ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF INFORMATION SCIENCE

# ARTIFICIAL NEURAL NETWORK BASED SHORT-TERM LOAD FORECASTING FOR THE ETHIOPIAN ELECTRIC AND POWER CORPORATION-(EEPCO)

A THESIS SUBMITTED IN PARTIAL FULLFILMENT OF THE REQUIRMENT
FOR
THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION SCIENCE

BY
MEKONNEN TADESSE
JULY 2004

**ADDIS ABABA UNIVERSITY**
**SCHOOL OF GRADUATE STUDIES**
**FACULTY OF INFORMATICS**
**DEPARTMENT OF INFORMATION SCIENCE**

# ARTIFICIAL NEURAL NETWORK BASED SHORT-TERM LOAD FORECASTING FOR THE ETHIOPIAN ELECTRIC AND POWER CORPORATION-(EEPCO)

*By*
***Mekonnen Tadesse***

Name and signature of the Examining Board

Ato Getachew Jemaneh, Chairman, Examining Board ____________________

Dr. Kumudha Raimond, Advisor ____________________

Dr. Osei Adjei, External Examiner ____________________

# DEDICATION

*"I have dedicated this work with love in the memory of the remembrance of my wife, Serawit Mekibib, who sadly passed away as a result of cardiac arrest"*

*And*

*"To our beloved daughter, Edelawit Mekonen, who lost her left eye due to eye cancer"*

# *ACKNOWLEDGMENTS*

*At the outset, I would like to thank God for giving me the strength to challenge all the problems I faced during my study periods. Had it not been for his support, it would have been impossible to accomplish my studies successfully.*

*I also would like to express my sincere gratitude and deep appreciation to my advisor, Dr. Kumudha Raimond, for her guidance, encouragement, and support throughout the project. I also have special thanks to her for accepting my request to advise me when I am in bad need of the same. I also want to thanks all my friends who have been supporting me morally and financially, particularly, Ato Tekilu Girma and Ato Fasil Tegegn, who have been very generous and cooperative since I new them. Thanks to both of you.*

*My thanks also extend to EEPCO staffs, Ato Girum Kebede and Ato Tigistu Wolde Mariam, for giving me their time in providing the necessary information and expertise advise throughout this research work.*

*My thanks also go to Ato Alemayehu Bekele, Ato Fasit Debele and Ato Teshome Mergia who helped me in the secretarial service of this research work.*

*Moreover, I would also like to give my special gratitude to my sister, Meseret Birhane and to my mother-in-law W/O Elizabeth Mengesha who have always been with me to share all my sadness.*

## LIST OF TABLES

# LIST OF FIGURES

## ABBREVATIONS

| | |
|---|---|
| *ACF:* | *Auto Correlation Function* |
| *ANN:* | *Artificial Neural Network* |
| *ARIMA:* | *Auto Regressive Integrated Moving Average* |
| *ARMA* | *Auto Regressive Moving Average* |
| *ARMAX:* | *Auto Regressive Moving Average with Exogenous Variables* |
| *AR:* | *Auto Regressive* |
| *EEPCO:* | *Ethiopian Electric and Power Corporation* |
| *GDP:* | *Gross Domestic Product* |
| *ICS:* | *Inter-Connected Systems* |
| *MAD:* | *Mean Absolute Deviation* |
| *MAPE:* | *Mean Absolute Percentage* |
| *MLP:* | *Multi Layer Perceptron* |
| *MSD:* | *Mean Square Deviation* |
| *SCS:* | *Self-Connected Systems* |
| *SPSS:* | *Statistical Packages for Social Science* |
| *STLF:* | *Short Term Load Forecast* |

# TABLE OF CONTENTS

Page



## CHAPTER ONE
## INTRODUCTION

## CHAPTER TWO

## ARTIFICIAL NEURAL NETWORKS



## CHAPTER THREE

## LOAD FORECASTING

## CHAPTER FOUR

## EXPERIMENTATION

## CHAPTER FIVE

## CONCLUSIONS AND RECOMMENDATIONS

## ***Abstract***

*Load forecasting has become, in recent years, one of the major areas of research. Most traditional forecasting and artificial intelligence researches have tried out this task. Artificial neural networks (ANNs) have lately received much attention, and successful experiments and practical tests have been reported. This work studies the applicability of this kind of model in load forecasting.*

*The multi-layered feed-forward neural network, that are capable of representing non-linear functional mappings between inputs and outputs was used to model the short-term load forecast for the Ethiopian electric and power corporation (EEPCO). The network was trained with the error back-propagation method. Two models were studied in this whole process. The first one is forecasting the load one hour ahead and secondly the daily peak load forecast.*

*The test results, based on historical demand, indicates that this methodology is capable of providing accurate forecasts with 1.1 % and 1.3 % average absolute forecast errors for the hourly and daily peak load forecasts respectively.*

## CHAPTER FIVE

## CONCLUSIONS AND RECOMMENDATIONS

# CHAPTER ONE

# INTRODUCTION

## 1.1. Background

Over the last decades, one has witnessed not only the growth of power transmission and distribution networks but also its increased complexity. In relation to these circumstances, many factors have become influential to the electric power generation and consumption. As a result, intelligent power management has become one of the major research fields in the electrical engineering. This constitutes an important tool for efficient planning and operation of power system and its significance has been intensifying particularly, because of the recent movement towards open energy market and the need to assure high standards on reliability. One of the central functions in power systems planning and operation is load forecasting.

Load forecasting can be classified into four types: 1) long-term forecasting which constitutes a period of 5 to 10 years, or even more, 2) medium-term forecasting (on monthly basis) (3) short-term forecasting (one hour to 24 hours ahead) and 4) very-short-term forecasting (less than one hour forecasting). In recant years the short-term load forecasting has been the main focus of electric power industries and as a result many researchers are dealing with this topic.

At present, the electric supply system is fully governed by the Ethiopian electric power corporation (EEPCO) which is a state owned organization responsible for the generation and distribution of electric power throughout the country. In the last three decades, several studies have been carried out to define Ethiopia's energy potential and to develop appropriate investment program for the power sector. A comprehensive master plan study for generation and high-voltage transmission was carried out for the Ethiopian Electric Power Corporation (EEPCO) in 1981 and 1982. [Acres, 1982]. Similar study to address the most effective means of meeting potential load shortages in the near-term, and finally an integrated plan for a short and long-term system development was executed. At present, the Government and EEPCO, have recognized the fact that electricity supply along with transportation, water supply and health service are a key elements of the physical infrastructure needed for successful improvement of the standard of living of the nation. Therefore, the government and EEPCO are about to launch in a short, a medium, and a long-term power sector development program.

In the above-mentioned program, one of the major areas of the study is, to review the previous load forecast and develop realistic energy and peak demand projection at branch substation, grid and national levels.

This research is intended to introduce Ethiopian Electric Power Corporation (EEPCO) the application of artificial neural networks to load forecasting, which will be put in practice for the whole system after analyzing the result of the experiment at the selected sub-station.

### 1.1.1. Profile Of EEPCO

The Ethiopian Electric Light and Power Authority, as it was named before, was established on 11$^{th}$ September 1955. After having undergone restructuring, has been reorganized as the Ethiopian Electric Power Corporation (EEPCO) on 7$^{th}$ July 1997. The responsibilities of EEPCO are generating, transmitting, distributing and selling of electricity nation-wide.

Presently the corporation maintains two different supply systems; namely, the interconnected system (ICS), which is mainly supplied from hydropower plants, and the self-contained system (SCS), which consists of mini-hydropower plants and a number of isolated diesel generating units that are widely spread over the country.

The ICS has a total installed capacity of 767 MW and an average energy capability of about 3250 GWH per year. However, the dependable capacity is reduced to less than 676 MW, which is 88% of the total installed capacity, due to derate and unit's unavailability due to forced outages. This mentioned capacity is provided by eight hydroelectric plants and two thermal plants of the ICS generation.

The self-contained system (SCS) has a total installed generation capacity of about 21.5 MW. The dependable capacity is reduced to about 20.5 MW due to derate. The SCS generation consists of isolated mini-hydro and diesel installations. Since the focus of this research is on the ICS generation, the SCS has been left unexplained further.

EEPCO is undergoing construction of new generations that are located at Dire-Dawa and Tekeze. The former is a 40 MW diesel generator plant with a dependable capacity of 38 MW and it is expected to be functional soon. The second plant is a hydroelectric plant with installed capacity of 300 MW and 962 GWH of energy. This plant is presently under construction and expected to enter the system beginning of the year 2008. Both plants together will increase EEPCO's system capacity to a total of 1022 MW and annual average generation capacity of 4036GWH.

The existing transmission system comprises of 6447 km of transmission lines, of which 1750 km are at 230 KV, 2514 km at 132 KV, 1742 km at 66 KV and 476 km at 45 KV levels. There are a total of 105 substations, of which 11 are at 230 KV, 45 are at 132 KV, 26 are at 66 KV, and 23 stations at 45 KV level. Two 230 KV, eight 132 KV and six 45 KV substations are presently feeding the Addis Ababa region load which is the largest single consumption center in the ICS system and accounts for half of the total sales. This research experiment is also conducted on one of the substations in this center, which cover the major consumption areas of the region.

The substation chosen for the research experiment, which is the Addis center substation, is located at the center of the town and is said to be the largest of all other substations in the town in terms of its area coverage and availability of transmission lines. Moreover, during a visit around all substations the researcher came to know that in this substation all the 24 hours load data are timely recorded.

### 1.1.2. Load Forecasting Techniques In EEPCO

At present, EEPCO is the only public agency that is responsible for the general management of the electricity throughout the country. The current electricity supply is accounted to a very few portion of what will be expected for the whole economic activities of the country. According to the data on the EEPCO's master plan study (Acer, 1995) the pattern of consumption on both power systems (ICS and SCS) show higher growth rate. This growth rate is related to the currently electrified target groups. The study indicates that the market penetration rate in already electrified towns is below 50%. Most of the rural areas are not connected to the system.

In view of the above facts, the primarily objective of the government of Ethiopia and EEPCO is to increase the electricity supply so as to meet the overall demand of the country. Based on this objective EEPCO have maintained a load forecast for the next 10 years and plan the construction of generation plants in the short, medium and long-term programs.

During discussions, with EEPCO experts it was understood that no forecast had been made so far for the short-term period, like one hour or one day ahead. The operational activities are carried out based on the expert's judgments that mostly rely on their past experience of power management. Moreover, the present records available at each substation in Addis Ababa show inconsistencies in registering the meter readings. Few, keep records for the entire 24 hours while others up to 8 pm with an hour discontinuity at mid-day.

However, EEPCO has gone through a considerable effort to maintain a long-term load forecast with the aim of fulfilling its objectives towards increasing the coverage of electricity supply in the country. To this end, EEPCO have carried out successive studies in the last three decades. The last comprehensive master plan study was carried out by Acer International to review and update the previous ones based on the present situations. The study presents the power market forecast for the next 10 years (2004-2014). In the general approach, quantitative analysis has been made on relevant information with additional qualitative judgments appropriate to the forecasting period.

In the forecast process, separate analysis has been made for currently connected and unconnected centers. For the connected centers, the analysis is based on the assessment of the demand within each principal power-consumer sector, and for the unconnected centers, the forecast is developed based on the estimates of population, population growth, the level of interconnections and consumption per connection.

First, a regression analysis has been made to determine the relationship of different independent variables with the electricity sales. The sales and number of customers database developed in the previous study was used for the analysis. The test in the correlation analysis indicates the existence of strong relationship between sales and gross domestic product, price, and/or the number of customers. Based on this result regression model has been built representing the following categories:

Model 1:- relates to branches that are electrified through the base ICS for the entire period and have typically expanded within the bound of each branch.

Model 2:- relates to branches which were added to the ICS in the first major extension to the system occurred in the early 1980s.

Model 3:- addresses the remaining extension to ICS.

(The SCS branches are handled separately)

The above models were prepared for the principal customer classes (Domestic, commercial, and industrial).

a) **Domestic sales**

As mentioned above the regression analysis indicates, typically to this category, a strong correlation of total sales with non-agricultural GDP, domestic sales elasticity and the number of customers. Based on these results, income and price elasticity were estimated for each models and accordingly the forecast model was built.

The form of the general model is:

$$s_t = s_{t-1} * r_c * \left( \frac{GDP_t}{GDP_{t-1}} \right)^a * \left( \frac{T_t}{T_{t-1}} \right)^b \qquad (1.1)$$

Where, $s_t$ are domestic sales in Kwh at time t

$r_c$ is customer growth rate

$GDP$ is non-agricultural component of GDP

$T_t$ is Tariff at time t

$a$ is income elasticity

$b$ is price elasticity

b) **Commercial sales**

With the same logic to the domestic sales, the correlation was also analyzed for this group of customers. The difference was only in the value of the calculated elasticity for each model.

c) **Industrial sales**

Here the analysis indicates that the dominant independent variable is non-agricultural GDP. No further significant tariff change was assumed to occur. Also neither price nor the number of customers yields significant correlations. Hence based on the income elasticity of the group, calculated for each model, a forecast has been made using the formula:

$$s_t = s_{t-1} * \left( \frac{GDP_t}{GDP_{t-1}} \right)^a \tag{1.2}$$

The part of public lighting consumption was assumed to represent fixed portion of total sales over the entire forecast period but the proportion varies for each models. Model 1, Model 2 and Model 3 assumes a proportion of 1%, 1.25% and 1.5% respectively. (Similarly, a forecast has been made for SCS in specific periods).

In all the processes two scenarios were also analyzed with different assumptions. The first is called Target growth scenario and assumes that the GDP growth rate for the forecast period will prevail to the GDP growth rate forecast for the years 2004 to 2006 and ICS include 100% of the new connections attributed to rural electrification. The forecast made with this scenario is called Target forecast.

The second scenario, which is called Moderate growth scenario, assumes that the available GDP growth rate will drop to the recent five years economic performance within 12 years of period. This was shown to be 5.1% for industrial and 5.7% for services. Moreover, the moderate growth scenario assumes that ICS includes 10% of the new connections attributed to rural electrification. The forecast with this scenario was known as Moderate forecast.

Finally, the ten-year forecasts have been made at branch level for the two kinds of forecasts separately.

## 1.2. Objectives

### 1.2.1. General objective:

The general objective of this research is to explore the application of artificial neural network in providing load forecast for a power system for planning, scheduling, or dispatching purpose.

### 1.2.2. Specific objectives:

Based on the general objectives, the research focuses on the following specific objectives:

- Thoroughly analyze different literatures in the area of artificial neural networks for forecasting loads in the power system operation.
- Identify the appropriate artificial neural network architecture to short-term load forecasting.
- Determine the type of data set and collect systematically from the pre-defined sources.
- Prepare the training and test set data for analysis, which includes cleaning and normalization.
- Training and testing the model using the appropriate software.

* Present the result reached at the end of the research and make appropriate conclusion and recommendation.

## 1.3. Statement Of The Problem

The principal task of power system is to deliver the electric power requested by the customers. This task has to be solved in a safe, reliable and economical manner. The electric power system represents complex system involving many electrical components whose operations must be planned, monitored, analyzed and controlled. The amount of information to be processed will continue to increase for a number of reasons. Improving power system efficiency and reliability will require increase control system complexity through the addition of new process components.

Especially on last decade, one has been witnessing not only the growing power transmission and distribution but also its increasing complexity. At the same time, there is a growing demand on environment preservation as well as the need for energy efficiency.

Load forecasting constitutes an important tool for efficient planning and exploration of power systems, and its significance has been intensifying particularly because of movement towards open energy markets and the need to ensure high standards on reliability [ Fidelgo, J,N, Peças Lopes, J.A, 2000]..

The motivation of accurate load forecast is three fold: generation expansion planning, transmission expansion planning, and financial planning. In addition, load forecasting helps the utility to determine the system's spinning reserve and full requirement, and plan their unit maintenance scheduling.

The timely availability of sufficiently reliable electricity supply is vital to the proper functioning of the whole economy. Generally, forced interruption of electric supply will lead to substantial losses in output. Accurate demand estimates provide the decision makers to undertake necessary measures to improve the reliability.

EEPCO, which is responsible for the electricity supply of the country as a whole is currently making a progress towards introducing new generation plants to meet the growing demand for the electricity. At present, the power installed for meeting the demand requirement is nearly in a full use. Episode of power interruption have been seen in the last few years due to the declining of generating power of the hydro thermals. This led EEPCO to dispatch power partially on the basis of different shift periods. During such kind of events the utilization of future load demand will have a paramount importance for efficient planning the distribution.

With such context, load forecasting at small interval of periods is important for higher precision in expressing instant load profiles. Such values are important for the decision-making process regarding greater power availability as well as ensuring a more effective load-demand management process.

It will be apparent from the above discussion that a reliable load forecast will benefit EEPCO for their operation planning.

To this end, this research is aimed to provide EEPCO accurate load forecasting technique using artificial neural networks (ANN) approaches. ANN has been replacing the traditional methods for load forecasting for their unique power in analyzing non-linear patterns.

## 1.4. Scope and Limitations of the research work

The focus of the research work is on providing accurate short-term load forecast technique for EEPCO. Two types of models are used with varying forecast period. The first model is intended to give a one-hour-ahead forecast, while the second model is restricted to the daily peak (maximum) load of the forecast day. Both models employ the ANN based load forecasting technique. The whole research work is performed on one of the sub-stations at Addis Ababa.

The inadequate literatures and books available in the department have hindered the work in exploring more about the research area. Moreover, the time allotted to the work was not sufficient enough to cover all the necessary ideas as an input to the research work and as a result some relevant parts are restricted to a few discussions.

It was previously agreed that the necessary amount of fund for the research would be released on time as per the requested budget. However, very limited amount of budget was given at the last period of the research work and this forced much of the work to be remained unfinished at the final stage.

## 1.5. Structure Of The Work

The following chapter concentrates on the concept of artificial neural networks. In the first part the fundamentals of neural networks (NN) is presented. Then, the most popular network type, the multilayer perceptron (MLP) is discussed in some details. The subject of load forecasting is discussed in the third chapter. It deals with the factors that affect load as it was described by different researchers and compare the major conventional forecasting techniques with the recent methods of ANN. Chapter four is devoted to the actual work of the research. First, it gives a general overview on the data collection and analysis, which is followed by the one hour ahead and daily peak load forecast models are studied. The research work ends by making a conclusion and some recommendations for further researches in the last chapter.

# CHAPTER TWO

# ARTIFICIAL NEURAL NETWORKS

## 2.1. Fundamentals

Artificial neural networks, which are commonly called as neural networks, are systems that are loosely modeled on the human brain. It can also be referred to as an adaptive statistical model based on an analogy with the structure of the brain. Its adaptive behavior comes from the way it can learn to estimate the parameters of some population using a small number of examples at a time.

Neural networks can be considered as a recent development but enjoys a resurgence of interest by many researchers in the field of computer engineering and artificial intelligence.

The interest in neural networks comes from its potential to find a solution to various problems of application domains. It has been used in large number of applications and has proven to be effective in performing complex functions in a variety of fields. These include pattern recognition, classification, vision, control system, and prediction [Fauset, L., 1994].

Various literatures have given different definitions to neural networks, the researcher has decided to provide the definition by [Hykin, s., 1999] for the simple reason that it is being comprehensive.

***"A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural property for storing experimental knowledge and making it available for use. It resembles the human brain in two respects:***

- ***Knowledge is acquired by the network from its environment through a learning process.***
- ***Interneuron connections strength known as synaptic weights, are used to store the acquired knowledge."***

## 2.1.1. Basic Structure Of A Neuron

### 2.1.1.1. The human brain

The human brain is estimated to contain about $10^{11}$ interconnected neurons. Each neuron is nearly connected to $10^4$ other neurons. The neural model shown in fig. 2.1 comprises the basic structure of the human neuron. The tree like structure is called the ***dendrites***. The neurons collect signals from other neurons through these dendrites. The neuron sends out spikes of electrical activity through a long, thin stand known as ***axon***, which splits into thousands of branches. At the end of each branch, a structure called a ***synapse*** converts the activity from the axon into electrical effects that inhibits or excites activity in the connected neuron. When a neuron receives excitory input that is sufficiently large compared with its inhibitory input, it ends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

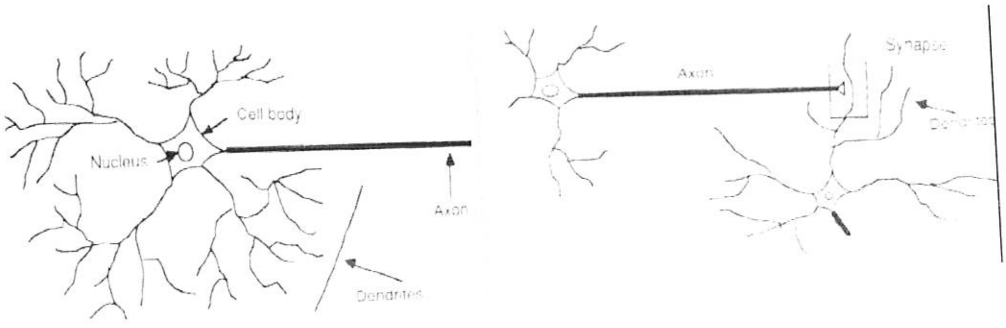



***Fig 2.1 Components of a biological neuron*** ***The synapse***

### 2.1.1.2. <u>Artificial neuron</u>

The biological neuron model is the foundation of an artificial neuron. It simulates the four basic components of the natural neurons: dendrites, soma (cell body), axon, and synapses.

In the ANN literatures, the unit analogous to the biological neuron is referred to as a processing element or unit. (Fig. 2.2 shows the general structure of processing element). A processing element has many input path and combines, usually by a simple summation, the values of these input paths.

