Development of Neural Network Model for Predicting Crucial Product Properties or Yield for Optimisation of Refinery Operation

by

Sharliza binti Mohamad

Dissertation submitted in partial fulfillment of
the requirement for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

January 2005

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Development of Neural Network Model for Predicting Crucial Product Properties or Yield for Optimisation of Refinery Operation

by

Sharliza binti Mohamad

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

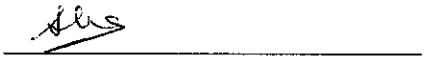
Approved by,

ASSOCIATE PROFESSOR DR MOHAMED IBRAHIM B. ABDUL MUTALIB

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
July 2004

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



SHARLIZA BINTI MOHAMAD

ABSTRACT

Refinery optimisation requires accurate prediction of crucial product properties and yield of desired products. Neural network modeling is an alternative approach to prediction using mathematical correlations. The project is an extension of a previous research conducted by the university on product yield and properties prediction using non-linear regression method. The objectives of this project are to develop a framework for the application of neural network modeling in predicting refinery product yield and properties, to develop neural network model for three case studies (predicting crude distillation yield, diesel pour point and hydrocracker total gasoline yield) and to evaluate the suitability of using neural network modeling for predicting refinery product yield and properties.

The project methodologies used are literature research and computer modeling using MATLAB neural network toolbox. The framework development for neural network modeling include aspects such as process understanding, data collection and division, input elements selection, data preprocessing, network type selection, design of network architecture, learning algorithm selection, network training, and network simulation using new data set. Various configurations of neural network model were tested to choose the best model to represent each case study. The model selected has the smallest mean squared error when simulated using test data.

The results are presented in the form of the network configuration that gives the smallest MSE, plots comparing the actual output with the output predicted by the neural network, as well as residual analysis results to determine the range of deviation between the actual and predicted output. Although the accuracy of the output predicted by the neural network model requires further improvement, in general, the study has shown the tremendous potential for the use of neural network for predicting refinery product yield and properties. Suggestions for future study in the area include improvement of the model accuracy using advanced methods such as cross-training and stacked network, integration of neural network with plant's Advanced Process Control as inferential property predictor, and study on inverted network for use in a neural network-based controller.

ACKNOWLEDGEMENTS

First and foremost, I would like to convey my utmost gratitude to my supervisor, Associate Professor Dr Mohamed Ibrahim Abdul Mutalib for his guidance and supervision throughout the duration of this project. His patience and support are deeply appreciated and is instrumental in the completion of the project. My special gratitude also to Mr Nooryusmiza Yusoff for his guidance and help in matters relating to neural network, and for introducing me to neural network modeling in the first place.

I would also like to thank the Final Year Research Coordinator, Ms Noryuliana Yuhana and Assistant Coordinator, Mr Bawadi Abdullah for the excellent coordination of the course on the whole, which allows us to focus on completing our projects within the specified time frame. My gratitude also to the Department of Chemical Engineering of Universiti Teknologi Petronas for providing the resources required for completing the project, especially the use of its computer lab for doing computer modeling work, and the literature references available in the library.

My appreciation also to my colleagues who were also doing computer simulation and modeling for their Final Year Research Project for their motivation and idea-sharing, but most of all for commiserating during my moments of need. If the battle cry of those conducting experimental research is "*when will my equipment/material arrive*?", our battle cry is most certainly *"why won't it converge?"*

Last but not least, my deepest gratitude to my family and friends for their support, and others directly or indirectly involved in allowing me to complete the project in time.

TABLE OF CONTENTS

LIST OF APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND OF STUDY

The ability to predict crucial product properties and yield of desired product is vital to a refinery in the effort to reduce cost and improve refining margin. A typical refinery has the goal of converting as much barrel of crude oil into transportation fuels (gasoline, kerosene and diesel) as is economically possible. Although a refinery can produce many profitable products, transportation fuels have the highest demand in terms of volume. The quality of refinery feedstocks (crude oil) affects the yield and properties of various products that can be obtained from the refining process. Crude assay data published by crude oil producers provide information on the crude properties, as well as the expected yield and properties of various fractions that can be obtained from crude distillation. The fractions correspond to boiling point ranges of products that refineries wish to produce. For processing units further downstream, the yield and properties of the products can also be predicted based on the unit feed properties.

In crude assays, a True Boiling Point (TBP) distillation provides the yield and properties of discrete crude fractions obtained by batch distillation of specific crudes. However, in actual operation, refineries usually run a mixture of different kinds of crudes as feedstock to the process units. The mixture changes frequently in terms of crude types and proportions depending on factors such as crude price and availability, as well as product requirements. Performing the TBP distillation is an expensive and time-consuming process, so a refinery cannot afford to run the test every time the proportions of the crude mixture are changed. Therefore, there is a need for a method to accurately estimate the yield and quality of products distilled from crude oils based on the properties of the crude. Previously, models to predict product yield and properties from crude distillation are formulated using mathematical correlations.

Development of mathematical correlations for predicting refinery product properties and yield was researched by a group of Universiti Teknologi Petronas lecturers in a study

entitled "Development of a Systematic Methodology for Predicting Crucial Properties or Yield for the Optimisation of Refinery Operation". The study introduced a framework for developing mathematical correlations using non-linear regression method, which could be used for a wide range of applications in refinery operation. As a mean to illustrate the application of the framework, two case studies were researched; prediction of diesel pour point, and prediction of distillate yield for atmospheric crude distillation (Abdul Mutalib, 2004).

This study aims to develop a systematic methodology for predicting refinery yield and properties using neural network. To illustrate the application of the methodology in a refinery-wide application, three case studies were conducted; prediction of crude distillation product yield based on crude properties, prediction of diesel pour point based on crude properties, and prediction of hydrocracker total gasoline yield based on gas oil feed properties. The complexity and non-linearity of the systems makes them good candidates for neural network modeling. Neural network is one of a group of intelligence technology for data analysis that differs from classical analysis methods by learning about the system from the data provided, rather than being programmed by user. A neural network models the system by detecting pattern and relationship in data, learning from the relationship, and adapting to change (Manning, 1998). Neural network has been applied successfully in process engineering for various applications such as predicting inferential properties from easily measurable parameters, monitoring and interpretation of process trends, and generating non-linear models for the design of model-predictive control system.

1.2 PROBLEM STATEMENT

1.2.1 Problem Identification

The ability to predict crucial product properties and yield for optimisation of refinery operation is vital to reduce cost and improve refinery profit margin. To tackle issues such as meeting customer's product specification and reducing quality giveaway, a refinery requires a reliable and accurate method for predicting product yield and properties based on the feedstock quality. Neural network modeling provides an alternative method to

generation of mathematical correlations. A refinery produces a wide range of products with many crucial properties; hence a systematic methodology is required to provide a framework for development of the neural network model for different types of products and properties.

1.2.2 Significance of Project

The project will study the application of neural network modeling for predicting crucial product properties and yield for a refinery-wide application. The results could be applied for in-house generation of neural network model by refinery personnel for optimisation of refinery operation. This would require less cost compared to use of commercially developed software, as well as being able to take into consideration any special characteristics of the refinery's process and unit operation.

1.3 OBJECTIVES AND SCOPE OF STUDY

The objectives of the study are:

i. To develop a systematic methodology for the application of neural network modeling for prediction of product yield and properties in a refinery
ii. To develop neural network model for three case studies:
    Case Study 1: Prediction of Product Yield for Crude Distillation based on Properties of Crude Oil
    Case Study 2: Prediction of Product Property (Diesel Pour Point) based on Properties of Crude Oil
    Case Study 3: Prediction of Total Gasoline Yield for Hydrocracker Unit based on Properties of Gas Oil Feed and Hydrogen Consumption
iii. To evaluate the feasibility of using neural network model for predicting refinery product yield and properties

The scope of study will focus on developing a systematic framework for the application of neural network modeling in refinery optimisation, illustrating the application of the framework by creating neural network model for the three case studies, and analysing the suitability of neural network modeling for predicting refinery product yield and properties.

1.3.1 The Relevancy of the Project

The project is an opportunity for the author to utilise the knowledge and insights obtained during industrial internship regarding refinery operation. From the university's perspective, the project will be an extension to the previous study using non-linear regression method. It will provide an alternative framework for development of prediction model for refinery product yield and properties.

1.3.2 Feasibility of the Project Within the Scope and Time Frame

The scope of the project is viable for completion in a one-semester research project. Approximately one-third of the duration was spent on studying the fundamentals, principles, applications and method of implementations of neural network modeling, another one-third on understanding the processes to be modeled, and the final one-third for actual computer modeling work.

CHAPTER 2

LITERATURE REVIEW AND THEORY

2.1 ARTIFICIAL NEURAL NETWORK

2.1.1 Definition

Artificial neural networks are inspired by the architecture of biological nervous systems, which consists of a large number of relatively simple nerve cells or neurons. The neurons function in parallel to facilitate rapid decisions. Similarly, artificial neural networks consist of a large number of computational elements, arranged in a massively parallel structure (Aldrich, 2001). Schalkoff (1997) defined artificial neural network as a structure composed of a number of interconnected units, also known as artificial neurons. Each unit has an input/output characteristics and implements a local computation or function. The output of any unit is determined by its input/output characteristics, its interconnection to other units and external inputs. The network develops an overall functionality through one or more form of training.

Neural networks follow some sort of training rule whereby the weights of connections between the units are adjusted based on the patterns existing in the data set presented to the network. In other words, neural networks learn from example, just like a child would learn to recognize a cat from examples of cats or a bird from examples of birds. Commonly, neural network is trained so that a particular input leads to a specific target outputs. The network is then adjusted based on comparison of the network output and the target until the network output matches the target (Demuth, 1998).

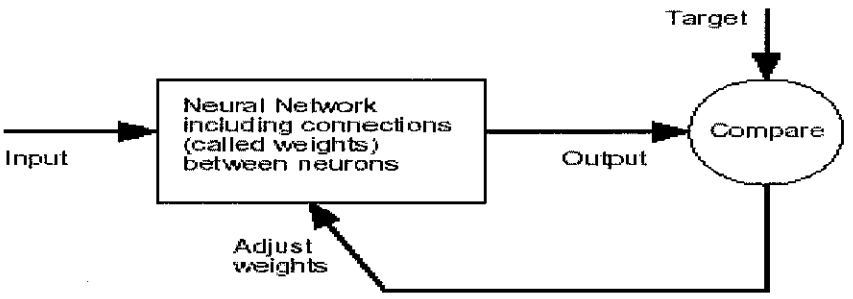


Figure 1: Training a Neural Network

In addition to taking its cue from biological neural network, neural network also has its basis in statistical problem solving. According to Bishop (1995), statistical framework is the most general and natural framework to formulate and understand solutions in neural network modeling. This is because statistical framework recognises the probabilistic nature of the information to be processed, and of the form in which the result is expressed. This is apparent in the set of data used for training the network. In most cases, it is impossible to collect all the data set representing the problem (in statistical terms, what is called the population). Consequently, the data set used for training the network must be a fair representation of the system to be modeled as a whole (a good sample of the population).

2.1.2 Neuron Model

To summarise Aldrich (2001) and Demuth (1998), fundamental understanding of the neuron model is required to understand how a neural network processes data. Neuron is the basic processing unit of a neural network. The neuron receives inputs from other neurons in the network, or from the outside, which are subsequently weighed and summed. A scalar bias, b is added to the weighted sum to become the argument n to the transfer function f. The transfer function f is typically a step function or a sigmoid function which takes the argument n and produces the output a. The weights, w and the bias, b are adjustable scalar parameters. The central idea of neural networks is that the weights and bias can be adjusted so that the network exhibits some desired behaviour. Hence, the network can be trained to perform a particular task by adjusting the weight and bias parameters to achieve a desired end.

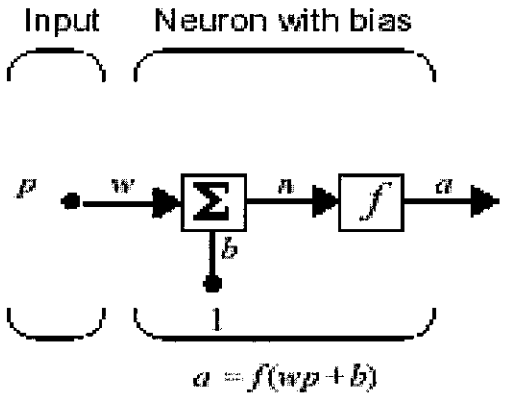


Figure 2: Simple Neuron Model

2.1.3 Network Architecture

The architecture of a neural network consists of a description of how many layers a network has, the number of neurons in each layer, the transfer function in each layer, and how the layers are connected to each other. The best architecture to be used depends on the type of problem, or input/output mapping represented by the network.

In a single layer network with multiple neurons, each element of the input vector **p** is connected to each neuron through the weight matrix **W.** Each neuron performs its own summing function to sum the weighted inputs and bias and form its own scalar input, n(i). The various n(i) from all the neurons taken together form the net input vector **n**. The net input vector becomes the argument to the transfer function *f* and the neuron layer outputs form a column vector **a** (Demuth, 1998). The network configuration is shown below:

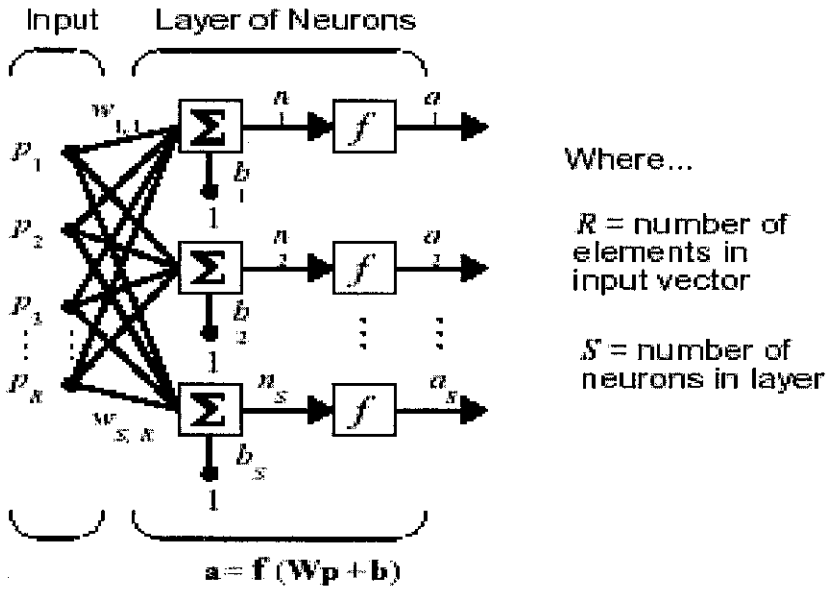


Figure 3: Single Layer Network

A network can also have several layers, in which case the outputs of each intermediate layer are the inputs to the next layer. To differentiate between weight matrices connected to inputs and weight matrices connected between the layers, the terms *input weights* and *layer weights* are used. The layers of a multilayer network play different roles. The layer

that produces the network output is called the *output layer,* while all other layers are called *hidden layer.*

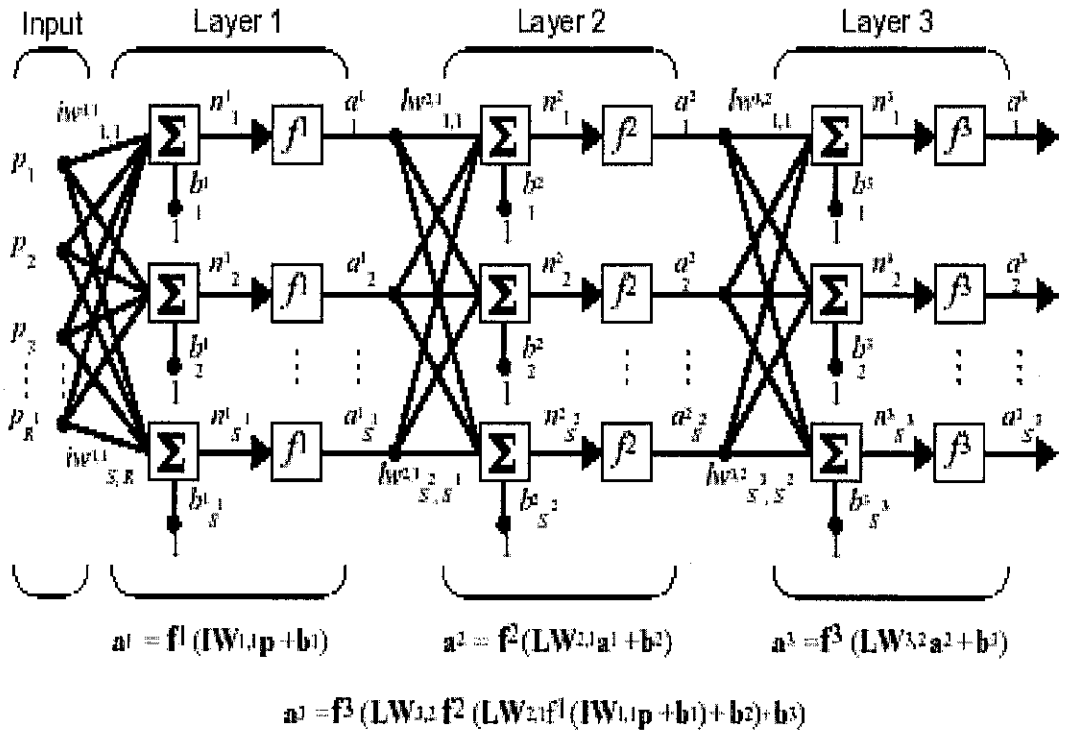


Figure 4: Multiple Layers Network

2.1.4 General Application of Neural Network

Neural network can be used to solve problems that cannot be tackled effectively using conventional computing approaches, which depends heavily on programming and specific instructions from user. Specifically, neural network is most useful to solve these types of problems:

Function Approximation

In certain cases, it is known that a relationship exists between some influencing factors or inputs and the observed behaviours or the outputs. However, due to the complexity or high degree of non-linearity of the problem, the relationship cannot be modeled mathematically. Neural network is useful to solve this type of problem, especially if a large number or examples of historical data which describes the relationship can be collected.

