

“ I hereby declared that I have read through this report entitled “*Short Term Load Forecasting With Feed Forward Neural Network*” and found that it has comply the partial fulfilment for awarding the degree of Bachelor in Electrical Engineering (Industrial Power)”

Signature :

Supervisor’s Name : Mr. Mohd Shahrieel Bin Mohd Aras

Date :

**SHORT TERM LOAD FORECASTING WITH FEED FORWARD NEURAL
NETWORK ALGORITHM**

MUHAMAD FAIRUZ BIN MUHAMAD RUSLAN

**This Report Is Submitted In Partial Fulfilment of Requirements For
The Degree of Bachelor in Electrical Engineering (Power Industry)**

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

May 2010

I declare that this report entitle “*Short Term Load Forecasting With Feed Forward Neural Network*” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature:

Name : MOHD FAIRUZ BIN MUHAMAD RUSLAN

Date : 22 April 2010

*Dedicated to my beloved Mother, my siblings
and all my friends
for their love and sacrifice.*

ACKNOWLEDGEMENT



Alhamdulillah, thanks to ALLAH S.W.T for His blessed at last I finished this project. First of all, I would like to take this opportunity to express my gratitude to my supervisors, Puan Intan Azmira Bt Abdul Razak and Mr. Mohd Shahrieel Bin Mohd Aras for encouragement supports critics and helps. Without their guidance and interest, this project will not be a success.

My sincere appreciation also extends to all my fellow friends for their assistance and motivation at various occasions. Their views and tips are very useful indeed. Last but not least, thank you to all people who in one way or another contribute to the success of this project.

I am also grateful to all my family members.

Thank you.

ABSTRACT

Load forecasting has become one of the major areas of research in electrical engineering in recent years. Several electric companies are now forecasting load power based on conventional method. However, since the relationship between load power and factor influencing load power is nonlinear, it is difficult to identify its nonlinearity by using conventional method. For this final year project, it involves Short Term Load Forecasting (STLF) with feed forward neural network algorithm. Artificial Neural Network (ANN) has been proved as a powerful alternative for STLF that it is not relying on human experience. This project deals with case study and simulation using Neural Network in Matlab software to forecast load in Peninsular Malaysia. The load data is taken for half hourly load because the aim is to get the minimum error about less or equals to 1.5%.

ABSTRAK

Peramalan beban telah menjadi salah satu bidang yang penting dalam penyelidikan didalam bidang kejuruteraan elektrik sejak kebelakangan ini. Seseengah syarikat elektrik pada masa kini meramalkan kuasa beban berdasarkan kaedah lazim. Walaubagaimanapun, hubungan diantara kuasa beban dan faktor yang mempengaruhi kuasa beban adalah tidak linear, ini menyebabkan kesukaran untuk mengenalpasti ketidaklurusan (*nonlinear*) dengan menggunakan kaedah lazim. Bagi projek sarjana muda ini, ia melibatkan peramalan beban dalam tempoh yang singkat (STLF) dengan menggunakan kaedah rangkaian saraf algoritma kehadapan. Jaringan saraf buatan (ANN) telah terbukti merupakan salah satu alternatif yang terbaik dalam peramalan beban tempoh yang singkat (STLF) tanpa mengharapkan pengalaman manusia. Projek ini membincangkan tentang kajian kes dan simulasi jaringan saraf dengan menggunakan perisian MATLAB untuk melakukan peramalan beban di seluruh Semenanjung Malaysia. Data beban yang diambil adalah pada setiap setengah jam, ini bertujuan bagi mendapatkan ralat yang paling minimum iaitu lebih atau kurang daripada 1.5% ralat.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	STUDENT DECLARATION	iii
	DEDICATION	iv
	ACKNOWLEDGEMENT	v
	ABSTRACT	vi-vii
	TABLE OF CONTENTS	viii
	LIST OF TABLES	xii
	LIST OF FIGURES	xiii
	LIST OF APPENDICES	xv
1	INTRODUCTION	1
	1.0 Problem Statement	1
	1.1 Project Objective	2
	1.2 Project Scope	2
	1.3 Report Online	3
2	LITERATURE REVIEW	4
	2.1 Neural Network in Load Forecasting	4
	2.2 Feed Forward Neural Network	5
	2.2.1 Description of the network	5
	2.2.2 Learning	7
	2.2.3 Generalization	8