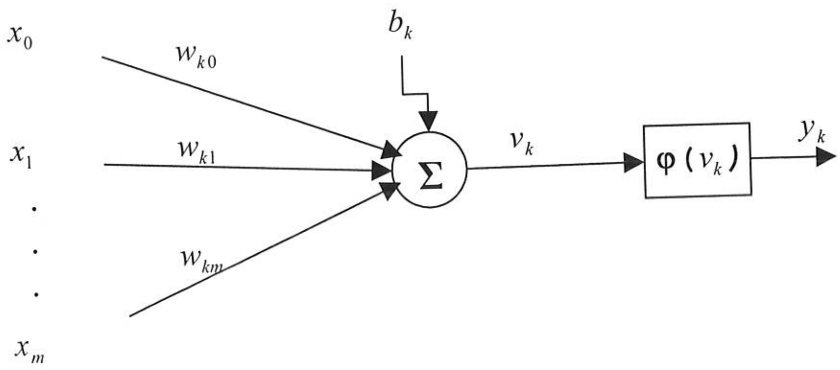



***Fig. 2.2 A processing element (unit)***

The result is an internal activity level $v$ for the process element. The combined input $v$ is then modified by a transfer function (also referred to as activation or gain function). This activation function can be a threshold function, which only passes information if the combined activity level reaches a certain level, or it can be a continuous function of the combined output. (Both types of functions will be discussed in the next section). The output value of the activation function is generally passed directly to the output path of the processing element. The model in fig. 2.2 also includes an externally applied bias, denoted by $b_k$. The bias $b_k$ has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively [Hykin, s., 1999]. The bias $b_k$ is an external parameter of artificial neuron k. The mathematical formulation will be then described as follows:

$v_k = \sum w_{kj} x_j$ *where,* $x_0 = +1$ , $w_{k0} = b_k$

$y_k = \varphi\,(v_k)$ *where,*

$v_{k,}$ *is the internal activity level of neuron k*

$x_j$ , *(j=1, 2, ...m) are the input signals*

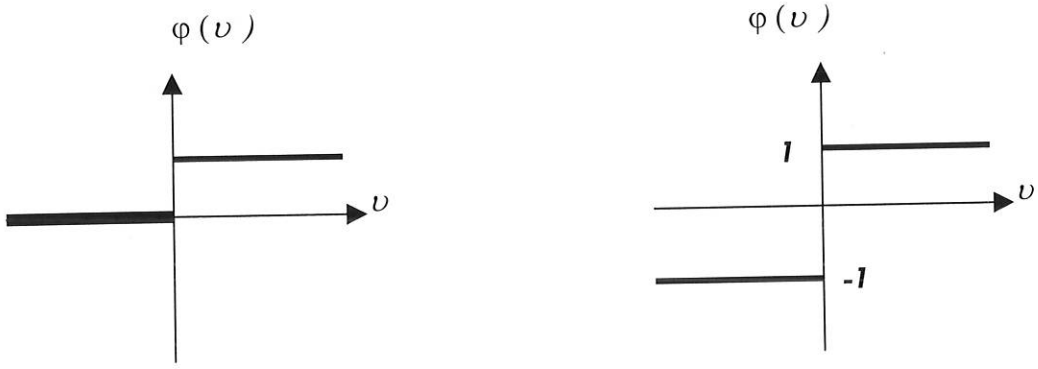
$w_{kj}$ , *(j=1, 2, ...m) are the synaptic weight of neuron k,*

$\varphi\,(v_k)$ *,activation function,*

$y_k$ *the output signals of the neuron k.*

### 2.1.1.3. <u>Types of activation functions</u>

The activation function $\varphi$ (.) performs a mathematical operation on the signal output. The most commonly used activation functions are described in the fig. 2.3 (a)-(e).




***Binary*** ***Bipolar***

***Fig. 2.3 (a) Threshold functions***

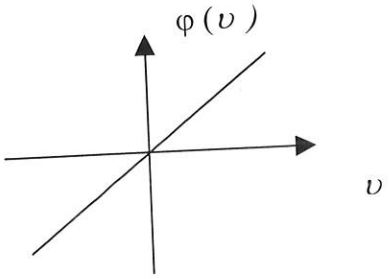



**Fig. 2.3 (b) linear function**

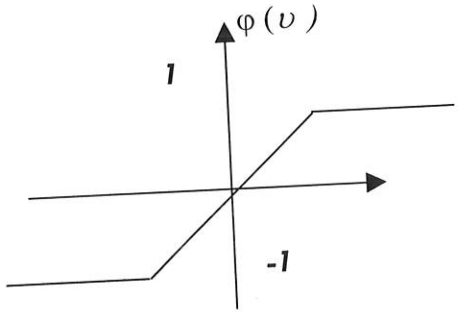



**Fig. 2.3 (c) Piecewise function**

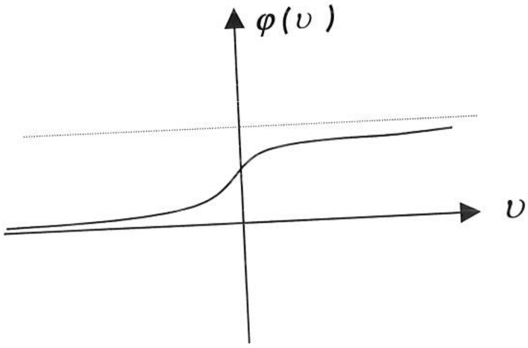



**Fig. 2.3 (d) sigmoid function**

### a) *Threshold function*

A threshold function, also known as Heavside function, is either binary type or a bipolar type. The output of the function is described as follows:

$$\varphi(\nu) = \begin{cases} 0 & if \quad \upsilon \leq 0 \\ 1 & if \quad \upsilon > 0 \end{cases} \quad \Rightarrow \text{binary type} \tag{2.1}$$

$$\varphi(\upsilon) = \begin{cases} +1 & if \quad \upsilon \geq 0 \\ -1 & if \quad \upsilon < 0 \end{cases} \quad \Rightarrow \text{bipolar type} \tag{2.2}$$

#### b) *Linear function*

$\varphi(\upsilon) = \alpha\upsilon$, $\alpha$ is the slope of the linear function when $\alpha = 1$ it is called an identity function which implies the output is equal to the input value.

#### c) *Piecewise function*

This type of activation function is also referred to as saturating linear function and can have either a binary or bipolar range for the saturation limits of output.

$$\varphi(\upsilon) = \begin{cases} -1 & if \quad \upsilon < -1 \\ \upsilon & if \quad -1 \geq \upsilon \geq \\ 1 & if \quad \upsilon \geq 1 \end{cases} \tag{2.3}$$

#### d) *Sigmoid (S shaped) function*

This nonlinear function is the most common type of activation used to construct the ANN. The sigmoid function is a strictly increasing continuous function, which can be differentiable. This property makes it appropriate for use in neural networks trained by back-propagation. A sigmoid function can be expressed in the form:

$$\varphi(\nu) = \frac{1}{1 + \exp^{-\phi\nu}} \qquad (2.4)$$

where, $0 \le \varphi(\upsilon) \le 1$

Where, $\alpha$ is the slope of the parameter of the sigmoid function. The output of the function varies between 0 and 1. Mathematically, this type of function is called the logistic function

Some more types of activation functions are also used in different areas of application of ANN. The choice of the activation function mainly depends on the distribution of the target value, which is generated by the output units. For continuous valued targets with a bound range, the sigmoid functions are preferable, provided that either the output or the targets to be scaled to the range of the output activation function [Sarls, w.s., 1977].

### 2.1.2. Network Architecture

The principal importance of a neural networks lies on their interconnections. Artificial neural networks have simple structures and are designed to mimic the function of biological neurons. The natural connection of a human brain is too complex to be implemented directly in neural networks. The structures, therefore studied so far are simple and easy to be implemented in digital computers.

Architecture of a neural network is the specific arrangement and connections of the neurons that are organized in the form of layers, make up the network. It is defined by the number of layers, the number of units per layers, and the interconnection pattern between layers. In general, ANN has a similar structure of topology. Basically, there are three types of layers each consisting of a group of neurons (units).

- The input layer consists of neurons that receive input from the external environment.
- The output layer consists of neurons that communicate the output of the system to the user or external environment.
- One or more hidden layers between the output and input layers.

When the input layer receives the input, its neurons produce output; this becomes input to the next layer of the system. The manner in which the neural networks are structured is internally linked with the learning algorithm used to train the network [Hykin, S., 1999]. In general, three classes of network architectures are in practice.

1. ***Single-layer feed forward networks***

A feed forward network is a network where the neurons on the first layer send their output to the neurons on the second layer, but they do not receive an input back from the neurons on the second layer.

In other words the network is strictly feed forward. A network with the input and output layers only is called single-layered network. A typical single-layered feed forward is shown in fig 2.4

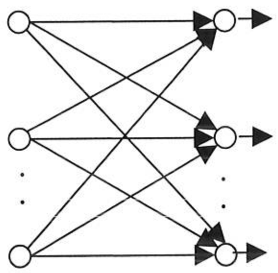



***Fig. 2.4 Feedforeward network with a single layer of neurons***

**2. *Multilayer feed forward networks***

The multilayer feed forward networks or the multilayer perceptions, as the name usually used, are the generalization of the single-layer perceptron. It includes one or more hidden layers each with their respective number of neurons. The function of the hidden neurons is to intervene between the external input and the network output in some useful manner. Fig 2.5 shows a typical multilayer perceptron neural networks structure.

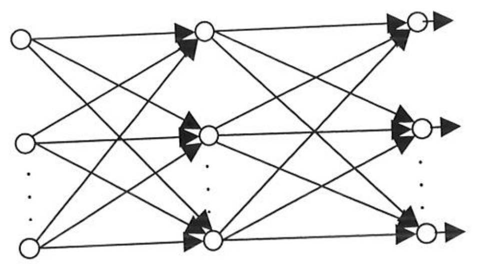



*Fig. 2.5 A fully connected feedforward network with one hidden layer and one output layer*

The next section (2.2) will briefly describe this network since it is the scope of the research work.

### 3. *Recurrent networks*

A recurrent network, which is sometimes referred to as feedback networks can have a signal traveling in both directions by introducing, loops in the network. There may exist one or more such loops. In this network a neuron of a layer after receiving input from another layer, send its output back to the previous layer neurons. Recurrent networks are said to be dynamic in that their state is changing continuously until they reach an equilibrium point. In some cases the activation values of the unit's undergo a relaxation process such that the network will evolve to a stable state in which activation does not change further. In other applications in which the dynamic behavior constitutes the output of the network, the changes of the activation values of the output units are significant. Graphical illustration of an example of recurrent neural network is presented in fig 2.6

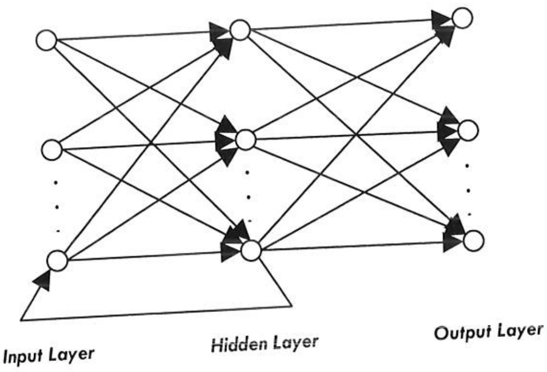



**Fig. 2.6 Recurrent neural network**

### 2.1.3. The Learning Process

The connections between neurons in a neural network are given adjustable weights or measure of importance. The process of adjusting the weights to make the network learn the relationship between the input and target is called learning. To be more precise a definition of learning that is adapted from Mendal and McClaren (1970) is given by [Hykin, S., 1999] as follows:

> ***"learning is a process by which the free parameters of a neural network are adopted through a process of stimulation by the environment in which the network is***

***embedded. The type of learning is determined by the manner in which the parameter changes take place."***

The process of making the network to learn the solution to a problem follows a set of well-defined rules called the learning algorithm. Many learning algorithms have been inverted to help find an optimum set of weights that result in the solution of the problem. In general, all the learning methods of an adaptive network can be classified into two major categories.

### a) *Supervised learning*

Most commonly, neural networks use supervised training. Here the network is trained by providing it with inputs and desired outputs. These input-output pairs are provided by an external teacher or by the system containing the network. For this reason it is also called learning with a teacher. The actual output of the net may or may not match the desired output, depending on the weights at the particular moment. In all cases the training algorithm modifies the weights with the idea that the next time the net sees the same input pattern, it will more closely (or exactly) reproduce the target output pattern as its actual output.

### b) *Un supervised learning*

In unsupervised learning, the network is not trained towards specific outputs. Instead, the net seeks to find patterns or regularity in the input data. It is self-organizing learning process that does not require external teacher. Once the network has become tuned to the statistical regularities of the input data, it develops the ability to form internal representation for encoding features of the input and thereby to create new classes automatically [Becker, S., 1991].

## 2.2. Multilayered Feed Forward Neural Networks

Multilayer feed forward neural networks or Multilayer perceptron network (MLP) is the most popular neural network type. It has been applied successfully to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as error back-propagation algorithm.

The typical network has an input layer, at least one hidden layer and an output layer. There is no limit on the number of hidden layer but most applications use just one or two.

### 2.2.1. General description

A fully connected MLP with 1 hidden layer is shown below.

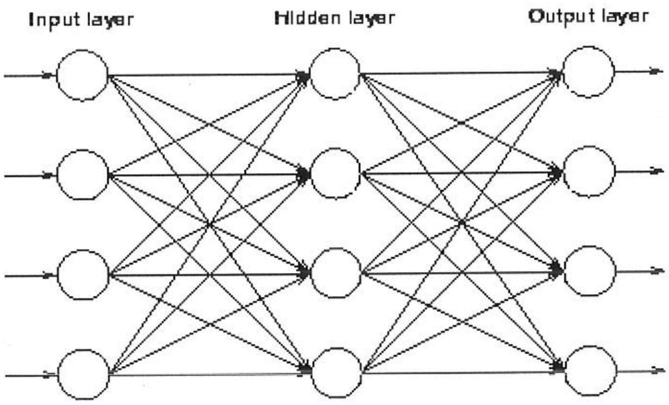



***Fig.2.7 A fully connected multilayer feedforward network with one hidden***

There is no quantifiable, best answer to the layout of the network for any particular application. There are only general rules picked up over time and followed by most researchers and engineers applying this architecture to their problems [Anderson, D. and Mcneil, G., 1992]:

*"Rule 1: As the complexity in the relationship between the input data and the desired output increases, then the number of the processing element in the hidden layer should also increase.*

*Rule 2: If the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required.*

*Rule 3: The amount of training data available sets an upper bound of processing element. To calculate this upper bound use the* *number of input-output pair examples in the training set and divide that number by the total number of input and output processing element in the network. Then divide that result again by scaling factor between five and ten. i.e.,*

*Let* $p$ *be the number of input-output examples,*

$m$ *be the number of input neurons,*

$n$ *be the number of output neurons,*

*Therefore number of neurons,* $H \in$

$$\left[\frac{p}{10(m+n)}, \frac{p}{5(m+n)}\right] \tag{2.5}$$

*Upper bound is* $\frac{p}{5(m+n)} = H$

*However, the best approach to find the optimal number of hidden neuron is trail and error. It is important to notice that using too few units in a completed data set may lead to under fitting and to the contrary, using too many units may cause over fitting, in which case the training set will be memorized and no generalization of the data trends will occur, making the network useless on new data sets."*

### 2.2.2. Back-Propagation

Back-propagation is the most commonly used method for training multilayer feed forward networks. This training scheme is used for adjusting the connection weights of each unit in such a way tat the error between the desired output and the actual output is reduced. More clearly, the neural network adjusts the connection weights to each unit, beginning with the connections between the last hidden layer and the output layer. After the network has made adjustment to this set of connections, it calculates error values for the next previous layer and makes adjustments. This scheme was developed independently by Webros, Parker, Rumehart, Hinton and Williams [Anderson, D., and McNell, G., 1992].

The error back-propagation method can be applied to any feed forward networks with differentiable activation function. The requirement of differentiability of the activation function is based on the following reason. As mentioned before, the difference between the response of an output unit and expected response is the error made by the network. The units of the output layer use this error directly to correct their connection weights, but this is not the same for the units of the hidden layer since they are not in direct contact with the error. As a result the units of hidden layers need to estimate their error using error back-propagation method. The amount of error made by the network is first converted into an error signal that is proportional to the rate of change of the nonlinear activation function. In fact, this implies that we want the correction to be proportional to the rate of change of the activation function. From the theory of calculus, the easiest way to fulfill this requirement is to make the correction to be proportional to the derivatives of the activation function, which justifies the need for differentiability of the activation function.

The following section discusses on back-propagation algorithm for a network where the units within the hidden and output layers are non-linear function called the sigmoid or logistic function. The generalized form can be found in [Mitchell, T.M., 1997].

#### 2.2.2.1. <u>The back-propagation algorithm with the sigmoid function</u>

- **<u>The Non-linear unit</u>**

The sigmoid function used as non-linear transfer (activation function) is described as:

$$\varphi(x) = \frac{1}{1+e^{-x}} \,, \qquad (2.6)$$

where, the function maps the set of all real numbers into [0, 1] range.

In a back-propagation network the units within the hidden layer and output layers are usually non-linear. This unit computes their activation level by adding all the weighted activations they receive and transfer this level into a response using the non-linear activation, in this case, the sigmoid function.

### ✠ <u>Back-propagation algorithm/BP/</u>

The rule used for back propagating the error is a generalization of the delta rule or widrow-Hoff learning rule [Abdi, et al, 1999]. It is simply the gradient descent method to minimize the total squared error of the output computed by the network.

### ✠ The BP algorithm involves two phases:

In the first phase, a forward flow of activation is generated from the input layer to the output layer through the hidden layer. Each unit in the hidden layer computes its activation level as a weighted sum of its inputs. For the $j^{th}$ unit this is expressed as:

$$h_j(n) = \sum_{i=0}^{m} w_{ji} x_i(n) \qquad (2.7)$$

$h_j(n)$ → activation level of jth hidden neuron at nth training example,

$w_{ji}$ → synaptic input weights (i = 0,1 …,nm)

$x_i(n)$ → $i^{th}$ input signals

m → total number of inputs

After calculating the activation level of the hidden neuron the result is transferred into the activation function of the neuron I.e.

$$y_i(n) = \varphi(h_j(n) \tag{2.8}$$

where, $y_j(n)$ → output of $j^{th}$ hidden neuron

$\varphi$ → the activation function.

The response of the hidden neuron $y_j$ (after a certain threshold), then will be the input to the output neuron. Similarly, the response of the output neuron for the $y_j$ input signals from the hidden neuron can be obtained as:

$$\hat{y}_p(n) = \sum_{j=1}^{k} z_{pj} y_j(n) \tag{2.9}$$

After transforming through the activation function the actual output of the network for the p<sup>th</sup> output neuron will be:

$$\hat{t}_p(n) = \varphi(\hat{y}_p(n)) \qquad (2.10)$$

In the second phase, the error term, $e_p$, will be determined by subtracting the actual output of the network from the desired.

i.e. $$e_p(n) = t_p(n) - \hat{t}_p(n) \qquad (2.11)$$

where, $t_p(n)$ is the desired output obtained from the $n^{th}$ training example

This error term is then transformed into an error signal, which takes into account the derivative of the unit activations, in this case the sigmoid function.

From (2.5) $$\varphi'(x) = \varphi(x) * [1 - \varphi(x)] \qquad (2.12)$$

Let $\delta_{output,p}(n)$ be the error signal obtained at the $p^{th}$ output neuron

Then, $$\delta_{output,p}(n) = \varphi'(\hat{y}_p(n)) * e_p(n)$$

From (2.10), (2.11) and (2.12) then,

$$\delta_{output,p}(n) = \hat{t}_p(n)[1 - \hat{t}_p(n)] * [t_p(n) - \hat{t}_p(n)] \qquad (2.13).$$

This error signal is then back propagated through the network layer by layer. The connection weights $w_i$' be adjusted accordingly so as to minimize the mean square error between the network output and the desired output.

The weight adjustment for the p$^{th}$ output neuron will be:

$$Z'_{pj} = Z_{pj} + \Delta Z_{pj} \tag{2.14}$$

$$\Delta Z_{pj} = \eta y_j \delta_{output,p} \tag{2.15}$$

where, $\Delta Z_{pj}$ is the correction weight for the p$^{th}$ output neuron.

$\eta \rightarrow$ a constant called learning rate

$y_j \rightarrow$ jth input signals of hidden neuron

The adjustment of the weights between the input and hidden neurons requires the amount of error term obtained at each hidden neuron. This error is estimated as the weight sum of the error signal of the out put layer unit. (i.e. the error signal of each output unit is weighted by the connection from the hidden unit).

Therefore,

$$\hat{e}_j(n) = \sum_p Z_j \delta_{output,p} \tag{2.16}$$

is the estimated error term of the $j^{th}$ hidden neuron and $z_i$ are the input signal weights of the $j^{th}$ neuron and $\delta_{output, p}$ is the error signal obtained at the $p^{th}$ output neuron.

Once the error terms has been estimated, the correction of the weight from input layer neuron to the hidden layer neuron is the same as for the out put layer neuron with the estimation of the error term replacing the actual error term.

i.e. $$\Delta w_{ji} = \eta x_i \delta_{hidden,j} \tag{2.17}$$

where,

$$\delta_{hidden,j} = \varphi'(h_j(n)) * \hat{e}_j$$

$$= h_j(n)[1 - h_j(n)] * \hat{e}_j(n)$$ therefore ,

$$w'_{ji} = w_{ji} + \Delta w_{ji} \tag{2.18}$$

is the adjusted weight for the $i^{th}$ input layer neuron.

### 2.2.3. Learning Rate And Momentum

The learning procedure requires that the change in weight is proportional to the error signals. The constant of proportionality is the learning rate η. It is a small positive number that don't lead to oscillation of the learning procedure of the network. Usually large η will lead to oscillation and can be avoided by making the change in weight dependent on the past weight change by adding momentum term [Kores, B.,and Vandher Smagt, P., 1996],

$$\Delta w_{ji}(n+1) = \eta x_i \delta + \alpha \Delta w_{ji}(n) \tag{2.19}$$

where, $\alpha$ is a small fraction number, usually $0 < \alpha < 1$

Eq.(12) is called the generalized delta rule.

The role of the learning rate and the momentum term are shown in fig.2.8. When no momentum is used, it typically takes a long time before the minimum is reached with a low learning rate (a), whereas for high learning rate the minimum may be never reached because of the oscillation (b). When adding the momentum term, the minimum will be reached faster (c).