Pattern Recognition/Classification

Neural network is useful to perform tasks such as handwriting recognition and signal and image identification by classifying and associating the inputs with a specific class of target.

Prediction of Trends and Future Events

Time-series prediction of future trends can be done using neural network, provided a lot of historical data is available. This is useful in fields such as the stock market for predicting share prices.

Clustering or Grouping of Data

A special type of neural network, the self-organising map, can be used to determine whether there is any structure or class/cluster/similarity within the values in a data set.

2.1.5 Application in Process Engineering

In process engineering, neural network has been applied in various problems such as process identification, inferential property prediction and model-based control strategy development. Various papers and studies have been published regarding the use of neural network modeling in refinery optimisation. Barsamian and Macias (1998) in their work on inferential property predictors studied the use of neural network to produce non-linear property correlation equations for boiling point, flash point, freeze point, Reid Vapour Pressure, asphalt penetration, yield and octane number prediction. They also designed an Inferential Property Estimation Software tool using neural network embedded in Excel spreadsheet. The results from the study indicate that their correlations for flash point and freeze point are accurate up to 3°C and the prediction for octane number is accurate up to 0.3 octane number. The findings from the study emphasised the importance of data validity and filtering to obtain high quality results and avoid the phenomenon of "garbage in-garbage out".

Ramos and Cunha (1998), two researchers from a Brazilian refinery studied two methods for predicting LPG 95% evaporation point, the inference curve method and neural network. For the neural network, they used a three-layer perceptron neural network with back

propagation. The neural network outperformed the inference curve method, efficiently predicting the evaporation point with little error margins.

Barbosa et.al. (2002) applied Bayesian neural networks on the inference of diesel 85% ASTM distillation from process operating conditions, and compared the results with traditional multi-layered perceptrons. Bayesian neural network uses probability densities instead of frequencies, assumes a particular model for the probability densities of data and network weights, and uses Bayes' rule to infer the optimum weights, given the available data. From the results, the best inference was obtained using a one-neuron Bayesian neural network with six input variables.

2.2 PETROLEUM REFINING

2.2.1 Refinery Feedstock

The basic raw material for refineries is petroleum, also known as crude oil. Crude oil is a naturally occurring mixture of hydrocarbon compounds that may include compounds of sulfur, nitrogen, oxygen, metals and other elements. The compounds boil at different temperatures, thus can be separated into fractions by distillation (Speight, 2001). The elementary composition of crude oils is quite uniform, falling within the following ranges:

Table 1: Elementary Composition of Crude Oils

Element	Percent by Weight
Carbon	84 – 87
Hydrogen	11 – 14
Sulfur	0 – 3
Nitrogen	0 – 0.6

Source: *Petroleum Refining Technology and Economics*

In terms of chemical composition, crude oil is a mixture of various hydrocarbon compounds. The hydrocarbons present in crude oil are classified into three classes:

i) Paraffins: saturated hydrocarbons with straight or branched chain, but without any ring structure, such as methane, ethane and propane

ii) Naphthenes: saturated hydrocarbons containing one or more rings, such as cyclopentane, cyclohexane and methylcyclopentane

iii) Aromatics: unsaturated hydrocarbons containing a benzene ring such as benzene, toluene and ethylbenze

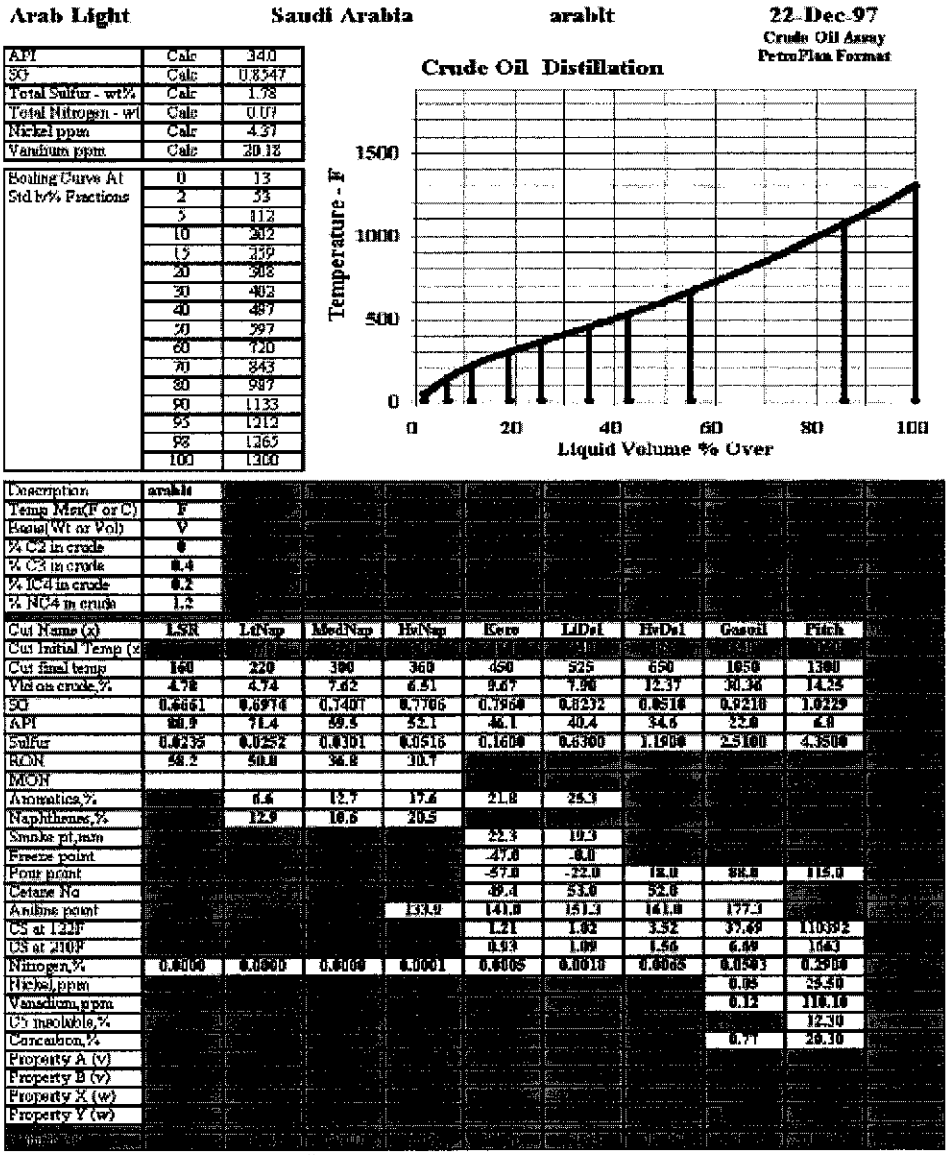
Wide variations in composition and properties are exhibited by crude oils, not only among those from different oilfields, but also in crude oils taken from different production depth in the same field. The differences in composition will affect the physical properties exhibited by different type of crude oils, as well as the expected yield and properties of various product fractions obtained after refining. Therefore, knowledge on the properties of the crude will allow a refinery to optimise its conversion to valuable products.

Evaluation of a crude for use as a refinery feedstock involves an examination of a few important properties. The value and suitability of a particular type of crude to a refinery depends on its quality and whether the refinery can obtain a satisfactory product yield and properties that meet market demand. Usually, the refinery is not concerned with the actual chemical nature of the crude, but is more interested in methods of analysis that would provide sufficient information to assess the potential quality of the crude in terms of product slate and suitable method for processing. This information can be obtained from one of two ways:

1. Preliminary assay inspection data
2. Full assay including preparation of a true boiling point curve and analysis of product fractions throughout the full range of the crude oil

The *preliminary assay* provides general data on the crude based on tests that are routine, simple and can be completed in a short time, such as specific gravity, pour point, sulfur content, viscosity and water content (Speight, 2001). The *preliminary assay* provides a useful general picture of the crude quality, but does not provide adequate data for the refinery to predict the yield and properties of the products obtained from the crude after processing.

A *full assay* of a crude is based on a true boiling point distillation of the crude, from which sufficient data is obtained to assess the yield and properties of the straight-run products obtained from distilling the crude. The distillation is carried out using the ASTM D-2892 method, also known as the *true boiling point distillation* (TBP) method. The method uses a column with 15 theoretical plates and a 5:1 reflux ratio; hence it is also known as the "15/5" method (Maples, 1997). Preparing a full assay is an expensive and time-consuming process, and a large quantity of sample is required to perform the TBP distillation.



API	Calc	34.0
SG	Calc	0.8547
Total Sulfur - wt%	Calc	1.78
Total Nitrogen - wt	Calc	0.07
Nickel ppm	Calc	4.37
Vanadium ppm	Calc	30.18

Boiling Curve At Std b/% Fractions	0	13
	2	53
	5	112
	10	202
	15	259
	20	308
	30	402
	40	487
	50	597
	60	720
	70	843
	80	987
	90	1133
	95	1212
	98	1265
	100	1300

Description	arablt								
Temp Met(F or C)	F								
Basis(Wt or Vol)	V								
% C2 in crude	0								
% C3 in crude	0.4								
% IC4 in crude	0.2								
% NC4 in crude	1.2								
Cut Name (x)	LSR	LtNap	MedNap	HvNap	Kero	LtDsl	HvDsl	Gasoil	Pitch
Cut Initial Temp (x									
Cut final temp	160	220	300	360	450	525	650	1050	1300
Vol% on crude,%	4.78	4.74	7.62	6.51	9.67	7.96	12.37	30.36	14.25
SG	0.6661	0.6974	0.7407	0.7706	0.7960	0.8232	0.8510	0.9210	1.0229
API	80.9	71.4	59.5	52.1	46.1	40.4	34.6	22.0	8.0
Sulfur	0.0236	0.0252	0.0301	0.0516	0.1600	0.6300	1.1900	2.5100	4.3800
RON	58.2	50.0	36.9	30.7					
MON									
Aromatics,%		0.6	12.7	17.6	21.0	25.3			
Naphthenes,%		12.9	10.6	20.5					
Smoke pt,mm					22.3	19.3			
Freeze point					-47.0	-8.0			
Pour point					-57.0	-22.0	18.0	88.0	115.0
Cetane No					49.4	53.0	52.0		
Aniline point				133.9	141.0	151.3	161.0	177.3	
CS at 122F					1.21	1.02	3.52	37.69	110392
CS at 210F					0.93	1.09	1.56	6.89	1643
Nitrogen,%	0.0000	0.0000	0.0000	0.0001	0.0005	0.0010	0.0035	0.0503	0.2900
Nickel,ppm								0.05	25.50
Vanadium,ppm								0.12	110.10
CS insoluble,%									12.30
Conradson,%								0.77	26.30
Property A (v)									
Property B (v)									
Property X (w)									
Property Y (w)									

Figure 5: Full assay and TBP distillation curve

Source: http://homeflash.net/~celjure/enginering/petroplan/assay

The important physical properties of crude oils include density (or API gravity), pour point and sulfur content. The properties are discussed below:

API Gravity

In the petroleum industry, the density of crude oil is usually expressed as API gravity rather than specific gravity. API gravity is related to specific gravity in such a way that an increase in API gravity corresponds to a decrease in specific gravity:

$$^{o}\text{API} = \frac{141.5}{specific\ gravity} - 131.5$$

The API gravity of a crude oil refers to a liquid sample of the crude at 60°F or 15.6°C. Crude oil gravity usually ranges between 10° API to 50° API. API gravity is used as an indicative measure of the proportions of various products that can be obtained from distilling the crude. A lighter crude with a higher API gravity, hence lower density, is expected to produce a higher yield in the low-boiling range upon distilling. Similarly, a heavier crude oil with lower API gravity, hence higher density, is expected to produce a higher yield in the high-boiling range.

Pour Point

Pour point is defined as the lowest temperature at which petroleum oil will flow or pour when it is chilled without disturbance at a controlled rate. The pour point of a crude oil is a rough indicator of the relative paraffinicity and aromaticity of the crude. The lower the pour point, the lower the paraffin content and the greater the content of aromatics.

Sulfur Content

Sulfur content and API gravity are the two properties which have the greatest influence on the value of crude oil. Sulfur content is expressed as percent sulfur by weight and usually varies from less than 0.1% to greater than 5%. Crudes with sulfur content more than 0.5% requires more extensive processing than those with lower sulfur content, and are known as "sour" crude.

2.2.2 Overall Refinery Process Flow

Each refinery has a unique processing scheme, which is determined by the process equipment available, crude oil characteristics, operating costs and product demands. The optimum flow pattern for any refinery is dictated by economic considerations. However, in general, the processing sequencing in a typical refinery follows the general pattern as described by Gary (2001).

Crude oil is heated in a furnace and charged to an atmospheric crude distillation tower, where it is separated into light gas, light naphtha, heavy naphtha, kerosene, diesel or atmospheric gas oil and reduced crude. The reduced crude is sent to a vacuum distillation tower and separated into vacuum gas oil streams and vacuum residue. The vacuum residue from the vacuum tower bottom is then thermally cracked in a delayed coker to produce wet gas, coker gasoline, coker gas oil and coke.

The atmospheric and vacuum crude distillation unit gas oils, along with the coker gas oil are used as feedstocks for the catalytic cracking or hydrocracking units. These units crack heavier molecules into lower molecular weight compounds boiling in the middle distillate fuel ranges. The light naphtha streams from atmospheric crude distillation, coker and cracking units are sent to an isomerisation unit to convert straight-chain paraffins into isomers that have higher octane number. The heavy naphtha streams are fed to a catalytic reformer to improve their octane number. The streams from the isomerisation unit and the catalytic reformer unit are blended into gasoline.

The gas streams are separated in the vapor recovery section into refinery fuel gas, liquefied petroleum gas (LPG), normal butane, isobutane and unsaturated hydrocarbons. The fuel gas is burned as fuel in refinery furnaces and the normal butane is blended into gasoline or LPG. The isobutane and unsaturated hydrocarbons are sent to an alkyllation unit, which uses sulfuric or hydroflouric acid as catalyst to react olefins with isobutanes to form isoparaffins boiling in the gasoline range. The middle distillates from the crude distillation unit, coker, and cracking units are blended into diesel and jet fuel (kerosene).