	2.2.4	Feed Forward Neural Network in Load Forecasting	10
	2.3	Literature Survey (Comparison Journal)	11
	2.3.1	Basic Feed Forward Neural Network Model	11
2	2.3.2	Peak, valley, and total load forecasting	12
	2.3.3	Hourly Forecasting	13
	2.3.4	Unsupervised learning models	14
3		METHODOLOGY	16
	3.1	Overview of Methodology	16
	3.1.1	Research on Feed Forward Neural Network Forecasting Method	18
	3.1.2	Data Collection	18
	3.1.3	Data Analysis	18
	3.1.3.1	Analysis of Data Load from the Graph	19
	3.1.4	Data Simulation	24
	3.1.4.1	Using Neural Network Tool	26
	3.1.4.2	Assemble Training Data	27
	3.1.4.3	Network or Data Created	28
	3.1.4.4	Train Network	29
	3.1.5	Simulate	32
	3.1.6	Output Data (Forecast Result)	33
4		RESULT	36
	4.1	Forecast Results of Power Consumption for Each Day	36
	4.1.1	Forecasting Result of Power Consumption On Monday	36
	4.1.2	Forecasting Result of Power Consumption On Tuesday	39
	4.1.3	Forecasting Result of Power Consumption On Wednesday	40

4.1.4	Forecasting Result of Power Consumption On Thursday	42
4.1.5	Forecasting Result of Power Consumption On Friday	44
4.1.6	Forecasting Result of Power Consumption On Saturday	46
4.1.7	Forecasting Result of Power Consumption On Sunday	48
5	ANALYSIS & DISCUSSION OF RESULT	51
5.1	Analysis of Forecast Result for Each Day	51
5.1.1	Monday Load Analysis	52
5.1.1.1	Mathematical Calculation for Monday Analysis	53
5.1.2	Tuesday Load Analysis	55
5.1.2.1	Mathematical Calculation for Tuesday Analysis	57
5.1.3	Wednesday Load Analysis	58
5.1.3.1	Mathematical Calculation for Wednesday Analysis	60
5.1.4	Thursday Load Analysis	61
5.1.4.1	Mathematical Calculation for Thursday Analysis	63
5.1.5	Friday Load Analysis	64
5.1.5.1	Mathematical Calculation for Friday Analysis	66
5.1.6	Saturday Load Analysis	67
5.1.6.1	Mathematical Calculation for Saturday Analysis	69
5.1.7	Saturday Load Analysis	70
5.1.7.1	Mathematical Calculation for Saturday Analysis	72

5.2	Analysis and Comparison the Feed Forward Neural Network Method with other Method	73
6	SUMMARY AND CONCLUSION	75
6.1	Summary	75
6.2	Conclusion	76
6.2.1	Recommendation	77
	REFERENCES	78
	APPENDICES	79

LIST OF TABLE

TABLE	TITLE	PAGE
3.1	Data Load and Forecast Result	34
4.1.1	Forecasting Result of Power Consumption on Monday	36
4.1.2	Forecasting Result of Power Consumption on Tuesday	39
4.1.3	Forecasting Result of Power Consumption on Wednesday	40
4.1.4	Forecasting Result of Power Consumption on Thursday	42
4.1.5	Forecasting Result of Power Consumption on Friday	44
4.1.6	Forecasting Result of Power Consumption on Saturday	46
4.1.7	Forecasting Result of Power Consumption on Sunday	48
5.1	Monday Load Over Than 1.5% Error	52
5.2	Monday Percentage Error Calculation	54
5.3	Tuesday Load Over Than 1.5% Error	56
5.4	Tuesday Percentage Error Calculation	57
5.5	Wednesday Load Over Than 1.5% Error	59
5.6	Wednesday Percentage Error Calculation	60
5.7	Thursday Load Over Than 1.5% Error	62
5.8	Thursday Percentage Error Calculation	63
5.9	Friday Load Over Than 1.5% Error	65
5.10	Friday Percentage Error Calculation	66
5.11	Saturday Load Over Than 1.5% Error	68
5.12	Saturday Percentage Error Calculation	69
5.13	Sunday Load Over Than 1.5% Error	71
5.14	Sunday Percentage Error Calculation	72
5.15	Average Percentage Error between 3 Method	74

LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	A Three Layer Feed Forward Neural Network	6
3.1	Flow Chart of Overall Methodology	17
3.2	Monday Load Pattern for 5 Week	19
3.3	Tuesday Load Pattern for 5 Week	20
3.4	Wednesday Load Pattern for 5 Week	20
3.5	Thursday Load Pattern for 5 Week	21
3.6	Friday Load Pattern for 5 Week	22
3.7	Saturday Load Pattern for 5 Week	22
3.8	Sunday Load Pattern for 5 Week	23
3.9	Flow Chart of Data Simulation	25
3.10	Neural Network Tool Window	26
3.11	Input and Target data	27
3.12	Import Window	27
3.13	Network/Data Manager	28
3.14	Network or Data Window	28
3.15	Network view (2 layer)	29
3.16	Training Info and Training Parameters	29
3.17	Neural Network Training (nntraintool)	30
3.18	Performance (plot perform)	30
3.19	Regression (plot regression)	31
3.20	Simulate Network	32
3.21	Network Outputs	32
3.22	Forecast Result	33
3.23	Output Data (forecast result)	33
3.24	Comparison between Forecast Load and Actual Load	35
4.1	Monday Load Forecast	38
4.2	Tuesday Load Forecast	40
4.3	Wednesday Load Forecast	42

FIGURE	TITLE	PAGE
4.4	Thursday Load Forecast	44
4.5	Friday Load Forecast	46
4.6	Saturday Load Forecast	48
4.7	Sunday Load Forecast	50
5.1	Comparison Load Pattern on Monday	52
5.2	Comparison Load Pattern on Tuesday	55
5.3	Comparison Load Pattern on Wednesday	58
5.4	Comparison Load Pattern on Thursday	61
5.5	Comparison Load Pattern on Friday	64
5.6	Comparison Load Pattern on Saturday	67
5.7	Comparison Load Pattern on Sunday	70

LIST OF APPENDICES

NO	TITLE	PAGE
A	Feed Forward Neural Network Method	79

CHAPTER 1

INTRODUCTION

This project is generally to achieve the forecasting result a week ahead for the Peninsular Malaysia electricity load based on 4 weeks previous data. This chapter will discuss the objective and the scope of the project.

1.0 Problem Statement

Load forecasting plays an important role in power system planning and operation. Basic operating functions such as unit commitment, economic dispatch, fuel scheduling and unit maintenance, can be performed efficiently with an accurate forecast [1]. Load forecasting is however a difficult task. First, because the load series is complex and exhibits several levels of seasonality. Second, the load at a given hour is dependent not only on the load at the previous day, but also on the load at the same hour on the previous day and previous week, and because there are many important exogenous variables that must be considered [1]. Various statistical forecasting techniques have been applied to short term load forecasting (STLF). Examples of such methods including, Time Series, Similar-day approach, Regression methods and expert systems. In general, these methods are basically linear models and the load pattern is usually a nonlinear function of the exogenous variables [1]. This final year project presents STLF with feed forward neural network algorithm to forecast future half hourly load demand for 24 hours or one week ahead with minimum error.

Generally, the load of an electric utility is composed of very different consumption units. A large part of the electricity is consumed by industrial activities. Another part is course used by private people in forms of heating, lighting, cooking, laundry, etc. The load is often quite steady, and it is possible to estimate its dependency on different production

levels. However, from the point of view of the utility selling electricity, the industrial units usually add uncertainty in the forecasts. The problem is the possibility of unexpected events, like machine breakdowns or strikes, which can cause large unpredictable disturbances in the load level. In the case of private people, the factors determining the load are much more difficult to define. Each person behaves in his own individual way, and human psychology is involved in each consumption decision. Many social and behavioural factors can be found. For example, big events, holidays, even TV-programs, affect the load. The weather is the most important individual factor, the reason largely being the electric heating of houses, which becomes more intensive as the temperature drops.[5],[7],[11]