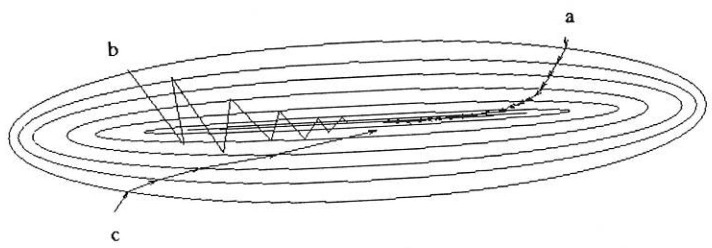



***Fig. 2.8 The descent vs. learning rate and momentum***

### 2.2.4. Generalization, Over Fitting And Stopping Criteria

Several techniques are available to address the problem of over fitting. One of the most successful methods for overcoming the over fitting problem is to simply provide a set of validation data to the algorithm in addition to the training data [Mitchell, M., 1997]. The generalization error will be then estimated. The most commonly used method for estimating generalization error in NNs is split-sample or hold-out validation.

A test set is reserved from the training data set and after training the network it will be run on the test set. The error on the test set provides an unbiased estimate of the generalization error.

An improvement of the hold-out validation is the cross-validation that allows using all of the data for training, i.e. in a k fold cross validation, the data is divided into k subsets of equal size (approximately). The network will be trained k times, each time leaving out one of the subsets from training for estimating the generalization error and the results are then averaged.

In the above methods of estimating generalization error, the training session is stopped periodically (i.e. every so many epochs) and the network is tested on the validation subset after each period of training. The validation error rate decreases monotonically to a minimum and starts to increase as the training goes on. Early stopping point is the point at which the validation error rate is minimum. Care should be taken in choosing the minimum point. The minimum reached by the network might not be the global but rather an inflection in the slope (or local minima) after which the error goes down toward another minimum. Using lots of hidden units will avoid getting trapped in a local minimum.

# CHAPTER THREE

# LOAD FORECASTING

## 3.1. Factors Affecting Load

Electric load is influenced by different factors with varying degrees on each consumption units. The magnitude and types of factors that determine the consumption level of each unit also varies with change in load pattern.

Generally, the load of an electricity utility is composed of very different units. A large part of the electricity is consumed by industrial activities. This part is usually characterized by the level of the production. The load is often quite steady, and it is possible to estimate its dependencies on different production levels. However, from the utility selling electricity point of view, 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 disturbance in the load level.

The other part is consumed by the residential units in the form of heating, lighting, cooking, laundry, etc. In the case of residential unit, many factors are influencing their electricity power consumption. Many social and behavioral factors can be found that varies according to each individual behavior. Big events, holidays, even TV programs, affect the load [Gross and Galiana, 1987].

The weather is the most important factor, especially temperature, which is addressed by many researchers.

The usual approach in load forecasting is to concentrate in the aggregate load of the whole utility. This reduces the number of factors that can be taken into account.

The most identified factors that influence load can be categorized into weather, calendar, and economical and random factors.

### a) ***Weather effects***

One of the most important factors that affect the load profile is the weather. Particularly, in the short run, weather causes large variations in the aggregate load. We can find that not all weather factors are important. Some are typically random during a period of, such as wind speed and thunderstorms [Bao, J., 2002]. Temperature and humidity are the most commonly considered factors for load forecasting. An electric load prediction survey indicate that of 22 research reports considered 13 made use of temperature only, 3 made use of temperature and humidity, 3 utilized additional weather parameters, and 3 used only load parameters [Hippart, H.S., et al., 2001]. The other weather effects that influence the load to a lesser extent include; cloud cover, dew point, rainfall, and snowfall. Temperature has a direct influence on many kind of electric consumption, such as air conditioner, heater and refrigerator. However, the leading weather influential factors for the specific consumer may be different. The individual consumers that belong to low income groups, like most of the population in Ethiopia, may not able to purchase those electrical units, and hence the variation on temperature may not be a crucial factor for the change in load consumption pattern of this particular groups (the relationship is examined in the next chapter).

#### b) *Calendar effects*

The calendar date is the other factor that determines the shape of the load profile. The day of the week, hour of the day, month of the year, weekends, weekdays and holidays are known to cause disturbance in load consumption pattern. The week day profiles are typically different from the weekend profiles. The demand for energy in the night time of the day is expected to be different than day time. The month of the year capture the seasonal effect. Hence, in load forecasting, making a distinction between loads data based on the calendar effect will help the network to understand the inputs belonging to each class. A number of researches have taken into account these effects in different manners. Some dealt only with the week day profile discarding the weekends and holidays [chow, T.W.S and Leung, C.T., 1996], [Park, D.C., et al., 1991]. Holidays usually pose a special problem. Some authors group them with weekend days; other reserve them special classes, or device to deal with them [Bakartzis, A.G., et al., 1995], [Khortanzad, A., et al., 1998], [Khotanzad, A., et al., 1995], [Papalexopoulos, S.H, and Peng, T.M, 1994].

**c)** ***Economic effects***

In the long run, economical effects such as price of electricity, industrial growth, economic trends (recession or expansion), service area demography (rural, residential), etc. factors play the most important role determining the evolution of the electricity demand.

**d)** ***Random factors***

These are usually small in the case of individual consumers. Social events like; sport and popular TV programs add uncertainty in the forecast. The start or stop of large loads from industrial units and strikes can cause relatively large disturbances.

In general, the above effects can be considered as influencing factors to the load forecast and can be possible candidates to be used as inputs for neural network trainings. In this work, those factors that are assumed to have a great impact on the daily load profile are dealt with and hence the economic and random factors will not be considered. Moreover, since all consumption units are combined into an aggregate load, it is assumed that the forecast will rest largely on the past behavior of the load.

## 3.2. Conventional Approaches To Load Forecasting

Forecasting in general, and load forecasting in particular has been studied by researchers over decades. Though the applications of neural networks to load forecasting is getting more attention in recent years, some authors remain skeptical and believe that the advantage of using NNs in load forecasting has not been systematically proved yet. The early day's approaches to the load forecasting have also remained applicable in the present days with varying degrees of success. Over the last few decades a number of forecasting methods have been introduced. Most of these methods use statistical techniques combined with artificial intelligence such as neural networks, fuzzy logic and expert systems. Most of these traditional techniques used for short-term load forecasting can be broadly classified as the stochastic time-series methods and the regression-based methods.

### 3.2.1. The regression models

Regression models normally assume that the load can be divided into a standard load components and a component linearly dependent on some explanatory variables, such as weather, day type, and customer class. The model can be written as:

$$z(t) = b(t) + \sum a_i y_i(t) + \varepsilon(t) \qquad (3.1)$$

where, b(t) is the standard load, $\varepsilon(t)$ is a white noise component, and $y_i(t)$ are the independent explanatory variables. The most typical explanatory variables are weather factors.

A typical regression model decomposes the load into basic and weather dependent components have been used by [Hyde, O. and Hodneth, P.F, 1997]. Engle, R.F, et al, (1992) presented several regression models for the next day peak forecasting. Their model incorporates deterministic influence such as holidays, stochastic influence such as average loads, and exogenous influence such as weather.

Regression models are among the oldest models suggested for load forecasting. They are quite insensitive to occasional disturbances in the measurements and can be implemented easily. However, they are basically linear devices, and the load series they try to explain are known to be distinctly non-linear functions of the exogenous (independent) variables.

### 3.2.2. ***Stochastic time series models***

Time series methods treat the load pattern as a time-series signal with known seasonal, weekly, and daily periodicities. These periodicities give a rough prediction of the load at a given season, day of the week, and time of the day. The difference between the prediction and the actual load can be considered as a stochastic process (random signal). The basic principle is that the load time series can first be transformed into a stationary time series (i.e. invariant with respect to time) by a suitable differencing. Then the remaining stationary series can be filtered into white noise. The models assume that the properties of the time series remain unchanged for the period used in model estimation, and all disturbances are due to this white noise component contained in the identified process. By the analysis of the white noise, a more accurate prediction can be obtained.

Several techniques have been used for the analysis of this white noise. Among them the Box-Jenkins class of models called ARMA (Autoregressive moving average) and ARIMA (autoregressive Integrated moving average) are the most widely used classical time series methods. ARMA models are usually used for stationary process while ARIMA is an extension of ARMA to no stationary process. Both use the time and load as the only input parameters. An extension to them that uses the weather and time of the day as exogenous variables is ARIMAX and is considered as the most natural tool for load forecasting.

The general form of ARMA and ARIMA models are described as follows:

- ***The ARMA model***

$$\phi_p(B)y_t = \theta_q(B)a_t \tag{3.2}$$

where, $\phi_p(B) = 1 - \phi_1(B) - \phi_2(B^2)\ldots\phi_p(B^p)$

$\theta_q(B) = 1 - \theta_1(B) - \theta_2(B^2)\ldots\theta_q(B^q)$

$B$ is the backward shift operator denoted by

$B\,y_t = y_{t-1}$ and hence $B^k y_{t-1} = y_{t-k}$

$\phi$ and $\theta$ are non-stationary model parameters

$y_t$ is the model time series, t=1,2, …, N

$a_t$ is the white noise sequence, i = 1, 2, …, N p and q are coefficients of autoregressive components $\phi(B)$ and moving average component $\theta(B)$ respectively.

- ***The ARIMA model***

$$\phi_p(B)\nabla^d y_t = \theta_q(B)a_t \tag{3.3}$$

where, $\nabla$ is the backward differencing operator and $\nabla$ = 1-B, i.e. $\nabla y_t=(1-B)y_t$.

d is the order of the model, i.e. $\nabla^d y_t=(1-B)^d y_t$

Vemuri, et. al, (1981) have identified ARMA models which describes the stochastic behavior of hourly load demand on a power system. The proposed ARMA model has gone two step procedures; first they identified the finite order AR model and next an equivalent ARMA model. Based on their findings a forecast for the Lincoln Electric system hourly loads (at three hour intervals) of one day have been made. These forecasts have 7.8% maximum error with a daily rms error of 5.7% for the AR model. In comparison, the corresponding ARMA model has maximum and daily rms error values of 6.9& and 4.5% respectively.

The experiment on the other time-series model that assumes exogenous variables (ARIMAX) have been conducted by Yang, et.

al, (1996). They used evolutionary programming (EP) approach to identify ARIMAX model parameters for one day to one week ahead hourly load demand forecasts. They used the ARIMAX model to represent relationship of load with temperature. Their experiment was tested on the Taipower system load data for one day and one week ahead hourly load forecasting. The possibility of improving the forecasting accuracy of the ARIMAX model has been proven through the proposed EP based approach compared to that of the traditional gradient search based approach.

## 3.3. Applications Of Neural Networks To Load Forecasts

The research in the field of ANN has a history of many decades, but after a diminishing interest in the 1970's a massive growth started in the early 1980's. It was in this period that ANN research reached power systems due mainly to the contribution of Sobajic and Pao [Damboreg, M.T., 1990]. Since then main additional applications have been followed to the power system areas.

The load forecasting is one of the major areas of research field in the electric engineering. The supply industry requires forecast with lead times that range from the short-term (a few minutes, hours, or day ahead) to the long-term (up to 20 years ahead). Short-term forecast, in particular has become increasingly important since the rise of the competitive energy markets.

In this research work, the literatures of neural networks in load forecasting are surveyed. This literatures offer the background for the rest of the work.

The interests in applying ANN to load forecasting have considerable variation among different researchers. But, most of the approaches reported since its uses, are based on the use of a MLP network as an approximation of an unknown non-linear relation. However, the number of different ways to use this type of networks seems to be unlimited on the basis of the articles.

The review of the previous research works are divided into three parts. The first part refers to articles discussed on short-term load forecasting with one output neuron. In the second part the models used to forecast load profiles are discussed. Finally, various suggestions to utilize ANN for load forecasts that are different from the previous parts are examined.

### 3.3.1. One-Hour Ahead And Day Peak Load Forecasts

Most of the research works in load forecasting focus on short-term load forecasting ranging from one hour ahead to daily peak ahead. Forecasts below one hour have also been made. This section focuses on the neural network architectures with only one output nodes used to forecast the next hour's load and next day's peak load.

Gavrilas, M.,(2000) has developed two types of neural networks for short-term load forecasting. First a one hour forecast type network using the MLP architecture and secondly the fuzzy SOFMs model is described to perform short-term and medium-term load forecasts respectively. This section deals only for the first type and the other is left for reading to those interested.

Gavrilas used the basic three layers MLP. Based on a correlation analysis the input variables chosen are: three types of historical load data; previous 5 hrs load values before the forecast hour (i.e. h-1, h-2, …, h-5), previous day hourly load values at the forecast hour and 2 hours before (i.e. h-24, h-23 and h-22) and the previous week hourly load at the forecast hour and 2 hours before (i.e. h-168, h-167, h-166), the forecasted day max, min and average temperature. Apart from the above variables, a day type code (where 0.5 indicating weekends and 1.0 indicating weekdays) and the hour code that takes five binary values (00001, 00010, …, 11000) for each hours of the day are included. The total number of input neurons is 20. This model is tested on a one week data and the result obtained indicate a forecast error ranging from -7.08 % to 2.71% for each day of the week.

Ho, K.L., et al., (1992) also proposed a MLP neural network that uses an adaptive learning algorithm to forecast peak and valley loads. The MLP type neural network that uses the back propagation learning algorithm to determine the connection weights assumes a constant learning rate and momentum. Ho, K.L., et al. demonstrates the effect of different learning rates and momentums on the convergence property of the learning process. The neural network has 46 input neurons containing the forecasted high (low) temperature in the three areas of Taiwan for the day load forecast being conducted, the recorded area high (low) temperatures in the previous day, and the recorded area high (low) temperatures and peak (valley) load in the past ten days with the same load pattern as the forecast day. The hidden layer constitutes 60 neurons. In the training process 30 input/output patterns are employed and the result obtained from the new approach was compared to the conventional method and is found that the adaptive algorithm converges much faster than the conventional. Moreover, the convergence property of the adaptive learning algorithm will not be affected by the learning rate and initial value of momentum.

Chen, S., et al. (1992) used model that consists of one fully connected main ANN and three supporting fully connected ANNs. These four ANNs are then combined to form a non-fully connected ANN. In the whole process 31 input variables are considered containing past load and temperature data, temperature forecast data, and hour-of-day and day-of-week information. The main ANN provides a basic forecasting reference. It establishes three feed forward connections. The first feed forward connection is formed between the current load input (neuron 31) and the output neuron. The other two feed forward connections are made between temperature inputs and the output neuron. These three feed forward connections provide information on the most recent load and temperature reference. Each supported network provide additional learning capacity to the main ANN. The first provide information about the relationship between temperature and the forecasted load, the second is used to capture the relationship between variation of load curve and the temperature for different days of the week, and the last supporting ANN is used to provide information about the effect of time on the load curve. All the outputs of these supporting ANNs are connected to the main ANN. The training is done on a two week data set and the result obtained shows an improvement over the ARIMA model, which is used as a comparison.

Senijyu, et, al. (2002) have recently reported a new concept to forecast future load using ANN. The paper focuses on two issues; first, it states a one-hour-ahead load forecast that uses the actual temperature of the forecast day instead of the forecasted temperature. Secondly it argues that the use of all similar days' data to learn the trend of similarity is a complex task and hence it is necessary to reduce the network structure and learning time.

By considering the above arguments, Senijiu, et al., have proposed a one hour ahead forecasting method using the correction of similar day's data. The selection of similar days is made using the Euclidian norm with weight factors. Their model is composed of three layers, which have feed forward connections. Nine input neurons (deviations in temperature between the forecast day and a similar day, deviations of power between the forecast day and on a similar day and the correction for forecast load), 20 hidden layer neurons and one output layer neuron. The forecasted load power is obtained by adding a correction to the selected similar day data. The correction is yielded from the NN. The network is trained using data of past 30 days the day before the forecast day and past 60 days before and after forecast day in previous year. The test result shows that the proposed model is much better than the simple regression model, NN using all similar days' data and using the NN, which adopts off-line learning.

### 3.3.2. Load Profile Forecasts

This section reviews some of the research works on short-term load forecasting using NNs that have several output neurons used to forecast sequence of hourly loads. Typically, the 24 output neurons used to forecast next day's 24 hours load, which is called as "load profile" will be given much emphasis.

Bakirtizs, A.G., et al., (1996) use a fully connected three layer feed forward ANN with 63 input neurons, 24 hidden neurons and 24 output neurons representing the next day's 24 hourly forecasted loads. Among the input neurons the first 48 were used to represent the hourly historical loads of the present and previous days. Minimum and maximum temperature for the forecast day and the present day that represents two weather stations are part of the input neurons. The day of the week's information is added to the ANN as coded bits in the remaining neurons. The model considers the holiday effect to the load forecast in a separate way. First the ANN is trained using only normal days (i.e. excluding the holidays in both input/output mapping). This basic ANN model is then adjusted to obtain holiday load forecast by increasing the ANN input vectors containing holiday load by some adjustment term before they are feed to the ANN for training. The overall performance of the model shows accurate load prediction for both normal day and holidays. It exhibits 2.24% and 3.50% average error for the normal day and holiday respectively.

Most of the researches on short-term load forecasting using ANN have confirmed a good performance for load patterns during common days. To the contrary, less care has been internationally taken to the application of ANNs to the forecasts of the anomalous load condition periods characterized by exponential circumstances of a national nature such as; strike, holidays and sport events. Lamedica, et. al (1996) combines the unsupervised and supervised learning techniques to take into account the anomality of the load. The Kohen's SOM is used to classify the different daily load profiles into different clusters. These clusters types are coded and provided as input pattern to the basic ANN that constitutes 48 historical hourly loads of 2 days before the forecast day, 31 hidden neurons and 24 output neurons comprising the hourly load of the forecast day. The daily load profiles and the binary codification of their clusters constitute the new training set of the ANN. Load data of one year is used for the training purpose. The combined procedure has shown a noticeable increase of the forecast accuracy in all the anomalous load conditions.

A different approach has been implemented by Khotanzad, A., et al., (1997) which is one of the ANNSTLF called the second generation engine. The approach relies on 24 small MLP's, one per hour of the day. The new extension to the previous method is the distinction made between forecast indicators for various hours of the day. Three categories of indicators are employed; past loads, past weather and the forecast weather for the coming day. These categories are supposed to have different impact on various hours of the day based on these observations, the hours of the day are divided into four categories and a different set of inputs is defined for the MLP's in each category. The grouping of the hours is as follows:

1. early morning (hour one to nine)
2. Mid-morning, early afternoon, and early night (hours 10 to 14 and 19 to 22)
3. Afternoon peak (hour 15 to 18)
4. Late night (hour 23 to 24)

Each category has 19, 25, 25 and 21 input variables where 17 INPUTS are similar for all categories. Additional inputs representing the temperature and humidity effect have been also included but, the values of the later inputs are obtained from a separate weather forecasting engine developed using ANN. The data for holidays is treated as outliers and not included in the training set. The decision is made because there are no enough examples about holidays in the total training set that helps the network to understand their effect. The performance of the model indicates that the second generation engine with its much smaller size and complexity actually outperforms the firs generation engine.

Chow, T.W.S et, al. (1996) have proposed a neural network model for STLF that utilizes the full dynamic range of ANN. The model is used to provide hourly load forecasts for one day ahead. The study focus on multilayer feed forward network. It uses a modified nonlinear autoregressive neural network model to take the non-stationary effect into account. Different weather factors are also included in the model. Temperature, relative humidity, rainfall and sunshine are the weather factors. The model constitutes 81 input neurons, 81 hidden neurons and 24 output neurons. The influences of weekends and standard holidays on the load are not considered in the STLF. The overall percentage error is 1.755 for the forecast of 24-hours ahead. Based on this result they concluded that the electric load demand in Hong Kong is basically non stationary in nature.

Alfuhaid, A.S., et al., (1997) has studied two different approaches of ANN to short-term load forecast for the electric power system of Kuwait. Both approach use the multilayer feed forward neural network technique. In the first approach, the input pattern consists of 104 variables of which 48 represent the past half-hourly load, 24 represent past hourly temperature, 24 represent past hourly humidity, forecasted maximum and minimum temperature for the forecast day, one forecasted humidity again for the forecast day and 5 variables denoting the month and type of day. In the second approach, this is the main focus of the research; present a cascade artificial neural network that captures the sensitivity of the nonlinear influence of temperature and humidity. Two ANNs are employed to support the main ANN; an additional network aiming to forecast the maximum, minimum and daily energy is modeled with input patterns consisting of total energy (MWH), load factor, maximum load, minimum load, temperatures and the time at the maximum and minimum load, maximum humidity and 4 digits for type of the day. The supporting ANN is structured with 1 hidden layer of 8 neurons and an output layer of 3 neurons. The three output neurons are used as input to the main ANN of the second approach (cascade ANN) in addition to the other inputs considered in the first approach. This network has 2 hidden layers with 35 neurons each and 48 output neurons representing the half-hour load of the forecast day. Both approaches are compared on same training set (365 input patterns) and the result indicates that approach two is more accurate than approach one.

### 3.3.3. <u>Other reported approaches</u>

Though, the number of approaches in applying neural networks to the electric load forecast based on MLP increase in recent years, other types of neural networks has also brought significant interest to the researchers in the area.

One of these methods is the self-organized map or unsupervised learning. The leading researcher in the unsupervised learning is Tuevo

Kohenen (Anderson, D. and McNeil, G, 1992). He has developed a self-organizing network, sometimes called an auto-associator that learns without the benefit of knowing the right answer.

Hsu, Y. and Yang, C., (1991) use Kohenens self-organizing map to identify the different day types based on their load profiles. The daily forecast is obtained into two phases. In the first phase, the forecast for the daily load shape is obtained by averaging some load patterns of the days belonging to the same day type as the target day. In the second phase, the daily peak and valley loads are forecasted using the MLP network. The actual forecast is then obtained linearly scaling the shape to match the peak and valley loads.

Yoo, h. and Pimmel R.L (1999) have also developed a self-supervised adaptive neural network to perform a STLF. They used the self-supervised network to extract correlation features from temperature and load data. They found a MAPE of 0.91 percent for one-hour-ahead and 1.92 percent for day-ahead forecasting.