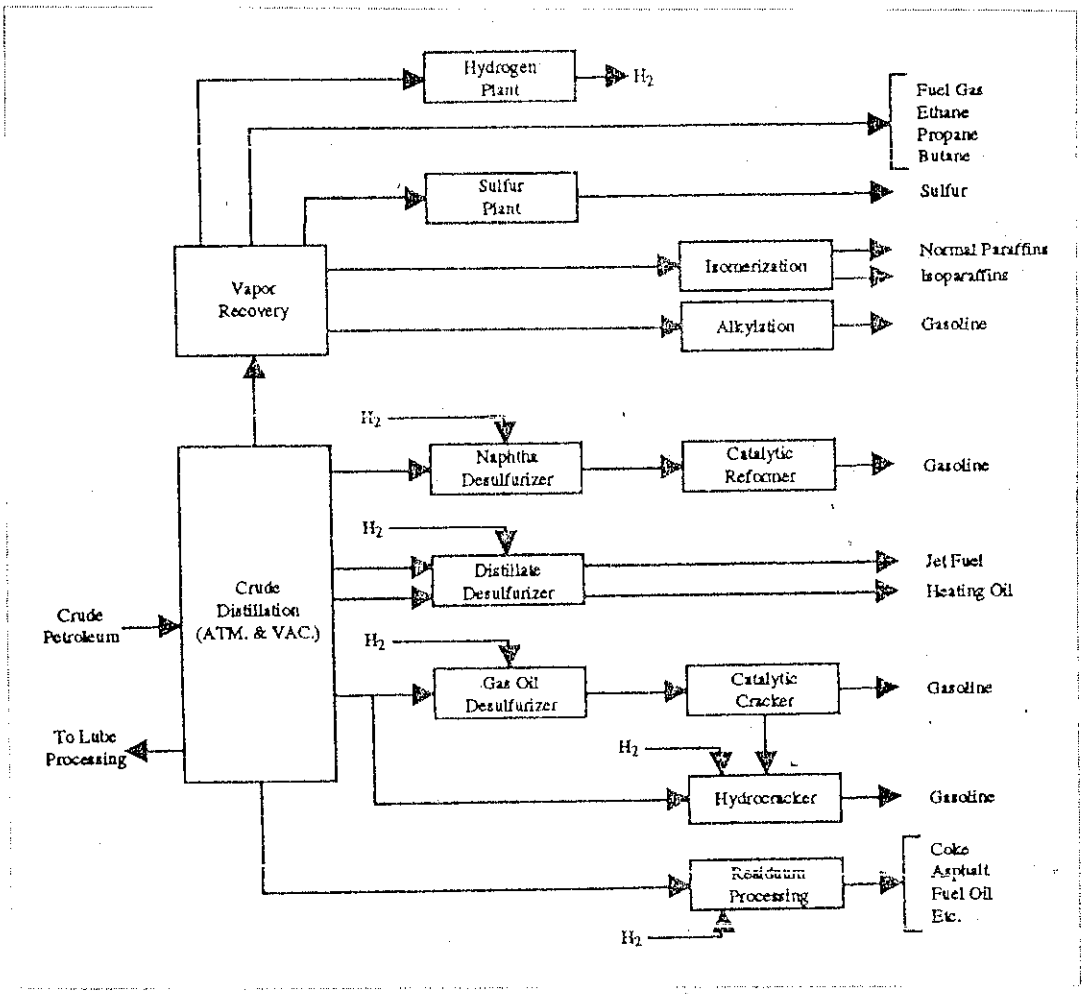


Figure 6: Overall Refinery Process Flow

Source: Petroleum Refining Technology and Economics

2.2.3 Crude Distillation Unit

The atmospheric crude distillation unit, the first major processing unit in a refinery, is used to separate crude oils into fractions by distillation according to boiling points. The TBP cut points for the various fractions corresponding to specific products will vary slightly among different refineries; however, the typical values are given in Table 2:

Table 2: TBP Cut Points for Various Crude oil Fractions

Cut	Initial Boiling Point (°C)	Final Boiling Point (°C)
Fuel Gas	C1	C2
LPG	C3	C4
Light Naphtha	36	180
Heavy Naphtha	180	320
Kerosene	320	450
Diesel	450	690
Reduced Crude	690	690+

Source: Philippine National Oil Company Reference Manual

The yields and properties of the desired fractions that can be obtained from a specific crude can be predicted from the TBP distillation curve and full assay. However, it is rare for a refinery to process only a single crude at a given time. Instead, a blend of a few different crudes is usually charged as the feedstock to the crude distillation unit. Performing a full assay of a crude is an expensive and time-consuming procedure. Furthermore, the crude blend being charged into crude distillation unit could change significantly in terms of crude types and proportions before an assay can be completed. Therefore, it would be impossible for a refinery to perform a full assay and TBP distillation each time there is a change in the crude blend. This creates a need for other means to estimate the yields and properties of the desired products that can be obtained from processing a specific blend of crude.

2.2.4 Hydrocracker

Hydrocracking is a catalytic cracking process conducted with a high hydrogen partial pressure. The objective of the process is to produce higher-value, lower molecular weight products such as gasoline, kerosene and diesel from low quality gas oils, which would otherwise have to be sold as low-priced distillate fuels. Hydrocracking increases the yield

of gasoline components, and produces gasoline blending components with a quality not obtainable from other process by recycling the gas oils through the cracking process that generated them (Leffler, 1979). In contrast to coking and deasphalting processes, hydrocracking decreases the carbon-to-hydrogen ratio by the addition of hydrogen, rather than the removal of carbon. An important feature of the hydrocracker is the gain in product volume compared to the feed volume, usually up to 25%. The cracking/hydrogenation process results in products whose average density is a lot lower than the feed. The primary product is a gasoline blending components called hydrocrackate.

2.2.5 Estimating Product Yield and Properties

In contrast to the *full assay,* a *preliminary assay* of a crude or a blend of crude can be completed in a short time since the tests required are relatively simple and routine. For this reason, various studies were conducted to predict the yield and properties of products distilled from a crude based on the properties of the crude itself. The approach used by most researchers is to correlate the yield and properties of selected products from the crude distillation unit to the properties of the crude obtained from the *preliminary assay.* For example, Al Soufi et. al. (1986) generated a linear correlation model for predicting distillate yield based on crude API gravity. In another study, non-linear regression method is used to correlate more crude properties (API gravity, sulfur content, pour point, viscosity, total acid number and Reid vapour pressure) to the yield of selected distillate products (Abdul Mutalib, 2004).

In both studies, the process operating conditions of the crude distillation unit is ignored in the development of the correlation. This is because the idea here is to come up with a generic correlation that does not depend on the specific conditions of the crude distillation unit, but is only dependent on the crude properties. Consequently, the model can be applied at any refinery for any crude distillation unit, instead of being valid only for the specific unit for which the model was developed. A similar approach is taken in this study, such that the neural network model will be developed to estimate the yields and

CHAPTER 3
METHODOLOGY AND PROJECT WORK

There are two major stages involved in carrying out the project. The first stage is to develop a systematic framework or methodology for designing a neural network model for the system. The second stage is the computer simulation of the neural network model using a commercial neural network software available in the market.

To accomplish the first stage, research and literature review was done to learn and understand the fundamental, principles and method of application of neural network modeling. The objective of the first stage is to come up with possible network configurations suited to model the problem, which would then be tested in the computer simulation stage to determine the best network configuration for the problem. The project methodology is summarised in the diagram included in Appendix II.

3.1 SYSTEMATIC METHODOLOGY FOR NEURAL NETWORK MODELING

3.1.1 Process Understanding

As is the case in any modeling approach, the first step in designing a neural network is to study and understand the process to be modeled. This starts by classifying the input/output problem to be solved as function approximation, pattern recognition, time-series prediction, or data clustering. The type of input/output mapping will have an impact on the type of network as well as network architecture that is suitable for modeling the process. For refinery optimisation problems, the input/output mapping generally falls under the function approximation classification, where the objective is to predict the value of certain output parameters, given the values of other parameters that are known to have an impact on the output. Sufficient understanding on the nature of the process, as well as the characteristics of the inputs and outputs are necessary prerequisites before proceeding to the next steps.

Process understanding will also helps in determining the characteristics of the model to be created. For example, for problems having multiple outputs, a decision must be made

whether to model each output separately using individual network, or to use a single network with multiple outputs. There are some limitations in neural network for modeling multiple output process. Bishop (1995) stated that defining a single network with multiple output variables causes the network to suffer from cross-talk, where the hidden neurons experience difficulty in learning as they are attempting to model at least two functions at once. However, for some process, the nature of the outputs is such that modeling each output individually is not suitable. Ultimately, the final decision can only be made after both approaches have been tested to determine which one gives a better output prediction.

3.1.2 Collection And Division Of Data

The data set available is divided into two sets; training data and testing data. The training data is used during the training phase to adjust the weights and biases of the network in order to minimise the mean squared error (MSE) between the network output and the target. After the training phase is completed, the network is run using testing data to see how well it performs. For network utilising the early stopping feature to improve generalisation, the data is divided into three different sets; training, testing and validation. In this case, training is interrupted periodically to run the network on the validation data.

The data for training must be representative of the process to avoid extrapolation when the model is used to predict output for data it has not seen before. Heuristics suggests an 80% - 20% division between the training data and the testing data (Schalkoff, 1997). However, in the case where limited data set is available, the proportion of the training data may be increased to ensure that the trained network is able to find a generalised pattern in the data set.

3.1.3 Selection Of Input Elements

In cases where the dimension of the input vector is large, some of the elements are probably highly correlated and redundant. Having redundant inputs to a network will reduce the performance of the network, as well as making the network unnecessarily large and complex. This effect is termed "*the curse of dimensionality*" by Bishop (1995), and the

end effect is that the number of training data required to specify the mapping increases exponentially with additional input elements.

The selection of which input elements to be included is done using the following approaches:

1. Based on multivariate analysis to eliminate elements that are highly correlated
2. Based on prior knowledge on the process to be modeled

Multivariate analysis is conducted by performing a linear regression for each input element with all the other elements, one after the other. When two input elements are found to be highly correlated with one another (having a high value of r^2), one of them is omitted from the model. The choice of which of the two elements is eliminated will depend on other factors such as the relative importance of the input element to the output, which in turn is to be determined from understanding of the process to be modeled.

Aldrich (2001) suggested that fundamental knowledge of the process to be modeled should always be included in the network. This can be done either by making use of hybrid neural network systems, in which neural networks are explicitly combined with first-principle process models, or by structuring the inputs to the network in such a way that previous knowledge on the process is incorporated in the network through the training process.

3.1.4 Data Preprocessing

According to Bishop (1995), data preprocessing is one of the most important stages in the development of a neural network model, and the choice of preprocessing steps has significant effects on generalisation performance. One form of preprocessing involves a reduction in the dimensionality of the input data by a process called *feature extraction*, in which modified inputs are formed from collections of the original inputs which might be combined in linear or non-linear ways. Another approach is to scale the inputs and targets so that they always fall within a specified range. This is relevant in cases where different input elements have values that differ by several orders of magnitude. If the data is not

scaled, the difference in magnitude may not reflect the relative importance of the input elements in determining the required outputs.

Data preprocessing also addresses the issue of data set with incomplete input values. Bishop (1995) suggested that if the quantity of data is sufficiently large, then the simplest solution is to omit the data sets with missing values from the network. However, when limited data set is available for training the network, it is important to make full use of the information that is potentially available from the incomplete sets. Various heuristics have been proposed for dealing with missing input data. The most basic is to replace the missing value with the mean of the variable over the available data sets. More complicated approaches include integration over the corresponding variables, weighted by the appropriate distribution, and filling in the missing data points with values drawn at random from available data sets.

3.1.5 Selection Of Network Type

Neural network can solve various problems classified as either function approximation, pattern recognition or classification problems. The problem for the case studies, which is to estimate the values of various outputs of a process, given certain inputs, can be classified as a function approximation problem. Based on literature review from previous research on neural network application in industry (Barbosa, 2002, Ramos, 1998), as well as heuristics given in the neural network toolbox manual (Demuth, 1998), two types of network have been identified as being suitable for modeling the yield prediction from crude properties; multilayer feedforward network with backpropagation algorithm, and Bayesian regularisation network.

Feedforward Network

Feedforward networks are layered, acyclic networks in which there is no path from a neuron back to itself. The neurons are partitioned into subsets called layer, where a connection is allowed from layer j to k only if layer j precedes layer k (Mahrotra, 1997). Some literature also refers to the feedforward network as multilayered perceptron (MLP). The network consists of one or more hidden layers with tangent-sigmoid or log-sigmoid

transfer function, and an output layer with a linear transfer function. The multiple layers of neurons with nonlinear differentiable transfer functions allow the network to learn nonlinear and linear relationships between the input and output vectors (Demuth, 1998).
Standard backpropagation utilises gradient descent algorithm to update the weights. Each neuron calculates its error derivatives which consist of partial derivatives of the neuron's error, E with respect to one of its weights, *wi*. The error derivatives computed propagate backwards such that each hidden neuron uses the error information from the neuron ahead of it to calculate its own error derivative. The error derivatives are used to calculate the amount to adjust the weights of the links (Winston, 1993).

Bayesian Regularisation

Bayesian regularisation is a special type of network in which the weights and biases of the network are assumed to be random variables with specified distribution. The parameters are estimated using statistical techniques. Bayesian network is created as a way to improve network generalisation by using a network that is just large enough to provide an adequate fit. One of the problems that occur during network training is called overfitting. A network that is overfitted has a very small error on the training set, but the error becomes very large when new data is presented to the network. In short, the network has memorised the training examples, but it has not learn to generalise to new situations. Bayesian networks ensure that the network size is just enough to learn the training data, and will not have enough power to overfit the data.

3.1.6 Formulation Of Network Architecture

The architecture of a neural network consists of a description of how many layers a network has, the number of neurons in each layer and the transfer function in each layer.

Number of Layers

For this project, neural networks with 1 and 2 hidden layers were tested. The basis for this choice is that the neural network toolbox manual claimed that a feedforward network with a single hidden layer could approximate any function, given sufficient number of neurons (Demuth, 1998). However, several researchers have indicated that two hidden layers

would work better, especially for problems with multiple outputs, which is the case here (Smith, 1993).

Number of Neurons in Each Layer

The number of neurons in the output layer must be equal to the number of elements in the output vector. The number of neurons in the hidden layers is determined from trial-and-error. The objective is to find the smallest number of neurons that will allow a specific network architecture to converge. Currently, no guideline or heuristics can be found with regards to determining the number of neurons in the hidden layers, and trial-and error is the prevalent method used. The approach used here is to determine a minimum and sufficient number of neurons for the task at hand. If too few neurons are used, the network will not converge during the training phase. However, if too many neurons are used, overfitting will occur, i.e. the network will model the training data well, but it will perform poorly when presented with data it has never seen before. To avoid overfitting, the number of weights must be kept smaller than the number of training data.

Transfer Function in Each Layer

The transfer function in the output layer is set as the linear transfer function to ensure that the outputs can take on any values, not just restricted to 0 and 1. The transfer function in the hidden layers must be a sigmoid function due to its differentiability. The differentiability property is an important requirement for the backpropagation algorithm to work, since the weights and biases are adjusted in the direction of the negative of the gradient of the function. If the function is not differentiable, the algorithm could not compute its gradient; hence the weights and biases cannot be adjusted. There are two options for the sigmoid functions, **tangent-sigmoid** and **log-sigmoid**, and both were tested to see which one gives a better model. For the two hidden layers group, 4 different combinations of the transfer functions are used:

i) **log-sigmoid** transfer function in both hidden layers

ii) **tangent-sigmoid** transfer function in both hidden layers

iii) **log-sigmoid** in the first hidden layer and **tangent-sigmoid** in the second hidden layer

iv) **tangent-sigmoid** in the first hidden layer and **log-sigmoid** in the second hidden layer

3.1.7 Training Of Network

12 different learning algorithms which seem to be suitable to the problem at hand are tested. The learning algorithms differ in terms of how the weights and biases of the network are calculated. The algorithms can be classified into 4 main categories:

i) Basic gradient descent algorithm (batch-gradient descent, **'traingd'** and batch gradient descent with momentum, **'traingdm'**)

ii) Variable learning rate algorithm (variable learning rate method, **'traingda'**, variable learning rate method with momentum, **'traingdx'** and resilient backpropagation, **'trainrp'**)

iii) Conjugate gradient algorithm (Fletcher-Reeves Update, **'traincgf'**, Polak-Ribiere Update, **'traincgp'**, Powell-Beale Restarts, **'traincgb'** and scaled conjugate gradient, **'trainscg'**)

iv) Quasi-Newton algorithm (BFGS algorithm, **'trainbfg'**, one step secant method, **'trainoss'**, and Levenberg-Marquardt algorithm, **'trainlm'**)

Each network configuration is also trained under two conditions; with early stopping, and without early stopping. Early stopping is another method used to improve generalisation. In this method, the data is divided into training, validation and testing sets. The training data is used for computing the gradient and updating the weights and biases. The error of the validation data is monitored during the training process. When the network starts to overfit the data, the error of the validation data set will increase. Training is stopped when the validation data error increases for a specific number of iterations, and the weights and biases at the minimum of the validation error are returned.

From the framework development of neural network model, a set of possible network configurations to model the case studies is obtained, summarised in Figure 8.