1.1 Project Objective

The main objectives for this project are:-

1. To design short term load forecasting with feed forward neural network algorithm using Matlab software with minimum error
2. To analyze the data load given from utility company and the result obtained

1.2 Project Scope

Scope and limitation for this project are :-

1. Analyze load pattern in Peninsular Malaysia for each period
2. Used Feed Forward Neural Network algorithm in Matlab software as a forecasting method.
3. Forecast future load demand for 24 hour ahead or 1 week with minimum error below 1.5%

1.3 Report Outline

This report was divided into 5 chapters where is consisting:

Chapter 1 : Introduction

Chapter 2 : Literature Review

Chapter 3 : Methodology

Chapter 4 : Result and Discussion

Chapter 5 : Conclusion and recommendation

Chapter 1 is about introduction of the project that has been conducted, problem statements, objective and scope of the project. The project was done according to the objective and scope of this project that state at earlier.

Chapter 2 was about the overview of this project based on literature review and theory background.

Chapter 3 shows the flow chart of methodology of the entire project that will conduct for this testing and with some brief explanation about the flow chart.

Chapter 4 was the listing of result and the discussion with the analysis from the tested sample of the oil.

Chapter 5 conclude all the experimental work and testing that conducted in this project. Some suggestion was given to help another student for their project reference

CHAPTER 2

LITERATURE REVIEW

Neural networks, or artificial neural networks (ANN) as they are often called, refer to a class of models inspired by biological nervous systems. The models are composed of many computing elements, usually denoted neurons, working in parallel. The elements are connected by synaptic weights, which are allowed to adapt through a learning process.

2.1 Neural Network in Load Forecasting

Neural networks can be interpreted as adaptive machines, which can store knowledge through the learning process. The research in the field has a history of many decades, but after a diminishing interest in the 1970's, a massive growth started in the early 1980's. Today, neural networks have applications, for example, in pattern recognition, identification, speech recognition, vision, classification, and control systems.[1],[7],[10]

There are many types of neural network models, the common feature in them being the connection of the ideas to biological systems. The models can be categorized in many ways. One possibility is to classify them on the basis of the learning principle. A neural network uses either supervised or unsupervised learning. In supervised learning, the network is provided with example cases and desired responses. The network weights are then adapted in order to minimize the difference between network outputs and desired outputs. In unsupervised learning the network is given only input signals, and the network weights change through a predefined mechanism, which usually groups the data into clusters of similar data. The most common network type using supervised learning is a feed-forward (signal transfer) network. The network is given an input signal, which is transferred forward through the network. Eventually, an output signal is produced. The network can be understood as a mapping from the input space to the output space, and this

mapping is defined by the free parameters of the model, which are the synaptic weights connecting the neurons.[3],[6]

The most popular of all neural networks is Feed Forward Neural Network (FFNN) type. The ANN models are researched in connection with many power system applications, short-term forecasting being one of the most typical areas. The attraction of FFNN has been explained by the ability of the network to learn complex relationships between input and output patterns, which would be difficult to model with conventional algorithmic methods. In the models, inputs to the network are generally present and past load values and outputs are future load values. The network is trained using actual load data from the past. In addition to FFNN forecasters, models based on unsupervised learning have also been suggested for load forecasting. The purpose of these models can be the classification of the days into different day types, or choosing the most appropriate days in the history to be used as the basis for the actual load forecasting[2],[3],[5].

2.2 Feed Forward Neural Network

2.2.1 Description of the Network

Feed Forward Neural Network (FFNN) is the most popular neural network type and most of the reported neural network short-term loads forecasting models are based on it. The basic unit (neuron) of the network is a perceptron. This is a computation unit, which produces its output by taking a linear combination of the input signals and by transforming this by a function called activity function. The output of the perceptron as a function of the input signals can thus be written as equation (2.1).

$$y = \sigma(\sum_i^n w_i x_i - \theta) \quad (2.1)$$

Where

y is the output

x_i are the input signals

w_i are the neuron weights

θ is the bias term (another neuron weight)

σ is the activity function

Possible forms of the activity function are linear function, step function, logistic function and hyperbolic tangent function. The Feed Forward Neural Network consists of several layers of neurons. Each neuron in a certain layer is connected to each neuron of the next layer. There are no feedback connections. A three-layer FFNN is illustrated in Figure 2.1 .

