FUZZY SET CLASSIFIED NEURAL NETWORK FOR SHORT-TERM LOAD FORECASTING

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Power System & Electric Drives

Thapar University, Patiala

By:
Gagan
(80641007)

Under the supervision of:
Dr. Sanjay K. Jain

September 2008

ELECTRICAL & INSTRUMENTATION ENGG. DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Fuzzy Set Classified Neural Network for Short –Term Load Forecasting"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Power Systems & Electric Drives* submitted in Electrical & Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of ***Dr. Sanjay K. Jain, Assistant Prof., EIED.***

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Gagan
(Gagan)
Regn. No. 80641007

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Sanjay 18 sept 08
(Dr. Sanjay K. Jain)
Assistant Prof., EIED
Thapar University
Patiala

Countersigned by

S. Ghosh 18/9/08
(Dr. Smarajit Ghosh)
Professor & Head
Electrical & Instrumentation Engg. Department
Thapar University
Patiala

RK Sharma
(Dr. R. K. Sharma)
Dean(Academic Affairs)
Thapar University
Patiala

i

ACKNOWLEDGMENTS

First of all, I thank the Almighty, who gave me the opportunity and strength to carry out this work.

I would like to thank **Dr. Sanjay K. Jain, Assistant Prof., EIED** for the opportunity to work with him, and also for his encouragement, trust and untiring support. **Dr. Sanjay K. Jain** has been an advisor in the true sense both academically and morally throughout this project work.

Much appreciations is expressed to **Dr. Abhijit Mukherjee, Director, Dr. K.K. Raina, Deputy Director and Dr. R.K. Sharma, Dean (Academic Affairs), Thapar University** to provide me much needed administrative support leading to complete the M.E. Thesis work.

Gratitude is accorded to Thapar University, Patiala, for providing all the necessary facilities to complete my M.E. Thesis work.

I thank to **Dr. Smarajit Ghosh, Professor & Head, EIED** for their continuous inspiration during this thesis work.

The paucity of words does not compromise for extending my thanks to all the staff members of EIED, Thapar University, my all family members and friends whose uninterrupted love, inspiration and blessings helped me in completing this research report.

I am very grateful to **Dr. Maneek Kumar, Professor & Head** and **Dr. Sarbjit Singh, Assistant Prof., Civil Engineering Department** for providing me the useful weather information.

I am also thankful to HPSEB officials, PSEB officials and Metrological department officials for providing me the required station data records.

I am also thankful to the researchers whose work has been consulted, utilized and cited in my dissertation.

(Gagan)

Regn. No. 80641007

ABSTRACT

Load forecasting is an important component for power system energy management system. Precise load forecasting helps the electric utility to make unit commitment decisions, reduce spinning reserve capacity and schedule device maintenance plan properly. Besides playing a key role in reducing the generation cost, it is also essential to the reliability of power systems. The system operators use the load forecasting result as a basis of off-line network analysis to determine if the system might be vulnerable. If so, corrective actions should be prepared, such as load shedding, power purchases and bringing peaking units on line.

Load forecasting plays an important role in power system planning, operation and control. Forecasting means estimating active loads at various load buses ahead of actual load occurrence. Planning and operational applications of load forecasting requires a certain 'lead time' also called forecasting intervals. On the basis of lead time, load forecasts can be divided into four categories: very short-term forecasts, short-term forecasts, medium-term forecasts and long-term forecasts. The forecasts for different time horizons are important for different operations within a utility company.

Since in power systems the next days' power generation must be scheduled everyday, day-ahead short-term load forecasting (STLF) is a necessary daily task for power dispatch. Its accuracy affects the economic operation and reliability of the system greatly. Under prediction of STLF leads to insufficient reserve capacity preparation and, in turn, increases the operating cost by using expensive peaking units. On the other hand, over prediction of STLF leads to the unnecessarily large reserve capacity, which is also related to high operating cost.

In this work, a Fuzzy Set Classified Neural Network Approach for Short Term Load Forecasting is attempted and implemented using Matlab 6.5. First of all, training data is classified using Fuzzy Set Based Classification Method. Temperature data is classified into five fuzzy sets (Very Cold, Cold, Normal, Hot and Very Hot). Relative Humidity is classified into four fuzzy sets (Very Dry, Dry, Humid and Very Humid). Day Type is classified into four fuzzy sets (Post-Holiday, Weekday, Pre-Holiday and

Holiday). So, depending upon the temperature, relative humidity and day type, data is classified into eighty classes. After the classification, the neural network is trained for various classes using the historical data. The multilayer neural network structure has been used and the training is imparted using back propagation algorithm. Thereafter, the trained neural network is used for load forecasting. The work presented here is divided into three steps:

1. Fuzzy Set Based Classification: Classification of training data using Fuzzy Set Based Classification Technique and also identify classes for all the 24 hours of the day for which the load is to be forecasted.
2. Training of Neural Network: Training of the neural network for each hour of each day for which the load is to be forecasted using the training data of that particular class to which that hour belongs.
3. Short term load forecasting: Forecasting of hourly load using trained neural network.