Vermaak, J. and Botha, E.C, (1998) suggest one of the members of a class of neural network called the fully connected recurrent neural network to the STLF problem. Their experiment shows that recurrent network can successfully capture the dynamic behavior of the load, resulting in a more complex and natural internal representation of the temporal information contained in load profile.

The use of functional-link and fuzzy set theory combined with neural networks are also proposed to the STLF problem by different articles [Dash, P.K., et al., 1997], [Daneshdonst, et. al, 1988].

In general the problem to STLF has been addressed by different researchers. The types of models used by each researcher are different in nature. It can be said that most of the models used so far are based on feed forward type MLP networks, but also models using unsupervised learning, fuzzy concepts and recurrent networks are described by a number of researchers. There is a single feature, which clearly divides the models into two distinct classes. Some models are based on the idea of producing the whole load curve of a day at one time while others able to forecast the hourly load ahead at any time of the day.

In this research work the experience in the different feed forward MLP networks are utilized and adopted wherever it is found appropriate.

# CHAPTER FOUR

# EXPERIMENTATION

## 4.1. Data Preparation Analysis And Processing

### Overview

One of the most important components in the success of any neural networks solution is the data. The quality, availability, reliability and relevance of the data used to develop and run the system are critical to its success. If input data has been processed in such a way that it clearly reveals the important information, the task in building a model can be said to go through half way with success. On the other hand, if the necessary input information is complex and confusing, the task of building the model will be difficult or even unsuccessful. The design process of the neural network involves the following steps as depicted in fig 4.1.

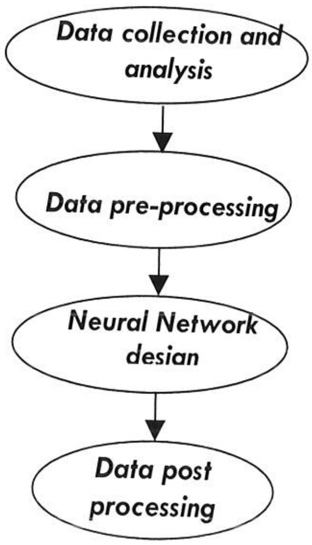



*Fig. 4.1 steps in the neural network design*

Data processing starts from the data collection and analysis followed by pre-processing and then fed to the neural networks. Finally, post-processing is needed to transform the outputs of the network to the required outputs, if necessary. The following section discusses the most important considerations involved in the above steps.

### 4.1.1 Types of variables

#### I. Categorical Variable

Categorical variable do not have a natural ordering and do not have relationships like "greater than" or "less than". Some of them came from some input values that do not have numeric values but have to transform to numeric values as input variables.

Categorical variable can be presented to the network with the *l-of-c* encoding scheme, which has as many units as there are values that the variable can take on. Exactly one of the units will be turned on according to the value of the variable, and all the other units will be turned off. Another way to encode categorical variable is to represent all the possible values to one continuous input variable. The later case imposes an artificial ordering on the data that does not exist. But for variables with a large number of categories, this can dramatically decrease the number of input units.

### II. Ordinal Variables

Ordinal variables have a natural ordering. Such data can simply be transformed directly into corresponding values of continuous variable, with or without scaling.

## 4.1.2 Data Collection

The data collection step comprises the following tasks: -

### I. Identifying the data requirement

The first thing to do when planning data collection is to decide what data will be needed to solve the problem. In general, it will be necessary to obtain the assistance of some experts in the field or go through various literatures in the problem area. One needs to know:

a) What data are definitely relevant to the problem;

b) What data may be relevant;

c) What data are collateral? Both relevant and possibly relevant data should be considered as inputs to the application.

## II. Identifying data sources

The next step is to decide from where the data will be obtained. There are two types of data sources:

a) Primary data sources are obtained from experiment or through observations and are not published;

b) Secondary data sources, where the data are available in the form of documents, research results etc.

Making use of secondary data requires being consistent, accurate, and applicable. Known sources of data will enable to alleviate difficulties that may arise during data analysis by consulting the expertise available in the source area.

## III. Determining the data quantity

A reasonable estimation of the data required in developing the neural network model is an important task to be considered at large. The amount of data used in the training process determines the NN to learn different properties of the data. If the data is collected is limited, it may not reflect the full range of properties that the network should be learning and this will limit its performance with unseen data. On the other hand, it is possible to introduce unnecessary expense by collecting too much data. In general, the quantity of data required is governed by the number of training cases that will be needed to ensure the network perform adequately. The intrinsic dimensionality of the data and the required resolution are the main factors determining the number of training cases and therefore, the quantity of the data required.

### 4.1.3. <u>Data Pre-Processing</u>

Today's real world databases are highly susceptible to noisy, missing and inconsistent data due to various reasons, such as attributes of interest may not always be available, relevant data may not be recorded due to misunderstanding of the subject under consideration, instrument used may be faulty, etc

There are a number of data preprocessing techniques. Data cleaning, data integration, data transformation and data reduction are the most commonly used techniques that help to improve the overall quality of the data.

Data cleaning helps to remove the noisy data and correct inconsistencies in the data. Noisy data can be smoothed by the binary method, clustering or regression methods. Some data inconsistencies can be corrected manually by consulting the domain experts. Data integration usually applied to data from multiple sources in order to match up entities having similar properties. The other task in the pre-processing step is the data transformation technique. It is used to transform the data into a form appropriate for analysis. One of the widely used methods of transformation is normalization, which helps to scale the data into specific ranges.

In the design process of the neural networks, the data pre-processing task is vital. Before data are ready to be used as input to a NN, the data should be pre-processed in order to make the network process more manageable. There are two basic techniques, which can be used to help us understand the data.

### a) Statistical analysis

Neural networks can be regarded as extension of standard statistical techniques. Most of the above methods are the variants of the statistical techniques, and so such tests can give us an idea of the performance the network is likely to achieve. Some statistical techniques can give useful clues about the appropriateness of the data.

### b) Data visualization

The distribution of the data can easily be seen when data is presented in a suitable format. Graphical description of data sets is one of the statistical methods used in various applications. Distinguishing features such as valleys and peaks, which characterized the data, can be spotted easily by plotting a graph.

The combination of visualization and statistical analysis give good results in the pre-processing step. Visualization gives an appraisal of the data, and ideas about the underlying patterns, while statistical analysis enables to test these ideas. Pre-processing, also frequently means partitioning the input space. It is the process of dividing the data into validation sets, training sets, and test sets. The primary concerns should be to ensure that: -

a. The training set contains enough data, and suitable data distribution to adequately demonstrate the properties we wish the network to learn;

b. There is no unnecessary similarity between data in different data sets.

Reducing the dimensionality of the input data and minimizing the number of inputs to the network is another task to be considered during data pre-processing. The process of reducing the input data set is called feature extraction. Prior knowledge about the problem area can simplify the selection process.

### 4.1.4 Data Post-Processing

Post processing covers any process that is applied to the output of the network. As with pre-processing, it is entirely dependent on the application, and it may include detecting when a parameter exceeds an acceptable range, sometimes it is, just the reverse process of data pre-process.

## 4.2 One-Hour Ahead Load Forecasting Model

In this and in the following section two types of short-term load forecasting models are built and tested. They are 1) the hourly load forecast 2) daily peak load forecasts. The models are mainly based on the most established ideas presented in the literature. In this section the process undertaken to build the hourly load forecast model will be discussed. This network has only one output node, which provides the forecast for the next hour. To obtain the forecast for a longer lead-time, the output is fed back to the same network along with other input variables to get the forecast for the next hour. By continuing, the forecast can be obtained for any arbitrary number of lead hours. At each step, the input vector includes at least one forecasted value. This kind of approach is taken by, e.g. (Park, et. al, 1991b), (Chen, et. al, 1992).

### 4.2.1 Input data collection and preparation

As mentioned earlier, the primary step in the data collection process is identification of the data requirement. To this end, different research works were examined and the knowledge from the domain experts is taken into account. Hourly historical load are the most relevant data to be considered as input data to the network under consideration. The next important factor to determine the shape of the load profile is the calendar date. The load profile will not be the same for all the days. Some days are typically different from the others. Therefore, appropriate classification of the day types is required. A complete one-day load profile is expected to have different pattern at different time of the day. The load at night, when every body is sleeping, surely consists of low load values than the daytime. So, the network should learn the hourly trend in some predefined manner. In the analysis of the one-hour ahead forecast, the weather effect is not considered following the result obtained from the daily peak load forecast model feature selection analysis (see section 4.3.2.2.). Moreover, the effect of holidays is not considered too, since there isn't enough training data representing the effect of the periods under consideration for the training data set.

According to the above considerations, the only data to be collected is limited to historical load values. These values came from the organization, the researcher focuses on. Prior to the data collection process, preliminary discussions were held with EEPCO officials on the type of data available at their premises. In addition to this, visual inspections of the available data have been made in all the substations found in Addis Ababa. It has been found that all the substations are recording the load manually from the meter reading at each hour intervals. There are remarkable numbers of substations, which keep the records partially for specific time of the day. Few substations are continuously recording the reading for the whole day. This research is based on one of these substations, which is also considered as the major distributor among the others. The nature of the available data has made the task of data collection to be difficult.

Using the above stated input data; the model can be set up, to include the following effects.

a. Historical load

b. Hour of day

c. Day-type

Based on the above decision, the data collection task starts manually using the pre-designed format for the same purpose. The record sheet, which is called as "the day log sheet", contains a number of information pertinent to their activities. In one single sheet, the incoming and outgoing load for each transmission line available to the particular station is registered periodically. The Addis Center substation, where this research is based on, is one of the most efficient stations (according to the researcher's view) to keep its records permanently. Since the research objective is to predict future load at small interval period of time, such as 1 hour, the 24 hours load pattern for a certain period of time is required for experimentation process.

Even though there is no definite method to determine the amount of training datasets required for the experimentation process, care should be taken not to use too many or too few examples.

The decision as to how much training set should be included to train the network and which time period data will be representative pattern is the difficult question to answer. However, in forecasting the hourly load using large set of data may cause the training process to be exhaustive requiring large network size. On the other hand, data concerning short time period, can be trained fast. The problem is the relatively fast changing of the load characteristics. If, for example, one or two preceding months are used for training, the conditions may already be quite different at the time of the forecasting. However, the training can be performed very often. Alternately, the network weight can be adapted continuously by training with the most recent data. For this particular research, load data from Jan.10, 2004 to March 09, 2004 have been used for the model building process. Apart from the above justification, other considerations have been taken into account to decide about the period and size of the data set.

- The size reflects the experience of various researchers reviewed in this research work.

- According to the customs of the Ethiopia people there are only 3 holiday events in a year that highly influence the load pattern (New Year, Christmas and Eastern). Since holiday effects are not considered here, the selection reflects holiday-free events.
- The most reasonable argument for the selection seems to be the insignificant change in the load pattern and slow rate of change on the electricity supply throughout the year. Figure 4.2 (a)-(f) depicts the daily peak load pattern for 6 consecutive months in the year 2003/4 (September 2003 - February 2004). It can be seen that on the average, the patterns are similar except for holidays and unforeseen circumstances.

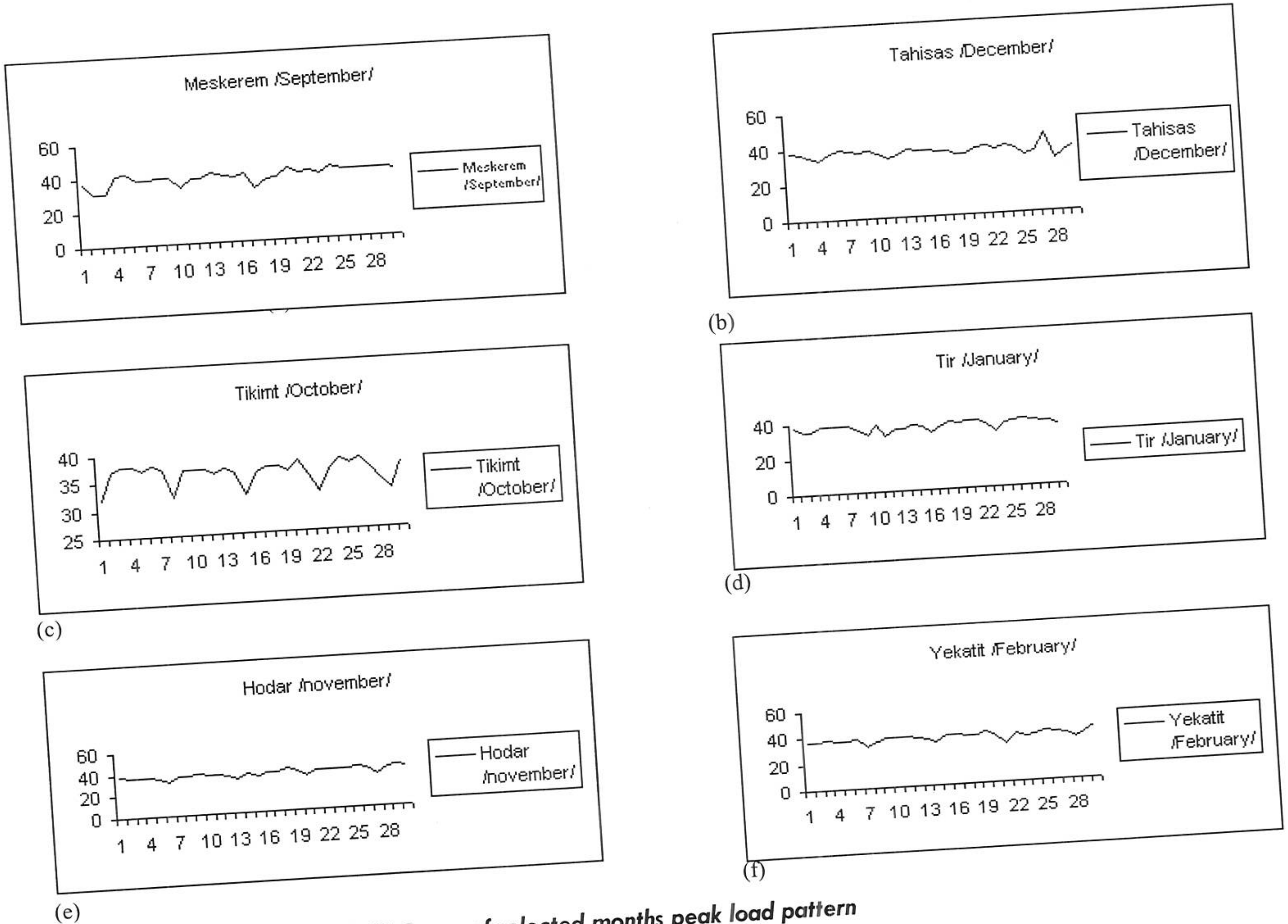



**Fig 4.2 (a)-(f) Curve of selected months peak load pattern**

After the initial data collection, (based on the pre-designed format) all the raw data is stored into a computer using MS-Excel sheet. The file consists of load data in MW, the corresponding day-type and hours of the day.

### 4.2.2 Data Pre-Processing Steps

The next task in the network design process is to decide on the data to be used for analysis. The following activities are undertaken during this phase:

#### a. Data Cleaning

This phase improves the data quality level as required by the selection analysis. Some includes the selection of data to be combined with the cleaning task (CRISP-DM, 2000). But, since the selection task involves different techniques, in this research work separate section is devoted to it.

By applying some statistical method, the data cleaning process helps to correct outliers and missing values, if any. In this regard, the data were carefully examined if such events exist. Since, all the data initially collected manually and entered to the computer with a great care no missing values have been found. But, as to the outliers, few records showed different patterns when the whole data was graphically visualized on a daily basis. The graph indicates unusual peak or valley at the points where the outliers exist. The corrections of such occurrences are made by revisiting the data sources and consulting the experts in the area. In some cases, the researcher himself corrects them. This is done by examining the neighboring records and referring to the additional field containing the load reading in ampere (AM), which was deliberately included for checking purposes. In all the circumstances, out of the total 1440 records only 17 of them were subjected to correction.

### b. Data Selection

During this phase, data to be used for the analysis were determined based on some criteria. As it was mentioned earlier, apart from the calendar effects, the model uses historical load value for the prediction purposes. The extent of historical data to be considered was an unanswered question. In other words, the inputs in relation to historical load should be decided first. In this regard, many literatures examine different combinations on several test sets and select the one with good result. To begin with, the researcher chooses to make some statistical analysis on the data to select the initial sets of data and adjust the inputs in the training stage based on the test results. The simplest way to identify the relationship of future load with previous load is to plot the daily load profile and find the existing patterns from the graph by visual inspection.

The most useful aid in measuring the relationship of successive observation is to examine the auto correlation coefficients of the series. Auto correlation coefficients measure the correlation between observations at different distance apart [Chatfield, C., 1996]. The coefficients are calculated using the following formula:

$$r_k = \sum \frac{(x_i - \bar{x}) \ (x_{i+k} - \bar{x})}{\sum (x_i - \bar{x})^2} \qquad (4.1)$$

where, $x_i$ (i=1,2 ……N) are the observations

$\bar{x}$, the mean of the observations,

k, the lag value

$r_k$, autocorrelation coefficient.

The auto-correction coefficients were calculated at different lag periods from 1 to 168. The results are shown in table 4.1.

| *Lags* | *ACF* | *Lags* | *ACF* | *Lags* | *ACF* | *Lags* | *ACF* |
|---|---|---|---|---|---|---|---|
| *1* | *0.893* | *48* | *0.864* | *96* | *0.831* | *144* | *0.804* |
| *2* | *0.693* | *49* | *0.783* | *97* | *0.755* | *145* | *0.734* |
| *3* | *0.435* | *50* | *0.596* | *98* | *0.576* | *146* | *0.562* |
| *4* | *0.166* | *51* | *0.354* | *99* | *0.340* | *147* | *0.338* |
| *5* | *-0.071* | *52* | *0.099* | *100* | *0.093* | *148* | *-0.103* |
| *6* | *-0.243* | *53* | *-0.124* | *101* | *-0.123* | *149* | *-0.103* |
| *7* | *-0.343* | *54* | *-0.288* | *102* | *-0.281* | *150* | *-0.254* |
| *8* | *-0.373* | *55* | *-0.378* | *103* | *-0.367* | *151* | *-0.337* |
| | | | | | | | |
| | | | | | | | |
| *24* | *0.884* | *72* | *0.844* | *120* | *0.820* | *168* | *0.834* |
| *25* | *0.803* | *73* | *.0769* | *121* | *0.746* | | |
| *26* | *0.613* | *74* | *.0588* | *122* | *0.572* | | |
| *27* | *0.365* | *75* | *.0351* | *123* | *0.342* | | |
| *28* | *0.107* | *76* | *.0101* | *124* | *0.101* | | |
| *29* | *-0.121* | *77* | *-0.119* | *125* | *-0.110* | | |
| *30* | *-0.285* | *78* | *-0.277* | *126* | *-0.262* | | |
| *31* | *-0.378* | *79* | *-0.366* | *127* | *-0.347* | | |

***Table 4.1 Results of ACF of lags 1-168***

From the results, it can be seen that short-term period show high correlation and the values for previous days exhibits such relationships. The weekly cyclic nature of the load pattern is also supported by the autocorrelation coefficient obtained at lag 168.

So, the historical load data lagged by 1, 2, 24, 48 and 168 hours are selected as input data and accordingly organized. Due to the lagged values, the total data set is reduced to 1272 after excluding the first 168 initial load values from the original data set.

### c. Data Transformation

The transformation process is carried out as per the requirement of neural network toolbox. This task includes encoding the categorical variables, day-type, and hour of the day, and normalizing the selected data set into a specific range suitable for the training process.

In the former case, the hours of the day are encoded using binary bits representation. Accordingly, five binary digits are used to accommodate the maximum hour value (0-23). So, each hour of the day has been converted to the binary equivalent digits. Similarly, the day-type was coded using the binary digits but the day classification should be considered first.

The day-type classification requires careful analysis of the load pattern at each day of the week. Typically, daily load profiles are classified as weekdays and weekends. Some authors [Srinivnsan, D., 1995] conducted analysis for each weekends and week days separately, while others dealt with separate analysis for 3 type of weekday i.e. a) Monday, b) Tuesday to Thursday c) Friday.

In this research work, the days are classified into four groups by analyzing the patterns of the days in different weeks. The following graph (Fig. 4.3) depicts the hourly load patterns of the days in a selected week.