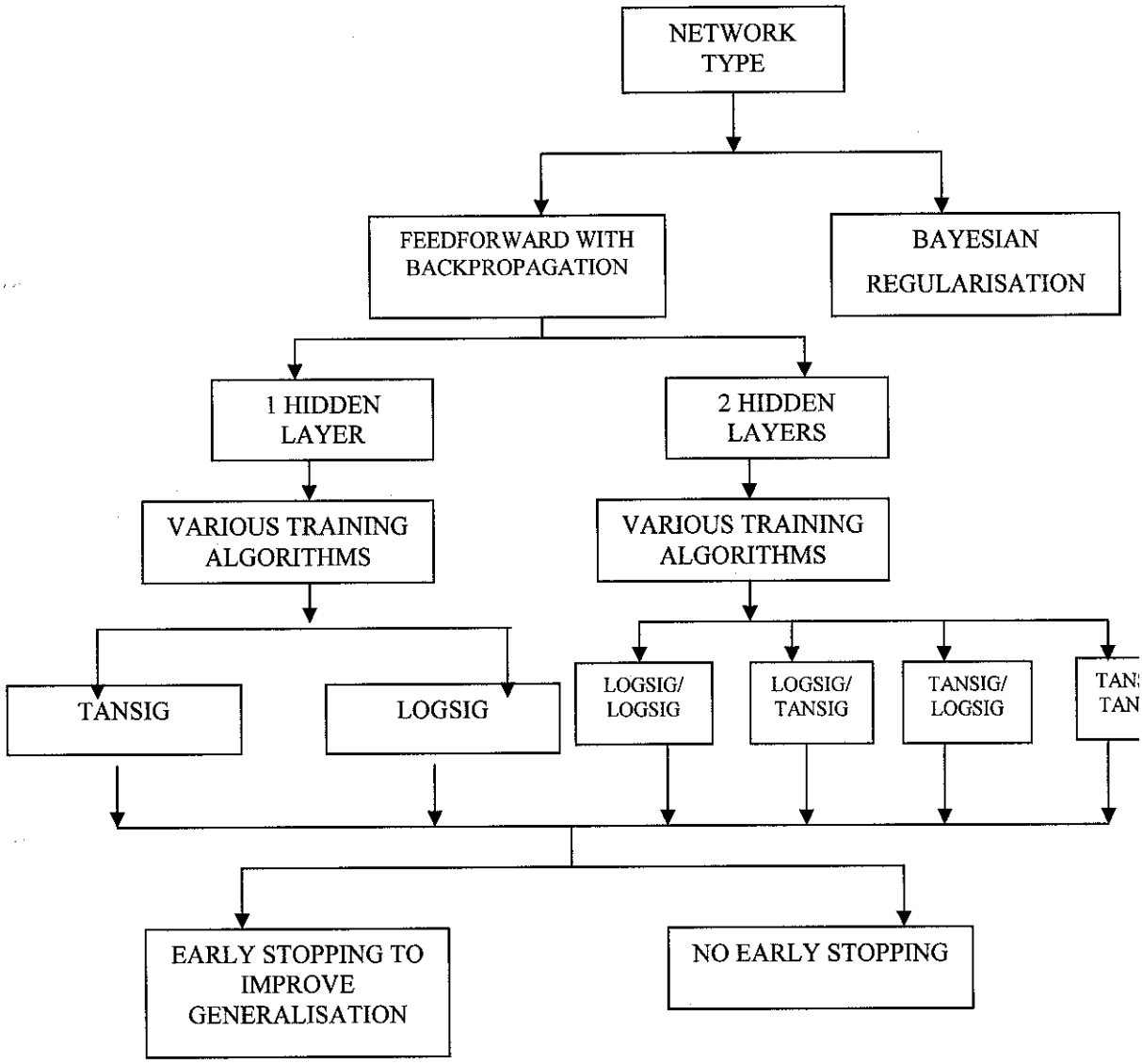


Figure 8: Process Flowsheet for Testing Network Configurations

3.2 COMPUTER SIMULATION AND MODELING

The second stage is computer simulation of the various network configurations to determine which configuration results in the best model for the process. The software chosen for this purpose is Neural Network Toolbox in MATLAB. The sample coding for creating, training and simulating the network is included in Appendix V.

The trained network is simulated using the test data to see how well it can predict the product yields from inputs it has not seen before. The different network architectures are compared and evaluated based on the following criteria:

i) Mean Squared Error (MSE) testing data set

ii) Size of network, i.e number of neurons in hidden layers

iii) Number of epochs required for convergence

iv) Speed of convergence

3.3 CASE STUDIES

The Case Studies are modeled to demonstrate the application of the systematic framework in predicting product yield and properties for refinery optimisation. The Case Studies are selected in such a way to illustrate the practical application of some of the issues touched upon in the framework development. For example, Case Study 1 shows the *multiple networks versus single network with multiple outputs* dilemma in action. Case Study 3 illustrates the effect of including data sets with missing values, while comparison between Case Study 2 and Case Study 3 can be used to demonstrate the effect of the amount of training data available on network performance.

3.3.1 Case Study 1: Prediction of Crude Distillation Product Yield

The objective of the model is to predict the product yield obtained from crude distillation, using crude properties as inputs to the model. For this case study, the data used is the same data that was used in the non-linear regression study, which was originally obtained from crude assay of 36 different crudes published in the Philippine National Oil Company Reference Manual (PNOC, 1987). Due to the limited availability of data, most of the data is reserved as training data, i.e. 30 data is used as training set, 3 for validation set and 3 for testing set. The division is done randomly.

For modeling the yield prediction of crude distillation products based on crude properties, the input vector is the properties of crude, and the output vector is the yield (in volume percent) of the various products. There are 7 elements in the output vector, corresponding

to the yield of the products, which are fuel gas, LPG, light naphtha, heavy naphtha, kerosene, diesel and reduced crude. The TBP cut points for the products are as given in Table 2. Two approaches are used for modeling the process, creating a single network having multiple outputs, and creating individual network for each product yield.

There are 9 crude properties reported in the PNOC assay; API gravity, pour point, sulfur content, basic sediment and water (BS&W), salt content, Reid vapor pressure (RVP), total acid number (TAN), viscosity at 100°F and viscosity at 122°F. From the multivariate analysis, it was found that the two viscosities at 100°F and 122°F are highly correlated (r^2 = 0.8617). The viscosity at 122°F is omitted from the model since it contains more missing data compared to the viscosity at 100°F. However, correlation results between API gravity with viscosity at 100°F and at 122°F also indicate some correlation between API gravity with viscosity (r^2 = 0.4273 for viscosity at 100°F and r^2 = 0.4500 for viscosity at 122°F). Compared to the r^2 values for the other elements, which are mostly less than 0.1, there appears to some justification for omitting viscosity altogether from the input. Moderately strong correlations were also found between sulfur content and viscosity, between basic sediment and water (BS&W) and salt content, and between salt content and total acid number (TAN). The result of the multivariate analysis is summarised in Appendix IV.

Incorporating the concept of prior knowledge regarding the process, the input elements to be included in the network are chosen based on consideration of the crude distillation process, i.e. what are the properties of crude that are expected to affect the yield of products most strongly?

Based on previous studies, API gravity is found to have the greatest impact on the expected yield of products from distillation. In fact, some researchers developed correlations using only API gravity as the input (Al Soufi, 1986, Maples, 1997). Therefore, API gravity is included as one of the input elements to the network. Pour point is also included because it has an effect on the relative paraffinicity and aromaticity of the crude, which in turn affects the product distribution.

The selection of input elements is also influenced by data availability factor. The crude assay set obtained from the PNOC Reference Manual contains many missing or unrecorded data for some of the properties. The data set is complete for all 36 crudes only for 3 properties, API gravity, pour point and sulfur content. Cursory examination of crude assay data from other sources also indicates that these 3 properties are the ones that are available for most assays, regardless of the source. Since the purpose of designing the neural network model is to predict product yield based on easily measurable crude properties, it is important to include only crude properties that are commonly recorded in a preliminary assay of a crude. This way, the model can be applied for a variety of situations. Properties such as total acid number, basic sediment and water, salt content, viscosity and Reid vapor pressure are less commonly included in the preliminary assay.

Based on the multivariate analysis and process understanding, API gravity, pour point and sulfur content are selected as input elements for the neural network model. Sample data set for the training data is included in Table 3. The complete data sets for training, testing and validation are included in Appendix III.

Table 3: Sample of Training Data Set

Input Parameters			Output Parameters						
API	Pour	Sulfur	FG	LPG	LN	HN	Kero	Diesel	Red.Crud
38.8	-30	1.1	0	2.6	7.6	14.4	15.7	25.4	34.3
30.8	-45	2.43	0.04	2.07	6.82	12.66	11.81	21.08	45.52
...	...	...	...	...	...	...	...	...	...

Notes:

1. API = API gravity of crude (°API)
2. Pour = Pour point of crude (°F)
3. Sulfur = sulfur content of crude (wt%)
4. FG = Fuel gas (C1/C2) yield (vol %)
5. LPG = LPG (C3/C4) yield (vol%)
6. LN = Light Naphtha (36-180°C) yield (vol%)
7. HN = Heavy Naphtha (180–320°C) yield (vol%)
8. Kero = Kerosene (320-450°C) yield (vol%)
9. Diesel = Diesel (450-690°C) yield (vol%)
10. Red.Crude = Reduced crude (690+) yield (vol%)

For the approach of creating a single network with multiple outputs, the output vector contains 7 elements, which are the percent yield of the various products. Therefore, the number of neurons in the output layer is set as 7. The data is preprocessed using the MATLAB function *premnmx*, which scale the inputs and outputs to values between –1 and

1. Neural network models using the configurations shown in Figure 7 are created and simulated in MATLAB, and the best model that gives the smallest Mean Squared Error (MSE) for the test data is selected.

3.3.2 Case Study 2: Prediction of Diesel Pour Point

The objective of the model is to predict the pour point of diesel product from crude distillation based on the properties of crude. The data is again obtained from PNOC assay, and consists of 9 crude properties that can be used to predict the pour point of diesel obtained from the crude distillation process. The crude properties are API gravity, pour point, sulfur content, basic sediment and water (BS&W), salt content, Reid vapor pressure (RVP), total acid number (TAN), viscosity at 100°F and viscosity at 122°F. This case study illustrates the possibility of predicting certain properties of a product from a unit operation, using the properties of the feed as the inputs.

The same rationale as in case study 1 applies for selection of input elements and preprocessing of data. Based on the multivariate analysis results, the input elements to the network are selected as crude API gravity, pour point and sulfur content. However, the output contains a single element, diesel pour point, so the number of neuron in the output layer is set as 1. From the 36 data sets available, 30 were used for training the network, 3 for validation and 3 for test data. The complete data set is included in Appendix III.

For division of data, instead of doing random division as in Case Study 1, for this case study, the test data were selected carefully to ensure that the values fall within the range of the training data. This means that the trained network is only asked to perform interpolation, not extrapolation. This is done to illustrate the effect of training data and test data range on the predictive capability of the network. It is expected that the prediction for Case Study 2 will be better than Case Study 1, since neural network performs better when predicting outputs for new inputs whose values are close to data sets that the network has been trained with.

3.3.3 Case Study 3: Prediction of Hydrocracker Total Gasoline Yield

The objective of the model is to predict the total gasoline yield from hydrocracking process, using gas oil feed properties and hydrogen consumption as the inputs. An important feature of hydrocracking process is the increase in volume of the product compared to the feed volume. Hence, the product yield is measured in liquid volume percent of feed (LV%), and the values can be more than 100%, owing to the increase in volume. The data is obtained from Maples (2000). The input elements are feed API gravity, feed K value (Watson characterisation factor), and amount of hydrogen consumed in standard cubic feet per barrel of feed (SCFB). The output is the total gasoline yield, which consists of the light hydrocrackate (C_5-180°F) and heavy hydrocrackate (180°F-450°C) fractions.

Some of the hydrocracking yield data from Maples contains missing values. There are 128 complete data sets, and 69 data sets with one or more missing values. The first approach taken is to omit the data sets containing missing values. The 80%-20% heuristics is followed such that 25 data sets are used for testing, and the rest for training and validation. The second approach used is to replace the missing values with the average of the corresponding variables, and to add the newly filled data sets to the training data. For example, for the API gravity of feed, the average value is 24.4. This value is used to fill in any missing API gravity value. For the first approach, 93 data sets are used for training, 10 for validation and 25 for testing. For the second approach, 162 data sets are used for training, 10 for validation and 25 for testing. Since the data sets are arranged according to decreasing hydrogen consumption in Maples (2000), random distribution of the data into training, validation and testing sets would results in data distribution that is not reflective of the system. The method used is to take every fifth and tenth data on the list as testing data, every ninth data as validation data, and the rest as training data.

CHAPTER 4
RESULTS AND DISCUSSION

The network performance was compared using the mean squared error (MSE) between the actual outputs and outputs predicted by the network for the test data. The mean squared error measures the error or difference between the output predicted by the neural network and actual output. The smaller the MSE, the better the network is at predicting the output. The mean squared error is given by the equation:

$$\text{MSE} = \frac{1}{Q}\sum_{k=1}^{Q} e(k)^2 = \frac{1}{Q}\sum_{k=1}^{Q} (t(k) - a(k))^2$$

Q = number of input patterns
t(k) = target (actual) output
a(k) = network output

To facilitate comparison between the different case studies on an equal basis, the MSE was computed for the preprocessed data (scaled to be within the interval –1 to 1). For each case study, the network configuration that gives the smallest MSE was selected as the best model for the problem.

For the models selected as the best network configuration for each case study, the results are also represented in the form of plots of predicted output versus actual outputs. If the model is able to predict the outputs perfectly, the plot will take the form of a straight line with slope of 1, and y-intercept of 0. Otherwise, the points will scatter across the line. Residual analysis was also conducted to find the absolute error as well as maximum deviation between the actual and predicted outputs. The purpose is to determine the accuracy of the model in predicting the outputs, i.e to estimate the confidence bound of the prediction.

4.1 NEURAL NETWORK MODEL FOR CASE STUDIES

4.1.1 Case Study 1: Prediction of Crude Distillation Product Yield

The modeling process for Case Study 1 was done using two different approaches; first by using a single network with multiple outputs, and then by using separate networks for each product yield. Various network configurations, as laid out in Figure 8, were tested and simulated in MATLAB to select the network which gives the smallest MSE. For the single network with multiple output approach, the configuration that gives the smallest Mean Squared Error (MSE) of the test data is a feedforward network with two hidden layers, 5-5-7 neurons configuration, tangent-sigmoid transfer function in both hidden layers, Levenberg-Marquardt learning algorithm, with early stopping. The network architecture is shown in Figure 9:

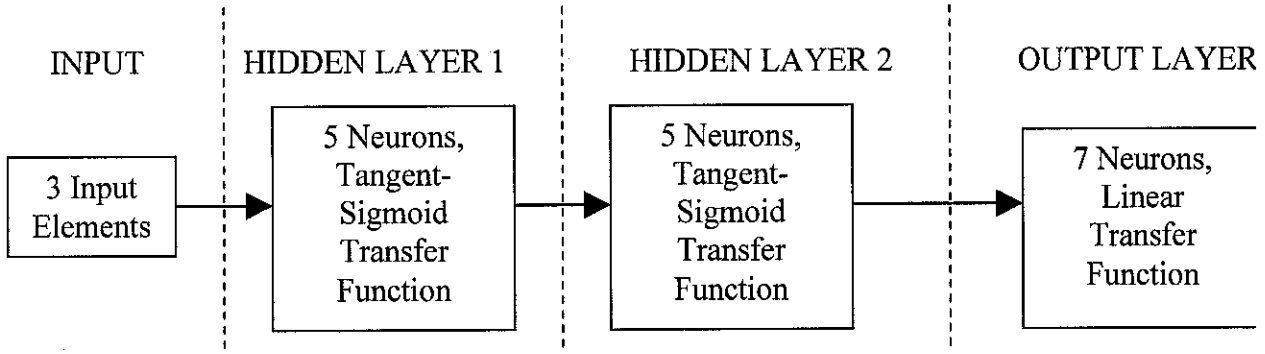


Figure 9: Selected Neural Network Architecture for Case Study 1

The network was trained for 104 epochs before stopping due to the increase in error in the validation data set. With early stopping, when the network was simulated using the training data, the MSE between the target (actual output) and the network output is 0.1397 and for testing data is 0.2157. The training data MSE is found to be larger when the network is trained using early stopping compared to without early stopping., which means that the network is not fitting the training data exactly. However, the advantage of using early stopping is that the MSE for the testing data is smaller, which means the network has a better generalisation and predictive capability when faced with data it has not seen before. To illustrate this, Table 4 shows the comparison between the MSE values for

network trained with early stopping, and without early stopping, using the same network configuration as described above.

