.3919, Grad 6.28e-001/1.00e-010, #Par 1.52e+001/29 Epoch 75/100, SSE 11.1703/0, SSW 31.7786, Grad 2.09e-001/1.00e-010, #Par 1.70e+001/29 Epoch 100/100, SSE 11.0522/0, SSW 33.516, Grad 2.23e-001/1.00e-010, #Par 1.75e+001/29
Trainr	369.18	0.38361	-
Trainc	463.215	0.45058	-

Table 4.3 above show that trainc is worse training function. With 463.215 s time consumption the analysis on it mse was not be able to provide. All training functions show in this figure is those according to the result produce high error and or high time consumption which should be reduced. Thus it not been suitable to be implemented.

**Table 4.4:** The best network choose to be used at second stage

Training Function	Configuration	Regression / R	Weight/bias
Trainbfg	3 2 2 1	0.78686 (Fig.4.2)	$IW = \begin{bmatrix} 0.8962 & 1.3351 & 1.8323 \\ 0.7661 & 1.6226 & 0.8849 \\ -1.1991 & -0.8702 & 0.4462 \end{bmatrix}$ $LW\{2,1\} = \begin{bmatrix} -1.1991 & 2.3165 & 1.3794 \\ 1.0035 & -0.6518 & 1.3154 \end{bmatrix}$ $LW\{3,2\} = \begin{bmatrix} 5.2843 & -2.8561 \\ -1.3583 & 2.7107 \end{bmatrix}$

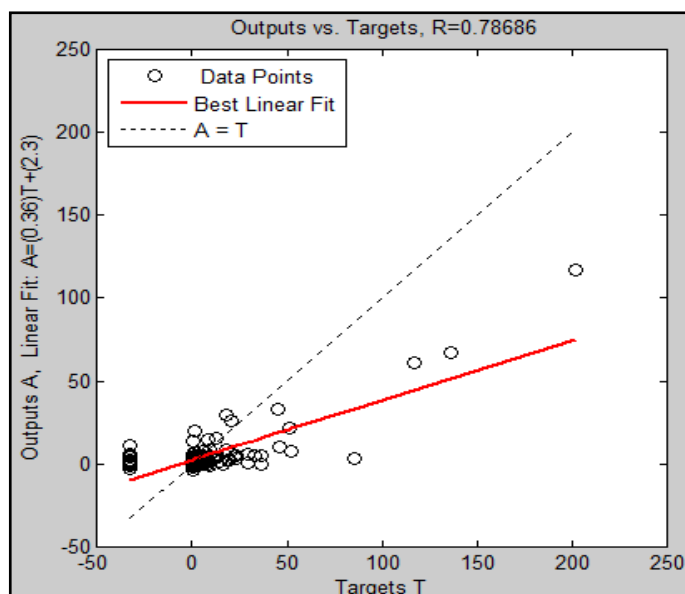
			$LW\{4,3\} = [ 4.3113 \quad -1.7096 ]$ $b\{1\} = \begin{bmatrix} -1.8386 \\ -1.7863 \\ -5.0238 \end{bmatrix}$ $b\{2\} = \begin{bmatrix} 0.3748 \\ -1.8658 \end{bmatrix}$ $b\{3\} = \begin{bmatrix} 1.7563 \\ 1.1867 \end{bmatrix}$ $b\{4\} = [0.8428]$
Trainseg	3 2 1	0.77729 (Fig.4.8)	$IW = \begin{bmatrix} -0.1251 & -1.7894 & 1.8462 \\ 0.6415 & 2.1492 & -0.4638 \\ 0.0914 & -0.2745 & -0.9062 \end{bmatrix}$ $LW\{2,1\} = \begin{bmatrix} 0.4867 & 0.9827 & 1.3557 \\ 1.3781 & 2.3515 & -0.3522 \end{bmatrix}$ $LW\{3,2\} = [-0.2572 \quad 1.5217]$ $b\{1\} = \begin{bmatrix} 1.9336 \\ -1.0687 \\ 2.2301 \end{bmatrix}$ $b\{2\} = \begin{bmatrix} 2.3812 \\ -1.3628 \end{bmatrix}$ $b\{3\} = [0.8235]$
Trainlm	2 2 2 2 1	0.77361 (Fig.4.5)	$IW = \begin{bmatrix} -0.4296 & -1.5140 & 0.7881 \\ -0.4236 & -1.3126 & 0.4253 \end{bmatrix}$ $LW\{2,1\} = \begin{bmatrix} -1.4776 & 2.5992 \\ -1.0255 & -1.0470 \end{bmatrix}$ $LW\{3,2\} = \begin{bmatrix} -1.4923 & -1.1633 \\ -2.7479 & -0.2769 \end{bmatrix}$ $LW\{4,3\} = \begin{bmatrix} 0.6071 & -2.9899 \\ 0.1848 & -1.9678 \end{bmatrix}$ $LW\{5,4\} = [-1.3877 \quad -0.1841]$ $b\{1\} = \begin{bmatrix} 0.9758 \\ 0.3331 \end{bmatrix}$ $b\{2\} = \begin{bmatrix} 1.4468 \\ -2.7809 \end{bmatrix}$ $b\{3\} = \begin{bmatrix} 3.1609 \\ 0.0423 \end{bmatrix}$ $b\{4\} = \begin{bmatrix} -1.6994 \\ 1.9755 \end{bmatrix}$ $b\{5\} = [0.5667]$
Trainoss	3 2 2 1	0.76511 (Fig.4.6)	$IW = \begin{bmatrix} 0.1010 & -1.0350 & 1.8001 \\ 0.5332 & 1.8871 & 0.4661 \\ -0.4794 & 0.4925 & -1.0977 \end{bmatrix}$ $LW\{2,1\} = \begin{bmatrix} -1.5155 & -1.7577 & -0.7465 \\ 0.3257 & 1.2799 & -1.6832 \end{bmatrix}$

			$LW\{3,2\} = \begin{bmatrix} -0.2655 & -1.1113 \\ 1.6265 & -2.0112 \end{bmatrix}$ $LW\{4,3\} = [ 0.3197 \quad -1.4453 ]$ $b\{1\} = \begin{bmatrix} 1.6438 \\ -0.7555 \\ -2.7845 \end{bmatrix}$ $b\{2\} = \begin{bmatrix} 1.3297 \\ -0.9282 \end{bmatrix}$ $b\{3\} = \begin{bmatrix} -3.0331 \\ 1.4973 \end{bmatrix}$ $b\{4\} = [0.8213]$
Traincgf	3 2 2 1	0.75397 (Fig.4.4)	$IW = \begin{bmatrix} -1.1560 & 0.0074 & -0.6104 \\ -0.1182 & -3.5330 & -1.0536 \\ 1.0373 & 1.1760 & -0.6578 \end{bmatrix}$ $LW(2,1) = \begin{bmatrix} 0.6918 & -0.4624 & -1.1334 \\ -0.7839 & -1.7898 & 0.5875 \end{bmatrix}$ $LW\{3,2\} = \begin{bmatrix} 0.7866 & 1.8173 \\ -0.8833 & -2.1204 \end{bmatrix}$ $LW\{4,3\} = [ -0.4521 \quad -1.3558 ]$ $b\{1\} = \begin{bmatrix} 2.8947 \\ 1.2326 \\ 2.6102 \end{bmatrix}$ $b\{2\} = \begin{bmatrix} -1.8675 \\ -2.0875 \end{bmatrix}$ $b\{3\} = \begin{bmatrix} -1.9802 \\ -1.6659 \end{bmatrix}$ $b\{4\} = [-0.1906]$
Traincgb	3 2 2 1	0.75317 (Fig.4.3)	$IW = \begin{bmatrix} 1.0984 & -1.0777 & -5.1767 \\ -1.6000 & 3.1188 & -2.1256 \\ 1.4279 & 2.7512 & 1.7846 \end{bmatrix}$ $LW(2,1) = \begin{bmatrix} -0.7322 & 3.8322 & 3.6968 \\ -0.3085 & -1.1960 & 1.2246 \end{bmatrix}$ $LW\{3,2\} = \begin{bmatrix} 0.6311 & 1.9438 \\ 2.1571 & 0.1296 \end{bmatrix}$ $LW\{4,3\} = [ 0.6586 \quad 0.9573 ]$ $b\{1\} = \begin{bmatrix} 6.9772 \\ 0.0579 \\ -2.2228 \end{bmatrix}$ $b\{2\} = \begin{bmatrix} -0.9790 \\ -1.7105 \end{bmatrix}$ $b\{3\} = \begin{bmatrix} -2.2303 \\ 2.0807 \end{bmatrix}$ $b\{4\} = [-0.0822]$

Trainrp	3 2 2 1	0.75095 (Fig.4.7)	$IW = \begin{bmatrix} 1.1105 & 5.0026 & -1.2660 \\ -0.5427 & -1.3409 & -2.0210 \\ 4.7867 & 2.0790 & -3.8728 \end{bmatrix}$ $LW(2,1) = \begin{bmatrix} 1.9315 & -2.4176 & 0.5486 \\ 1.4088 & -1.5272 & 0.3613 \end{bmatrix}$ $LW\{3,2\} = \begin{bmatrix} -2.1865 & -0.7800 \\ -1.3919 & -0.2494 \end{bmatrix}$ $LW\{4,3\} = \begin{bmatrix} -1.0890 & -0.6896 \end{bmatrix}$ $b\{1\} = \begin{bmatrix} -1.7387 \\ 0.1065 \\ 1.8432 \end{bmatrix}$ $b\{2\} = \begin{bmatrix} -2.8793 \\ 1.6981 \end{bmatrix}$ $b\{3\} = \begin{bmatrix} 2.1181 \\ -1.7273 \end{bmatrix}$ $b\{4\} = \begin{bmatrix} -0.0670 \end{bmatrix}$

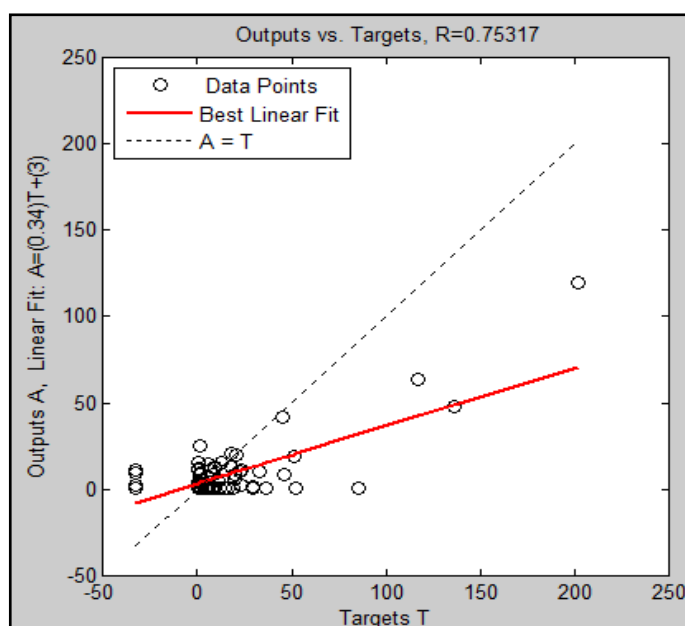
Table 4.4 above are the detail of weight and bias value corresponding to their training types that been used in stage 2.