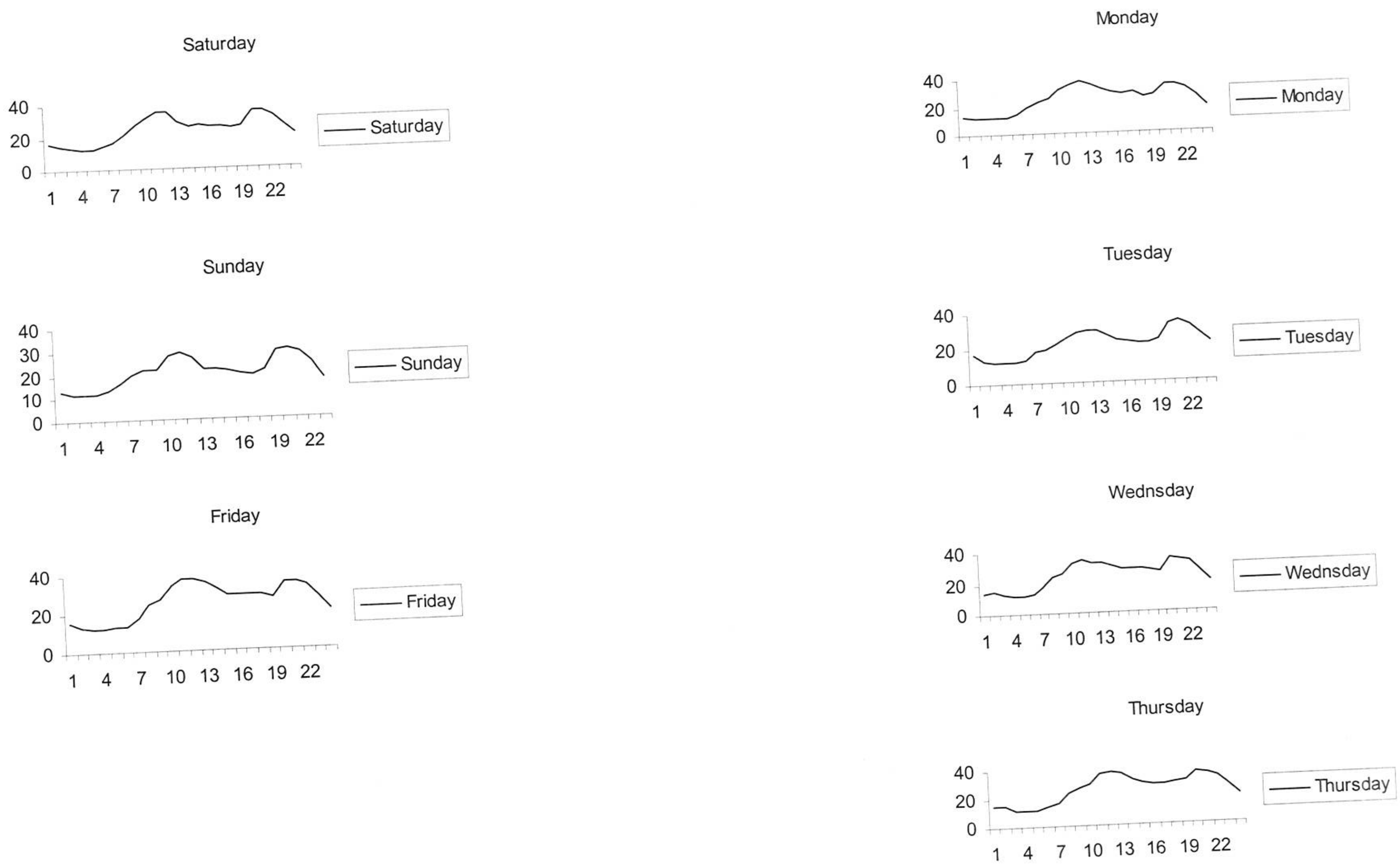



***Fig. 4.4 Hourly Load patterns in a selected week***

From the graph, the decision to take the proposed classification can be asserted easily. The weekends clearly have different load patterns than the weekdays and the shape of the load profile on Friday varies from the others, the rest show more or less similar pattern. Therefore, the day-types are classified as follows: -

| | | | |
|---|---|---|---|
| Group | 1 | = | Sunday |
| " | 2 | = | Monday to Thursday |
| " | 3 | = | Friday, and |
| " | 4 | = | Saturday |

The *1-of-c* encoding method is used according to the above ordering. Therefore, four bits are required and the encoding looks like as follows.

| | |
|---|---|
| ○ Sunday | 0001 |
| ○ Monday to Thursday | 0010 |
| ○ Friday | 0100 |
| ○ Saturday | 1000 |

Then, normalization procedure has followed the encoding task. The normalization process will help the network to speed up the learning phase. There are many methods for data normalization. The Min-max and Z-score techniques are the most frequently used normalization methods. The Min-max normalization maps the original values to the range of 0 to 1 while Z-score coverts the original distribution to a normal distribution with mean 0 and standard deviation 1. The former technique has been used here to linearly scale all load data between 0 and 1 using the minimum and maximum values of the corresponding input vector. The formula used is shown below.

$$v' = \frac{v - \min_v}{\max_v - \min_v}, \tag{4.2}$$

| | | | |
|---|---|---|---|
| where | $v$ | = | the original value |
| | $\min_v$ | = | Minimum value of the input data |
| | $\max_v$ | = | Maximum value of input data |
| | $v'$ | = | The normalized value of the input data |

### 4.2.3 The Network Design Process

This phase is the main part of the whole process. Several different neural networks models are built and tested. The models are mainly based on the most established ideas presented in the literatures and the findings of this research works. In this and the subsequent sections of the chapter, the implementation is carried out using the Matlab Neural network toolbox.

As discussed earlier, the MLP network is used in forecasting both the hourly and daily peak loads. The first subsection describes the network architecture, followed by the training and testing processes.

#### 4.2.3.1 Description of the MLP Network

A fully connected three layer feed forward ANN was used in the development of the networks. Generally, neural networks with three layers have approximation capability for a non-linear function by using hidden neurons with varying numbers (Senjyu, et.al, 2002)

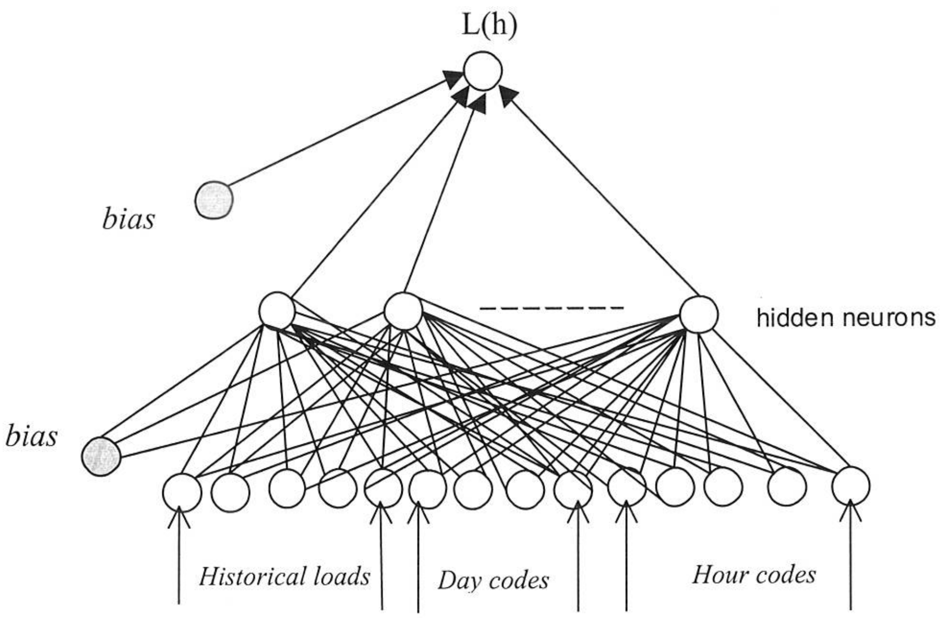



***Fig 4.4 structure of the hourly load forecast model***

Therefore, the model shown in fig 4.5 is composed of three layers, and each layer has a feed forward connection. The model consists of one output node, which corresponds to L(h), the load at hour h. As mentioned in the selection process, the input contains load data of previous days as well as previous week in addition to the most recent hours. Based on the autocorrelation results presented in section 4.2, the most suitable input structures are the one, where the load inputs are:-

$$L_k = L\,(h-k), \text{ where } K = 1,2$$

$$L_3 = L\,(h-24),$$

$$L_4 = L\,(h-48), \text{ and}$$

$$L_5 = L\,(h-168)$$

- L1 and L2 are the previous load values at one and two hours before the forecast hour.
- L3 is the previous day load at the same hour as the forecast day.
- L4 is the previous two-day load at the same hour as the forecast day.
- L5 is the previous week load at the same hour as the forecast day.

In addition to the load values, the network needs inputs indicating the day type and hours of the day.

As it was mentioned before, the day-types are classified into four classes and each class is coded with four binary digits. Therefore, for the day-type representation, the network will take four input nodes. Similarly, the hours of the day is fed to the network with five binary digits, which corresponds to the binary equivalent of the particular hour.

Therefore, the model has 14 input variables and one output variable.

Determining the number of neurons in the hidden layer is more difficult than determining the size of the input or the output layers. Generally, if they are very few, it will not be flexible enough to model the data well; if they are too many, the model will over fit the data. The researcher chose the following numbers by trial and error. The simulation was run with the few alternative numbers (10, 15, and 20) and then found out the one with best prediction performance. (The next section will deal with this matter)

#### 4.2.3.2 <u>Training And Testing The Networks</u>

After the MLP has been designed, it must be trained (i.e. its parameters must be estimated). The training algorithm must first be selected. The most common in use is the back propagation algorithm, based on a steepest descent method that performs stochastic gradient descent on the error surface.

In both the hidden and output units, the sigmoid function is used as the activation function. This function maps the data between 0 and 1. To enable the convergence of training within reasonable time, the desired output values should be scaled on this interval (Haykin, S., 1999).

The proposed training algorithm is known to be iterative; therefore, some criteria should be set up to stop the iterations.

In most of the papers reviewed, training was stopped after a fixed number of iterations (epochs), or after the error decreased below some specified tolerance. These criteria are not adequate, as they insure that the model fits closely to the training data, but do not guarantee good out-of- sample performance, they may lead to over fixing of the model. One way to avoid over fitting is by using cross-validation (see 2.2.4).

The models will be tested for its goodness of fit by examining their testing errors.

Based on the above facts, the data set that contains 1272 samples (input vectors) was partitioned into 3 sets; training, validation and test sets. Since the learning process is supervised, the network was also fed with the desired output.

Before it is imported to the neural network toolbox, it was re-organized in a format suitable for the network. Hence, the final structure of the data set appeared to be as follows.(sample views)

| *L(h)* | *L(h-1)* | *L(h-2)* | *L(h-24)* | *L(h-48)* | *L(h-168)* | *Hour- code* | | | | | *Day-type code* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *0.2414* | *0.3103* | *0.5517* | *0.0000* | *0.2500* | *0.2759* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *1* |
| *0.0690* | *0.2414* | *0.3103* | *0.0357* | *0.0714* | *0.1379* | *0* | *0* | *0* | *0* | *1* | *0* | *0* | *0* | *1* |
| *0.0690* | *0.0690* | *0.2414* | *0.0357* | *0.0357* | *0.0690* | *0* | *0* | *0* | *1* | *0* | *0* | *0* | *0* | *1* |
| *0.0690* | *0.0690* | *0.0690* | *0.0714* | *0.0357* | *0.0690* | *0* | *0* | *0* | *1* | *1* | *0* | *0* | *0* | *1* |
| *0.0690* | *0.0690* | *0.0690* | *0.1429* | *0.0357* | *0.0690* | *0* | *0* | *1* | *0* | *0* | *0* | *0* | *0* | *1* |
| *0.1379* | *0.0690* | *0.0690* | *0.2500* | *0.0714* | *0.1379* | *0* | *0* | *1* | *0* | *1* | *0* | *0* | *0* | *1* |
| *0.2414* | *0.1379* | *0.0690* | *0.3929* | *0.2857* | *0.3103* | *0* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *1* |
| *0.4138* | *0.2414* | *0.1379* | *0.5357* | *0.5000* | *0.4828* | *0* | *0* | *1* | *1* | *1* | *0* | *0* | *0* | *1* |
| *0.5172* | *0.4138* | *0.2414* | *0.6071* | *0.6429* | *0.6207* | *0* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *1* |
| *0.7241* | *0.5172* | *0.4138* | *0.8571* | *0.8214* | *0.7931* | *0* | *1* | *0* | *0* | *1* | *0* | *0* | *0* | *1* |

***Table 4.2 sample view of the hourly load dataset***

All the values in the first column indicate the desired output while columns 2 to 5 are historical loads and the others are hour and day-type codes.

### a. Partitioning the data set

Since neural networks are "data-driven" method, they require large samples in training. In this research work, the total training data size, which is 1272, is assumed to be sufficient for the networks to predict the hourly load forecasts. The partitioning of the data set into the 3 sets was performed as follows.

One month data from 08/05/1996 to 07/06/1996 were used as training set, 8 days data from 08/06/1996 to15/06/1996 as validation sets and the last 15 days from 16/06/1996 to 30/06/1996 for testing purpose.

In general the partitioned data is presented below in table 4.3

| Sets | Inputs date | Size | Targets/desired output/ date | Size |
|---|---|---|---|---|
| Training | 08/05/1996 to 07/06/1996 | 720 | 08/05/1996 to 07/06/1996 | 720 |
| Validation | 08/06/1996 to15/06/1996 | 192 | 08/06/1996 to15/06/1996 | 192 |
| Test | 16/06/1996 to 30/06/1996 | 360 | 16/06/1996 to 30/06/1996 | 360 |

*Table 4.3 Partitioned hourly datasets*

## b. Network Parameter Settings

Once the dataset partitioned, the optimal number of hidden units were obtained in a trail and error process. The training starts with a pre-determined set of hidden units i.e. 10, 15 and 20, with two different epochs limit.

In order to study the influence of the learning rate on the training time, three learning rates with constant momentum were used for each training cases. The training performances obtained in all the cases were compared in order to select the optimum hidden units. Table 4.4 shows the results.

# Network performances of different hidden neuron sizes

| Hidden neurons | Epochs | *Network performance errors /10$^{-2}$/* | | |
|---|---|---|---|---|
| | | *Learning rates /constant momentum=0.5/* | | |
| | | *0.3* | *0.5* | *1.2* |
| *10* | *10000* | *0.4600* | *0.4100* | *0.3800* |
| | *20000* | *0.3900* | *0.4100* | *0.3100* |
| *15* | *10000* | *0.4900* | *0.4000* | *0.3200* |
| | *20000* | *0.3600* | *0.3300* | *0.2500* |
| *20* | *10000* | *0.4600* | *0.4000* | *0.2900* |
| | *20000* | *0.3600* | *0.3400* | *0.2700* |

*Table 4.4 Network performance for varying hidden neurons at different learning rates*

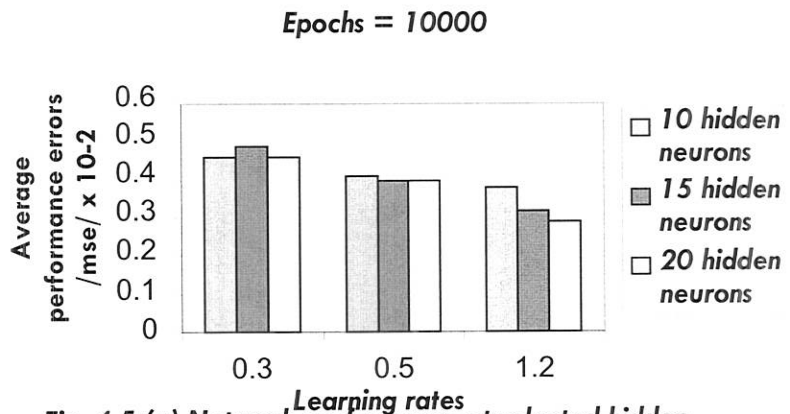



*Fig. 4.5 (a) Network performance at selected hidden neurons and Epochs limit 20000*

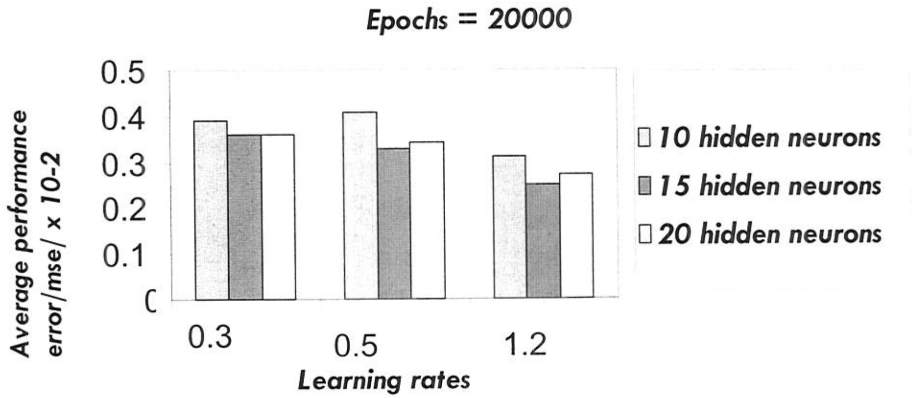



*Fig. 4.5(b) Network performance at selected hidden neurons and Epochs limit 20000*

From table 4.4 and fig 4.5 (a) and (b), it can be seen that the training with 15 hidden units gives in general, better results than the others.

### c. Training and testing

On the basis of the previous results, 15 hidden units were chosen for the network model and successive trainings were performed for different combinations of the learning rate and momentum. The training process stops upon hitting the error tolerance, or reaching the epochs limits. The error tolerance was set to 0.001 in all the training cases. First, the network was trained with 10 different combinations of the learning rate and momentums with fixed epoch limit i.e. 20,000. After finishing the training, a generalization test was performed for each training result to see whether the models are sufficiently general.

| Networks | Learning rate | Momentum | MSE |
|---|---|---|---|
| *1* | *0.01* | *0.9* | *0.0150* |
| *2* | *0.2* | *0.3* | *0.0048* |
| *3* | *0.3* | *0.3* | *0.0042* |
| *4* | *0.5* | *0.3* | *0.0046* |
| *5* | *0.5* | *0.5* | *0.0045* |
| *6* | *0.7* | *0.3* | *0.0038* |
| *7* | *0.8* | *0.5* | *0.0034* |
| *8* | *0.9* | *0.5* | *0.0034* |
| *9* | *1.0* | *0.5* | *0.0031* |
| *10* | *1.2* | *0.3* | *0.0024* |

***Table 4.5 Network Results for hourly load forecast with 15 hidden neurons and various combinations of learning rates and momentums***

The tests were carried out using the test set. The results of the tests are displayed in table 4.5. From the table, it can be seen that the network with 1.2 learning rate and momentum 0.3 was found to be the least generalization error. Next, to the selection process, the network that was evaluated best was again trained with a high epoch limits, which was 30,000. It exhibits a much better result than the previous one.

### 4.2.3.3 Analysis of the results

The performance measure for the effectiveness of the network varies in different literatures. Most of the literatures reported only the Mean Absolute Percentage Errors (MAPE); few also reported the standard deviation of the errors [Alfuhad, A.S, et, al, 1997]. Although MAPE has become somewhat of a standard in the electricity industry, it is clearly not enough to depend only on it. Measures based on squared errors are sometimes suggested, as they penalize large errors. Some papers reported that utilities would rather evaluate forecasting systems by the absolute errors produced and this suggests that the Mean Absolute Errors could be useful [Papadakis, S.E. et. al, 1998].

In any case, error measures are only intended as summaries for the error distribution. The shape of the distribution also suggests the performance of the networks. In this research, three measures were conducted; MAPE (Mean Absolute Percentage Error), MAD (Mean Absolute Deviations) and MSD (Mean Squared Devotions). Furthermore, the distributions of MAPEs were analyzed together with the others.

The mathematical definition of the measures is given by the following formulas [Abranovic, W.A, 1997].

$$\mathbf{MAPE} = \frac{\sum\left|\frac{y_i - \hat{y}_i}{y_i}\right| \mathbf{x100}}{\mathbf{N}} \qquad (4.3)$$

$$\mathbf{MAD} = \frac{\sum|y_i - \hat{y}_i|}{N} \qquad (4.4)$$

$$\mathbf{MSD} = \frac{\sum(y_i - \hat{y}_i)^2}{N} \qquad (4.5)$$

**where, $y_i$ are the desired outputs**

**$\hat{y}_i$ are forecasted values**

**N -number of test examples**

The model is tested using the test dataset and its performance was evaluated by comparing the hourly forecast output with the desired. The result is shown in table 4.6. Similarly the distribution of the absolute errors was also presented in fig. 4.6. and table 4.7

| Measures | Normalized | Un-normalized |
|---|---|---|
| **MAPE** | *** | *5.03%* |
| **MAD** | *0.0376* | *1.0912* |
| **MSD** | *0.0025* | *2.0606* |

***Table 4.6 Selected network error measures***

It shows the percentage absolute errors. It can be seen that 86%of the errors are below 2.001 of absolute forecast errors and the forecast errors with large values (greater than 3.000) constitutes 2.5% only. The forecasted hourly loads are graphically presented with the actual hourly loads in fig 4.7.below.

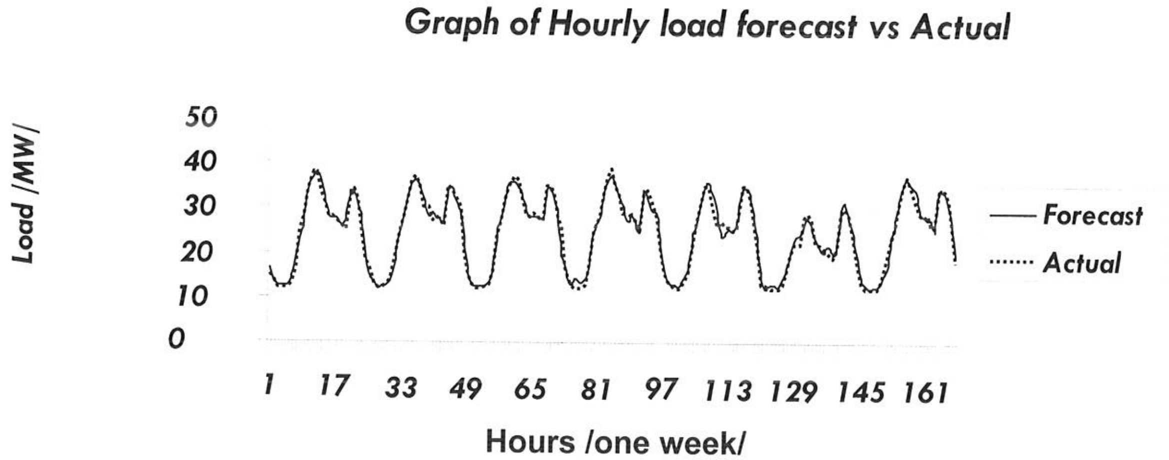



*Fig 4.7 graphical representation of the hourly load forecast vs Actual for the first weeks of the test period*

According to the results in the three measures of the performance errors, the proposed ANN model can be considered as a good model for EEPCO hourly load forecasts.

## 4.3 Model For The Daily Peak Load Forecasts

The research work extends its experiment to design a neural network model for the daily peak load forecast. Several neural network models were trained and tested in order to come up with the one having the smallest performance errors.

The objectives of this experiment is to identify the assumed dependency of the daily peak load on temperature, and include the holiday and seasonal effects

The same procedures, similar to the hourly forecasts, were followed, and hence some discussions are generalized to avoid redundancies.