Table 4: Comparison between Early Stopping and No Early Stopping Network

Network	MSE (Training)	MSE (Testing)
Early Stopping	0.1397	0.2157
No Early Stopping	9.86E-04	2.0104

For comparison with the second approach (individual network with a single output for each product yield), MSE for each output element were also determined. The same network configuration was used to run the individual networks, with the exception that the number of neurons in the output layer is now reduced to 1. Comparison of the MSE values between the two approaches is shown in Table 5:

Table 5: Comparison of MSE for Multiple and Single Output Networks

Product	Single Network, Multiple Outputs	Individual Network
	MSE	MSE
Fuel Gas	0.851	5.064
LPG	0.561	0.118
Light Naphtha	0.279	0.699
Heavy Naphtha	0.313	0.058
Kerosene	0.278	0.474
Diesel	0.462	0.472
Reduced Crude	0.078	0.044

Based on the MSE comparison in Table 5, it is seen that using individual network for each product yield does not significantly improve the prediction. In fact, for some products, the individual networks actually give a worse prediction (fuel gas, light naphtha, kerosene and diesel) than the single network with multiple outputs. Improvement in prediction is observed for LPG, heavy naphtha and reduced crude. Theoretically, it is expected that using a separate network for each product yield would improve the predictive capability of the network. This is because by using a separate network with only one output, the network size, which is the number of weights in the network, is reduced from 75 to 45. The number of weights is calculated as follows:

No of weights = (No of elements in input vector) x (No of neuron in first layer) +
(No of neuron in first layer) x (No of neurons in second layer) +
(No of neurons in second layer) x (No of neurons in output layer)

For the network with multiple outputs, the output layer has seven neurons, corresponding to the seven outputs of the network, while the individual network has only one neuron in the output layer. The reduction in the network size is expected to increase the generalisation capability of the network and avoid overfitting. However, in this case, the number of training data is so limited (30 data sets) that the reduction in network size does not guarantee an improvement in the network performance.

The plots of actual yield the yield predicted using the two approaches are shown in Figures 10-16. The straight line represents the ideal situation where the predicted output is equal to the actual output. From the graphs, it is seen that for both approaches, the worst prediction is for fuel gas, and the best is for reduced crude. The unsatisfactory prediction for fuel gas yield is probably due to the fact that the actual yield is equal to or very close to zero, which causes difficulty for the network to adjust the weights and biases during training.

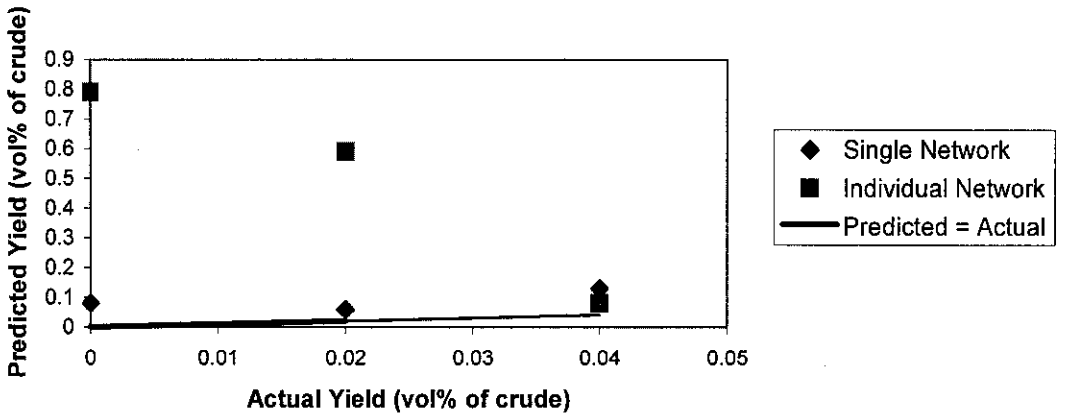


Figure 10: Actual versus Predicted Yield of Fuel Gas

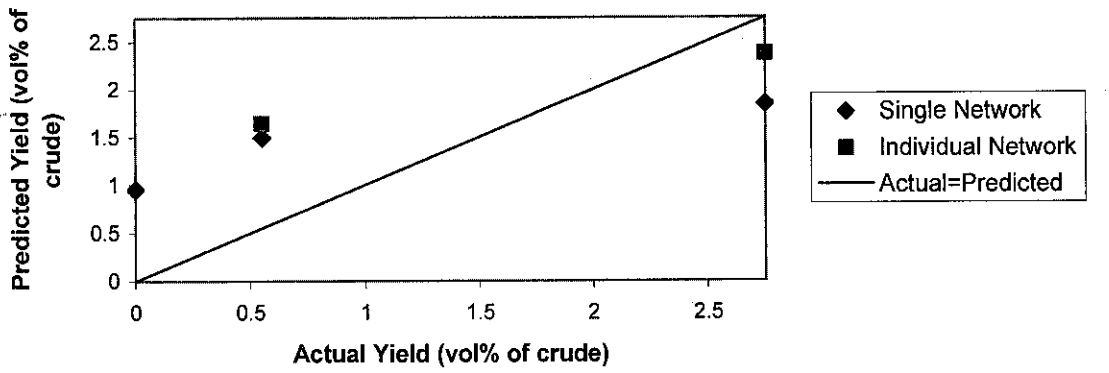


Figure 11: Actual versus Predicted Yield of LPG

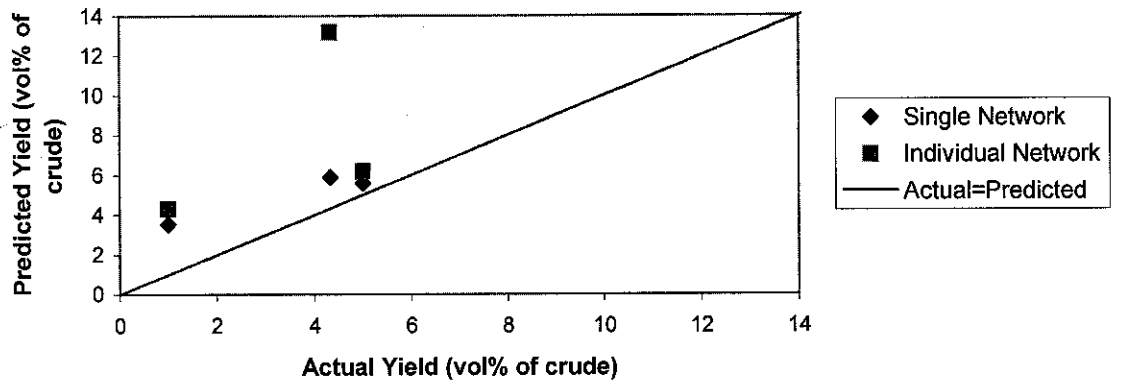


Figure 12: Actual versus Predicted Yield of Light Naphtha

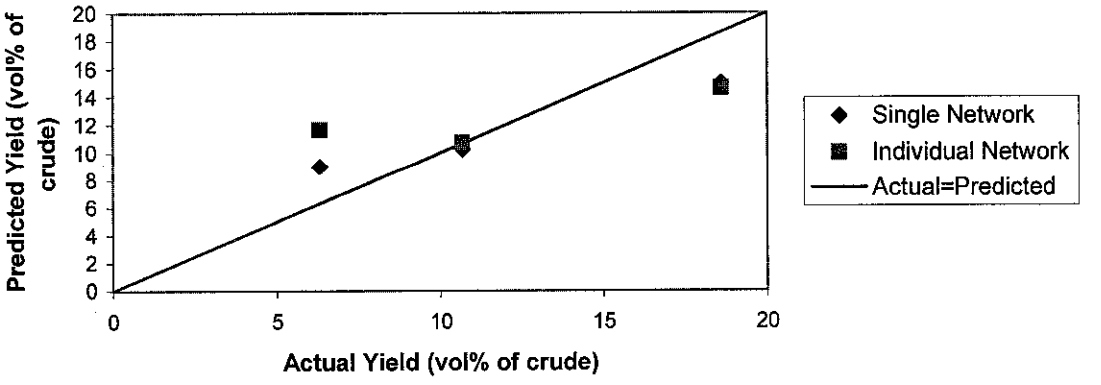


Figure 13: Actual versus Predicted Yield of Heavy Naphtha

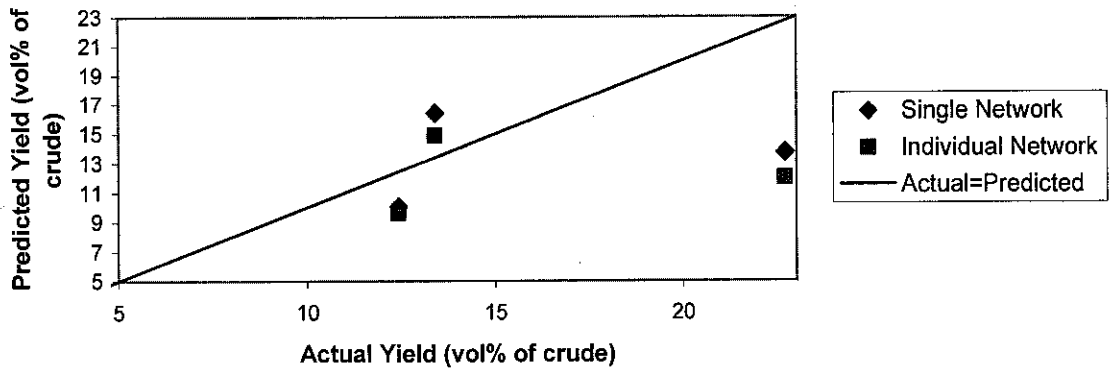


Figure 14: Actual versus Predicted Yield of Kerosene

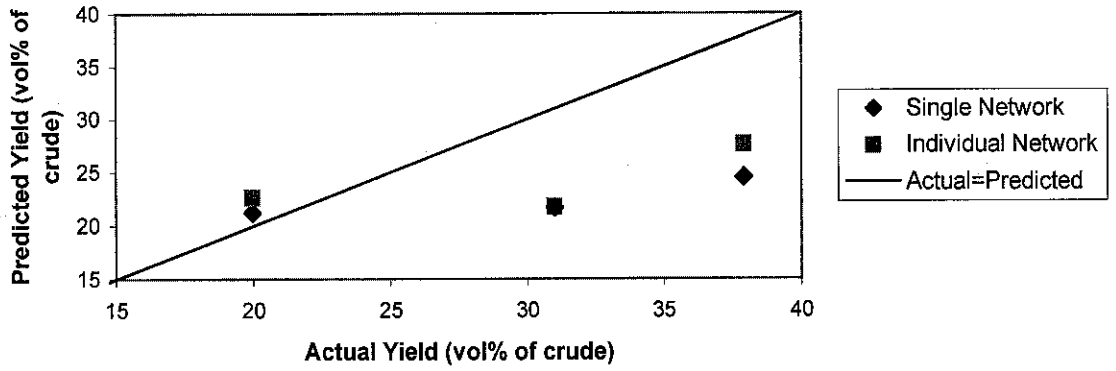


Figure 15: Actual versus Predicted Yield of Diesel

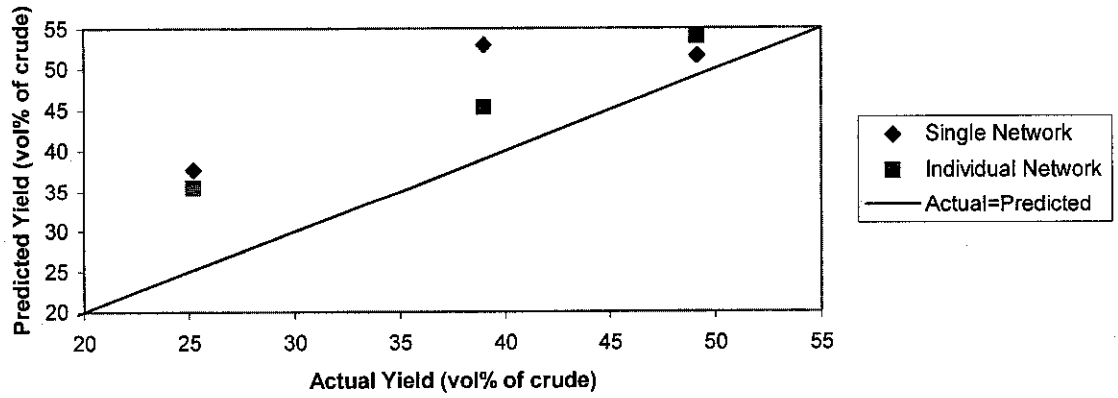


Figure 16: Actual versus Predicted Yield of Reduced Crude

Appendix VI shows the values for the actual and predicted yield, as well as deviation between the actual and predicted. Residual analysis is conducted to calculate the range of deviation for each product, as shown in Table 6:

Table 6: Deviation Range For Case Study 1

Product	Mean Actual Yield	Deviation Range	
		Single Network	Individual Network
FG	0.02	+0.04 to +0.09	+0.04 to +0.79
LPG	1.10	-0.91 to +0.95	-0.11 to +1.09
LN	3.44	+0.59 to +2.54	+1.18 to +8.84
HN	11.86	-3.63 to +2.69	-3.99 to +5.37
Kerosene	16.18	-8.98 to + 3.01	-10.7 to +2.81
Diesel	29.62	-13.37 to +1.24	-10.25 to +2.70
Reduced Crude	37.78	+2.46 to +14.00	+4.88 to +10.33

For the purpose of refinery optimisation, the large range in the deviation between the actual and predicted value shown above is clearly not satisfactory. However, it has to be noted that the result is obtained for a data set that has some very important constraints. The main constraint is the very small number of data sets used for training the network. The second constraint involves the spread of the data, i.e. whether the data set is representative of the whole range of possible input/output combinations. In an actual refinery, the data set available would be significantly larger, and consequently more representative of the whole population.

4.1.2 Case Study 2: Prediction of Diesel Pour Point

The network configuration which gives the smallest MSE for test data is a feedforward network with two hidden layers, 5-10-1 neurons configuration, log-sigmoid transfer function in both hidden layers, Levenberg-Marquardt learning algorithm, with early stopping. The network was trained for 66 epochs before stopping due to increase in MSE of validation data. The MSE for training data is 0.1088, for validation data is 0.0027 and for testing data is 0.0392. The network architecture is shown in Figure 17:

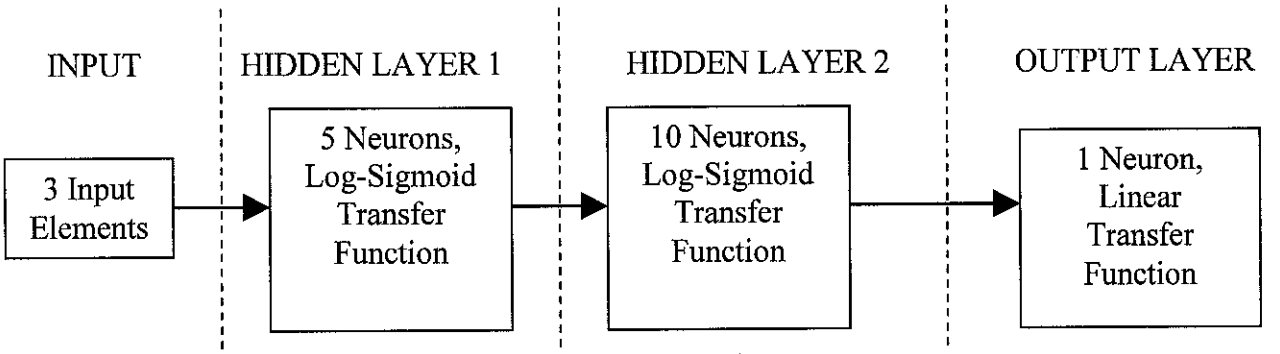


Figure 17: Selected Neural Network Architecture for Case Study 2

Comparison between actual and predicted diesel pour point is shown in Table 7 and the plot in Figure 18. The selected model gives prediction which is consistently lower than the actual values. The maximum deviation is –7.0°F.