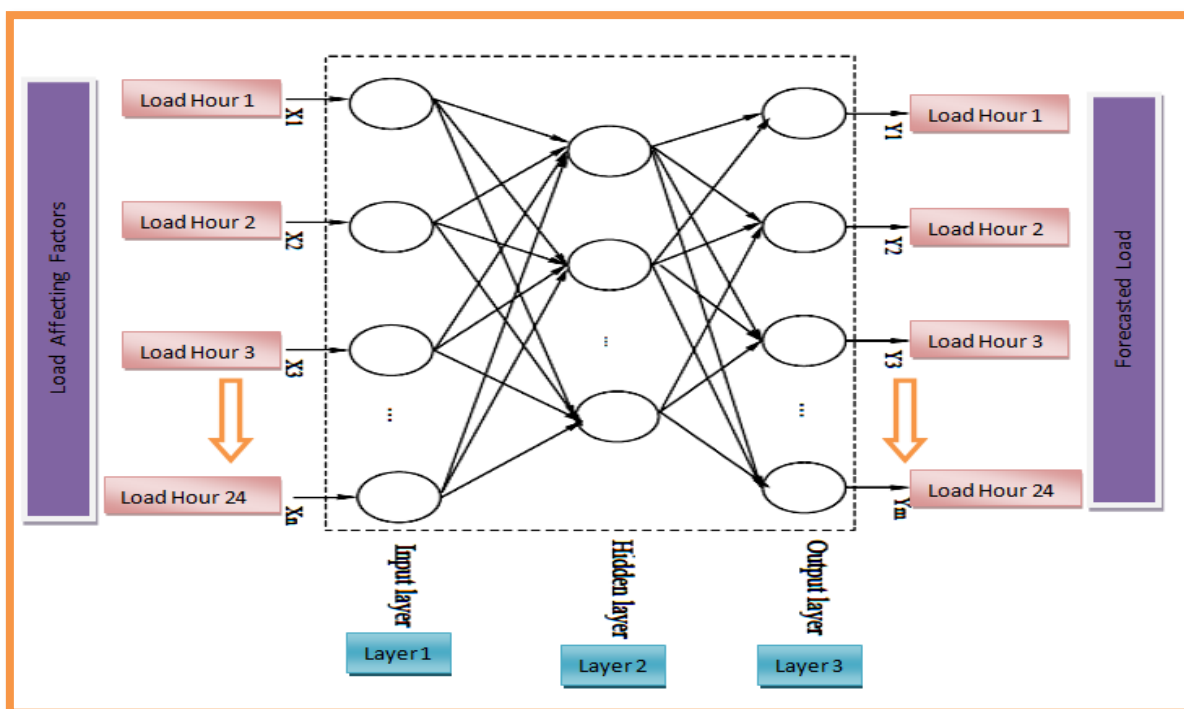


Figure 2.1: The Three Layer Feed Forward Neural Network[10].

As an N-dimensional input vector is fed to the network, an M-dimensional output vector is produced. The network can be understood as a function from the N dimensional input space to the M-dimensional output space. This function can be written in the form as equation (2.2).

$$y = f(x; \mathbf{W}) = \sigma(\mathbf{W}_n \sigma(\mathbf{W}_n \sigma(\dots \sigma(\mathbf{W}_1 \mathbf{x}) \dots))) \quad (2.2)$$

Where

\mathbf{y} is the output vector

\mathbf{x} is the input vector

\mathbf{W}_i is a matrix containing the neuron weights of the hidden layer. The neuron weights are considered as free parameters.