Result from table 4.2 arranged training types with ascending time consumption. From the result, shows that trainrp is the fastest with 1.92 s. However according to performance analysis trainlm is the best with mse value 0.0287225 which is 11.23 % smaller than trainrp comparing with its time consumption which is 1.76 s longer.



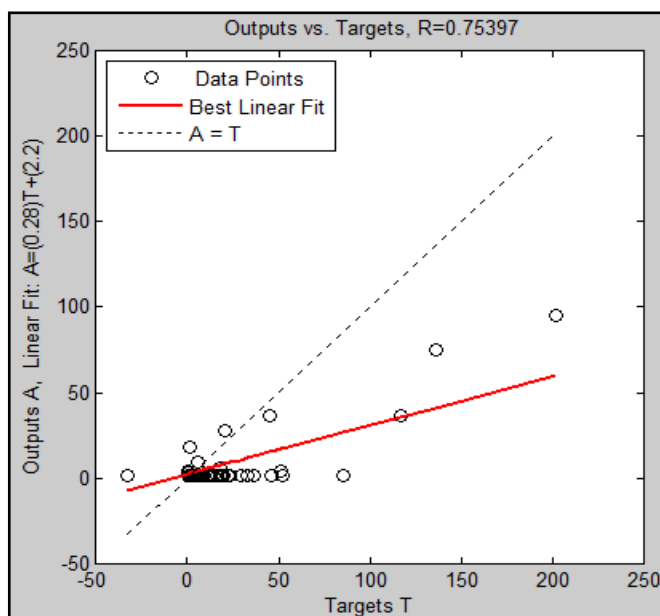
**Figure 4.2:** BFGS quasi-Newton backpropagation Training Function

Figure 4.2 is the result of BFGS quasi-newton backpropagation training function regression plot corresponding to the real data of 2009. The plot shows that it could produce up to 78.69% accuracy.



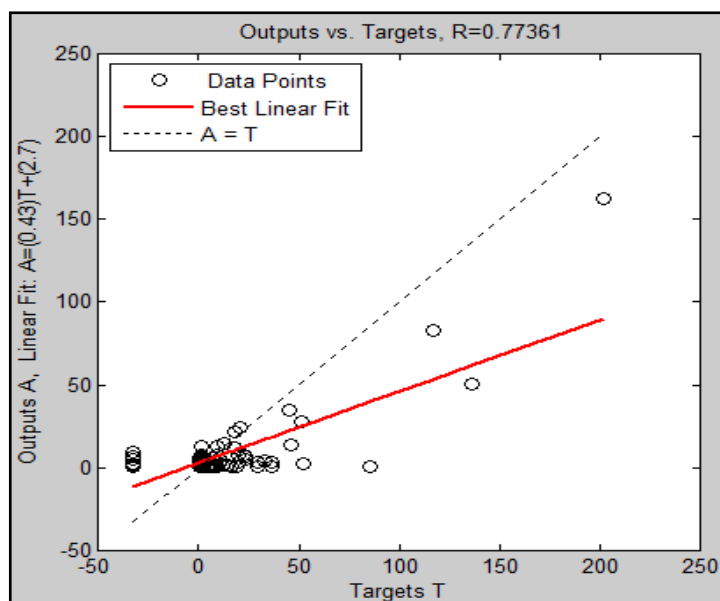
**Figure 4.3:** Powell-Beale conjugate gradient backpropagation Training Function

Figure 4.3 is the result of Powell-Beale conjugate gradient backpropagation training function regression plot corresponding to the real data of 2009. The plot shows that it could produce up to 75.32% accuracy.



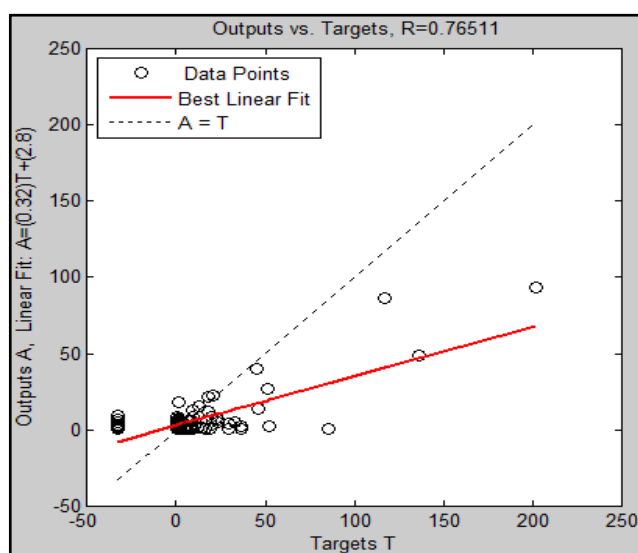
**Figure 4.4:** Fletcher-Powell conjugate gradient backpropagation Training Function

Figure 4.3 is the result of Fletcher-Powell conjugate gradient backpropagation training function regression plot corresponding to the real data of 2009. The plot shows that it could produce up to 75.40% accuracy.



**Figure 4.5:** Levenberg Marquardt Training Function

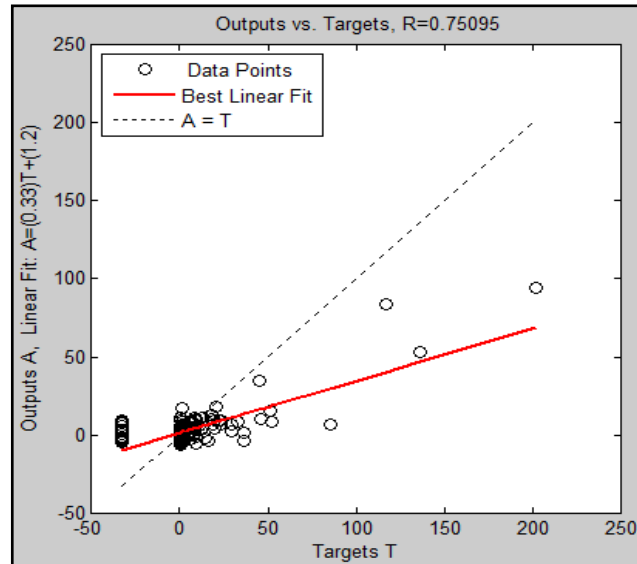
Figure 4.5 is the result of Levenberg Marquardt training function regression plot corresponding to the real data of 2009. The plot shows that it could produce up to 77.36% accuracy.



**Figure 4.6:** One step secant backpropagation Training Function

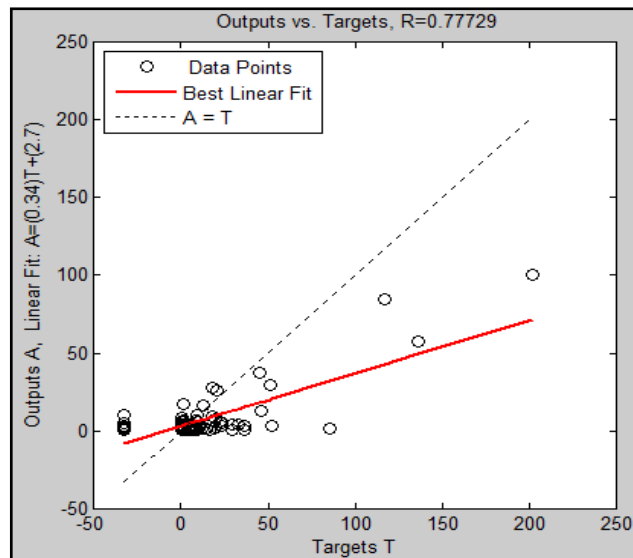


Figure 4.6 is the result of One step secant backpropagation training function regression plot corresponding to the real data of 2009. The plot shows that it could produce up to 76.51% accuracy.



**Figure 4.7:** Resilient backpropagation Training Function

Figure 4.7 is the result of Resilient backpropagation training function regression plot corresponding to the real data of 2009. The plot shows that it could produce up to 76.51% accuracy.



**Figure 4.8:** Scaled conjugate gradient backpropagation Training Function

Figure 4.8 is the result of Scaled conjugate gradient backpropagation training function regression plot corresponding to the real data of 2009. The plot shows that it could produce up to 76.51% accuracy.

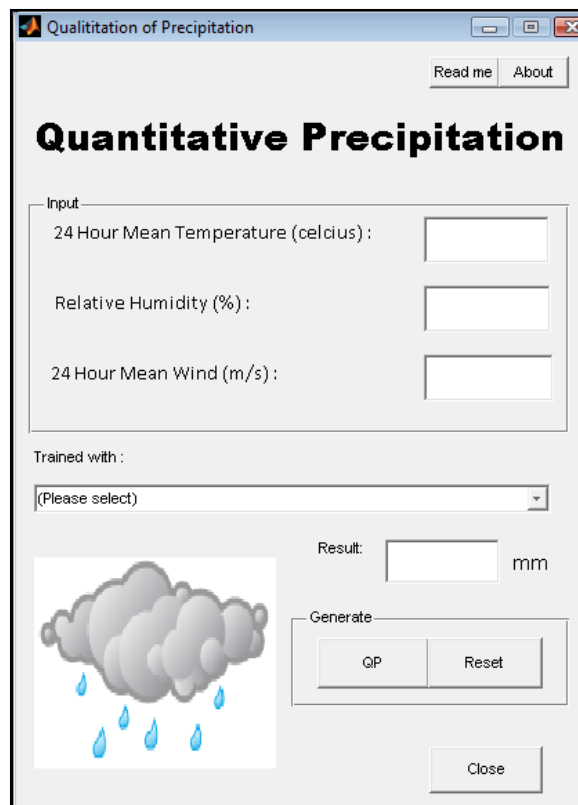
From the figures above, trainbfg provide the best regression value that is 78.69% accuracy. If compared with its time consumption from Table 4.2, it utilized 1.26 s times longer to trainrp and its accuracy which 4.56 % better.