Table 7: Deviation Results for Case Study 2

Actual	Predicted	Deviation
25	20.1	-4.9
35	29.3	-5.7
20	13.0	-7.0

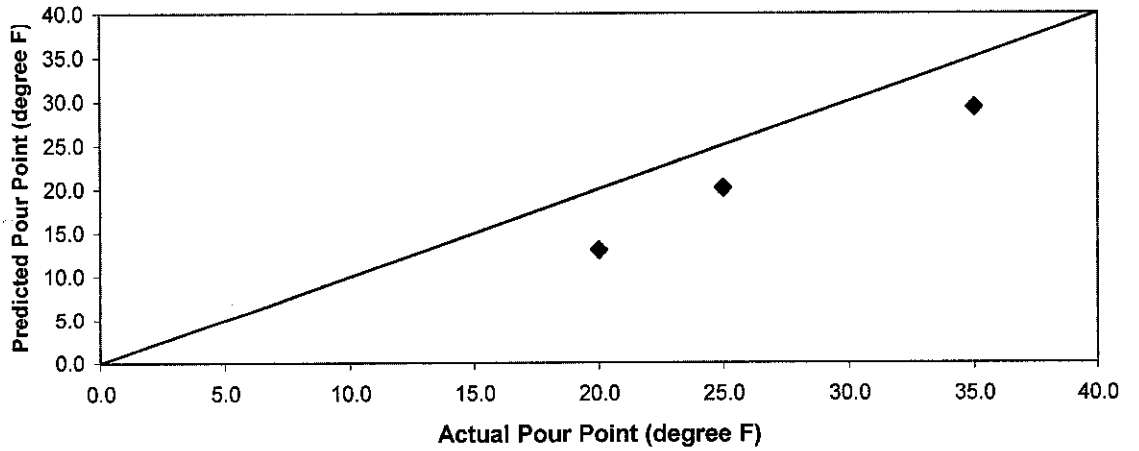


Figure 18: Actual versus Predicted Diesel Pour Point

Compared to the first case study, the prediction for the diesel pour point is slightly better in terms of MSE and maximum deviation. Division of data into training, validation and

testing data for Case Study 1 was done randomly, and some of the test data are outside the range of the training data. On the other hand, for the prediction of diesel pour point, the test data were carefully selected to ensure that they are within the range of the training data. This confirms the hypotheses that neural network model performs better when tested with data that is within the range of the data that the network was trained with.

The better prediction for Case Study 2 compared to Case Study 1 could also be attributed to the relative characteristics of the data. For Case Study 1, the output values, which are the yields of various products, are more distributed across many different values. For diesel pour point, the network is trained using data with outputs having values that are fairly consistent, i.e. multiples of five. The consistent characteristics of the outputs used to train the network probably made it easier for the neural network to adjust the weights and biases during the training process, resulting in a better prediction.

4.1.3 Case Study 3: Prediction of Hydrocracker Total Gasoline Yield

The network configuration which gives the smallest test data MSE prediction of hydrocracker total gasoline yield is a feedforward network with two hidden layers, 3-5-1 neurons configuration, log-sigmoid transfer function in both hidden layers, Levenberg-Marquardt learning algorithm, with early stopping. The network was trained for 162 epochs before stopping due to increase in MSE of validation data. The MSE for training data is 0.0720, for validation data is 0.2409 and for testing data is 0.2461. The network architecture is shown in Figure 19:

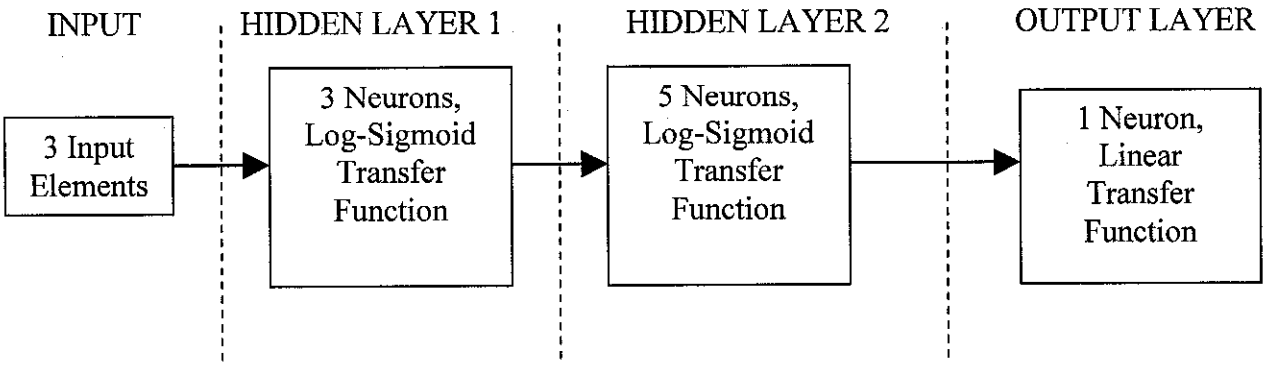


Figure 19: Selected Neural Network Architecture for Case Study 3

Using the same network configurations as above, the training data was increased to include data sets containing missing values of one of the input parameters. The missing values are substituted with the average of the variables across the whole data set. For example, for API gravity, the average value is 24.4, so if a data set does not contain the value of API gravity, 24.4 is used as its API gravity. The average value for Watson characterisation factor, K, and hydrogen consumption in SCFB is 11.5 and 1990, respectively. The network was trained for 136 epochs before stopping due to increase in MSE of validation data. The MSE for training data is 0.1133, for validation data is 0.1321 and for testing data is 0.2229.

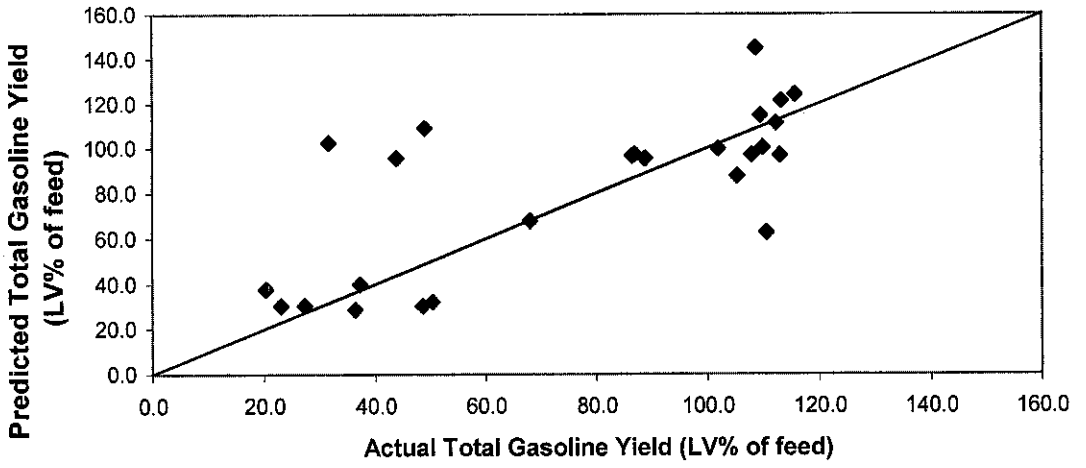


Figure 20: Actual versus Predicted HCK Gasoline Yield (Original Data Set)

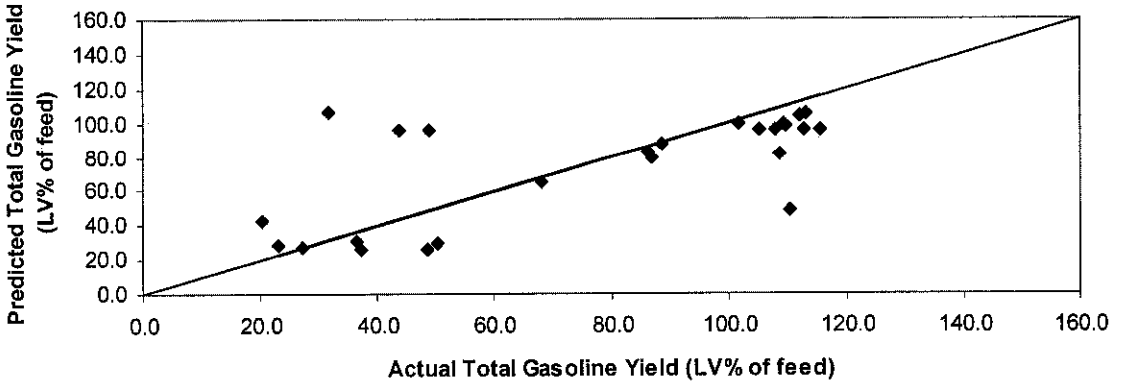


Figure 21: Actual versus Predicted HCK Gasoline Yield (With Additional Data)

Table 8: Comparison of Results using Original and Additional Data for Case Study 3

Property	Original Data	With Added Data
MSE	0.2461	0.2229
Maximum Negative Deviation	-47.9	-62.1
Maximum Positive Deviation	71.0	75.4

Comparing the results obtained using the original and additional training data set (Table 8), it is observed that the additional data set gives a slightly better prediction in terms of the mean squared error value. However, the spread in terms of maximum positive and negative deviation is slightly larger for the case with additional data set. Comparison of the plots of predicted versus actual output (Figures 20 and 21) indicates that the outputs predicted by the network in both cases do not differ by much, except for a few data points. This is confirmed by the plot in Figure 22, which plots the two predictions on the same graph.

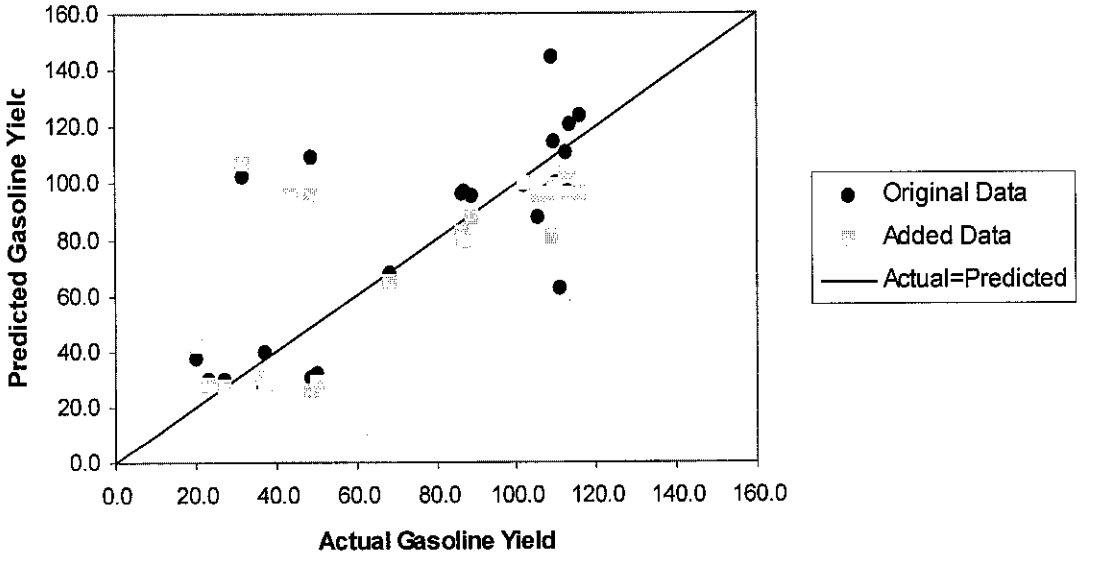


Figure 22: Actual Versus Predicted for Case Study 3

4.2 COMPARISON OF FEEDFORWARD NEURAL NETWORK TO BAYESIAN REGULARISATION

For all three case studies, multilayered feedforward network with early stopping gives a better prediction than Bayesian regularisation for the same network architecture (same number of hidden layers, number of neuron, and transfer function). Table 9 shows the comparison between the results obtained using feedforward network with early stopping, and Bayesian regularisation, for the same network configuration as selected before:

Table 9: Comparison of MSE between Early Stopping and Bayesian Regularisation

Case Study	MSE	
	Early Stopping	Bayesian Regularisation
Case Study 1	0.1397[a]	0.1465[a]
Case Study 2	0.0392	0.0825
Case Study 3	0.2461[b]	0.2625[b]

Note: a: Refers to MSE for the single network with multiple outputs approach
b: Refers to MSE for training with the original data set

Early stopping and Bayesian regularisation are both methods for improving the generalisation capability of the network, i.e. to avoid the network from overfitting the training data. The advantage of Bayesian regularisation is that no data sets have to be set aside for validation purpose, as is the case for early stopping This means more data can be used for training purpose, since the validation data is now included in the training data set. For the case studies used in this project, the number of data used for validation is small (3 each for Case Study 1 and 2, and 10 for Case Study 3), so the effect of increasing the number of training data by using Bayesian regularisation is not very pronounced. However, the difference in the MSE values are not very significant, so Bayesian regularisation should also be considered as a good option for the model.

4.3 COMPARISON OF DIFFERENT NETWORK ARCHITECTURE

4.3.1 One Hidden Layer versus Two Hidden Layers

For all three case studies, networks with two hidden layers perform better than networks with 1 hidden layer in terms of the MSE of test data. The disadvantage of the two hidden

layer, however, lies in terms of the slightly longer time required for the solution to converge, due to the increase in the number of weights and biases. However, this difference in convergence time is almost negligible, especially if the network is simulated on a computer with high memory capacity.

4.3.2 Number of Neurons

The optimal number of neurons is selected on a trial-and-error basis. The goal is to find the minimum number of neuron for a given configuration that will give the smallest MSE when the network is simulated using the test data. For the case studies, for each network configuration, the number of neurons tested starts from 1 neuron, and is increased consecutively by 1 neuron each time. It is observed that as the number of neurons is increased initially, the MSE of both training data and test data decreases. However, as the number of neurons increases beyond a certain point, the MSE of test data starts to increase even as the MSE of training data continues to decrease. This indicates that the network size has become too large, and the network is no longer generalising the function, but merely fitting the training data (as indicated by the increase in test data MSE and decrease in training data MSE). The optimum number of neurons for the network configuration is the one where the test data MSE is at a minimum. This is illustrated in Table 10, which shows the MSE for different number of neurons for the single hidden layer configuration for Case Study 3. The shaded part shows the optimum number of neurons, which is 5 in this case:

Table 10: Effect of Number of Neurons on Network Performance

Number of Neurons	MSE Training Data	MSE Test Data
1	0.0759	0.2849
2	0.0490	0.2352
3	0.0443	0.2988
4	0.1225	0.2755
5	0.1382	0.2501
6	0.0637	0.2904
7	0.0323	0.3032
8	0.0287	0.3206

4.3.3 Transfer Function

For network that uses backpropagation algorithm for updating the weights and biases, the type of transfer function used in the hidden layers must be a sigmoid function. The software used in modeling the case studies, MATLAB Neural Network Toolbox provides two options, the tangent-sigmoid function, and the log-sigmoid function. The tangent-sigmoid transfer function squashes the inputs to nonlinear range –1 to 1, while the log-sigmoid to range 0 to 1. From modeling the case studies, it is observed that the choice of transfer function in the hidden layer does not affect the network performance very much.

4.3.4 Training Algorithm

The choice of which training algorithm is used does not affect the network performance very much in terms of the test data MSE. The effect is more on the time required for the network to converge. For modeling the case studies, it is observed that for most of the network configurations tested, the Levenberg-Marquardt learning algorithm gives the fastest convergence.

4.4 FURTHER OPTIONS FOR IMPROVING MODEL PERFORMANCE

Various methods are suggested in literature to improve to performance of network with limited data set, such as cross-training (leave one out basis) and stacked neural network. In cross-training, the network is trained on all available data, except one which is held out for validating the performance of the network. This procedure is repeated until the network has been validated against all the data. The purpose is to use as much of the data as training data as possible in order to provide sufficient information for close simulation of the input/output mapping (Aldrich, 2001). To address the problem of limited data set, Zhang (2001) proposed the method of bootstrap aggregated neural network model, in which the process data is randomly re-sampled to form a number of different training and testing data sets. Neural networks are then developed for each re-sampled data set. However, instead of selecting the network perceived as the best, several networks are combined to form the model for prediction purposes. The combined network is called "stacked network". Unfortunately, due to time constraint, these approaches could not be incorporated in the project.

4.5 FEASIBILITY OF NEURAL NETWORK MODELING FOR REFINERY OPTIMISATION

At this point, it would be appropriate to comment on the suitability of using neural network modeling for the purpose of predicting refinery product yield and properties. In general, the prediction results obtained for the 3 case studies above are not very satisfactory. However, the results have to be interpreted within the context of the constraints faced during the project implementation. The author has elected to use data available from literature instead of data from an actual plant. The advantage of this approach is that the study is able to benefit from the insights provided by the respective authors on the processes modeled in the case studies. Consequently, theoretical and analytical understanding of the process can be included in the formulation of the neural network model, as opposed to proceeding in a purely empirical manner. On the other hand, the major disadvantage is that a limited number of data sets are available for training the network, which severely affects the predictive capability of the model. The other constraint is to complete the project within the three months duration (1 university semester).