### 4.3.1 Input Data Collection And Preparation

It has been noted that during the course of the previous works on literature review that the forecast of the daily load mainly depend on historical load and temperature data. Offcourse, there are other factors apart from these that are considered in different research works.

In this research work, as an initial test, only temperature was considered to check its dependency on load patterns. Like the hourly forecast, the calendar effect has also been considered with the additional focus concerning the monthly seasons of the year. The load pattern is expected to have different monthly rhythms in a given year. Therefore, the month-of-year captures the seasonal effects. Holidays are again special days; they tend to produce behavior that is different from weekdays.

Since the aim of this particular research is to forecast the daily peak load, the maximum and minimum temperature of the days under consideration will be tested for their dependency on peak load. The historical daily peak loads are assumed to be auto correlated and hence, all the lag series with relatively high ACF will be considered.

The additional input data related to the temperature were obtained from previously collected data by a student in the same department for his research work on temperature forecasts. The historical peak load data came from the same source as before.

Generally, the daily peak load forecast model will be set up to include the following effects.

- Historical daily peak load
- Historical daily maximum temperature
- Historical daily minimum temperature
- Month-of-year
- Day-type

Based on the above facts, a pre-designed format was used to collect the required data. The question of how much data set is required and which period of time to be included remains challenging here too. But, as to the historical load, which offcourse determines the other variables, the following considerations were taken.

- First, the decision as to how much data set is required for the whole process is related to the possible number of holiday's events to be included in the sample. The more the patterns reflecting this effect in the training set, the better is the network ability to extract their characteristics.

- Secondly, in recent years (2002/2003) as a result of the decrease in the hydroelectric resources, major interruption of electricity supply were seen which led to a planned load shedding throughout the year. After the problem was resolved, and a new generation plant was installed, the full distribution of electric power has been maintained starting from July, 2003.

- To find the seasonal effects, all the months of the year should be included adequately in the sample data.

Based on the above grounds, two years of daily peak load data that include the periods from September 9, 2000 to September 10, 2003 (Meskerem 1, 1993-Pagume 5, 1994) were used for the training process. Moreover, the load data corresponding to the recent periods were collected for further analysis.

All the raw data were initially stored in the MS-Excel as before.

### 4.3.2 The data pre-processing steps

The data to be used in the training process should pass the pre-processing steps in order to improve the quality of the data and prepare as per the requirements of the network and the software in use. The following activities ensure these principles.

#### a. Data cleaning

Here, the data cleaning process were simplified to detecting and correcting outliers only. No missing values were identified. The load data in MW that do not correspond to the column value measured in Ampere (current), were identified as outliers and the correction was done by changing the current into MW. A registration error was also found that wrongly specified the holiday date and adjustment was made by referring to the calendar dates. The total corrections made were insignificant compared to the previous case. Finally, the total number of data sets was found to be 934.

### b. Data Selection

The final dataset that is to be used for the training process should be selected first. To this end, the inputs in relation to the historical daily peak load are to be decided first and the others will be automatically determined. The past load data that was graphically inspected did not explicitly shows the exact trend and hence the ACF was calculated for further analysis. This was accomplished using the statistical software, SPSS. The ACF obtained for the first 10 lags are shown below.

| Lags | ACF |
|---|---|
| 1 | 0.374 |
| 2 | 0.207 |
| 3 | 0.198 |
| 4 | 0.223 |
| 5 | 0.202 |
| 6 | 0.319 |
| 7 | 0.648 |
| 8 | 0.373 |
| 9 | 0.186 |
| 10 | 0.173 |

***Table 4.8 The ACF results for10 lags for the historical peak load datsets***

Unlike the hourly load data, the daily peak load didn't show perfect relationship between sequences of data. But, the existence of weekly peak load dependency seems to be real. To avoid the risk of discarding lagged variables that showed no significant linear correlation in the load, but which were strongly nonlinearly correlated to it, lag one and the corresponding related weekly load are taken as inputs to the network. Therefore, the load input data are decided to be; previous one-day peak load, the peak loads corresponding to the previous $7^{th}$ & $8^{th}$ day from the forecast day. The relationship between the peak load and the corresponding maximum and minimum temperature were analysed using the correlation coefficient, which is given by;

$$r_{yx} = \frac{s_{yx}}{s_y \quad s_x}, \tag{4.6}$$

where, $$s_{yx} = \frac{\sum (y_i - \bar{y}) \quad (x_i - \bar{x})}{(n-1)}, \tag{4.7}$$

$$s_y = \frac{\sum (y_i - \bar{y})^2}{(n-1)} \tag{4.8}$$

$$s_x = \frac{\sum (x_i - \bar{x})^2}{(n-1)} \tag{4.9}$$

The result obtained indicates that the maximum temperature of the day is not linearly correlated with the daily peak load ($r_{yx}$ = -0.04) and the negative value of $r_{yx}$ seems to be unrealistic. Minimum temperature of the day show slightly better correlation than the maximum resulting in positive value ($r_{yx}$ =0.24), but still the value can not be considered as significant. In all the circumstances, the findings are the indications of the existing realities. It was supposed to have some relationship between the temperature and the load since it has been known that the demand rises on cold days because of the use of electric heater and water-heating devices, and on hot days, because of air conditioning. But, the fact that most of the people being categorized as low income groups, those above-mentioned devices are extremely expensive for this dominant group of people. In addition to this, the prevailing climatic conditions of the country didn't exhibit the expected relationships. In most cases in the year, the daily average temperature lies between 20 and 27 $^0$c in Addis Ababa, which is considered as one of the highland regions in the country. However, since the function that relates the temperature to the load is assumed to be nonlinear [Hagan, M.T, and Behr, S.M, 1987] and have U shaped pattern, further analysis was conducted by looking at the distribution of sample values of temperature with respect to loads. Here also, it doesn't exhibit the required relationships. Finally the researcher chooses to exclude the temperature data and process with the remaining effects.

### c. **Data transformation**

The selected data were transformed according to the requirements of the network and the previously proposed ideas. Three categorical variables that correspond to the month-of-year, day-type and holidays were coded using the binary bits 0 and 1. The months were code by using four binary digits. The classification of the day-types remains the same as before except the coding method differs from the *1-of-c* technique. Here, holidays are considered as part of the day types forming the fifth group in the classification. As mentioned earlier, three holiday events were selected that were found to be highly influencing the load. The influence of the New Year and Easter holidays appear on the eve of the days, while on the holiday, it tends to drop fast. Unlike the others, the load tends to increase on the Christmas day. The holidays are grouped with Sundays, while the eve days and Christmas day as the fifth additional group.

In general, the result of the classification is presented as follows:

- ✓ Group 1 :- Sunday, new-year day, Easter day
- ✓ Group 2 :- Monday, Tuesday, Wednesday and Thursday
- ✓ Group 3 :- Friday
- ✓ Group 4 :- Saturday
- ✓ Group 5 :- New-year eve, Easter eve and Christmas day, and

Corresponding codes are indicated below.

- Group 1 001
- Group 2 010
- Group 3 011
- Group 4 100
- Group 5 101

Next, the load data was normalized using the min-max technique and the complete set of data was stored in the Ms-Excel format.

### 4.3.3 The network design process

The network design process for the daily peak load forecast starts in the same way as the hourly load forecast.

#### 4.3.3.1 Description of the MLP network

Three layers fully connected feed forward ANN was implemented. The structure of the model is presented in fig 4.8. The model has one output node L (d) corresponding to the peak load forecast at day d. The inputs consists of 10 units (nodes), where the first three nodes represent the historical peak load at day d-1, d-7, and d-8, the next four nodes are the bit codes for the month-of-year and finally three nodes for the bit codes indicating the day type and holiday events.

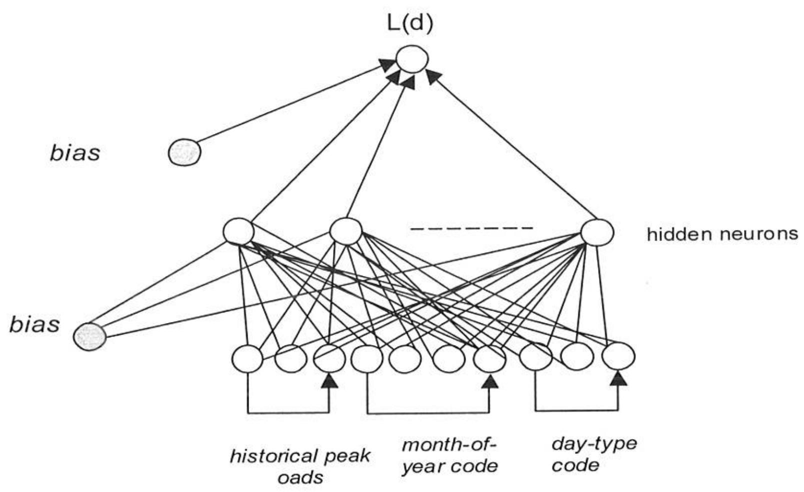



***Fig 4.8 structure of daily peak load model***

The determination of number of hidden neurons was based on trail and errors as before.

### 4.3.3.2 Training and testing the network

Similar training algorithm was selected as before and the activation function at both the hidden and output nodes was chosen to be the sigmoid function. The cross-validation technique was also applied to overcome the over fitting problem and separate test dataset was used to evaluate the network's goodness-of-fit. The dataset contains 722 examples. After the final arrangement, all the data were stored in the Excel sheet as follows:

| *L(d)* | *L(d-1)* | *L(d-7)* | *L(d-8)* | *Monthly code* | | | | *Day type code* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *0.5000* | *0.4333* | *0.2000* | *0.0667* | *0* | *0* | *0* | *1* | *0* | *1* | *0* |
| *0.5333* | *0.5000* | *0.4000* | *0.2000* | *0* | *0* | *0* | *1* | *0* | *1* | *0* |
| *0.5000* | *0.5333* | *0.5167* | *0.4000* | *0* | *0* | *0* | *1* | *0* | *1* | *0* |
| *0.5000* | *0.5000* | *0.4667* | *0.5167* | *0* | *0* | *0* | *1* | *0* | *1* | *1* |
| *0.4333* | *0.5000* | *0.4000* | *0.4667* | *0* | *0* | *0* | *1* | *1* | *0* | *0* |
| *0.1667* | *0.4333* | *0.5667* | *0.4000* | *0* | *0* | *0* | *1* | *0* | *0* | *1* |
| *0.4000* | *0.1667* | *0.4333* | *0.5667* | *0* | *0* | *0* | *1* | *0* | *1* | *0* |
| *0.5000* | *0.4000* | *0.5000* | *0.4333* | *0* | *0* | *0* | *1* | *0* | *1* | *0* |

***Table 4.9 sample peak load dataset***

#### a) Partitioning the dataset

The first column that indicates the peak load at day d is the desired output, while the rest are the input vectors. The training inputs were taken to accommodate the whole two year data and the recent year data was left for the other two sets i.e. validation and test. The result of the partitioning is shown below.

| Sets | Inputs date | Size | Targets/desired output/ date | Size |
|---|---|---|---|---|
| Training | 09/01/1993 to 05/13/1994 | 722 | 09/01/1993 to 05/13/1994 | 722 |
| Validation | 01/01/1996 to 30/02/1996 | 60 | 01/01/1996 to 30/02/1996 | 60 |
| Test | 01/03/1996 to 30/04/1996 | 60 | 01/03/1996 to 30/04/1996 | 60 |

***Table 4.10 peak Load original partitioning set***

### b) Network parameter setup

The main activity that further remains is to set the optimum number of hidden units. This is a trial and error process. By starting from a small number of hidden units and gradually increasing the number until the minimum training error is achieved, the required number of hidden units can be identified. The different combinations of the network parameters and the error results are shown below.

| *Networks* | *Learning rate* | *Momentum* | *Hidden neuron* | *Epochs* | *MSE* |
|---|---|---|---|---|---|
| *1* | *0.01* | *0.9* | *10* | *Stops at 142* | *0.069* |
| *2* | *0.3* | *0.2* | *10* | *Stops at 6* | *0.117* |
| *3* | *1.2* | *0.3* | *10* | *Stops at 6* | *0.059* |
| *4* | *0.01* | *0.9* | *15* | *20000* | *0.029* |
| *5* | *0.3* | *0.2* | *15* | *Stops at 12* | *0.05* |
| *6* | *1.2* | *0.3* | *15* | *Stops at 12* | *0.05* |
| *7* | *0.01* | *0.9* | *20* | *Stops at 6* | *0.1* |
| *8* | *0.3* | *0.2* | *20* | *12515* | *0.0159* |
| *9* | *1.2* | *0.3* | *20* | *Stops at 9* | *0.107* |
| *10* | *0.01* | *0.9* | *25* | *20000* | *0.0168* |
| *11* | *0.3* | *0.2* | *25* | *Stops at 6* | *0.146* |
| *12* | *1.2* | *0.3* | *25* | *Stops at 6* | *0.066* |

***Table 4.11 Initial network training results***

The training results do not indicate a good performance with the different combinations. In most cases, the training stops at the beginning due to increase in error on validation. It can be seen that the performance of the training is not even better for the rest of the network structures.

Therefore, deciding the number of hidden units on those results obviously will lead to improper architecture of the network, and hence the researcher chose to further analyse the original grounds considered in the design process.

The first consideration was made on the selection of the datasets. Since the training datasets include the two years of data while the validation set consist of the recent data after the shedding problem resolved and a new generation plant enters the system, there may be unexplained patterns in the training sets. The statistical analysis on the Pearson correlation coefficient (2-tailed test at $\alpha$ =0.01) shows a significant correlation between the 1993 and 1994 data only, which indicates that there is a high variability between training and validation dataset. Despite this, the performance did not improve when adding some of the new dataset into the training dataset, which may be due to the small size of the added data.

Therefore the next measures taken by the researcher is to only consider the previous year's sets of data.

One year data from 09/01/1993 to 08/01/1994 is used for the training input and the remaining for validation and testing. The new partitioning sets are;

| Sets | Inputs date | Size | Targets/desired output/ date | Size |
|---|---|---|---|---|
| Training | 09/01/1993 to 08/01/1994 | 365 | 09/01/1996 to 08/01/1993 | 365 |
| Validation | 09/01/1994 to 08/09/1994 | 240 | 09/01/1994 to 08/09/1994 | 240 |
| Test | 09/09/1994 to 05/13/1994 | 117 | 09/09/1994 to 05/13/1994 | 117 |

***Table 4.12 New Peak load Partitioning set***

The above changes in the dataset showed a remarkable increase in the training performance. The new result is shown below.

| *Networks* | *Learning rate* | *Momentum* | *Hidden neuron* | *Epochs* | *MSE* |
|---|---|---|---|---|---|
| *1* | *1.2* | *0.3* | *15* | *20000* | *0.0084* |
| *2* | *1.2* | *0.3* | *20* | *20000* | *0.0078* |
| *3* | *0.3* | *0.8* | *25* | *20000* | *0.0049* |
| *4* | *0.5* | *0.3* | *25* | *20000* | *0.0049* |
| *5* | *1.0* | *0.7* | *30* | *20000* | *0.005* |
| *6* | *1.2* | *0.3* | *30* | *20000* | *0.0048* |

***Table 4.13 Adjusted network training results***

It can be seen that the performance is better when the number of hidden units is increasing. Therefore, the training and testing process was continued by varying the number of hidden units to 15, 20, and 25.

c) **Training and testing**

The network that can generalize well was determined on the selected different sizes of the hidden units by testing the performance on the test dataset. The network was trained along with different combinations of learning rate and momentum. The corresponding testing performance is shown below in table 4.14.

| Networks | Learning rate | Momentum | Hidden neuron | Epochs | MSE |
|---|---|---|---|---|---|
| 1 | 0.2 | 0.3 | 15 | 20000 | 0.0040 |
| 2 | 0.5 | 0.7 | 15 | 20000 | 0.0038 |
| 3 | 0.6 | 0.3 | 15 | 20000 | 0.0037 |
| 4 | 0.8 | 0.4 | 15 | 20000 | 0.0037 |
| 5 | 0.3 | 0.7 | 20 | 20000 | 0.0042 |
| 6 | 0.8 | 0.4 | 20 | 20000 | 0.0037 |
| 7 | 0.5 | 0.9 | 25 | 20000 | 0.0036 |

***Table 4.14 Results of trained networks for the selected model***

#### 4.3.3.3 Analysis of the test results

In general, as can be seen from the table, the average percentage error decreases as the number of hidden unit increase. Based on this the architecture was fixed with 25 number of hidden neurons along with

0.5 as learning rate and 0.9 as momentum and its results were promising as shown below in table 4.15.

| Measures | Normalized | Unnormalized |
|---|---|---|
| MAPE | *** | 3.73% |
| MAD | 0.0446 | 1.3390 |
| MSD | 0.0036 | 3.2630 |

*Table 4.15 Selected network error measures of the peak load forecasts*

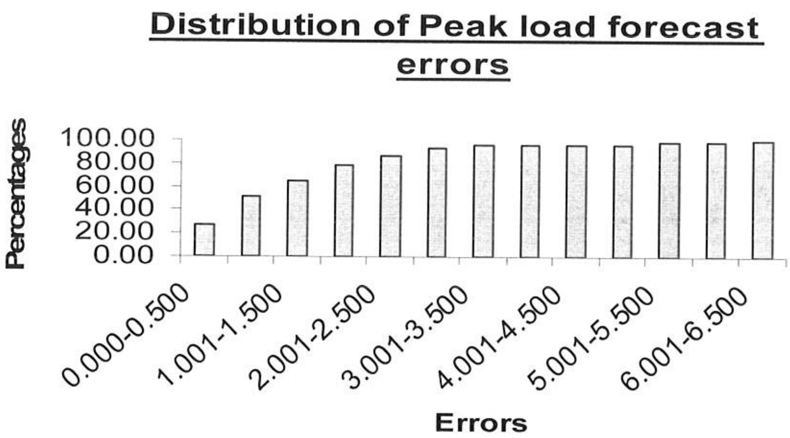



*Fig. 4.9 Percentage distribution of the mean absolute error of the daily peak load forecast*

**Absolute Error Distribution of selected Peak load forecast Model**

| Error Range | Frequency | cumm. Frequency | Relative frequency |
|---|---|---|---|
| 0.000-0.500 | 32 | 32 | 27.35 |
| 0.501-1.000 | 28 | 60 | 51.28 |
| 1.001-1.500 | 16 | 76 | 64.96 |
| 1.501-2.000 | 15 | 91 | 77.78 |
| 2.001-2.500 | 10 | 101 | 86.32 |
| 2.501-3.000 | 8 | 109 | 93.16 |
| 3.001-3.500 | 3 | 112 | 95.73 |
| 3.501-4.000 | 0 | 112 | 95.73 |
| 4.001-4.500 | 0 | 112 | 95.73 |
| 4.501-5.000 | 1 | 113 | 96.58 |
| 5.001-5.500 | 2 | 115 | 98.29 |
| 5.501-6.000 | 0 | 115 | 98.29 |
| 6.001-6.500 | 2 | 117 | 100.00 |
| | 117 | | |

***Table 4.16 Distribution of the absolute errors of the peak load***

The distribution of the Absolute errors is shown in table 4.16 and in fig 4.9. It can be seen that, 93% of the errors are less than 2.5%. The forecast values that exhibit large errors were insignificant (less than 3% accounts for errors greater than 5) and it is suspected that improper registration of the readings might have contributed to these errors. The test result of the selected network with the desired outputs was also graphically presented in fig4.10.

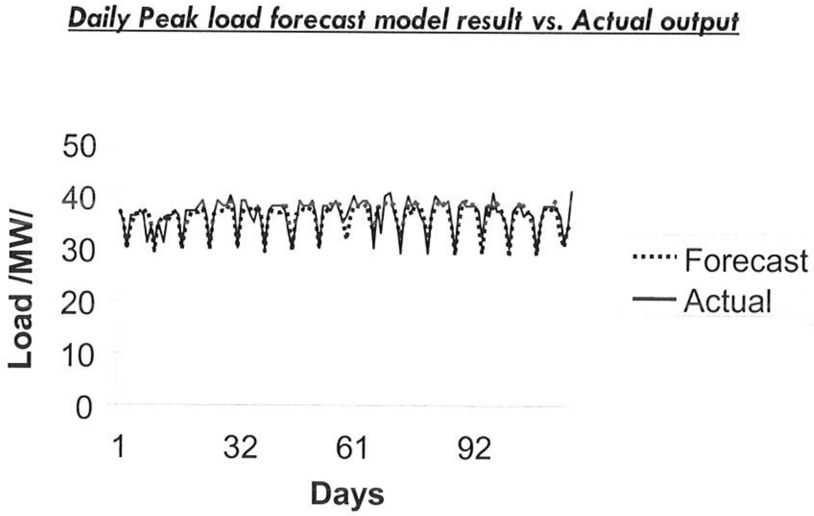



*Fig 4.10 forecasted daily peak load vs. actual*

# CHAPTER FIVE

# CONCLUSIONS AND RECOMMENDATIONS

## 5.1 Conclusions

Neural networks can learn to approximate any function and can behave like associative memories by using just examples of data that are representatives of the desired task. They are model free estimators and are capable of solving complex problems based on presentation of a large number of training data. This gives them a key advantage over traditional approaches such as statistical methods. Neural networks estimate a function without a mathematical description of how the outputs functionally depend on the inputs - they represent a good approach that is potentially robust and fault tolerant.

In this thesis, the properties of the feed-forward neural networks and the process of determining the appropriate network inputs and architectures are examined, and built a short-term load forecast models. Several network models were studied in the work. The techniques were divided into two classes; models for hourly load forecasts that utilize historical load data and models for the daily peak load forecast, which takes into account the seasonal effects.

The methodology employed follows three basic steps in general: data collection and analysis, data pre-processing and model building.

For the one hour-ahead load forecast experiment an average forecast error of 5.03 % was obtained. The distribution of the absolute errors (the absolute difference between the actual and forecasted values) indicate that of the total test sets (360), which is used for generalization purpose, 310 forecasts (or above 86%) made by the selected neural network model exhibits below 2.000 absolute errors when they are compared with their respective actual values. On the other hand, the forecasts that exceed an absolute error of 4.000 are found to be only 5 patterns (or less than 2 %).