All training types listed on Table 4.4 is among the best ANN configuration that can be generated. All its information is used to create GUI.

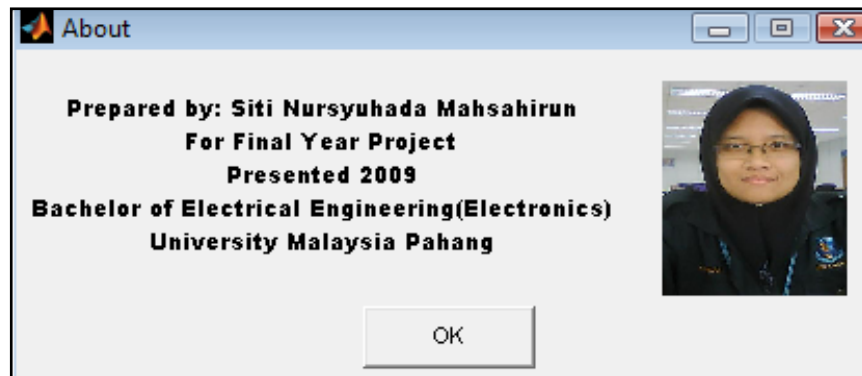
## 4.3 Graphical User Interface

### 4.3.1 Layout

Figure 4.9 shows the main window of the quantitative precipitation calculator. The basic uicontrol such as Readme, About and Close pushbutton was included. Input textboxes was grouped together using panel function. For multiple choice network type, pop-up menu was used. The output display using textbox to standardize with input uicontrol. Reset and QP button function are mutually close related. Thus, it was group together near each other using button group.

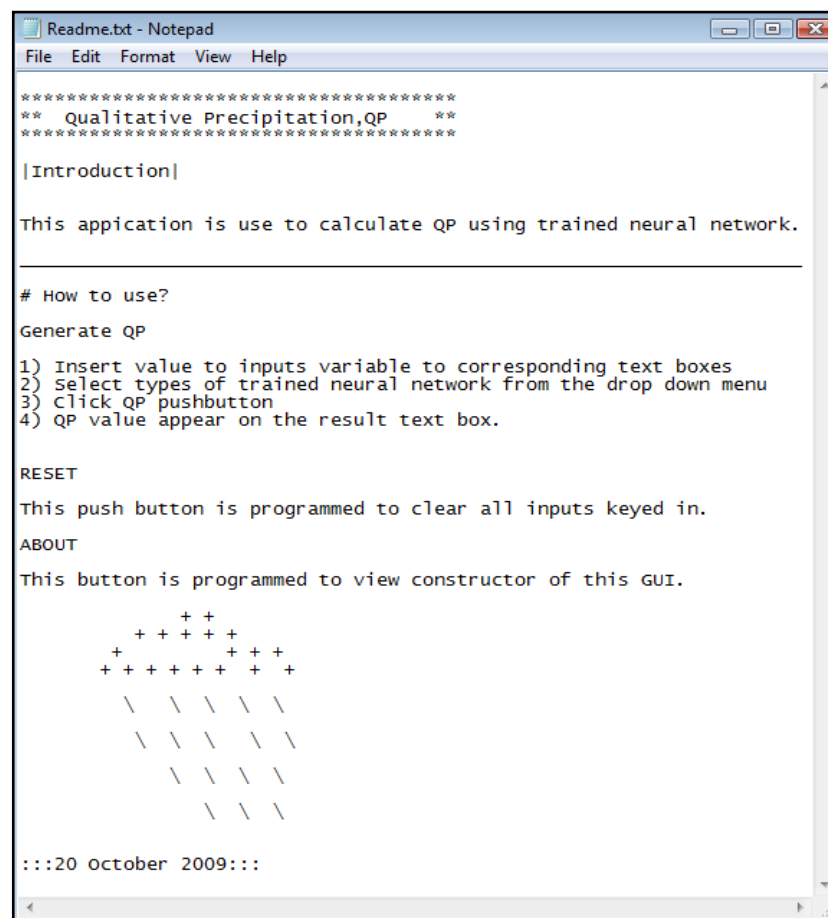


**Figure 4.9:** Graphical User Interface representation



**Figure 4.10:** About window

About dialog window was programmed separately from the quantitative precipitation window (parent window). It was just being called back using About pushbutton programmed at the parent window.



**Figure 4.11:** Read me text

Read me was simply wrote in notepad and save using .txt format. It just called back from the parent window using Read me push button.

## CHAPTER 5

### CONCLUSION AND RECOMMENDATION

#### 5.1 Conclusion

As qualitative precipitation with regard temperature, humidity and wind speed as input variables shows that BFGS quasi-Newton FFBP neural network has provide 78.69 % accuracy with high speed training 3.18 s.

There are also (3) training function which the result was failed to established. Those are the Sequential order incremental training w/learning functions, Random order incremental training w/learning function and Cyclical order incremental training w/learning function as none acceptable regression value could generated by the software used. Thus, the worse algorithm in term of accuracy is unable to determine. Both Random and Cyclical order incremental training w/learning function consume worse speeds which are 369.18 and 463.215 s respectively. Besides, the worse configuration also is not being able to determine because the training of variety hidden layer could take up days of time for each of them.

**Table 5.1:** Summary of the Analysis

	<b>Accepted value</b>	<b>Best</b>	<b>Worse</b>
<b>Accuracy</b>	> 70 percent	Trainbfg	
<b>Speed</b>	< 5 second	Trainrp	Trainc
<b>Network Configuration</b>	Configuration that software managed to produced result.	3 2 2 1	

ANN is suitable in controlling and understanding the way of non-linear set of data and variables to become mutually correlated with each other. It is a powerful yet effective method in embedding intelligent system to application.

Graphical User Interface is the best approach on developing user friendly ANN application. It is able make civilian event without ANN knowledge to use the network constructed easily. Thus, the benefits of ANN understanding and knowledge could be shared broaden.

## **5.2 Future Recommendation**

For future, research on generating ANN algorithm that show relationship between weather parameter could bring the new approach of designing weather hardware equipment. By embedding ANN system in monitoring weather equipment, cost can be reduced due to expensive hardware. This can be done by extracting mathematical equation from the best trained network with high performance and accuracy. The research result could also become better as many data used for the training process. Hence, the more reliable network can be constructed.

Research on this field could also improved by combining it with Fuzzy Logic System. Thus, expert body on weather could also involve in developing weather

forecasting based on Neuro-Fuzzy System, NFS that recently has been used broadly in so many application including forecasting.

Knowledge on ANN can also be implemented in other application especially in medical field. For example in diagnosing patient, it was usually involves history and symptoms which is in ANN language is *Pattern Recognition*. Or in pharmaceutical when prescribing drugs which *Function Approximation* problem type in ANN application. Thus, lack of medical officer problem can be overcome.

### **5.3 Costing and Commercialization**

This project is involving software and simulation only. Software product is the core material in this project, besides trained and test data for delivery and transportation. The total cost of developing the project is affordable which is around RM150.

Weather forecasting using ANN system could actually brings of beneficial in for meteorology department to implement further research and development in developing intelligent tools and devices. Besides, it may also may used in business such as agriculture and sea product, aviation, and building contractor which weather could determined their activities plus effecting their revenue.



## REFERENCE

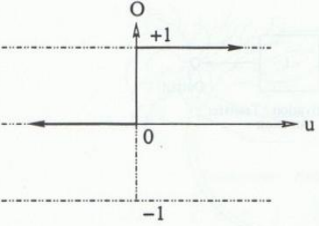
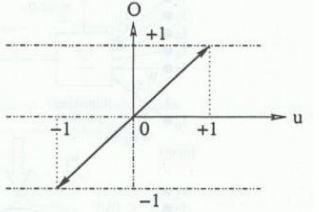
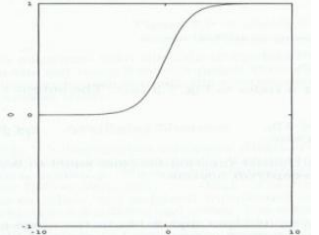
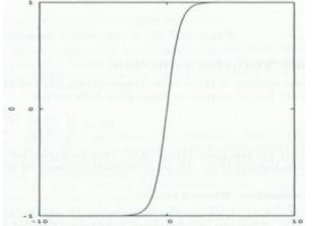
- [1] D.K. Pratihari, Soft Computing, United Kingdom: Alpha Science International Ltd. Oxford, 2008.
- [2] Devendra K. Chaturvedi, Soft Computing Techniques and its Applications in Electrical Engineering, Germany: Springer-Verlag Berlin Heidelberg, 2008.
- [3] Geller, T., Envisioning the Wind: Meteorology Graphics at Weather Underground, Computer Graphics and Application, IEEE, 2007. (Vol 27, Pg 92-97)
- [4] Ghanbarzadeh, A., Noghrehabadi A.R., Assareh E., Behrang M.A., Solar radiation forecasting based on meteorological data using artificial neural networks, INDIN 2009 7<sup>TH</sup> IEEE International Conference, 2009. (Pg 227-231)
- [5] Haykin, S., Neural Networks, New York: Macmillan College Publishing Company Inc, 1994.(pg 36-37)
- [6] Howard Demuth, Mark Beale and Morgan Hagan, Neural Network Toolbox 6, Natick MA, US: The MathWorks, Inc., 2009.
- [7] Jiansheng Wu., A Novel Artificial Neural Network Ensemble Model Based on K--Nearest Neighbor Nonparametric Estimation of Regression Function and Its Application for Rainfall Forecasting, International Joint Conference on Computational Science and Optimization, 2009. (Vol 2, Pg 44-48)
- [8] Karmakar, S.; Kowar, M.K.; Guhathakurta, P., Development of an 8-Parameter Probabilistic Artificial Neural Network Model for Long-Range Monsoon Rainfall Pattern Recognition over the Smaller Scale Geographical Region,

ICETET '08. First International Conference on Emerging Trends in Engineering Technology, 2008. (Pg 569)