As stated in the objectives, the main goal of this project is to focus on the systematic methodology for neural network modeling. The aim is to gain sufficient insights regarding the theoretical background and fundamental principles of neural network modeling, its methods of implementation, as well as how the results should be interpreted, for the purpose of using neural network modeling to predict product yield and properties for refinery optimisation. The case studies are included to illustrate how the methodology can be implemented. As such, the results from the case studies, in terms of the accuracy of the predicted outputs, cannot be interpreted as representative of how neural network models for other processes will behave. In a modern refinery, the widespread use of computerised data storage and management means that the historical data available for any process to be modeled will be very large, going back years into the past. Therefore, problem of limited data set will not occur.

As is the case in most modeling method, whenever the model output does not predict the actual result very well, the cause can be traced to either one of two factors. The first factor is a mismatch or incompatibility between the process characteristics and the modeling approach itself. An example would be using linear regression technique to model a process that is inherently non-linear. The second factor is related to the degree to which the characteristics of the process are accurately represented in the model. A model is essentially a simplification of the actual process, and it would be impossible to include all the characteristics of the actual process. However, enough information must be retained in the model so that the model does not become a gross oversimplification of the actual process. To this end, a high degree of understanding of the process is essential. In neural network modeling, process understanding is especially important during the input selection stage, in order to determine which inputs actually have an effect on the process output. The basic assumption in neural network modeling is that a relationship actually exists between the network inputs and the network outputs. The relationship could be very complex or weak, but it must be there. Otherwise, the network output will fall victim to *the garbage in, garbage out* phenomena.

These two factors must be weighed carefully when determining the suitability of a particular modeling approach for solving a problem. The fundamental question here seems to be; is the prediction inaccurate due to the inherent incompatibility between the process and the modeling approach, or is it due to flaws and shortcomings in incorporating the process characteristics into the model, as well as during the modeling process itself? In this case, since the research was carried out by a beginner student in the field of neural network, as well as in the study of refinery processes used in the case studies, the latter would be the more probable reason.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

The study has achieved its objective of developing a systematic methodology or framework for designing neural network model for product yield and properties prediction in a refinery. The framework focuses on aspects such as data collection and division, input elements selection, design of network architecture, learning algorithm selection, network training, and network simulation using new data set. The methodology was applied for the problem of predicting product yield and diesel pour point from crude distillation unit based on properties of crude, and the prediction of gasoline yield from hydrocracker based on feed properties and hydrogen consumption. The network architecture was designed and the network created, trained and simulated using Neural Network Toolbox in MATLAB software.

To evaluate the performance of the neural network model, the trained network was simulated using data that the network has not seen before (the test data). The output predicted by the neural network model is compared with the actual output. The mean squared error (MSE) between the predicted and actual outputs provides an indication of how well the model is predicting the output. For the three case studies, the MSE ranges from 0.1397 for the first case study, 0.0392 for the second case study and 0.2461 for the case studies.

Several conclusions can be drawn based on the results obtained from modeling the three case studies. It is found that in general, the performance of a neural network model is limited when the number of data available for training the network is small. The network also performs worse when asked to predict the output for test data whose input values do not falls within the range of the training data. On the other hand, when presented with test data whose value is close to specific data set with which the network is trained, the network predicts the output with greater accuracy. The specific network configuration most suitable for a specific problem will depend on the nature of the problem itself.

In conclusion, the study has shown the tremendous potential of neural network modeling in predicting crucial product yield and properties. Careful considerations of the limitations of neural network, some of them illustrated in this study, will allow for the formulation of a better model with greater accuracy.

RECOMMENDATIONS FOR FUTURE WORK

Future study on application of neural network modeling for prediction of product yield and properties in a refinery could focus on several aspects, as follows:

- Incorporation of techniques for improving the performance of network with limited data set, such as the cross-training method, and stacked neural network.
- Integration of neural network model into plant's Advanced Process Control strategy. The neural network model could be used to analyse process data available from Distributed Control System (DCS) to get inferential property predictions for properties that are hard to measure on-line, such as composition, flash point etc. This will translate to savings in terms of time and cost from reduction in lab analysis or use of on-line analysers.
- Study on the inversion property of neural network. The inversion process takes a neural network that maps input to output and invert it. The inverted network will give the set of inputs necessary to achieve a desired output. The inverted model can be used to control the process in a neural network-based controller.

REFERENCES

Abdul Mutalib, M.I., Tang, C.S., and Mohd Shariff, A., 2004, "Development of a Systematic Methodology for Predicting Crucial Properties or Yield for the Optimisation of Refinery Operation," Universiti Teknologi Petronas, to be published in SOMCHE 2004

Aldrich, C., and Slater, M.J., 2001, "Simulation of Liquid-Liquid Extraction Data with Artificial Neural Networks", *Application of Neural Networks and Other Learning Technologies in Process Engineering,* , London, Imperial College Press

Al-Soufi, H.H., Al-Najjar, H.S., Al-Duri, S.F., and Mohammed, H.F., 1986, , "Volume Distilled can be Calculated from Crude API Gravity", *Oil and Gas Journal,* vol 84, 78-80

Barbosa, C.H., Melo, B., Vellasco, M., Pacheco, M., and Vasconcellos, L.P., 2002, "Bayesian Neural Network on the Inference of Distillation Product Quality," *Proceedings of the VII Brazilian Symposium on Neural Networks*

Barsamian, A. and Macias, J., 1998, "Inferential Property Predictors Using Neural Network," *Hydrocarbon Processing* 77(10): 107

Bishop, C.M., 1995, *Neural Network for Pattern Recognition,* New York, Oxford University Press

Demuth, H. and Beale, M., 1998, *Neural Network Toolbox: For Use with MATLAB,* Massachusetts, The MathWorks, Inc.

Gary, J.H. and Handwerk, G.E., 2001, *Petroleum Refining: Technology and Economics,* New York, Marcel Dekker Inc.

http://homeflash.net/~celjure/enginering/petroplan/assay, 24 Aug 2004

Leffler, W.L., 1979, *Petroleum Refining for Non-Technical Person,* Oklahoma, PenWell Publishing

Mahrotra, K., 1997, *Elements of Artificial Neural Networks,* Massachusetts, The MIT Press

Manning, B., 1998, "Artificial Neural Network in Architecture," *Engineering Technology*

Maples R.E., 1997, "Linear Regression Determines Crude Properties at Variable Boiling Points", *Oil and Gas Journal,* Nov 3, 1997: 72

Maples, R.E., 2000, *Petroleum Refinery Process Economics,* Oklahoma, Penn Well Corporation

Moreno, A., Alba, M.C., Manriquez, L., and Mendoza, P., 1995, "Correlations Estimate Volume Distilled Using Gravity, Boiling Point", *Oil and Gas Journal,* Oct 23, 1995: 94

PNOC (The Energy Company), 1987, *Petroleum Downstream Logistics Course,* Philippine National Oil Company

Ramos, A.C.B. and Cunha, A.M., 1998, "Using Neural Nets to Predict LPG Weathering," *Industrial Applications of Neural Networks,* New Jersey, World Scientific Publishing

Schalkoff, R.J., 1997, *Artificial Neural Networks,* New York, McGraw Hill

Smith, M., 1993, *Neural Networks for Statistical Modeling,* New York, Van Nostrand Reinhold

Speight, J.G., 2001, *Handbook of Petroleum Analysis,* New York, John Wiley and Sons, Inc.

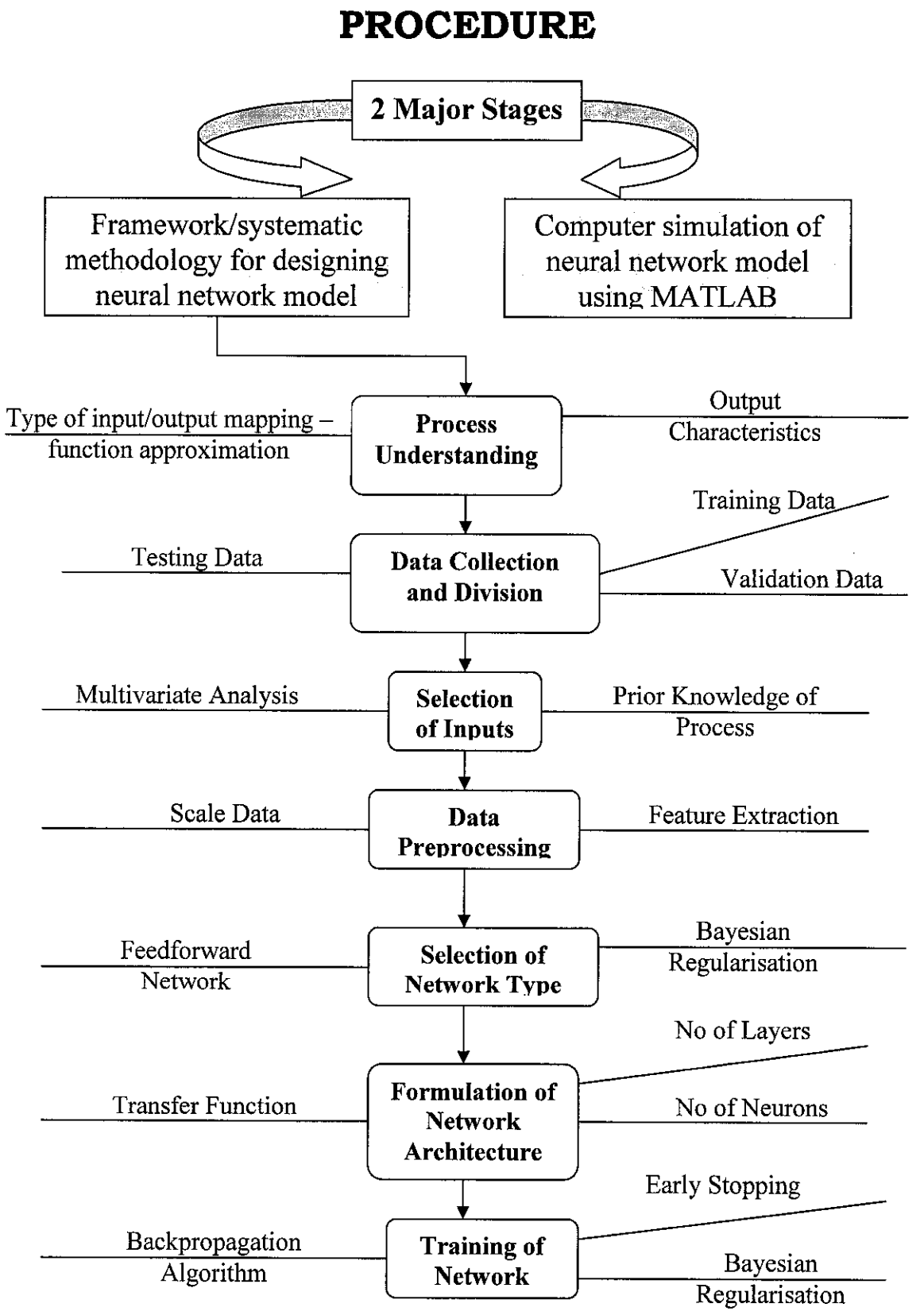
Winston, P.H., 1993, *Artificial Intelligence,* New York, Addison Wesley

Zhang, J., Morris, A.J., 2001, "Inferential Estimation and Optimal Control of a Batch Polymerisation Reactor Using Stacked Neural Networks", *Application of Neural Networks and Other Learning Technologies in Process Engineering,* London, Imperial College Press

APPENDIX I: PROJECT SCHEDULE

No	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	Selection of Project Topic	■														
	-Topic assigned to student															
	-First meeting with supervisor															
2	Preliminary Research Work		■	■												
	-Introduction to project scope and objectives															
	-Literature review															
	-Project Planning															
3	Submission of Preliminary Report			●6/8												
4	Project Work				■	■	■	■								
	-Literature Review															
	-Practical work (simulation)															
5	Submission of Progress Report							●3/9								
6	Project work continue								■	■	■	■				
7	Submission of Dissertation Draft to Supervisor											● 7/10				
8	Submission of Dissertation Final Draft													● 21/10		
9	Oral Presentation															● 22-26/11

APPENDIX II: PROJECT METHODOLOGY



APPENDIX III: DATA SET FOR CASE STUDIES

Case Study 1: Prediction of Crude Distillation Product Yield

Training Data Set

Input Parameters			Output Parameters						
API	Pour	Sulfur	FG	LPG	LN	HN	Kero	Diesel	Red. Crud
38.8	-30	1.1	0	2.6	7.6	14.4	15.7	25.4	34.3
30.8	-45	2.43	0.04	2.07	6.82	12.66	11.81	21.08	45.52
23	60	1.12	0.02	0.3	1.64	4.09	5.1	16.79	72.06
21.8	-35	3.34	0.33	1.81	3.74	9.88	9.35	18.68	56.21
46.2	-30	0.6	0.03	1.62	13.13	29.08	17.27	29.77	9.1
44.8	40	0.06	0.02	0.82	3.7	18.13	23.82	38.31	15.1
35.3	95	0.07	0	0.65	2.33	8.44	10.36	23.73	54.49
47.6	0	0.023	0.07	2.92	10.25	35.36	23.15	24.29	3.96
21.1	54	0.21	0	0.02	0.48	3.2	6	14.7	75.6
32.5	-5	1.68	0.06	2.45	6.63	12.33	13.67	24.82	40.04
36.3	34	0.08	0.03	2.57	10	26.2	20	29.7	11.5
31.4	-5	2.56	0.12	3.09	6.63	11.54	11.95	20.44	46.23
33.9	5	1.44	0.26	1.48	5.97	14.88	14.79	24.74	37.88
42.1	-10	0.028	0.04	3.12	8.82	29.85	22.92	26.54	8.71
39.3	-30	0.81	0.1	1.44	6.63	18.01	16.08	27.22	30.52
33.9	-15	2.03	0	0	6.5	12.5	11	22	48
41.3	5	1.37	0.05	3.9	10.65	15.7	15.8	24.6	29.3
35.2	75	0.105	0.5	2.1	7.96	17.65	13.25	27.68	30.88
27.2	20	2.03	0.03	0	0	10.97	14	38	37
34.8	-10	0.97	0.04	0.82	6.11	13.45	14.07	25.08	40.43
31.3	90	0.08	0.07	1.02	1.37	9.8	14.5	35.5	37.73
38	-15	0.49	0	0	0.43	16.63	22.96	34	25.98
33.5	-25	1.41	0.37	2.3	6.83	13.2	11.67	21.53	44.1
36.2	-20	1.95	0	2.48	7.21	15.58	15.19	24.36	35.18
36.2	35	0.07	0.07	1.66	8.27	16.15	19.59	33.02	21.24
33.3	0	1.95	0.06	2.1	7.96	13.29	13.33	23.07	40.19
38.9	-45	1.79	0.01	1.4	6.48	14.13	13.55	24.62	39.81
36.9	55	0.1	0.09	2.55	6.25	18.16	17.38	30.1	25.47
44.5	45	0.64	0.08	0.89	7.87	24.58	21.61	31.03	13.94
28	-30	2.82	0.09	2.35	5.9	10.45	10.2	20.53	50.48

Validation Data Set

Input Parameters			Output Parameters						
API	Pour	Sulfur	FG	LPG	LN	HN	Kero	Diesel	Red. Crud
40.4	30	0.21	0.04	2.37	8.76	16.53	14.72	24.34	33.24
35.4	20	0.68	0.02	0.71	4.78	15.94	16.93	29.98	31.64
31	0	1.62	0.04	2.73	6.73	12.9	11.18	19.22	47.2

Testing Data Set

Input Parameters			Output Parameters						
API	Pour	Sulfur	FG	LPG	LN	HN	Kero	Diesel	Red. Crude
26.5	25.2	0.13	0	0	1	6.3	22.7	31	39
33.1	50	0.08	0.02	0.55	4.32	18.61	13.4	37.9	25.2
28	-35	2.85	0.04	2.75	5	10.68	12.43	19.95	49.15

Case Study 2: Prediction of Diesel Pour Point

Training Data Set

Input Parameters			Output
API	Pour	Sulfur	Diesel Pour Point
38.8	-30	1.1	5.00
30.8	-45	2.43	10.00
23	60	1.12	20.00
21.8	-35	3.34	5.00
46.2	-30	0.6	20.00
44.8	40	0.06	20.00
35.3	95	0.07	40.00
47.6	0	0.023	5.00
21.1	54	0.21	5.75
32.5	-5	1.68	10.00
36.3	34	0.08	8.60
31.4	-5	2.56	15.00
33.9	5	1.44	10.00
42.1	-10	0.028	25.00
39.3	-30	0.81	20.00
33.9	-15	2.03	20.00
41.3	5	1.37	20.00
35.2	75	0.105	65.00
27.2	20	2.03	40.00
34.8	-10	0.97	5.00
31.3	90	0.08	16.00
38	-15	0.49	5.00
33.5	-25	1.41	15.00
36.2	-20	1.95	16.00
36.2	35	0.07	16.00
33.3	0	1.95	5.00
38.9	-45	1.79	10.00
36.9	55	0.1	25.00
44.5	45	0.64	15.00
28	-30	2.82	10.00