The most often used FFNN consists of three layers: an input layer, one hidden layer, and an output layer. The activation function used in the hidden layer is usually nonlinear (sigmoid or hyperbolic tangent) and the activation function in the output layer can be either nonlinear or linear network.[1],[7],[8],[11]

2.2.2 Learning

The network weights are adjusted by training the network. It is said that the network learns through examples. The idea is to give the network input signals and desired outputs. To each input signal the network produces an output signal, and the learning aims at minimizing the sum of squares of the differences between desired and actual outputs. From here on, this function call the sum of squared errors. The learning is carried out by repeatedly feeding the input-output patterns to the network. One complete presentation of the entire training set is called an epoch. The learning process is usually performed on an epoch-by-epoch basis until the weights stabilize and the sum of squared errors converges to some minimum value.[2][4]

The most often used learning algorithm for the Feed Forward Neural Networks is the back propagation algorithm. This is a specific technique for implementing gradient descent method in the weight space, where the gradient of the sum of squared errors with respect to the weights is approximated by propagating the error signals backwards in the network. The derivation of the algorithm is given, for example, in Haykin (1994). Also some specific methods to accelerate the convergence are explained there. A more powerful algorithm is obtained by using an approximation of Newton's method called Levenberg-Marquardt. In applying the algorithm to the network training, the derivatives of each sum of squared error such as with each training case to each network weight are approximated and collected in a matrix. This matrix represents the Jacobian of the minimized function.

The Levenberg-Marquardt approximation is used in this work to train the FFNN. In essence, the learning of the network is nothing but estimating the model parameters. In the case of the FFNN model, the dependency of the output on the model parameters is however very complicated as opposed to the most commonly used mathematical models for example regression models.[4],[7]

This is the reason why the iterative learning is required on the training set in order to find suitable parameter values. There is no way to be sure of finding the global minimum of the sum of squared error. On the other hand, the complicated nonlinear nature of the input-output dependency makes it possible for a single network to adapt to a much larger scale of different relations than for example regression models. That is why the term learning is used in connection with neural network models of this kind.[8]

2.2.3 Generalization

The training aims at minimizing the errors of the network outputs with regard to the input-output patterns of the training set. The success in this does not, however, prove anything about the performance of the network after the training. More important is the success in generalization. A network is said to generalize well, when the output is correct (or close enough) for an input, which has not been included in the training set. A typical problem with network models is over fitting, also called memorization in the network literature. This means that the network learns the input-output patterns of the training set, but at the same time unintended relations are stored in the synaptic weights. Therefore, even though the network provides correct outputs for the input patterns of the training set, the response can be unexpected for only slightly different input data.

Generalization is influenced by three factors such as the size and efficiency of the training set, the model structure (architecture of the network), and the physical complexity of the problem at hand [4]. The latter of these cannot be controlled, so the means to prevent over fitting are limited to affecting the first two factors. The larger the training set, the less likely the over fitting is. However, the training set should only include input-output patterns that correctly reflect the real process being modelled. Therefore, all invalid and irrelevant data should be excluded.

The effect of the model structure in the generalization can be seen in two ways. First, the selection of the input variables is essential. The input space should be reduced to a reasonable size compared to the size of the training set. If the dimension of the input space is large, then the set of observations can be too sparse for a proper generalization. Therefore, no unnecessary input variables should be included, because the network can learn dependencies on them that do not really exist in the real process. On the other hand, all factors having a clear effect on the output should be included. The larger number of free parameters in the model, the more likely the over fitting is. Then speak of over-parameterization. Each hidden layer neuron brings a certain number of free parameters in the model, so in order to avoid over-parameterization; the number of hidden layer neurons should not be too large. There is a rough rule of thumb for a three-layered Feed Forward Neural Network. [4]

The number of free parameters is roughly $W=H(N+M)$. This should be smaller than the size of the training set, preferably about $T/5$. Thereby, the size of the hidden layer should be approximately:

$$H \approx \frac{T}{5(N+M)} \quad (2.3)$$

H = number of hidden layer neurons

N = size of the input layer

M = size of the output layer

T = size of the training set

In order to be sure of a proper generalization, the network model, like any mathematical model, has to be validated. This is a step in system identification, which should follow the choosing of the model structure and estimating the parameters. The validation of a neural network model can be carried out on the principle of a standard tool in statistics known as cross-validation. This means that a data set, which has not been used in parameter estimation such that training the network, is used for evaluation of the performance of the model.[4],[9]