- [9] Kilon, J.; Saeed-Baginska, A.; Sienkiewicz, J.; Mosdorf, R., Nonlinear and Frequency Analysis of the Weather in Bialystok, IEEE: Conference Information Systems and Industrial Management Applications, 6th International Conference, 2007. (Pg 310-315)
- [10] Marzban, C., G. J. Stumpf, A neural network for tornado prediction Based on Doppler Radar-derived Attributes., Journal of Applied Meteorology, 1996. (Vol. 35, 617)
- [11] Methaprayoon, K., Wei-Jen Lee, Rasmiddatta, S., Liao, J.R., Ross, R.J., Multistage Artificial Neural Network Short-Term Load Forecasting Engine With Front-End Weather Forecast, Industry Applications, IEEE Transactions, 2007. (Vol 43, 1410-1416)
- [12] National Weather Service Arkansas-Red Basin River Forecast Center, “ABRFC Forecast Methodology” [online], Last modified 6<sup>th</sup> January 2009, [cited 2009] <http://www.srh.noaa.gov/abrfc/fcstmethods.shtml>
- [13] Takahide Niimura, Ryuichi Yokoyama, Water Level Control of Small-scale Hydro-Generating Units by Fuzzy Logic, Canada/Japan: Universiti of British Columbia Canada and Tokyo Metropolitan University Hachioji Tokyo Japan, 1995.
- [14] The MathWorks, MATLAB 7 Creating Graphical User Interfaces, Natick MA, US: The MathWorks, Inc., 2009.
- [15] Tony Hall et al., “Precipitation Forecasting Using Neural Network” [Online documanet], 1999, [cited 2009 July], Available HTTP: [http://www.nssl.noaa.gov/users/brooks/public\\_html/papers/hallbrooksoswell.pdf](http://www.nssl.noaa.gov/users/brooks/public_html/papers/hallbrooksoswell.pdf)

## APPENDIX 1

### Transfer Function Types

NO	TYPE	OUTPUT VALUE, O	REPRESENTATION	
			MATHEMATICAL	GRAPHICAL
1	Hard-limit Transfer Function	1.0 or 0.0	$O = \begin{cases} 0.0 & \text{if } u < 0.0 \\ 1.0 & \text{otherwise} \end{cases}$	
2	Linear Transfer Function	-1.0 to 1.0	$O = u$	
3	Log-Sigmoid Transfer Function	0.0 to 1.0	$O = \frac{1}{1 + e^{-au}}$	
4	Tan-Sigmoid Transfer Function	-1.0 to 1.0	$O = \frac{e^{au} - e^{-au}}{e^{au} + e^{-au}}$	

**APPENDIX 2****Stage 1: ANN Training Codes**

```
load Year08.dat
load Year09.dat
p=Year08(1:366,4:6);
t=Year08(1:366,3);
a=Year09(1:151,4:6);
s=Year09(1:151,3);
[pn,minp,maxp,tn,mint,maxt]=premnmx(p',t');
[an,mina,maxa]=premnmx(a');
net=newff(minmax(pn),[* *
*],{'***','***','***'},'****');
[net tr]=train(net,pn,tn);
y=sim(net,an);
n=postmnmx(y',mint,maxt);
[m,b,r]=postreg(n',s');

% notes: * imply the manipulated parameter
```

## APPENDIX 3

### Stage 2: GUI Parent Window Codes

```

%***** THIS IS THE PROGRAM OF THE PARENT WINDOW
%*****%

function varargout = weather(varargin)
% WEATHER M-file for weather.fig
% WEATHER, by itself, creates a new WEATHER or raises the
existing
% singleton*.
%
% H = WEATHER returns the handle to a new WEATHER or the handle
to
% the existing singleton*.
%
% WEATHER('CALLBACK',hObject,eventData,handles,...) calls the
local
% function named CALLBACK in WEATHER.M with the given
input_panel arguments.
%
% WEATHER('Property','Value',...) creates a new WEATHER or
raises the
% existing singleton*. Starting from the left, property value
pairs are
% applied to the GUI before weather_OpeningFunction gets
called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to weather_OpeningFcn via
varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help weather

% Last Modified by GUIDE v2.5 20-Oct-2009 23:14:07

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @weather_OpeningFcn, ...
                  'gui_OutputFcn', @weather_OutputFcn, ...

```

```

        'gui_LayoutFcn', [] , ...
'gui_Callback', []);if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
Else

gui_mainfcn(gui_State, varargin{:});

end

% End initialization code - DO NOT EDIT

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes just before weather is made visible.
function weather_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to weather (see VARARGIN)

% Choose default command line output for weather
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes weather wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = weather_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% imageaxes %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%display image on axes
[x]=imread('ShowersScattered','BMP');
image(x)
set(gca,'visible','off')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% pushbutton %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in readme_Pb.
function readme_Pb_Callback(hObject, eventdata, handles)
% hObject    handle to readme_Pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

dos('notepad D:\psm\psm2\NNdatabase\Readme.txt&');

% --- Executes on button press in About_Pb.
function About_Pb_Callback(hObject, eventdata, handles)
% hObject    handle to About_Pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Figure(about)
pos_size = get(handles.figure1, 'Position');
% Call modaldlg with the argument 'Position'.
user_response = about('Title', 'Confirm Close');

% --- Executes on button press in Reset_Pb.
function reset_pb_Callback(hObject, eventdata, handles)
% hObject    handle to Reset_Pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.processDataCompleted = 0;

%set the popupmenu to default value
set(handles.TrainingTypes, 'Value', 1);

%clears the contents of the textbox
set(handles.temp, 'String', '');
set(handles.humid, 'String', '');
set(handles.wind, 'String', '');
set(handles.result, 'string', '');

%updates the handles structure
guidata(hObject, handles);

```

```

% --- Executes on button press in Close_Pb.
function Close_Pb_Callback(hObject, eventdata, handles)
% hObject    handle to Close_Pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get the current position of the GUI from the handles structure
% to pass to the modal dialog.
pos_size = get(handles.figure1, 'Position');
% Call modaldlg with the argument 'Position'.
user_response = Dialog('Title', 'Confirm Close');
switch user_response
case 'No'
    % take no action
case 'Yes'
    % Prepare to close GUI application window
    %
    %
    %
    delete(handles.figure1)
end

```

```

% --- Executes on button press in qp_pb.
function qp_pb_Callback(hObject, eventdata, handles)
% hObject    handle to qp_pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

HH1 = findobj('Tag', 'temp');
HH2 = findobj('Tag', 'humid');
HH3 = findobj('Tag', 'wind');
HH4 = findobj('Tag', 'result');

v1 = str2double(get(HH1, 'string'));
v2 = str2double(get(HH2, 'string'));
v3 = str2double(get(HH3, 'string'));
mint=-33.3;
maxt=218.6;

popup_sel_index = get(handles.TrainingTypes, 'value');

switch popup_sel_index

case 1 % User selects
    set(HH4, 'string', 'xxxxx');

case 2 % User selects
    op=bfq(v1, v2, v3);
    n=postmmx(op, mint, maxt);

```



```

        set(HH4, 'string', n);

    case 3 % User selects
        op=cgb(v1,v2,v3);
        n=postmnmx(op',mint,maxt);
        set(HH4, 'string', n);

    case 4 % User selects
        op=cgf(v1,v2,v3);
        n=postmnmx(op',mint,maxt);
        set(HH4, 'string', n);

    case 5 % User selects
        op=lm(v1,v2,v3);
        n=postmnmx(op',mint,maxt);
        set(HH4, 'string', n);

    case 6 % User selects
        op=oss(v1,v2,v3);
        n=postmnmx(op',mint,maxt);
        set(HH4, 'string', n);

    case 7 % User selects
        op=rp(v1,v2,v3);
        n=postmnmx(op',mint,maxt);
        set(HH4, 'string', n);

    case 8 % User selects
        op=scg(v1,v2,v3);
        n=postmnmx(op',mint,maxt);
        set(HH4, 'string', n);

    otherwise
end

% Save the handles structure.
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function qp_pb_CreateFcn(hObject, eventdata, handles)
% hObject    handle to qp_pb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% --- Executes on selection change in TrainingTypes.
function TrainingTypes_Callback(hObject, eventdata, handles)
% hObject    handle to TrainingTypes (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns TrainingTypes
contents as cell array
% contents{get(hObject,'Value')} returns selected item from
TrainingTypes

% --- Executes during object creation, after setting all properties.
function TrainingTypes_CreateFcn(hObject, eventdata, handles)
% hObject handle to TrainingTypes (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

function temp_Callback(hObject, eventdata, handles)
% hObject handle to Temp (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Temp as text
% str2double(get(hObject,'String')) returns contents of Temp
as a double

%store the contents of input1_editText as a string. if the string
%is not a number then input_panel will be empty
input = str2num(get(hObject,'String'));

%checks to see if input_panel is empty. if so, default
input1_editText to zero
if (isempty(input))
set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function temp_CreateFcn(hObject, eventdata, handles)
% hObject handle to Temp (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function humid_Callback(hObject, eventdata, handles)
% hObject      handle to Humid (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Humid as text
%       str2double(get(hObject,'String')) returns contents of Humid
as a double

%store the contents of input1_editText as a string. if the string
%is not a number then input_panel will be empty
input = str2num(get(hObject,'String'));

%checks to see if input_panel is empty. if so, default
input1_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function humid_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Humid (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function wind_Callback(hObject, eventdata, handles)
% hObject      handle to Wind (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Wind as text
%         str2double(get(hObject,'String')) returns contents of Wind
as a double

%store the contents of input1_editText as a string. if the string
%is not a number then input_panel will be empty
input = str2num(get(hObject,'String'));

%checks to see if input_panel is empty. if so, default
input1_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function wind_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Wind (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function result_Callback(hObject, eventdata, handles)
% hObject      handle to result (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of result as text
%         str2double(get(hObject,'String')) returns contents of
result as a double

% --- Executes during object creation, after setting all properties.
function result_CreateFcn(hObject, eventdata, handles)
% hObject      handle to result (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUiControlBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
%*****  
%***** END OF THE PROGRAM *****  
%*****
```

## APPENDIX 4

### Stage 2: About Window Codes

```

% ***** THIS IS THE PROGRAM OF ABOUT WINDOW ***** %

function varargout = about(varargin)
% ABOUT M-file for about.fig
%     ABOUT, by itself, creates a new ABOUT or raises the existing
%     singleton*.
%
%     H = ABOUT returns the handle to a new ABOUT or the handle to
%     the existing singleton*.
%
%     ABOUT('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in ABOUT.M with the given input
arguments.
%
%     ABOUT('Property','Value',...) creates a new ABOUT or raises
the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before about_OpeningFunction gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to about_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help about

% Last Modified by GUIDE v2.5 20-Oct-2009 23:41:07

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @about_OpeningFcn, ...
                  'gui_OutputFcn',  @about_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