Validation Data Set

Input Parameters			Output
API	Pour	Sulfur	Diesel Pour Point
40.4	30	0.21	15
35.4	20	0.68	15
31	0	1.62	10

Training Data Set

Input Parameters			Output
API	Pour	Sulfur	Diesel Pour Point
26.5	25.2	0.13	25
33.1	50	0.08	35
28	-35	2.85	20

Case Study 3: Prediction of Hydrocracker Total Gasoline Yield

Training Data Set

Input Parameters			Output Parameter
API	K	SCFB H2	Total gasoline yield (LV%)
6.4	10.40	3700	40.10
18.0	10.70	3650	93.60
18.0	10.70	3380	112.40
18.0	10.70	3350	104.10
22.2	11.22	2950	113.60
18.5	10.60	2715	110.00
20.4	10.94	2690	108.90
29.7	11.20	2570	115.60
17.0	11.50	2543	116.40
28.9	11.35	2500	107.40
21.9	10.70	2460	101.60
21.2	11.77	2437	110.30
20.3	11.37	2430	112.40
22.5	11.40	2425	36.40
25.8	11.30	2400	110.80
23.4	10.83	2400	12.40
23.3	11.00	2380	93.20
25.8	11.23	2320	63.40
22.3	11.75	2310	105.30
21.9	10.70	2267	104.10
23.2	10.80	2240	101.00
22.8	11.75	2180	100.70
24.2	11.00	2160	104.40
22.3	11.75	2150	110.80

29.0	11.33	2150	108.90
22.3	11.85	2150	110.80
24.1	11.10	2143	91.90
24.0	11.00	2130	92.20
20.3	11.37	2110	50.90
25.4	11.30	2100	109.10
29.6	11.50	2100	96.30
29.8	11.35	2100	93.20
21.4	11.30	2100	52.00
21.4	10.97	2090	87.70
22.6	11.45	2090	109.60
27.8	11.30	2050	104.30
24.1	11.10	2036	102.10
18.8	11.40	2020	23.40
19.2	11.45	2020	39.70
27.6	11.30	1950	104.90
20.3	11.37	1950	52.40
24.0	11.00	1950	103.20
22.3	11.74	1930	37.20
19.7	11.80	1920	36.30
25.8	12.08	1900	107.60
29.6	11.50	1900	99.70
17.7	10.45	1900	69.00
25.8	12.10	1900	107.60
23.3	12.70	1900	67.90
32.8	11.67	1876	96.90
21.2	11.77	1833	51.40
30.1	11.66	1820	99.80
27.4	11.68	1815	108.40
27.1	11.25	1800	71.00
27.1	11.25	1800	71.00
32.2	11.80	1800	107.90
23.3	12.70	1800	60.50
19.7	11.80	1780	23.40
21.2	11.77	1769	42.10
27.3	11.85	1760	59.10
20.3	11.37	1750	64.10
29.7	11.28	1730	102.20
21.2	11.77	1705	26.00
30.1	11.66	1700	103.10
22.4	11.93	1675	16.70
22.3	11.75	1660	45.20
18.8	11.40	1648	23.30
29.2	11.25	1640	85.50
31.8	12.16	1640	106.00
22.3	11.85	1630	46.10

22.3	11.75	1600	39.50
23.3	12.70	1600	23.10
27.5	11.45	1581	105.90
25.8	11.65	1580	32.00
22.4	11.93	1550	10.50
21.2	11.77	1541	44.10
27.3	11.85	1530	23.40
32.8	11.69	1526	101.20
37.0	11.86	1500	93.90
20.0	11.30	1500	17.80
28.7	11.75	1500	69.00
22.3	11.75	1480	31.10
22.3	11.75	1450	23.00
32.2	11.80	1410	55.90
22.3	11.74	1400	27.30
27.1	11.25	1400	50.00
29.7	11.28	1390	78.80
21.8	11.85	1350	13.30
22.3	11.85	1350	25.40
29.5	11.12	1340	100.50
28.6	11.30	1840	100.10
19.2	10.85	2380	107.30
22.2	11.20	1760	55.00

Additional Training Data (Missing values substituted with average of variables)

Input Parameters			Output Parameter
API	K	SCFB/B2	Total gasoline yield (V %)
17.1	11.50	2705	111.10
21.4	11.50	2410	107.90
22.8	11.50	2410	111.40
22.8	11.50	2370	111.10
19.2	11.50	2305	110.50
22.8	11.50	2250	103.30
22.8	11.50	2225	109.50
22.8	11.50	2165	110.40
23.0	11.50	2160	114.90
5.8	11.50	2105	23.12
26.8	11.50	2090	106.70
28.2	11.50	2070	107.00
24.2	11.50	2060	104.40
19.7	11.50	2000	38.20
25.5	11.50	1790	105.30
21.1	11.50	1650	26.90
27.6	11.50	1550	108.00
5.8	11.50	1530	13.81

29.2	11.50	1520	104.20
19.3	11.50	1500	12.70
20.8	11.50	1490	27.20
28.6	11.50	1360	36.70
5.8	11.50	1335	10.36
27.6	11.50	1310	60.50
28.1	11.50	1300	8.80
29.2	11.50	1990	26.80
29.7	11.44	1990	104.30
8.4	10.70	1990	40.60
19.2	11.50	1990	107.30
30.1	11.42	1990	104.40
28.6	12.33	1990	22.30
31.1	11.47	1990	105.70
30.7	11.42	1990	105.80
29.7	11.44	1990	103.00
34.8	11.62	1990	98.00
21.7	11.85	1990	2.40
18.6	10.50	1990	109.90
25.5	11.50	2170	113.00
23.0	11.50	2260	104.00
23.3	11.50	2250	102.00
23.0	11.50	220	100.00
22.8	11.50	2170	110.40
25.5	11.50	1790	105.30
26.8	11.50	1565	104.70
33.2	11.50	2020	108.00
29.3	11.50	1600	60.90
29.3	11.50	1600	60.10
17.7	11.50	2050	33.75
20.4	11.50	1100	12.70
22.8	11.50	1050	7.50
22.7	11.50	1860	23.90
22.7	11.50	1550	17.30
8.4	11.50	2500	40.60
25.8	11.50	2050	104.60
23.2	11.50	1950	108.20
38.8	11.50	1250	96.40
26.9	11.50	1900	80.20
26.9	11.50	1950	74.80
26.9	11.50	1760	49.88
23.4	11.50	2250	107.10
23.4	11.50	2300	99.80
23.4	11.50	1500	44.90
23.4	11.50	1550	49.50
19.5	11.50	2008	112.10

19.5	11.50	2847	103.50
20.3	11.50	1954	40.80
20.3	11.50	1989	48.20
20.3	11.50	1774	35.40
20.3	11.50	1813	46.60

Validation Data

Input Parameters			Output Parameter
API	K	SCFB/BBL	Total gasoline yield (LV%)
22.5	11.40	2630	49.5
25.8	11.23	2400	110.8
25.9	10.96	2240	94.7
27.9	11.15	2130	101.2
29.3	11.95	2080	51.8
23.3	12.70	1950	106.9
22.4	11.93	1890	30.2
29.6	11.50	1800	100.5
32.8	11.69	1701	99.4
29.4	11.69	1630	102.1

Testing Data

Input Parameters			Output Parameter
API	K	SCFB/BBL	Total gasoline yield (LV%)
19.0	11.08	3200	115.7
21.2	11.77	2590	108.7
19.0	11.10	2440	48.9
20.0	11.30	2400	109.5
23.3	11.00	2310	105.3
26.5	11.44	2213	109.9
27.8	11.28	2150	108.0
22.6	11.85	2120	112.3
27.1	11.25	2100	113.0
27.6	11.85	2070	113.3
22.3	11.75	2000	110.60
22.3	11.85	1930	37.20
26.4	11.59	1900	43.80
17.0	11.50	1876	36.40
32.4	11.65	1800	102.00
20.0	11.30	1800	20.30
32.2	11.80	1750	31.60
24.3	11.83	1700	50.30
21.1	11.80	1640	48.60

29.2	11.25	1620	87.00
29.2	11.25	1570	86.50
29.7	11.28	1510	88.80
22.3	11.74	1450	23.00
22.3	11.75	1400	27.30
29.8	11.35	1300	68.00

APPENDIX IV: MULTIVARIATE ANALYSIS RESULTS

Input Element	Correlated With	r^2 value
API gravity	Pour point	0.0049
	Sulfur Content	0.2089
	BS&W	0.0492
	Salt content	0.0104
	RVP	0.0578
	TAN	0.1056
	Viscosity at 100°F	0.4273
	Viscosity at 122°F	0.4500
Pour point	Sulfur Content	0.1330
	BS&W	0.0355
	Salt content	0.0001
	RVP	0.0827
	TAN	0.0136
	Viscosity at 100°F	0.0272
	Viscosity at 122°F	0.0467
Sulfur Content	BS&W	0.0002
	Salt content	0.0074
	RVP	0.2117
	TAN	0.0010
	Viscosity at 100°F	0.2792
	Viscosity at 122°F	0.3921
BS&W	Salt content	0.3881
	RVP	0.0210
	TAN	0.0325
	Viscosity at 100°F	0.0015
	Viscosity at 122°F	0.0018
Salt Content	RVP	0.0841
	TAN	0.3404
	Viscosity at 100°F	0.2181
	Viscosity at 122°F	0.2457
RVP	TAN	0.0124
	Viscosity at 100°F	0.0484
	Viscosity at 122°F	0.1489
TAN	Viscosity at 100°F	0.0509
	Viscosity at 122°F	0.1144
Viscosity at 100°F	Viscosity at 122°F	0.8617

APPENDIX V: SAMPLE MATLAB CODING

Coding for no early stopping

```
% Preprocess data so that all values falls between –1 to 1
[pn, minp, maxp, tn, mint, maxt] = premnmx (p,t);

% Create network
net=newff(minmax(pn), [5  5  7],{'tansig', 'tansig', 'purelin'},'trainlm');

% Initialise weights and biases to zero
net=init(net);

% Set maximum number of epochs and error goal
net.trainParam.show=100;
net.trainParam.epochs=1000;
net.trainParam.goal=1e-5;
net.trainParam.mu=1;


% Train the network
[net,tr]=train(net,pn, tn);

% Plot error as a function of epoch
plot(tr.epoch,tr.perf,tr.epoch,tr.vperf,tr.epoch,tr.tperf)
legend('Training','Validation','Test',-1);
ylabel('Squared Error');xlabel('Epoch')


% Simulate the trained network using training data set
an = sim(net,pn);
a = postmnmx (an, mint, maxt);
MSE(tn-an)
MSE(t-a)

% Simulate network using testing data set
testPn = tramnmx(testP, minp, maxp);
testTn = tramnmx(testT,mint, maxt);
a_testn = sim(net, testPn);
a_test = postmnmx(a_testn, mint, maxt);
MSE(testTn-a_testn)
MSE(testT-a_test)
```

Coding for early stopping

```
% Preprocess data so that all values falls between –1 to 1
[pn, minp, maxp, tn, mint, maxt] = premnmx (p,t);
valPn = tramnmx(valP, minp, maxp);
valTn = tramnmx(valT,mint, maxt);
val.P = valPn;
val.T = valTn;
testPn = tramnmx(testP, minp, maxp);
testTn = tramnmx(testT,mint, maxt);
test.P = testPn;
test.T = testTn;

% Create network
net=newff([minmax(pn)],[5 5 7],{'tansig',  'tansig', 'purelin'},'trainlm');

% Initialise weights and biases to zero
net=init(net);

% Set maximum epoch and error goal
net.trainParam.show=100;
net.trainParam.epochs=1000;
net.trainParam.goal=1e-5;
net.trainParam.mu = 1;
net.trainParam.mu_dec = 0.8;
```

```
net.trainParam.mu_inc = 1.5;
net.trainParam.max_fail = 100;
net.trainParam.min_grad = 0;

% Train the network
[net,tr]=train(net,pn,tn,[ ],[ ],val, test);

% Plot graph of training, validation and testing mean squared error
plot(tr.epoch,tr.perf,tr.epoch,tr.vperf,tr.epoch,tr.tperf)
legend('Training','Validation','Test',-1);
ylabel('Squared Error');xlabel('Epoch')

% Simulate trained network using training data
an=sim(net,pn);
a = postmnmx (an, mint, maxt);
MSE(tn-an)
MSE(t-a)


%Simulate network using validation data
a_valn = sim(net, valPn);
a_val = postmnmx (a_valn, mint, maxt);
MSE(valTn-a_valn)
MSE(valT-a_val)


%Simulate network using testing data
a_testn = sim(net, testPn);
a_test = postmnmx(a_testn, mint, maxt);
MSE(testTn-a_testn)
MSE(testT-a_test)


%Save network output in Excel format
A=[a_test', testT'];
save results.dat A -ascii
```

APPENDIX VI: DETAIL RESULTS

Case Study 1: Actual versus Predicted Yield

Product	Test Data 1				
	Actual	Single Network	Error (Prediction-Actual)	Individual Network	Error (Prediction-Actual)
FG	0.00	0.08	0.08	0.79	0.79
LPG	0.00	0.95	0.95	-0.11	-0.11
LN	1.00	3.54	2.54	4.31	3.31
HN	6.30	8.99	2.69	11.67	5.37
Kerosene	22.70	13.72	-8.98	12.00	-10.70
Diesel	31.00	21.75	-9.25	21.78	-9.22
Reduced Crude	39.00	53.00	14.00	45.35	6.35
Product	Test Data 2				
	Actual	Single Network	Error (Prediction-Actual)	Individual Network	Error (Prediction-Actual)
FG	0.02	0.06	0.04	0.59	0.57
LPG	0.55	1.49	0.94	1.64	1.09
LN	4.32	5.87	1.55	13.16	8.84
HN	18.61	14.98	-3.63	14.62	-3.99
Kerosene	13.40	16.41	3.01	14.87	1.47
Diesel	37.9	24.53	-13.37	27.65	-10.25
Reduced Crude	25.2	37.72	12.52	35.53	10.33
Product	Test Data 3				
	Actual	Single Network	Error (Prediction-Actual)	Individual Network	Error (Prediction-Actual)
FG	0.04	0.13	0.09	0.08	0.04
LPG	2.75	1.84	-0.91	2.37	-0.38
LN	5.00	5.59	0.59	6.18	1.18
HN	10.68	10.25	-0.43	10.74	0.06
Kerosene	12.43	10.05	-2.38	9.62	-2.81
Diesel	19.95	21.19	1.24	22.65	2.70
Reduced Crude	49.15	51.61	2.46	54.03	4.88

Case Study 2: Actual versus Predicted Pour Point

Actual	Predicted	Deviation
25	20.1	-4.9
35	29.3	-5.7
20	13.0	-7.0

Case Study 3: Actual versus Predicted Total Gasoline Yield

Actual	Original Data		Added Data With Missing Values	
	Predicted	Deviation	Predicted	Deviation
115.7	124.0	8.3	96.4	-19.30
108.7	144.9	36.2	81.4	-27.30
48.9	109.1	60.2	95.6	46.70
109.5	114.6	5.1	99.5	-10.00
105.3	87.7	-17.6	95.7	-9.60
109.9	100.2	-9.7	98.5	-11.40
108.0	97.2	-10.8	96.5	-11.50
112.3	111.4	-0.9	104.0	-8.30
113.0	96.9	-16.1	96.5	-16.50
113.3	121.3	8.0	105.0	-8.30
110.6	62.7	-47.9	48.5	-62.10
37.2	39.9	2.7	26.5	-10.70
43.8	95.6	51.8	95.5	51.70
36.4	28.6	-7.8	30.7	-5.70
102.0	99.7	-2.3	100.0	-2.00
20.3	37.7	17.4	42.2	21.90
31.6	102.6	71.0	107.0	75.40
50.3	32.1	-18.2	29.2	-21.10
48.6	30.3	-18.3	25.8	-22.80
87.0	97.3	10.3	79.4	-7.60
86.5	96.6	10.1	82.5	-4.00
88.8	95.5	6.7	88.0	-0.80
23.0	30.2	7.2	28.0	5.00
27.3	30.2	2.9	27.8	0.50
68.0	67.8	-0.2	65.0	-3.00