Similarly, the selected model for the daily peak load forecast shows an average percentage error of 3.73%. Here, the distributions of the absolute errors indicate that out of the 117 test sets used for generalization purpose, the neural network model forecasts a total of 101 patterns (above 93%) with less than 3.000 absolute errors. Likewise, 5 forecasts made by the selected model show an absolute error greater than 4.000.

During the model building, the researcher observed that the nature of neural network model for load prediction require going through a number of trail and errors so that much improvement can be achieved in the process. The selection of appropriate inputs relevant to the application and selecting the representative samples are the decisive part of the process. Moreover, the data preparation step should be considered as an important activity in the general process.

In this case, much attention was given to the data preparation and analysis steps as a result; encouraging results were found in all the cases.

It is important to note that, the model building process for the prediction of future load using neural networks is not a one-time job. Successive adjustments to the designed model, taking into account of the newly arrival input data, should be considered at large, in order to improve the forecast accuracies.

## 5.2 Recommendations

On the basis of the findings of this research work and the various reviewed literatures, the researcher would like to make the following recommendations in relation to artificial networks in the load forecast problems.

The researcher would also like to note that this research, as an academic exercise, should only be considered as a preliminary effort to assess the applicability of neural networks in load forecasting for EEPCO system. Accordingly, the findings of this research can fairly be considered as a contribution towards a more in depth and comprehensive study in the area. The following areas can be taken as a future research work that helps to initiate more research activities for others.

- ✓ **Comparative analysis of load forecasting**

  A method is considered to have been properly validated if; I) its performance is comparable to that of well-accepted methods; ii) the comparison is based on the performance on different test samples; iii) the size of the training samples is adequate, so that some inference might be drawn. In this regard, the findings of this research work can be compared with the statistical forecasting techniques like ARMAX or regression models. It is also possible to consider the load forecasting method applied in this work with a recent and large sample data.

✓ **Other types of neural network architectures**

In this research only one output node is considered in the two NN models. Forecasting the load profile for the whole day can be considered further with similar procedures. Typically, to forecast the next day's 24 hours load can be developed using neural network techniques.

✓ **Other artificial neural networks**

There are different artificial intelligence techniques employed in the load forecast problem. This includes:

- Fuzzy logic in combination with neural networks were found to provide a good forecasting results,
- Kohenen self-organizing map (SOM) that uses unsupervised learning process was used in various research works. The performances were found to be promising. It also helps in classifying the daily load profiles for improving the accuracy of the forecast in the models,

- Expert systems were proposed in the forecasting area. A typical approach is to try imitating the reasoning of human operator. The idea is then to reduce the analogical thinking behind the intuitive forecasting to formal steps of logic. A possible method to create the forecast is to search in history database for a day that corresponds to the target day with regard to the day type, social factors and weather factors. Then the load values of this similar day can be taken as the basis for the forecast.

  On the other hand, the expert systems can consist of a rule base defining relationships between external factors and daily load shapes.

# REFERENCES


Abdi, H. Valentin, D. and Edelman, B. "Neural networks-Series: Quantitative applications in the social science". *The international publishers, Thousands Oaks, London, New Delhi; SAGE publications, 1999.*

Abranovic, W.A., "Statistical thinking and data analysis methods for managers". *1st ed. Addisson-Wesley Educational publishers, Inc., 1997.*

Acers, *"Power study (PPS)-EEPCO", 1982.*

Acers, *"Power study (PPS)-EEPCO", 1996.*

Alfuhaid, A.S., et al. "Cascaded artificial neural networks for short-term load forecasting". *IEEE Trans. on power systems, vol. 12, no. 4, Nov. 1997, and pp. 1524-1529.*

Anderson, D. and McNeil, G. "Artificial neural networks Technology. Data analysis centre for software", *Rome, NY. Rome laboratory, 1992.*

Bakirtzis, A.G., et al. "A neural network short term load forecasting model for Greek power system". *IEEE. Trans. on power systems, vol. 11, no. 2, May, 1996, pp. 858-863.*

Bakirtzis, A.G., et al. "Short-term load forecasting using fuzzy neural networks". *IEEE Trans. on power systems, vol. 10, no. 3, 1995, pp. 1518-1524.*

Bao, J., "Short-term load forecasting based on neural network and moving average".URL:http://www.public.iastate.edu/~baojie/pub/2002-05-8_stlf.pdf

Becker, S., "Unsupervised learning procedures for neural networks: *International journal of neural systems, vol. 2, no. 1& 2, 1991, pp. 17-33.*

Chatfield, C., "The analysis of time series: An introduction". *Fifth ed. London Chapman & HALL/CRC, 1996.*

Chen, S., et al. "Weather sensitive short-term load forecasting using non-fully connected artificial neural network". *IEEE Trans. on power systems, vol. 7, no. 3, 1992, pp. 1098-1105.*

Chow, T.W.S. and Leung, C.T. "Neural network based short-term load forecasting using weather compensation". *IEEE Trans. on power systems, vol. 11, no. 4, Nov. 1996, and pp. 1736-1741.*

Damborg, M.T, et al. « Potential of artificial neural networks to power system operation". *Proceeding of the 1990 IEEE international symposium on circuit and systems. New Orleans, Louisiana, May 1-3, pp. 2933-2937.*

Daneshdoost, et al. "Neural network with fuzzy set-based classification for short-term load forecasting". *IEEE. Trans. on power systems, vol. 13, no. 4, Nov. 1988, pp. 1387-1391.*

Dash, P.K., et al. *"*A Real time short-term load forecasting system using Functional Link network*". IEEE, Trans. on power systems, vol. 12, no. 2, May, 1987, pp. 675-680.*

Engle, et al. "Modelling peak electricity demand". *J. forecast, vol. 11, no. 2, 1994, pp. 241-251.*

Fauset, L. "Foundation of neural networks". *New Jersey, prentice-Hall, 1994.*

Gavrillas, M. "Neural network based forecasting for electricity markets".

URL: http://www.telmark.org/2002Sep/3-6_Gavrilas.pdf

Gross, G. and Galliana, F.D. "Short-term load forecasting"., *Proceedings of the IEEE, vol. 75, no. 12, Dec. 1987, pp. 1558-1573.*

Hippart, H.S., et al. "Neural networks for short-term load forecasting: A review and evaluation". *IEEE Trans. on power systems, vol. 16, 2001, pp. 44-55.*

Ho, K.L., et al. "Short term load forecasting using a multilayer neural network with an adaptive learning algorithm". *IEEE Trans. on power systems, vol. 7, no, 1, Feb. 1992, pp. 141-149.*

Hyde, O and Hodnett, P.F. "An adaptive automated procedure for short-term electricity forecasting". *IEEE, Trans. on power systems, vol. 17, no. 2, Feb. 1997, pp. 48.*

Hykin, S. "Neural networks- A comprehensive foundation". *2nd. ed. Upper saddle River, NJ; prentice Hall, 1999.*

Khotanzad, A., et al. "An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities". *IEEE Trans. on power systems, vol. 10, no.3, Aug. 1995, pp. 1716-1722.*

Khotanzad, A., et al. "ANNSTLF A neural network based electric load forecasting". *IEEE Trans. Neural nets, vol. 8, no. 4, July 1997, pp. 835-846.*

Khotanzad, A., et al. "ANNSTLF-Artificial neural network short-term load forecaster-generation Three". *IEEE Trans. on power systems, vol. 13, no. 4, Nov. 1998, and pp. 1413-1421.*

Kores, B. and Vander Smagt, P. "An introduction to neural networks". *8th ed., 1996.*

Lamedica, R. et al. "A neural network based Technique for short-term forecasting of anomalous load periods". *IEEE. Trans power systems, vol. 11, no. 4, Nov. 1996, pp. 1749-1755.*

Mitchell, J.M. "Machine learning". *McGraw-Hill companies, Inc., 1997.*

Papadakis, S.E., et al., "A novel approach to short-term load forecasting using fuzzy neural networks". *IEEE Trans. on power systems, vol. 13, no. 2, 1998, pp. 480-492.*

Papalaxopoulos, A.D., et al. "An implementation of a neural network based load forecasting model for the EMS". *IEEE Trans. on power systems, vol. 9, no. 4, 1994, pp. 442-449.*

Park, D.C., et al. "Electric load forecasting using artificial neural network". *IEEE Trans. on power systems, vol. 6, no. 2, May 1991, pp. 442-449.*

Sarles, W.S. "Neural network FAQ, Periodic posting to the Usenet news group comp.ai.neural-net". URL:ftp//ftp.sas.com/pub/neural/FAQ.html, 1997.

Senjyu, et al. "One-hour-ahead load forecasting using neural network". *IEEE Trans. on power systems, vol. 17, no. 1, Feb. 2002, pp. 113-118.*

Srinnivasan, D., et al., "Demand forecasting using fuzzy neural computation, with emphasis on weekend and public holiday". *IEEE Tran on power systems, vol. 10, no. 4, Nov., 1995, pp. 1897-1903.*

Vemuri, et al. "On-line algorithm for forecasting hourly loads of an electric utility". *IEEE. Trans. vol. PAS-100, no. 8, Aug. 1981, pp.3775-3784.*

Vermaak, J. and Bothe, E.C. "Recurrent neural networks for short-term load forecasting". *IEEE. Trans. on power systems, vol. 13, no. 1, Feb. 1998, pp. 126-131.*

Yang, H.T., et al. "Identification of ARMAX model for short-term load forecasting: An Evolutionary programming Approach". *IEEE. Trans. on power systems, vol. 11, no. 1, Feb. 1996, pp. 403-408.*

Yoo, H. and Pimmel, R.L. "Short-term load forecasting using a self-supervised adaptive neural network". *IEEE. Power systems, vol. 14, no. 2, May 1999, and pp. 729-785.*

# ANNEXES

## Annex 1: Sample dataset of the original hourly load

| Initial input variables /original/ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| day | day | hour | day- type | load (h) | load (h-1) | load (h-2) | load (h-3) | load (h-4) | load (h-24) | load (h-48) | load (h-168) |
| 1996-05-01 | 2004-01-10 | 00 | Saturday | 18 | | | | | | | |
| 1996-05-01 | 2004-01-10 | 01 | Saturday | 14 | 18 | | | | | | |
| 1996-05-01 | 2004-01-10 | 02 | Saturday | 12 | 14 | 18 | | | | | |
| 1996-05-01 | 2004-01-10 | 03 | Saturday | 12 | 12 | 14 | 18 | | | | |
| 1996-05-01 | 2004-01-10 | 04 | Saturday | 12 | 12 | 12 | 14 | 18 | | | |
| 1996-05-01 | 2004-01-10 | 05 | Saturday | 14 | 12 | 12 | 12 | 14 | | | |
| 1996-05-01 | 2004-01-10 | 06 | Saturday | 19 | 14 | 12 | 12 | 12 | | | |
| 1996-05-01 | 2004-01-10 | 07 | Saturday | 24 | 19 | 14 | 12 | 12 | | | |
| 1996-05-01 | 2004-01-10 | 08 | Saturday | 28 | 24 | 19 | 14 | 12 | | | |
| 1996-05-01 | 2004-01-10 | 09 | Saturday | 33 | 28 | 24 | 19 | 14 | | | |
| 1996-05-01 | 2004-01-10 | 10 | Saturday | 36 | 33 | 28 | 24 | 19 | | | |
| 1996-05-01 | 2004-01-10 | 11 | Saturday | 38 | 36 | 33 | 28 | 24 | | | |
| 1996-05-01 | 2004-01-10 | 12 | Saturday | 35 | 38 | 36 | 33 | 28 | | | |
| 1996-05-01 | 2004-01-10 | 13 | Saturday | 30 | 35 | 38 | 36 | 33 | | | |
| 1996-05-01 | 2004-01-10 | 14 | Saturday | 28 | 30 | 35 | 38 | 36 | | | |
| 1996-05-01 | 2004-01-10 | 15 | Saturday | 26 | 28 | 30 | 35 | 38 | | | |
| 1996-05-01 | 2004-01-10 | 16 | Saturday | 25 | 26 | 28 | 30 | 35 | | | |
| 1996-05-01 | 2004-01-10 | 17 | Saturday | 25 | 25 | 26 | 28 | 30 | | | |
| 1996-05-01 | 2004-01-10 | 18 | Saturday | 26 | 25 | 25 | 26 | 28 | | | |
| 1996-05-01 | 2004-01-10 | 19 | Saturday | 35 | 26 | 25 | 25 | 26 | | | |
| 1996-05-01 | 2004-01-10 | 20 | Saturday | 34 | 35 | 26 | 25 | 25 | | | |
| 1996-05-01 | 2004-01-10 | 21 | Saturday | 33 | 34 | 35 | 26 | 25 | | | |
| 1996-05-01 | 2004-01-10 | 22 | Saturday | 28 | 33 | 34 | 35 | 26 | | | |
| 1996-05-01 | 2004-01-10 | 23 | Saturday | 23 | 28 | 33 | 34 | 35 | | | |
| 1996-05-02 | 2004-01-11 | 00 | Sunday | 16 | 23 | 28 | 33 | 34 | 18 | | |
| 1996-05-02 | 2004-01-11 | 01 | Sunday | 15 | 16 | 23 | 28 | 33 | 14 | | |
| 1996-05-02 | 2004-01-11 | 02 | Sunday | 12 | 15 | 16 | 23 | 28 | 12 | | |
| 1996-05-02 | 2004-01-11 | 03 | Sunday | 12 | 12 | 15 | 16 | 23 | 12 | | |
| 1996-05-02 | 2004-01-11 | 04 | Sunday | 10 | 12 | 12 | 15 | 16 | 12 | | |
| 1996-05-02 | 2004-01-11 | 05 | Sunday | 13 | 10 | 12 | 12 | 15 | 14 | | |
| 1996-05-02 | 2004-01-11 | 06 | Sunday | 15 | 13 | 10 | 12 | 12 | 19 | | |
| 1996-05-02 | 2004-01-11 | 07 | Sunday | 19 | 15 | 13 | 10 | 12 | 24 | | |

| Initial input variables /original/ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| day | day | hour | day- type | load (h) | load (h-1) | load (h-2) | load (h-3) | load (h-4) | load (h-24) | load (h-48) | load (h-168) |
| 1996-05-02 | 2004-01-11 | 08 | Sunday | 22 | 19 | 15 | 13 | 10 | 28 | | |
| 1996-05-02 | 2004-01-11 | 09 | Sunday | 25 | 22 | 19 | 15 | 13 | 33 | | |
| 1996-05-02 | 2004-01-11 | 10 | Sunday | 23 | 25 | 22 | 19 | 15 | 36 | | |
| 1996-05-02 | 2004-01-11 | 11 | Sunday | 24 | 23 | 25 | 22 | 19 | 38 | | |
| 1996-05-02 | 2004-01-11 | 12 | Sunday | 28 | 24 | 23 | 25 | 22 | 35 | | |
| 1996-05-02 | 2004-01-11 | 13 | Sunday | 25 | 28 | 24 | 23 | 25 | 30 | | |
| 1996-05-02 | 2004-01-11 | 14 | Sunday | 22 | 25 | 28 | 24 | 23 | 28 | | |
| 1996-05-02 | 2004-01-11 | 15 | Sunday | 20 | 22 | 25 | 28 | 24 | 26 | | |
| 1996-05-02 | 2004-01-11 | 16 | Sunday | 20 | 20 | 22 | 25 | 28 | 25 | | |
| 1996-05-02 | 2004-01-11 | 17 | Sunday | 21 | 20 | 20 | 22 | 25 | 25 | | |
| 1996-05-02 | 2004-01-11 | 18 | Sunday | 30 | 21 | 20 | 20 | 22 | 26 | | |
| 1996-05-02 | 2004-01-11 | 19 | Sunday | 35 | 30 | 21 | 20 | 20 | 35 | | |
| 1996-05-02 | 2004-01-11 | 20 | Sunday | 30 | 35 | 30 | 21 | 20 | 34 | | |
| 1996-05-02 | 2004-01-11 | 21 | Sunday | 24 | 30 | 35 | 30 | 21 | 33 | | |
| 1996-05-02 | 2004-01-11 | 22 | Sunday | 18 | 24 | 30 | 35 | 30 | 28 | | |
| 1996-05-02 | 2004-01-11 | 23 | Sunday | 16 | 18 | 24 | 30 | 35 | 23 | | |
| 1996-05-03 | 2004-01-12 | 00 | Monday | 14 | 16 | 18 | 24 | 30 | 16 | 18 | |
| 1996-05-03 | 2004-01-12 | 01 | Monday | 10 | 14 | 16 | 18 | 24 | 15 | 14 | |
| 1996-05-03 | 2004-01-12 | 02 | Monday | 10 | 10 | 14 | 16 | 18 | 12 | 12 | |
| 1996-05-03 | 2004-01-12 | 03 | Monday | 10 | 10 | 10 | 14 | 16 | 12 | 12 | |
| 1996-05-03 | 2004-01-12 | 04 | Monday | 10 | 10 | 10 | 10 | 14 | 10 | 12 | |
| 1996-05-03 | 2004-01-12 | 05 | Monday | 12 | 10 | 10 | 10 | 10 | 13 | 14 | |
| 1996-05-03 | 2004-01-12 | 06 | Monday | 18 | 12 | 10 | 10 | 10 | 15 | 19 | |
| 1996-05-03 | 2004-01-12 | 07 | Monday | 24 | 18 | 12 | 10 | 10 | 19 | 24 | |
| 1996-05-03 | 2004-01-12 | 08 | Monday | 26 | 24 | 18 | 12 | 10 | 22 | 28 | |
| 1996-05-03 | 2004-01-12 | 09 | Monday | 23 | 26 | 24 | 18 | 12 | 25 | 33 | |
| 1996-05-03 | 2004-01-12 | 10 | Monday | 34 | 23 | 26 | 24 | 18 | 23 | 36 | |
| 1996-05-03 | 2004-01-12 | 11 | Monday | 35 | 34 | 23 | 26 | 24 | 24 | 38 | |
| 1996-05-03 | 2004-01-12 | 12 | Monday | 34 | 35 | 34 | 23 | 26 | 28 | 35 | |
| 1996-05-03 | 2004-01-12 | 13 | Monday | 29 | 34 | 35 | 34 | 23 | 25 | 30 | |
| 1996-05-03 | 2004-01-12 | 14 | Monday | 29 | 29 | 34 | 35 | 34 | 22 | 28 | |
| 1996-05-03 | 2004-01-12 | 15 | Monday | 29 | 29 | 29 | 34 | 35 | 20 | 26 | |
| 1996-05-03 | 2004-01-12 | 16 | Monday | 29 | 29 | 29 | 29 | 34 | 20 | 25 | |
| 1996-05-03 | 2004-01-12 | 17 | Monday | 27 | 29 | 29 | 29 | 29 | 21 | 25 | |
| 1996-05-03 | 2004-01-12 | 18 | Monday | 26 | 27 | 29 | 29 | 29 | 30 | 26 | |
| 1996-05-03 | 2004-01-12 | 19 | Monday | 35 | 26 | 27 | 29 | 29 | 35 | 35 | |
| 1996-05-03 | 2004-01-12 | 20 | Monday | 34 | 35 | 26 | 27 | 29 | 30 | 34 | |
| 1996-05-03 | 2004-01-12 | 21 | Monday | 33 | 34 | 35 | 26 | 27 | 24 | 33 | |
| 1996-05-03 | 2004-01-12 | 22 | Monday | 30 | 33 | 34 | 35 | 26 | 18 | 28 | |