```

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before about is made visible.
function about_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to about (see VARARGIN)

% Choose default command line output for about
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes about wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = about_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%display image on axes
[x]=imread('myImage','jpg');
image(x)
set(gca,'visible','off')

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

close

% ***** THIS IS THE EN OF THE PROGRAM ***** %

```

## APPENDIX 5

### Stage 2: Close Dialog Box Codes

```

% ***** THIS IS THE PROGRAM OF CLOSE DIALOG WINDOW *****
%
function varargout = Dialog(varargin)
% DIALOG M-file for Dialog.fig
%     DIALOG by itself, creates a new DIALOG or raises the
%     existing singleton*.
%
%     H = DIALOG returns the handle to a new DIALOG or the handle
to
%     the existing singleton*.
%
%     DIALOG('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in DIALOG.M with the given input
arguments.
%
%     DIALOG('Property','Value',...) creates a new DIALOG or raises
the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before Dialog_OpeningFunction gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Dialog_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Dialog

% Last Modified by GUIDE v2.5 19-Oct-2009 14:03:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Dialog_OpeningFcn, ...
                  'gui_OutputFcn',  @Dialog_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

```



```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Dialog is made visible.
function Dialog_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Dialog (see VARARGIN)

% Choose default command line output for Dialog
handles.output = 'Yes';

% Update handles structure
guidata(hObject, handles);

% Insert custom Title and Text if specified by the user
% Hint: when choosing keywords, be sure they are not easily confused
% with existing figure properties.  See the output of set(figure)
for
% a list of figure properties.
if(nargin > 3)
    for index = 1:2:(nargin-3),
        if nargin-3==index, break, end
        switch lower(varargin{index})
            case 'title'
                set(hObject, 'Name', varargin{index+1});
            case 'string'
                set(handles.text1, 'String', varargin{index+1});
            end
        end
    end
end

% Determine the position of the dialog - centered on the callback
figure
% if available, else, centered on the screen
FigPos=get(0, 'DefaultFigurePosition');
OldUnits = get(hObject, 'Units');
set(hObject, 'Units', 'pixels');
OldPos = get(hObject, 'Position');
FigWidth = OldPos(3);
FigHeight = OldPos(4);
if isempty(gcbf)
    ScreenUnits=get(0, 'Units');
    set(0, 'Units', 'pixels');
    ScreenSize=get(0, 'ScreenSize');

```

```

set(0, 'Units', ScreenUnits);

FigPos(1)=1/2*(ScreenSize(3)-FigWidth);
FigPos(2)=2/3*(ScreenSize(4)-FigHeight);
else
GCBFOldUnits = get(gcbf, 'Units');
set(gcbf, 'Units', 'pixels');
GCBFPos = get(gcbf, 'Position');
set(gcbf, 'Units', GCBFOldUnits);
FigPos(1:2) = [(GCBFPos(1) + GCBFPos(3) / 2) - FigWidth / 2, ...
              (GCBFPos(2) + GCBFPos(4) / 2) - FigHeight / 2];
end
FigPos(3:4)=[FigWidth FigHeight];
set(hObject, 'Position', FigPos);
set(hObject, 'Units', OldUnits);

% Show a question icon from dialogicons.mat - variables
questIconData
% and questIconMap
load dialogicons.mat

IconData=questIconData;
questIconMap(256,:) = get(handles.figure1, 'Color');
IconCMap=questIconMap;

Img=image(IconData, 'Parent', handles.axes1);
set(handles.figure1, 'Colormap', IconCMap);

set(handles.axes1, ...
      'Visible', 'off', ...
      'YDir'    , 'reverse' , ...
      'XLim'    , get(Img, 'XData'), ...
      'YLim'    , get(Img, 'YData') ...
      );

% Make the GUI modal
set(handles.figure1, 'WindowStyle', 'modal')

% UIWAIT makes Dialog wait for user response (see UIRESUME)
uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Dialog_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

```

```

% The figure can be deleted now
delete(handles.figure1);

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.output = get(hObject, 'String');

% Update handles structure
guidata(hObject, handles);

% Use UIRESUME instead of delete because the OutputFcn needs
% to get the updated handles structure.
uiresume(handles.figure1);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.output = get(hObject, 'String');

% Update handles structure
guidata(hObject, handles);

% Use UIRESUME instead of delete because the OutputFcn needs
% to get the updated handles structure.
uiresume(handles.figure1);
% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

if isequal(get(handles.figure1, 'waitstatus'), 'waiting')
    % The GUI is still in UIWAIT, us UIRESUME
    uiresume(handles.figure1);
else
    % The GUI is no longer waiting, just close it
    delete(handles.figure1);
end

% --- Executes on key press over figure1 with no controls selected.
function figure1_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```
% handles      structure with handles and user data (see GUIDATA)

% Check for "enter" or "escape"
if isequal(get(hObject,'CurrentKey'),'escape')
    % User said no by hitting escape
    handles.output = 'No';

    % Update handles structure
    guidata(hObject, handles);

    uiresume(handles.figure1);
end

if isequal(get(hObject,'CurrentKey'),'return')
    uiresume(handles.figure1);
end
% ***** THIS IS THE END OF THE PROGRAM ***** %
```

## APPENDIX 6

### Function Created Codes

```
function [qp]=bfg(temp,humid,wind)
```

```
load newton.mat  
i=premnmx([temp,humid,wind]);  
qp=sim(net,i');
```

```
function [qp]=cgb(temp,humid,wind)
```

```
load pobe  
i=premnmx([temp,humid,wind]);  
qp=sim(net,i');
```

```
function [qp]=cgf(temp,humid,wind)
```

```
load flepo  
i=premnmx([temp,humid,wind]);  
qp=sim(net,i');
```

```
function [qp]=lm(temp,humid,wind)
```

```
load levmar  
i=premnmx([temp,humid,wind]);  
qp=sim(net,i');
```

```
function [qp]=oss(temp,humid,wind)
```

```
load astepsec  
i=premnmx([temp,humid,wind]);  
qp=sim(net,i');
```

```
function [qp]=rp(temp,humid,wind)
```

```
load resilient  
i=premnmx([temp,humid,wind]);  
qp=sim(net,i');
```

```
function [qp]=scg(temp,humid,wind)
```

```
load scalconj  
i=premnmx([temp,humid,wind]);  
qp=sim(net,i');
```

## APPENDIX 7

## Weather Data

**JABATAN METEOROLOGI MALAYSIA**  
**Records of Meteorological Data**

**Station : Kuantan**

**Lat. : 3° 47'N**

**Long. : 103°**

**13'E**

**Ht. Above M.S.L. : 15.3 m.**

Stnno	Year	Month	Day	Rainfall ( 08-08 MST ) ( mm )	24 Hour	Relative	24 Hour
					Mean Temp. ( ° C )	Humidity ( % )	Mean Wind ( m/s )
48657	2008	1	1	6.6	23.9	92.0	2.0
48657	2008	1	2	28.6	24.0	91.7	2.0
48657	2008	1	3	9.2	24.1	94.6	1.6
48657	2008	1	4	-33.3	25.1	89.5	1.7
48657	2008	1	5	1.0	25.2	90.3	1.7
48657	2008	1	6	3.6	25.0	91.7	1.6
48657	2008	1	7	6.8	25.4	90.0	1.8
48657	2008	1	8	3.2	25.5	92.4	1.8
48657	2008	1	9	1.4	25.9	90.6	1.6
48657	2008	1	10	0.0	26.4	87.6	1.6
48657	2008	1	11	0.0	27.0	85.3	1.5
48657	2008	1	12	0.0	26.9	85.5	1.5
48657	2008	1	13	0.0	26.6	86.3	1.6
48657	2008	1	14	0.0	26.3	86.8	1.5
48657	2008	1	15	0.0	26.8	83.7	1.5
48657	2008	1	16	0.0	27.1	78.4	1.4
48657	2008	1	17	153.6	26.6	82.8	2.2
48657	2008	1	18	11.8	23.7	95.3	2.7
48657	2008	1	19	6.0	25.4	91.1	1.8
48657	2008	1	20	0.4	26.1	86.8	2.0
48657	2008	1	21	3.6	26.0	89.2	2.0
48657	2008	1	22	-33.3	26.2	86.8	2.1
48657	2008	1	23	9.2	26.4	87.1	2.3
48657	2008	1	24	0.4	26.3	89.6	1.9
48657	2008	1	25	0.0	26.5	87.9	1.7
48657	2008	1	26	0.2	26.3	87.6	2.3
48657	2008	1	27	8.8	25.8	90.5	2.0
48657	2008	1	28	17.6	25.8	91.6	1.3
48657	2008	1	29	-33.3	26.6	87.8	1.7
48657	2008	1	30	0.0	26.9	85.4	1.8
8657	2008	1	31	0.0	26.8	86.6	1.9
48657	2008	2	1	45.6	26.1	89.2	1.9
48657	2008	2	2	64.4	24.9	95.6	1.5
48657	2008	2	3	16.8	26.1	90.8	1.9
48657	2008	2	4	0.4	25.7	92.1	1.8
48657	2008	2	5	0.4	26.0	89.5	1.9
48657	2008	2	6	0.0	26.1	85.0	1.8
48657	2008	2	7	0.0	25.7	84.8	1.9
48657	2008	2	8	0.0	26.1	85.5	2.0

48657	2008	2	9	-33.3	25.9	82.7	2.0
48657	2008	2	10	0.0	25.6	77.3	2.1
48657	2008	2	11	0.0	25.3	83.6	1.8
48657	2008	2	12	0.0	25.2	82.7	1.8
48657	2008	2	13	0.0	25.6	83.5	2.2
48657	2008	2	14	0.0	25.9	83.7	2.1
48657	2008	2	15	0.0	25.9	82.8	2.3
48657	2008	2	16	0.0	25.1	84.3	2.2
48657	2008	2	17	0.0	25.6	81.0	2.3
48657	2008	2	18	0.0	25.5	81.4	2.4
48657	2008	2	19	0.0	25.5	79.2	2.4
48657	2008	2	20	0.0	26.0	79.7	2.7
48657	2008	2	21	0.0	25.8	83.2	1.7
48657	2008	2	22	0.0	26.2	80.9	1.3
48657	2008	2	23	0.0	26.4	82.7	1.3
48657	2008	2	24	0.0	26.9	85.5	1.4
48657	2008	2	25	1.2	26.2	87.4	1.6
48657	2008	2	26	0.0	26.1	87.5	1.8
48657	2008	2	27	47.0	26.7	84.4	2.4
48657	2008	2	28	218.6	24.2	96.5	1.9
48657	2008	2	29	7.6	23.6	97.5	1.5
48657	2008	3	1	45.6	24.2	96.4	1.0
48657	2008	3	2	0.0	26.1	87.0	2.2
48657	2008	3	3	0.0	25.5	86.2	1.9
48657	2008	3	4	29.2	25.5	85.3	1.9
48657	2008	3	5	77.6	23.9	95.3	1.7
48657	2008	3	6	18.8	25.6	90.7	1.7
48657	2008	3	7	-33.3	26.3	88.2	2.1
48657	2008	3	8	5.4	26.4	87.2	2.3
48657	2008	3	9	3.6	26.1	88.5	2.6
48657	2008	3	10	3.0	26.0	89.6	2.4
48657	2008	3	11	87.0	25.6	91.3	1.9
48657	2008	3	12	8.6	26.2	91.2	2.2
48657	2008	3	13	0.2	26.7	88.2	2.0
48657	2008	3	14	22.6	26.4	89.7	1.7
48657	2008	3	15	2.2	26.7	88.3	2.1
48657	2008	3	16	-33.3	26.4	87.0	1.7
48657	2008	3	17	0.0	26.8	85.9	1.8
48657	2008	3	18	0.0	26.9	86.1	1.9
48657	2008	3	19	0.0	26.9	85.8	1.6
48657	2008	3	20	0.0	26.9	84.3	1.3
48657	2008	3	21	0.0	27.2	85.0	1.5
48657	2008	3	22	8.4	25.8	90.2	1.3
48657	2008	3	23	0.0	27.1	84.5	1.2
48657	2008	3	24	5.6	27.6	83.7	1.8
48657	2008	3	25	19.8	27.0	87.8	2.1
48657	2008	3	26	7.8	26.4	89.6	1.2
48657	2008	3	27	0.0	27.5	84.9	1.6
48657	2008	3	28	0.0	28.0	87.0	1.6
48657	2008	3	29	0.0	27.9	87.8	1.7
48657	2008	3	30	0.0	27.7	86.6	1.4
48657	2008	3	31	-33.3	27.4	85.7	1.4
48657	2008	4	1	0.0	26.7	89.3	1.2
48657	2008	4	2	0.0	26.5	88.5	1.3
48657	2008	4	3	4.2	26.8	88.2	1.3
48657	2008	4	4	0.8	26.6	90.3	1.2
48657	2008	4	5	0.0	27.6	86.0	1.7
48657	2008	4	6	5.0	26.7	91.0	0.8
48657	2008	4	7	0.0	27.9	86.2	1.7
48657	2008	4	8	3.0	27.7	87.8	1.5

48657	2008	4	9	0.0	28.0	84.8	1.1
48657	2008	4	10	0.0	28.1	84.9	1.5
48657	2008	4	11	0.0	27.7	84.3	1.1
48657	2008	4	12	0.0	27.5	84.3	1.3
48657	2008	4	13	0.0	27.8	83.5	1.0
48657	2008	4	14	0.0	28.2	81.9	1.3
48657	2008	4	15	-33.3	26.4	88.0	0.7
48657	2008	4	16	0.0	28.2	84.1	1.1
48657	2008	4	17	-33.3	28.6	84.5	1.1
48657	2008	4	18	39.2	28.0	83.6	1.4
48657	2008	4	19	-33.3	28.4	81.6	1.6
48657	2008	4	20	31.2	28.2	82.6	1.8
48657	2008	4	21	0.0	28.3	83.7	1.1
48657	2008	4	22	0.4	27.5	85.6	1.1
48657	2008	4	23	6.2	27.4	86.1	1.9
48657	2008	4	24	1.8	25.3	93.1	1.0
48657	2008	4	25	0.0	27.0	87.5	0.7
48657	2008	4	26	21.0	27.7	87.1	1.5
48657	2008	4	27	27.6	27.9	86.8	1.4
48657	2008	4	28	0.6	27.9	84.8	1.5
48657	2008	4	29	3.0	26.9	90.0	0.8
48657	2008	4	30	44.0	26.2	90.0	1.1
48657	2008	5	1	5.0	27.8	85.2	1.7
48657	2008	5	2	1.4	27.6	85.5	1.7
48657	2008	5	3	23.8	26.0	92.5	1.3
48657	2008	5	4	2.4	27.2	86.9	1.3
48657	2008	5	5	22.6	27.4	86.5	1.5
48657	2008	5	6	6.6	27.1	87.0	1.5
48657	2008	5	7	0.0	27.2	86.2	1.3
48657	2008	5	8	9.0	27.3	86.4	1.5
48657	2008	5	9	0.0	27.6	82.3	1.8
48657	2008	5	10	0.0	27.4	83.5	1.5
48657	2008	5	11	-33.3	27.0	87.0	1.7
48657	2008	5	12	0.0	27.8	81.9	1.7
48657	2008	5	13	5.4	28.4	82.3	1.6
48657	2008	5	14	1.8	27.8	87.0	1.3
48657	2008	5	15	0.0	28.6	80.6	1.7
48657	2008	5	16	0.0	28.9	83.8	1.3
48657	2008	5	17	0.6	28.3	85.0	1.6
48657	2008	5	18	7.6	27.9	86.0	1.6
48657	2008	5	19	-33.3	27.9	84.6	1.4
48657	2008	5	20	0.2	27.1	88.2	1.3
48657	2008	5	21	-33.3	27.3	86.5	1.3
48657	2008	5	22	-33.3	27.8	85.5	1.4
48657	2008	5	23	-33.3	28.5	83.5	1.2
48657	2008	5	24	0.8	27.4	86.5	0.9
48657	2008	5	25	0.0	27.6	84.7	1.2
48657	2008	5	26	0.0	28.2	81.8	1.4
48657	2008	5	27	0.0	27.5	82.8	1.6
48657	2008	5	28	0.0	28.3	79.2	2.1
48657	2008	5	29	7.6	28.5	80.5	1.4
48657	2008	5	30	5.6	26.9	86.5	1.6
48657	2008	5	31	0.0	27.2	84.8	1.5
48657	2008	6	1	-33.3	27.6	85.1	1.1
48657	2008	6	2	1.2	28.0	84.0	1.1
48657	2008	6	3	-33.3	27.3	85.4	1.5
48657	2008	6	4	0.0	28.3	79.2	1.2
48657	2008	6	5	0.0	28.0	82.9	1.0
48657	2008	6	6	42.2	27.9	84.2	1.8
48657	2008	6	7	-33.3	26.3	89.9	1.3



48657	2008	6	8	-33.3	26.2	87.4	2.2
48657	2008	6	9	-33.3	26.9	85.5	1.4
48657	2008	6	10	-33.3	27.9	79.6	2.2
48657	2008	6	11	0.2	27.2	83.8	1.8