| Initial input variables /original/ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| day | day | hour | day- type | load (h) | load (h-1) | load (h-2) | load (h-3) | load (h-4) | load (h-24) | load (h-48) | load (h-168) |
| 1996-05-03 | 2004-01-12 | 23 | Monday | 20 | 30 | 33 | 34 | 35 | 16 | 23 | |
| 1996-05-04 | 2004-01-13 | 00 | Tuesday | 15 | 20 | 30 | 33 | 34 | 14 | 16 | |
| 1996-05-04 | 2004-01-13 | 01 | Tuesday | 13 | 15 | 20 | 30 | 33 | 10 | 15 | |
| 1996-05-04 | 2004-01-13 | 02 | Tuesday | 13 | 13 | 15 | 20 | 30 | 10 | 12 | |
| 1996-05-04 | 2004-01-13 | 03 | Tuesday | 13 | 13 | 13 | 15 | 20 | 10 | 12 | |
| 1996-05-04 | 2004-01-13 | 04 | Tuesday | 13 | 13 | 13 | 13 | 15 | 10 | 10 | |
| 1996-05-04 | 2004-01-13 | 05 | Tuesday | 15 | 13 | 13 | 13 | 13 | 12 | 13 | |
| 1996-05-04 | 2004-01-13 | 06 | Tuesday | 28 | 15 | 13 | 13 | 13 | 18 | 15 | |
| 1996-05-04 | 2004-01-13 | 07 | Tuesday | 23 | 28 | 15 | 13 | 13 | 24 | 19 | |
| 1996-05-04 | 2004-01-13 | 08 | Tuesday | 27 | 23 | 28 | 15 | 13 | 26 | 22 | |
| 1996-05-04 | 2004-01-13 | 09 | Tuesday | 23 | 27 | 23 | 28 | 15 | 23 | 25 | |
| 1996-05-04 | 2004-01-13 | 10 | Tuesday | 35 | 23 | 27 | 23 | 28 | 34 | 23 | |
| 1996-05-04 | 2004-01-13 | 11 | Tuesday | 38 | 35 | 23 | 27 | 23 | 35 | 24 | |
| 1996-05-04 | 2004-01-13 | 12 | Tuesday | 37 | 38 | 35 | 23 | 27 | 34 | 28 | |
| 1996-05-04 | 2004-01-13 | 13 | Tuesday | 33 | 37 | 38 | 35 | 23 | 29 | 25 | |
| 1996-05-04 | 2004-01-13 | 14 | Tuesday | 32 | 33 | 37 | 38 | 35 | 29 | 22 | |
| 1996-05-04 | 2004-01-13 | 15 | Tuesday | 31 | 32 | 33 | 37 | 38 | 29 | 20 | |
| 1996-05-04 | 2004-01-13 | 16 | Tuesday | 30 | 31 | 32 | 33 | 37 | 29 | 20 | |
| 1996-05-04 | 2004-01-13 | 17 | Tuesday | 31 | 30 | 31 | 32 | 33 | 27 | 21 | |
| 1996-05-04 | 2004-01-13 | 18 | Tuesday | 34 | 31 | 30 | 31 | 32 | 26 | 30 | |
| 1996-05-04 | 2004-01-13 | 19 | Tuesday | 36 | 34 | 31 | 30 | 31 | 35 | 35 | |
| 1996-05-04 | 2004-01-13 | 20 | Tuesday | 33 | 36 | 34 | 31 | 30 | 34 | 30 | |
| 1996-05-04 | 2004-01-13 | 21 | Tuesday | 22 | 33 | 36 | 34 | 31 | 33 | 24 | |
| 1996-05-04 | 2004-01-13 | 22 | Tuesday | 25 | 22 | 33 | 36 | 34 | 30 | 18 | |
| 1996-05-04 | 2004-01-13 | 23 | Tuesday | 19 | 25 | 22 | 33 | 36 | 20 | 16 | |
| 1996-05-05 | 2004-01-14 | 00 | Wednesd | 17 | 19 | 25 | 22 | 33 | 15 | 14 | |
| 1996-05-05 | 2004-01-14 | 01 | Wednesd | 12 | 17 | 19 | 25 | 22 | 13 | 10 | |
| 1996-05-05 | 2004-01-14 | 02 | Wednesd | 12 | 12 | 17 | 19 | 25 | 13 | 10 | |
| 1996-05-05 | 2004-01-14 | 03 | Wednesd | 12 | 12 | 12 | 17 | 19 | 13 | 10 | |
| 1996-05-05 | 2004-01-14 | 04 | Wednesd | 13 | 12 | 12 | 12 | 17 | 13 | 10 | |
| 1996-05-05 | 2004-01-14 | 05 | Wednesd | 14 | 13 | 12 | 12 | 12 | 15 | 12 | |
| 1996-05-05 | 2004-01-14 | 06 | Wednesd | 18 | 14 | 13 | 12 | 12 | 28 | 18 | |
| 1996-05-05 | 2004-01-14 | 07 | Wednesd | 25 | 18 | 14 | 13 | 12 | 23 | 24 | |
| 1996-05-05 | 2004-01-14 | 08 | Wednesd | 28 | 25 | 18 | 14 | 13 | 27 | 26 | |
| 1996-05-05 | 2004-01-14 | 09 | Wednesd | 34 | 28 | 25 | 18 | 14 | 23 | 23 | |
| 1996-05-05 | 2004-01-14 | 10 | Wednesd | 37 | 34 | 28 | 25 | 18 | 35 | 34 | |
| 1996-05-05 | 2004-01-14 | 11 | Wednesd | 38 | 37 | 34 | 28 | 25 | 38 | 35 | |
| 1996-05-05 | 2004-01-14 | 12 | Wednesd | 36 | 38 | 37 | 34 | 28 | 37 | 34 | |
| 1996-05-05 | 2004-01-14 | 13 | Wednesd | 32 | 36 | 38 | 37 | 34 | 33 | 29 | |

| Initial input variables /original/ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| day | day | hour | day- type | load (h) | load (h-1) | load (h-2) | load (h-3) | load (h-4) | load (h-24) | load (h-48) | load (h-168) |
| 1996-05-05 | 2004-01-14 | 14 | Wednesd | 30 | 32 | 36 | 38 | 37 | 32 | 29 | |
| 1996-05-05 | 2004-01-14 | 15 | Wednesd | 28 | 30 | 32 | 36 | 38 | 31 | 29 | |
| 1996-05-05 | 2004-01-14 | 16 | Wednesd | 33 | 28 | 30 | 32 | 36 | 30 | 29 | |
| 1996-05-05 | 2004-01-14 | 17 | Wednesd | 31 | 33 | 28 | 30 | 32 | 31 | 27 | |
| 1996-05-05 | 2004-01-14 | 18 | Wednesd | 30 | 31 | 33 | 28 | 30 | 34 | 26 | |
| 1996-05-05 | 2004-01-14 | 19 | Wednesd | 36 | 30 | 31 | 33 | 28 | 36 | 35 | |
| 1996-05-05 | 2004-01-14 | 20 | Wednesd | 35 | 36 | 30 | 31 | 33 | 33 | 34 | |
| 1996-05-05 | 2004-01-14 | 21 | Wednesd | 32 | 35 | 36 | 30 | 31 | 22 | 33 | |
| 1996-05-05 | 2004-01-14 | 22 | Wednesd | 28 | 32 | 35 | 36 | 30 | 25 | 30 | |
| 1996-05-05 | 2004-01-14 | 23 | Wednesd | 23 | 28 | 32 | 35 | 36 | 19 | 20 | |
| 1996-05-06 | 2004-01-15 | 00 | Thursday | 18 | 23 | 28 | 32 | 35 | 17 | 15 | |
| 1996-05-06 | 2004-01-15 | 01 | Thursday | 13 | 18 | 23 | 28 | 32 | 12 | 13 | |
| 1996-05-06 | 2004-01-15 | 02 | Thursday | 12 | 13 | 18 | 23 | 28 | 12 | 13 | |
| 1996-05-06 | 2004-01-15 | 03 | Thursday | 12 | 12 | 13 | 18 | 23 | 12 | 13 | |
| 1996-05-06 | 2004-01-15 | 04 | Thursday | 12 | 12 | 12 | 13 | 18 | 13 | 13 | |
| 1996-05-06 | 2004-01-15 | 05 | Thursday | 13 | 12 | 12 | 12 | 13 | 14 | 15 | |
| 1996-05-06 | 2004-01-15 | 06 | Thursday | 19 | 13 | 12 | 12 | 12 | 18 | 28 | |
| 1996-05-06 | 2004-01-15 | 07 | Thursday | 25 | 19 | 13 | 12 | 12 | 25 | 23 | |
| 1996-05-06 | 2004-01-15 | 08 | Thursday | 29 | 25 | 19 | 13 | 12 | 28 | 27 | |
| 1996-05-06 | 2004-01-15 | 09 | Thursday | 34 | 29 | 25 | 19 | 13 | 34 | 23 | |
| 1996-05-06 | 2004-01-15 | 10 | Thursday | 37 | 34 | 29 | 25 | 19 | 37 | 35 | |
| 1996-05-06 | 2004-01-15 | 11 | Thursday | 38 | 37 | 34 | 29 | 25 | 38 | 38 | |
| 1996-05-06 | 2004-01-15 | 12 | Thursday | 37 | 38 | 37 | 34 | 29 | 36 | 37 | |
| 1996-05-06 | 2004-01-15 | 13 | Thursday | 35 | 37 | 38 | 37 | 34 | 32 | 33 | |
| 1996-05-06 | 2004-01-15 | 14 | Thursday | 32 | 35 | 37 | 38 | 37 | 30 | 32 | |
| 1996-05-06 | 2004-01-15 | 15 | Thursday | 31 | 32 | 35 | 37 | 38 | 28 | 31 | |
| 1996-05-06 | 2004-01-15 | 16 | Thursday | 30 | 31 | 32 | 35 | 37 | 33 | 30 | |
| 1996-05-06 | 2004-01-15 | 17 | Thursday | 29 | 30 | 31 | 32 | 35 | 31 | 31 | |
| 1996-05-06 | 2004-01-15 | 18 | Thursday | 29 | 29 | 30 | 31 | 32 | 30 | 34 | |
| 1996-05-06 | 2004-01-15 | 19 | Thursday | 36 | 29 | 29 | 30 | 31 | 36 | 36 | |
| 1996-05-06 | 2004-01-15 | 20 | Thursday | 35 | 36 | 29 | 29 | 30 | 35 | 33 | |
| 1996-05-06 | 2004-01-15 | 21 | Thursday | 33 | 35 | 36 | 29 | 29 | 32 | 22 | |
| 1996-05-06 | 2004-01-15 | 22 | Thursday | 28 | 33 | 35 | 36 | 29 | 28 | 25 | |
| 1996-05-06 | 2004-01-15 | 23 | Thursday | 24 | 28 | 33 | 35 | 36 | 23 | 19 | |
| 1996-05-07 | 2004-01-16 | 00 | Friday | 11 | 24 | 28 | 33 | 35 | 18 | 17 | |
| 1996-05-07 | 2004-01-16 | 01 | Friday | 12 | 11 | 24 | 28 | 33 | 13 | 12 | |
| 1996-05-07 | 2004-01-16 | 02 | Friday | 12 | 12 | 11 | 24 | 28 | 12 | 12 | |
| 1996-05-07 | 2004-01-16 | 03 | Friday | 13 | 12 | 12 | 11 | 24 | 12 | 12 | |
| 1996-05-07 | 2004-01-16 | 04 | Friday | 15 | 13 | 12 | 12 | 11 | 12 | 13 | |

| Initial input variables /original/ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| day | day | hour | day- type | load (h) | load (h-1) | load (h-2) | load (h-3) | load (h-4) | load (h-24) | load (h-48) | load (h-168) |
| 1996-05-07 | 2004-01-16 | 05 | Friday | 18 | 15 | 13 | 12 | 12 | 13 | 14 | |
| 1996-05-07 | 2004-01-16 | 06 | Friday | 22 | 18 | 15 | 13 | 12 | 19 | 18 | |
| 1996-05-07 | 2004-01-16 | 07 | Friday | 26 | 22 | 18 | 15 | 13 | 25 | 25 | |
| 1996-05-07 | 2004-01-16 | 08 | Friday | 28 | 26 | 22 | 18 | 15 | 29 | 28 | |
| 1996-05-07 | 2004-01-16 | 09 | Friday | 35 | 28 | 26 | 22 | 18 | 34 | 34 | |
| 1996-05-07 | 2004-01-16 | 10 | Friday | 36 | 35 | 28 | 26 | 22 | 37 | 37 | |
| 1996-05-07 | 2004-01-16 | 11 | Friday | 38 | 36 | 35 | 28 | 26 | 38 | 38 | |
| 1996-05-07 | 2004-01-16 | 12 | Friday | 35 | 38 | 36 | 35 | 28 | 37 | 36 | |
| 1996-05-07 | 2004-01-16 | 13 | Friday | 32 | 35 | 38 | 36 | 35 | 35 | 32 | |
| 1996-05-07 | 2004-01-16 | 14 | Friday | 29 | 32 | 35 | 38 | 36 | 32 | 30 | |
| 1996-05-07 | 2004-01-16 | 15 | Friday | 29 | 29 | 32 | 35 | 38 | 31 | 28 | |
| 1996-05-07 | 2004-01-16 | 16 | Friday | 29 | 29 | 29 | 32 | 35 | 30 | 33 | |
| 1996-05-07 | 2004-01-16 | 17 | Friday | 27 | 29 | 29 | 29 | 32 | 29 | 31 | |
| 1996-05-07 | 2004-01-16 | 18 | Friday | 27 | 27 | 29 | 29 | 29 | 29 | 30 | |
| 1996-05-07 | 2004-01-16 | 19 | Friday | 35 | 27 | 27 | 29 | 29 | 36 | 36 | |
| 1996-05-07 | 2004-01-16 | 20 | Friday | 34 | 35 | 27 | 27 | 29 | 35 | 35 | |
| 1996-05-07 | 2004-01-16 | 21 | Friday | 31 | 34 | 35 | 27 | 27 | 33 | 32 | |
| 1996-05-07 | 2004-01-16 | 22 | Friday | 27 | 31 | 34 | 35 | 27 | 28 | 28 | |
| 1996-05-07 | 2004-01-16 | 23 | Friday | 20 | 27 | 31 | 34 | 35 | 24 | 23 | |
| 1996-05-08 | 2004-01-17 | 00 | Saturday | 18 | 20 | 27 | 31 | 34 | 11 | 18 | 18 |
| 1996-05-08 | 2004-01-17 | 01 | Saturday | 13 | 18 | 20 | 27 | 31 | 12 | 13 | 14 |
| 1996-05-08 | 2004-01-17 | 02 | Saturday | 13 | 13 | 18 | 20 | 27 | 12 | 12 | 12 |
| 1996-05-08 | 2004-01-17 | 03 | Saturday | 13 | 13 | 13 | 18 | 20 | 13 | 12 | 12 |
| 1996-05-08 | 2004-01-17 | 04 | Saturday | 13 | 13 | 13 | 13 | 18 | 15 | 12 | 12 |
| 1996-05-08 | 2004-01-17 | 05 | Saturday | 15 | 13 | 13 | 13 | 13 | 18 | 13 | 14 |
| 1996-05-08 | 2004-01-17 | 06 | Saturday | 18 | 15 | 13 | 13 | 13 | 22 | 19 | 19 |
| 1996-05-08 | 2004-01-17 | 07 | Saturday | 23 | 18 | 15 | 13 | 13 | 26 | 25 | 24 |
| 1996-05-08 | 2004-01-17 | 08 | Saturday | 26 | 23 | 18 | 15 | 13 | 28 | 29 | 28 |
| 1996-05-08 | 2004-01-17 | 09 | Saturday | 32 | 26 | 23 | 18 | 15 | 35 | 34 | 33 |
| 1996-05-08 | 2004-01-17 | 10 | Saturday | 35 | 32 | 26 | 23 | 18 | 36 | 37 | 36 |
| 1996-05-08 | 2004-01-17 | 11 | Saturday | 35 | 35 | 32 | 26 | 23 | 38 | 38 | 38 |
| 1996-05-08 | 2004-01-17 | 12 | Saturday | 34 | 35 | 35 | 32 | 26 | 35 | 37 | 35 |
| 1996-05-08 | 2004-01-17 | 13 | Saturday | 30 | 34 | 35 | 35 | 32 | 32 | 35 | 30 |
| 1996-05-08 | 2004-01-17 | 14 | Saturday | 27 | 30 | 34 | 35 | 35 | 29 | 32 | 28 |
| 1996-05-08 | 2004-01-17 | 15 | Saturday | 26 | 27 | 30 | 34 | 35 | 29 | 31 | 26 |
| 1996-05-08 | 2004-01-17 | 16 | Saturday | 25 | 26 | 27 | 30 | 34 | 29 | 30 | 25 |
| 1996-05-08 | 2004-01-17 | 17 | Saturday | 25 | 25 | 26 | 27 | 30 | 27 | 29 | 25 |
| 1996-05-08 | 2004-01-17 | 18 | Saturday | 26 | 25 | 25 | 26 | 27 | 27 | 29 | 26 |
| 1996-05-08 | 2004-01-17 | 19 | Saturday | 35 | 26 | 25 | 25 | 26 | 35 | 36 | 35 |

| Initial input variables /original/ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| day | day | hour | day- type | load (h) | load (h-1) | load (h-2) | load (h-3) | load (h-4) | load (h-24) | load (h-48) | load (h-168) |
| 1996-05-08 | 2004-01-17 | 20 | Saturday | 34 | 35 | 26 | 25 | 25 | 34 | 35 | 34 |
| 1996-05-08 | 2004-01-17 | 21 | Saturday | 32 | 34 | 35 | 26 | 25 | 31 | 33 | 33 |
| 1996-05-08 | 2004-01-17 | 22 | Saturday | 28 | 32 | 34 | 35 | 26 | 27 | 28 | 28 |
| 1996-05-08 | 2004-01-17 | 23 | Saturday | 22 | 28 | 32 | 34 | 35 | 20 | 24 | 23 |
| 1996-05-09 | 2004-01-18 | 00 | Sunday | 16 | 22 | 28 | 32 | 34 | 18 | 11 | 16 |
| 1996-05-09 | 2004-01-18 | 01 | Sunday | 13 | 16 | 22 | 28 | 32 | 13 | 12 | 15 |
| 1996-05-09 | 2004-01-18 | 02 | Sunday | 12 | 13 | 16 | 22 | 28 | 13 | 12 | 12 |
| 1996-05-09 | 2004-01-18 | 03 | Sunday | 12 | 12 | 13 | 16 | 22 | 13 | 13 | 12 |
| 1996-05-09 | 2004-01-18 | 04 | Sunday | 12 | 12 | 12 | 13 | 16 | 13 | 15 | 10 |
| 1996-05-09 | 2004-01-18 | 05 | Sunday | 13 | 12 | 12 | 12 | 13 | 15 | 18 | 13 |
| 1996-05-09 | 2004-01-18 | 06 | Sunday | 16 | 13 | 12 | 12 | 12 | 18 | 22 | 15 |
| 1996-05-09 | 2004-01-18 | 07 | Sunday | 19 | 16 | 13 | 12 | 12 | 23 | 26 | 19 |
| 1996-05-09 | 2004-01-18 | 08 | Sunday | 23 | 19 | 16 | 13 | 12 | 26 | 28 | 22 |
| 1996-05-09 | 2004-01-18 | 09 | Sunday | 26 | 23 | 19 | 16 | 13 | 32 | 35 | 25 |
| 1996-05-09 | 2004-01-18 | 10 | Sunday | 28 | 26 | 23 | 19 | 16 | 35 | 36 | 23 |
| 1996-05-09 | 2004-01-18 | 11 | Sunday | 29 | 28 | 26 | 23 | 19 | 35 | 38 | 24 |
| 1996-05-09 | 2004-01-18 | 12 | Sunday | 27 | 29 | 28 | 26 | 23 | 34 | 35 | 28 |
| 1996-05-09 | 2004-01-18 | 13 | Sunday | 21 | 27 | 29 | 28 | 26 | 30 | 32 | 25 |
| 1996-05-09 | 2004-01-18 | 14 | Sunday | 22 | 21 | 27 | 29 | 28 | 27 | 29 | 22 |
| 1996-05-09 | 2004-01-18 | 15 | Sunday | 20 | 22 | 21 | 27 | 29 | 26 | 29 | 20 |
| 1996-05-09 | 2004-01-18 | 16 | Sunday | 19 | 20 | 22 | 21 | 27 | 25 | 29 | 20 |
| 1996-05-09 | 2004-01-18 | 17 | Sunday | 21 | 19 | 20 | 22 | 21 | 25 | 27 | 21 |
| 1996-05-09 | 2004-01-18 | 18 | Sunday | 25 | 21 | 19 | 20 | 22 | 26 | 27 | 30 |
| 1996-05-09 | 2004-01-18 | 19 | Sunday | 32 | 25 | 21 | 19 | 20 | 35 | 35 | 35 |
| 1996-05-09 | 2004-01-18 | 20 | Sunday | 32 | 32 | 25 | 21 | 19 | 34 | 34 | 30 |
| 1996-05-09 | 2004-01-18 | 21 | Sunday | 30 | 32 | 32 | 25 | 21 | 32 | 31 | 24 |

## nnex 2: Sample dataset of original peak load

| | date | | day | load(d) | load (d-1) | load (d-7) | load (d-8) |
|---|---|---|---|---|---|---|---|
| Ieskerem | 2000-09-11 | 1993-01-01 | Monday | 20 | | | |
| eskerem | 2000-09-12 | 1993-01-02 | Tuesday | 24 | 20 | | |
| eskerem | 2000-09-13 | 1993-01-03 | Wednesday | 30 | 24 | | |
| eskerem | 2000-09-14 | 1993-01-04 | Thursday | 33.5 | 30 | | |
| skerem | 2000-09-15 | 1993-01-05 | Friday | 32 | 33.5 | | |
| skerem | 2000-09-16 | 1993-01-06 | Saturday | 30 | 32 | | |
| kerem | 2000-09-17 | 1993-01-07 | Sunday | 35 | 30 | | |
| kerem | 2000-09-18 | 1993-01-08 | Monday | 31 | 35 | 20 | |
| kerem | 2000-09-19 | 1993-01-09 | Tuesday | 33 | 31 | 24 | 20 |
| erem | 2000-09-20 | 1993-01-10 | Wednesday | 34 | 33 | 30 | 24 |
| erem | 2000-09-21 | 1993-01-11 | Thursday | 33 | 34 | 33.5 | 30 |
| erem | 2000-09-22 | 1993-01-12 | Friday | 33 | 33 | 32 | 33.5 |
| erem | 2000-09-23 | 1993-01-13 | Saturday | 31 | 33 | 30 | 32 |
| rem | 2000-09-24 | 1993-01-14 | Sunday | 23 | 31 | 35 | 30 |
| rem | 2000-09-25 | 1993-01-15 | Monday | 30 | 23 | 31 | 35 |
| rem | 2000-09-26 | 1993-01-16 | Tuesday | 33 | 30 | 33 | 31 |
| em | 2000-09-27 | 1993-01-17 | Wednesday | 25 | 33 | 34 | 33 |
| em | 2000-09-28 | 1993-01-18 | Thursday | 30 | 25 | 33 | 34 |
| m | 2000-09-29 | 1993-01-19 | Friday | 32 | 30 | 33 | 33 |
| m | 2000-09-30 | 1993-01-20 | Saturday | 30 | 32 | 31 | 33 |
| n | 2000-10-01 | 1993-01-21 | Sunday | 24 | 30 | 23 | 31 |
| n | 2000-10-02 | 1993-01-22 | Monday | 30 | 24 | 30 | 23 |
| n | 2000-10-03 | 1993-01-23 | Tuesday | 30 | 30 | 33 | 30 |
| | 2000-10-04 | 1993-01-24 | Wednesday | 32 | 30 | 25 | 33 |
| | 2000-10-05 | 1993-01-25 | Thursday | 32.5 | 32 | 30 | 25 |
| | 2000-10-06 | 1993-01-26 | Friday | 30 | 32.5 | 32 | 30 |
| | 2000-10-07 | 1993-01-27 | Saturday | 29 | 30 | 30 | 32 |
| | 2000-10-08 | 1993-01-28 | Sunday | 25 | 29 | 24 | 30 |
| | 2000-10-09 | 1993-01-29 | Monday | 31 | 25 | 30 | 24 |
| | 2000-10-10 | 1993-01-30 | Tuesday | 33 | 31 | 30 | [illegible] |
| | 2000-10-11 | 1993-02-01 | W[illegible] | [illegible] | [illegible] | [illegible] | [illegible] |