48657	2008	6	12	0.0	28.1	77.7	2.1
48657	2008	6	13	0.0	28.4	80.0	1.6
48657	2008	6	14	-33.3	28.6	80.3	1.4
48657	2008	6	15	0.0	28.5	76.5	2.0
48657	2008	6	16	17.4	26.9	79.7	2.2
48657	2008	6	17	0.0	27.7	79.3	1.0
48657	2008	6	18	-33.3	27.6	83.1	1.4
48657	2008	6	19	4.0	26.8	82.9	1.5
48657	2008	6	20	0.0	27.4	81.1	1.3
48657	2008	6	21	0.0	28.5	77.6	2.0
48657	2008	6	22	0.0	27.5	82.4	1.8
48657	2008	6	23	-33.3	27.6	79.2	2.1
48657	2008	6	24	-33.3	27.2	79.6	1.6
48657	2008	6	25	-33.3	28.1	79.5	1.6
48657	2008	6	26	2.6	27.4	85.2	1.1
48657	2008	6	27	1.2	27.3	83.6	1.5
48657	2008	6	28	-33.3	28.5	80.0	1.8
48657	2008	6	29	14.6	27.2	86.0	1.5
48657	2008	6	30	0.0	27.5	80.6	1.6
48657	2008	7	1	0.6	27.2	80.8	1.9
48657	2008	7	2	0.0	27.9	81.9	1.7
48657	2008	7	3	18.2	26.4	86.9	1.3
48657	2008	7	4	0.2	25.8	90.0	1.7
48657	2008	7	5	0.0	27.6	81.3	1.6
48657	2008	7	6	0.0	28.0	82.5	1.5
48657	2008	7	7	0.0	27.5	85.4	1.5
48657	2008	7	8	9.6	26.9	85.5	1.4
48657	2008	7	9	-33.3	26.3	84.7	1.4
48657	2008	7	10	6.8	26.5	84.5	1.9
48657	2008	7	11	0.0	26.9	82.5	1.5
48657	2008	7	12	5.0	27.6	81.4	1.7
48657	2008	7	13	-33.3	27.8	81.4	1.6
48657	2008	7	14	0.0	27.7	80.3	1.5
48657	2008	7	15	0.0	27.6	81.2	2.3
48657	2008	7	16	0.0	27.5	80.5	2.2
48657	2008	7	17	0.0	27.7	78.2	1.8
48657	2008	7	18	-33.3	25.8	87.6	1.7
48657	2008	7	19	20.0	25.5	88.7	1.2
48657	2008	7	20	1.6	26.7	85.5	0.9
48657	2008	7	21	-33.3	26.6	86.5	1.3
48657	2008	7	22	27.6	26.6	84.0	2.2
48657	2008	7	23	20.2	26.3	83.2	1.6
48657	2008	7	24	17.6	26.0	85.8	2.0
48657	2008	7	25	21.6	25.6	89.8	1.2
48657	2008	7	26	93.8	26.0	86.7	1.9
48657	2008	7	27	-33.3	25.3	86.8	1.2
48657	2008	7	28	0.0	27.2	83.2	1.3
48657	2008	7	29	0.0	28.0	81.7	1.3
48657	2008	7	30	0.0	28.8	80.7	1.9
48657	2008	7	31	0.0	28.8	80.0	1.9
48657	2008	8	1	0.0	28.9	80.7	2.0
48657	2008	8	2	0.0	28.9	80.6	1.8
48657	2008	8	3	-33.3	28.1	81.4	1.7
48657	2008	8	4	0.0	27.9	82.3	1.7
48657	2008	8	5	25.4	28.1	78.9	2.3
48657	2008	8	6	0.0	27.3	80.0	1.9

48657	2008	8	7	8.6	26.2	85.8	1.9
48657	2008	8	8	0.0	27.2	82.1	2.1
48657	2008	8	9	13.4	27.4	87.3	0.8
48657	2008	8	10	9.0	26.7	82.1	1.9
48657	2008	8	11	88.0	27.3	81.7	1.4
48657	2008	8	12	0.8	26.3	85.4	2.1
48657	2008	8	13	0.0	27.2	80.9	1.4
48657	2008	8	14	27.4	26.0	89.0	1.2
48657	2008	8	15	45.6	26.7	85.8	1.9
48657	2008	8	16	4.4	26.9	84.8	1.8
48657	2008	8	17	0.0	27.5	81.7	1.7
48657	2008	8	18	1.4	27.3	82.9	1.7
48657	2008	8	19	3.2	27.1	83.9	1.6
48657	2008	8	20	4.2	27.0	86.2	1.4
48657	2008	8	21	4.6	26.6	87.1	1.5
48657	2008	8	22	0.6	27.0	82.9	2.3
48657	2008	8	23	0.0	26.4	84.4	1.9
48657	2008	8	24	3.4	26.8	83.9	2.2
48657	2008	8	25	37.6	26.4	86.5	1.5
48657	2008	8	26	16.4	26.5	87.6	1.2
48657	2008	8	27	-33.3	26.7	84.3	1.3
48657	2008	8	28	0.0	27.3	83.0	1.7
48657	2008	8	29	0.0	27.4	83.0	1.9
48657	2008	8	30	-33.3	27.4	84.8	1.1
48657	2008	8	31	14.8	27.7	83.7	1.2
48657	2008	9	1	1.2	25.9	92.4	0.6
48657	2008	9	2	0.0	26.8	83.6	1.4
48657	2008	9	3	0.0	27.1	82.0	1.9
48657	2008	9	4	-33.3	26.6	85.4	1.1
48657	2008	9	5	76.6	25.9	87.9	1.5
48657	2008	9	6	3.0	25.6	89.3	1.8
48657	2008	9	7	0.2	26.1	85.9	0.9
48657	2008	9	8	0.4	26.5	86.9	1.0
48657	2008	9	9	0.0	27.4	85.1	0.9
48657	2008	9	10	-33.3	27.8	80.6	1.6
48657	2008	9	11	0.0	28.1	80.2	3.0
48657	2008	9	12	0.0	28.2	78.7	2.0
48657	2008	9	13	60.8	28.0	80.4	1.9
48657	2008	9	14	-33.3	26.5	86.7	1.3
48657	2008	9	15	3.2	27.4	83.6	1.6
48657	2008	9	16	12.6	27.8	80.4	1.8
48657	2008	9	17	0.0	28.0	82.8	1.7
48657	2008	9	18	-33.3	27.7	81.6	1.7
48657	2008	9	19	0.0	28.5	80.9	2.0
48657	2008	9	20	0.0	28.6	77.3	1.5
48657	2008	9	21	9.0	26.9	84.1	1.4
48657	2008	9	22	0.0	27.0	81.3	1.4
48657	2008	9	23	2.2	28.3	79.8	1.5
48657	2008	9	24	0.0	29.0	76.0	2.3
48657	2008	9	25	0.0	28.5	76.4	1.3
48657	2008	9	26	54.8	27.1	85.0	1.5
48657	2008	9	27	0.0	27.6	81.8	1.6
48657	2008	9	28	-33.3	27.9	81.1	2.1
48657	2008	9	29	2.6	27.2	84.8	1.7
48657	2008	9	30	70.8	27.3	83.5	1.7
48657	2008	10	1	0.0	27.3	85.7	1.2
48657	2008	10	2	0.0	28.3	81.1	1.2
48657	2008	10	3	-33.3	27.6	83.7	1.3
48657	2008	10	4	8.8	26.9	84.3	1.8
48657	2008	10	5	0.0	27.5	81.9	1.4

48657	2008	10	6	15.8	27.0	84.3	1.3
48657	2008	10	7	0.0	26.4	86.1	1.1
48657	2008	10	8	0.0	27.3	81.7	1.3
48657	2008	10	9	-33.3	26.9	85.6	1.0
48657	2008	10	10	0.0	27.9	80.2	1.1
48657	2008	10	11	0.0	28.0	82.3	1.3
48657	2008	10	12	0.6	27.1	86.0	1.2
48657	2008	10	13	0.4	27.7	83.6	1.2
48657	2008	10	14	0.0	28.8	81.2	1.3
48657	2008	10	15	-33.3	28.2	83.9	1.1
48657	2008	10	16	0.0	27.9	82.7	1.1
48657	2008	10	17	1.0	27.8	83.4	1.1
48657	2008	10	18	29.6	27.0	88.5	1.1
48657	2008	10	19	0.4	27.6	84.8	1.1
48657	2008	10	20	7.0	28.2	82.4	1.1
48657	2008	10	21	45.2	26.5	90.9	1.1
48657	2008	10	22	6.8	26.5	88.3	1.5
48657	2008	10	23	32.2	25.9	90.8	1.5
48657	2008	10	24	-33.3	25.8	88.1	2.3
48657	2008	10	25	0.4	28.2	79.2	1.4
48657	2008	10	26	2.4	26.3	89.6	1.1
48657	2008	10	27	0.6	26.4	88.3	1.0
48657	2008	10	28	0.0	27.5	82.0	1.0
48657	2008	10	29	39.1	26.6	86.3	1.4
48657	2008	10	30	-33.3	27.1	89.0	0.6
48657	2008	10	31	0.6	27.0	90.0	1.0
48657	2008	11	1	36.2	26.7	87.2	1.3
48657	2008	11	2	2.2	27.0	83.6	1.4
48657	2008	11	3	15.2	26.5	87.8	0.7
48657	2008	11	4	0.0	27.3	85.0	1.5
48657	2008	11	5	1.4	27.1	84.7	0.9
48657	2008	11	6	9.8	27.3	85.3	1.1
48657	2008	11	7	57.6	26.4	89.6	1.0
48657	2008	11	8	0.0	27.2	85.2	1.2
48657	2008	11	9	0.0	27.6	84.1	0.9
48657	2008	11	10	52.0	26.2	87.2	1.5
48657	2008	11	11	-33.3	26.2	87.0	1.2
48657	2008	11	12	2.6	26.7	86.0	1.5
48657	2008	11	13	0.4	26.0	89.2	1.0
48657	2008	11	14	2.0	27.1	84.4	1.3
48657	2008	11	15	5.8	26.1	90.0	0.9
48657	2008	11	16	0.0	27.2	85.7	0.9
48657	2008	11	17	14.2	27.1	86.5	1.4
48657	2008	11	18	5.2	26.7	88.7	0.9
48657	2008	11	19	100.6	26.0	92.0	1.9
48657	2008	11	20	14.2	25.5	91.6	1.1
48657	2008	11	21	0.0	26.7	86.1	1.8
48657	2008	11	22	0.0	26.8	85.8	1.6
48657	2008	11	23	28.4	26.7	88.0	1.4
48657	2008	11	24	4.0	26.4	91.0	0.9
48657	2008	11	25	13.8	26.9	89.2	1.0
48657	2008	11	26	49.0	25.4	95.0	1.1
48657	2008	11	27	-33.3	25.0	94.7	0.6
48657	2008	11	28	0.0	25.7	89.9	0.9
48657	2008	11	29	-33.3	26.9	87.0	1.0
48657	2008	11	30	0.0	26.6	88.0	1.3
48657	2008	12	1	2.2	26.1	89.8	1.7
48657	2008	12	2	12.8	24.9	93.7	1.7
48657	2008	12	3	6.0	26.1	89.7	1.6
48657	2008	12	4	3.4	26.4	86.7	1.2

48657	2008	12	5	22.6	26.8	83.5	1.4
48657	2008	12	6	27.8	24.4	95.8	1.1
48657	2008	12	7	129.4	24.1	97.0	2.0
48657	2008	12	8	52.0	23.3	97.4	1.3
48657	2008	12	9	0.2	25.6	86.6	1.2
48657	2008	12	10	-33.3	26.1	84.0	1.4
48657	2008	12	11	-33.3	25.9	86.8	1.6
48657	2008	12	12	0.0	26.4	82.6	1.4
48657	2008	12	13	18.0	26.4	85.8	1.7
48657	2008	12	14	13.1	25.2	93.7	1.0
48657	2008	12	15	0.6	26.0	85.6	1.6
48657	2008	12	16	0.8	26.0	84.3	1.6
48657	2008	12	17	-33.3	25.6	85.6	1.5
48657	2008	12	18	3.0	24.8	88.0	1.5
48657	2008	12	19	0.0	25.8	81.2	2.0
48657	2008	12	20	0.0	25.8	82.3	1.9
48657	2008	12	21	-33.3	26.1	82.3	2.0
48657	2008	12	22	5.4	25.7	85.5	1.6
48657	2008	12	23	0.2	25.6	78.9	2.2
48657	2008	12	24	2.2	25.4	80.0	1.9
48657	2008	12	25	9.2	24.3	90.9	0.9
48657	2008	12	26	6.0	25.3	88.5	1.2
48657	2008	12	27	37.8	25.4	89.2	1.3
48657	2008	12	28	78.4	25.9	90.3	1.1
48657	2008	12	29	0.5	25.3	92.3	1.1
48657	2008	12	30	55.4	25.6	90.9	0.9
48657	2008	12	31	4.8	25.6	90.3	1.1
48657	2009	1	1	117.4	25.1	94.2	1.6
48657	2009	1	2	136.6	24.0	96.7	1.5
48657	2009	1	3	202.0	23.3	97.2	2.7
48657	2009	1	4	1.6	24.3	92.6	1.8
48657	2009	1	5	-33.3	25.1	89.1	1.6
48657	2009	1	6	0.0	26.0	81.5	1.8
48657	2009	1	7	0.0	26.0	80.2	2.0
48657	2009	1	8	0.0	25.4	83.4	2.0
48657	2009	1	9	1.4	24.7	87.3	1.8
48657	2009	1	10	0.6	24.2	84.3	1.8
48657	2009	1	11	0.0	24.6	82.2	1.8
48657	2009	1	12	1.4	24.7	86.3	1.6
48657	2009	1	13	5.6	23.5	91.5	1.9
48657	2009	1	14	0.6	23.4	91.3	1.6
48657	2009	1	15	0.8	23.9	85.8	1.9
48657	2009	1	16	-33.3	24.4	82.6	2.1
48657	2009	1	17	0.0	25.0	80.7	2.2
48657	2009	1	18	0.0	25.0	80.1	1.9
48657	2009	1	19	0.0	24.7	80.0	1.9
48657	2009	1	20	0.0	25.1	79.0	1.9
48657	2009	1	21	0.0	25.2	78.7	2.1
48657	2009	1	22	0.0	25.4	79.8	1.8
48657	2009	1	23	-33.3	26.1	80.6	2.2
48657	2009	1	24	1.0	26.2	84.6	1.9
48657	2009	1	25	8.4	25.5	89.9	1.9
48657	2009	1	26	0.0	25.8	80.6	2.3
48657	2009	1	27	0.0	25.6	79.3	1.9
48657	2009	1	28	0.0	25.4	78.0	1.9
48657	2009	1	29	0.0	25.3	79.6	1.7
48657	2009	1	30	0.0	25.8	79.5	1.7
48657	2009	1	31	-33.3	25.4	82.0	1.8
48657	2009	2	1	-33.3	25.2	82.3	2.1
48657	2009	2	2	0.0	25.4	78.7	1.9

48657	2009	2	3	0.0	25.5	78.9	1.5
48657	2009	2	4	0.0	25.8	80.0	1.7
48657	2009	2	5	0.0	25.9	81.0	1.6
48657	2009	2	6	0.0	25.8	81.1	1.7
48657	2009	2	7	0.0	25.5	78.5	1.9
48657	2009	2	8	0.0	25.8	76.7	2.1
48657	2009	2	9	0.0	26.0	75.9	2.2
48657	2009	2	10	0.0	26.2	76.4	1.7
48657	2009	2	11	16.0	26.1	82.6	1.5
48657	2009	2	12	0.0	26.4	83.6	1.3
48657	2009	2	13	2.0	27.3	86.7	1.0
48657	2009	2	14	23.2	26.1	89.6	1.0
48657	2009	2	15	0.0	27.1	86.3	1.0
48657	2009	2	16	0.0	26.7	81.6	1.5
48657	2009	2	17	0.0	26.6	80.5	1.9
48657	2009	2	18	0.0	26.8	76.2	2.0
48657	2009	2	19	0.0	26.4	78.8	1.9
48657	2009	2	20	0.0	26.7	79.4	1.7
48657	2009	2	21	0.0	27.2	80.0	2.0
48657	2009	2	22	9.0	25.6	90.3	1.2
48657	2009	2	23	6.2	25.7	89.5	1.1
48657	2009	2	24	0.2	26.4	84.8	1.1
48657	2009	2	25	0.6	26.8	85.4	1.5
48657	2009	2	26	-33.3	26.7	84.4	1.6
48657	2009	2	27	0.0	27.5	78.3	1.2
48657	2009	2	28	2.2	27.3	83.5	1.5
48657	2009	3	1	1.6	26.7	86.4	1.6
48657	2009	3	2	52.4	26.7	86.9	1.8
48657	2009	3	3	20.6	24.6	94.9	1.3
48657	2009	3	4	11.8	26.7	85.6	1.6
48657	2009	3	5	-33.3	26.9	88.2	1.5
48657	2009	3	6	-33.3	27.1	86.2	1.6
48657	2009	3	7	0.0	27.2	84.5	1.3
48657	2009	3	8	5.8	26.3	87.3	1.5
48657	2009	3	9	0.8	25.1	90.6	1.8
48657	2009	3	10	-33.3	27.0	82.6	1.8
48657	2009	3	11	-33.3	27.4	79.9	1.8
48657	2009	3	12	-33.3	27.6	81.8	1.3
48657	2009	3	13	11.0	26.4	87.6	1.3
48657	2009	3	14	5.8	26.2	83.5	1.5
48657	2009	3	15	45.6	24.3	93.6	1.8
48657	2009	3	16	1.4	25.2	88.5	1.4
48657	2009	3	17	1.0	26.9	83.3	1.4
48657	2009	3	18	0.6	27.5	85.9	0.9
48657	2009	3	19	-33.3	26.3	88.7	1.2
48657	2009	3	20	0.0	26.8	86.1	1.2
48657	2009	3	21	7.6	27.2	85.9	1.4
48657	2009	3	22	0.0	26.9	83.7	1.4
48657	2009	3	23	2.6	28.2	82.4	1.3
48657	2009	3	24	32.6	25.5	89.2	1.4
48657	2009	3	25	0.0	26.2	83.3	1.3
48657	2009	3	26	36.2	27.0	82.5	1.4
48657	2009	3	27	-33.3	27.0	84.6	1.4
48657	2009	3	28	23.0	26.9	87.3	1.3
48657	2009	3	29	20.0	26.4	88.8	1.3
48657	2009	3	30	1.4	26.6	88.5	1.2
48657	2009	3	31	0.0	27.4	80.3	1.4
48657	2009	4	1	3.8	27.4	82.1	1.7
48657	2009	4	2	0.0	27.9	82.9	1.5
48657	2009	4	3	7.2	27.9	82.4	1.8

48657	2009	4	4	-33.3	27.8	83.3	1.8
48657	2009	4	5	9.4	26.3	90.7	1.1
48657	2009	4	6	-33.3	27.6	83.7	1.2
48657	2009	4	7	17.8	26.3	90.5	1.1
48657	2009	4	8	5.2	27.2	85.3	1.6
48657	2009	4	9	18.0	26.3	93.3	1.1
48657	2009	4	10	-33.3	27.9	83.2	1.3
48657	2009	4	11	2.0	27.8	84.1	1.4
48657	2009	4	12	12.4	26.3	92.5	1.0
48657	2009	4	13	1.0	27.5	86.9	1.3
48657	2009	4	14	0.0	28.1	85.1	1.2
48657	2009	4	15	51.0	26.0	92.1	1.3
48657	2009	4	16	8.0	26.5	85.9	1.6
48657	2009	4	17	0.0	28.3	80.7	1.5
48657	2009	4	18	0.0	28.7	82.2	1.2
48657	2009	4	19	5.6	27.7	83.1	1.9
48657	2009	4	20	8.0	28.1	80.7	1.7
48657	2009	4	21	-33.3	27.3	82.7	1.5
48657	2009	4	22	0.0	28.7	74.6	1.4
48657	2009	4	23	2.4	28.4	79.8	1.5
48657	2009	4	24	14.2	27.4	84.7	1.3
48657	2009	4	25	0.0	28.4	76.8	1.9
48657	2009	4	26	0.0	29.1	78.9	1.5
48657	2009	4	27	0.0	28.7	79.3	1.4
48657	2009	4	28	0.0	27.5	85.3	1.0
48657	2009	4	29	5.2	28.4	79.8	1.7
48657	2009	4	30	2.2	27.9	81.6	1.6
48657	2009	5	1	8.8	28.4	79.4	0.9
48657	2009	5	2	2.2	27.7	82.3	1.7
48657	2009	5	3	0.0	27.9	79.6	2.0
48657	2009	5	4	0.0	28.2	80.8	1.0
48657	2009	5	5	2.8	27.4	86.9	1.1
48657	2009	5	6	0.0	27.4	86.1	1.3
48657	2009	5	7	1.2	28.1	83.2	1.1
48657	2009	5	8	7.2	27.6	83.7	1.5
48657	2009	5	9	-33.3	27.2	83.5	1.4
48657	2009	5	10	19.6	26.5	87.1	1.2
48657	2009	5	11	22.0	26.2	89.1	1.2
48657	2009	5	12	0.0	28.0	83.1	1.4
48657	2009	5	13	36.4	28.0	84.9	1.4
48657	2009	5	14	-33.3	27.6	82.7	1.5
48657	2009	5	15	0.8	27.8	84.5	1.2
48657	2009	5	16	0.0	27.8	83.4	1.2
48657	2009	5	17	85.6	26.8	85.2	1.8
48657	2009	5	18	-33.3	25.6	89.5	1.1
48657	2009	5	19	0.0	28.0	81.9	1.2
48657	2009	5	20	0.0	28.8	80.7	1.1
48657	2009	5	21	0.6	27.8	86.2	1.3
48657	2009	5	22	29.0	28.5	81.2	1.7
48657	2009	5	23	0.0	26.9	85.5	1.4
48657	2009	5	24	0.0	28.2	84.2	1.3
48657	2009	5	25	1.4	28.7	84.3	1.5
48657	2009	5	26	29.2	26.5	87.5	1.5
48657	2009	5	27	9.4	27.7	83.3	1.5
48657	2009	5	28	18.6	27.4	84.3	1.8
48657	2009	5	29	45.8	26.8	89.1	1.3
48657	2009	5	30	0.4	26.7	88.3	0.8
48657	2009	5	31	0.0	28.5	84.2	1.2

Note : MST - Malaysian Standard Time

Definition : -33.3 - Trace ( Rainfall amount less than 0.1 millimetre )

The daily rainfall amount ( 0800 - 0800 MST ) for a particular day is the amount collected over the 24 - hour period beginning from 0800 a.m. on that day. For example,

the daily rainfall amount for 25th. December, 2004 is the amount collected over the

## APPENDIX 8

### Data Distribution Conditions of Malaysian Meteorology Department

#### Condition of Supply of Climatological Data

These climatological data are supplied by the Malaysian Meteorological Department on condition :-

- a) that the data shall not be utilised for other project or study unless fresh application has been made with the M.M.D.
- b) that acknowledgement for the use of the data obtained from the M.M.D. will be suitable made in any report, paper or publication in which such data have been quoted or utilised and a copy of such report, paper or publication be extended to M.M.D. free of charge.

#### DATA SUPPLIED

( Data copied in Microsoft Excel )

**Station :**

**Kuantan**

**Period**

2008 - May  
2009

1. Daily Rainfall Amount
2. Daily 24 hour Mean Temperature
3. Daily 24 Hour Mean Relative Humidity
4. Daily Mean Wind Speed