The test cases studied in this work to validate the accuracy of the proposed technique are as:-

- Case-1: Summer and Post-Holiday
- Case-2: Summer and Pre-Holiday
- Case-3: Winter and Weekday
- Case-4: Summer and Holiday
- Case-5: Rainy season and Weekday

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW

The utilities are required to provide reliable power to customers. In the design stages, utilities need to plan ahead for anticipated future load growth under different possible scenarios. Their decisions and designs can affect the gain or loss of crores of rupees for their companies/utilities as well as customer satisfaction and future economic growth in their area. Accurate models for electric power load forecasting are essential to the operation and planning of a utility company. Load forecasting helps an electric utility to make important decisions including decisions on sale, purchase, banking of power (with other companies or utilities of same state or the neighboring states) and generating electric power, load switching, and infrastructure development. For sale or purchase, short-term load forecasting is used and for banking, generally long-term load forecasting is used. Load forecasts are extremely important for energy suppliers, and other participants in electric energy generation, transmission, distribution, and markets.

In spite of the numerous literatures on STLF published since 1960s, the research work in this area is still a challenge to the electrical engineering scholars because of its high complexity. How to estimate the future load with the historical data has remained a difficulty up to now, especially for the load forecasting of holidays, days with extreme weather and other anomalous days. With the recent development of new mathematical, data mining and artificial intelligence tools, it is potentially possible to improve the forecasting result.

With the recent trend of deregulation of electricity markets, STLF has gained more importance and greater challenges. In the market environment, precise forecasting is the basis of electrical energy trade and spot price establishment for the system to gain the minimum electricity purchasing cost. In the real-time dispatch operation, forecasting error causes more purchasing electricity cost or breaking-contract penalty cost to keep the electricity supply and consumption balance.

1.2 LITERATURE REVIEW

The literature on the load forecasting and methods is very diversified and it is not possible to complement them in the limited time span. Therefore, in this section, the literature on Short term Load Forecasting is briefly revived. This literature review offers the background for the thesis work. The published literature has been classified into six main categories:

Regression Methods

Engle et al. [1] presented several regression models for the next day load forecasting. Their models incorporate deterministic influences such as holidays, stochastic influences such as average loads, and exogenous influences such as weather. [2], [3], [4] and [5] describe other applications of regression models applied to load forecasting.

Time Series

Time series method is based upon the assumption that the data has some internal structure such as autocorrelation, trend or seasonal variation. Most commonly used classical time series methods are ARMA (autoregressive moving average), ARIMA (autoregressive integrated moving average), ARMAX (autoregressive moving average with exogenous variables), and ARIMAX (autoregressive integrated moving average with exogenous variables). Fan and McDonald [6] and Cho et al. [7] described implementations of ARIMAX models for load forecasting. Yang et al. [8] used an evolutionary programming (EP) approach to identify the ARMAX model parameters for one day to one week ahead hourly-load-demand-forecasting. The evolutionary programming is a method for simulating evolution and constitutes a stochastic optimization algorithm. Yang and Huang [9] proposed a fuzzy autoregressive moving average with exogenous input variables (FARMAX) for one day ahead hourly load forecasting.

Expert Systems

Ho et al. [10] proposed a knowledge-based expert system for the short-term load forecasting of the Taiwan power system. Operators' knowledge and the hourly observation of system load over the past five years are employed to establish eleven day-types. Weather parameters were also considered. Rahman and Hazim [11] developed a site-independent technique for short-term load forecasting. Knowledge about the load and the factors affecting it is extracted and represented in a parameterized rule base. This rule-based system is complemented by a parameter database that varies from site to site. The technique is tested in different sites in the United States with low forecasting errors. The load model, the rules and the parameters presented in the paper have been designed using no specific knowledge about any particular site. Results can be improved if operators at a particular site are consulted.

Fuzzy Logic

One of the advantages of the use of fuzzy logic is the absence of a need for a mathematical model mapping inputs to outputs and the absence of a need for precise inputs. With such generic conditioning rules, properly designed fuzzy logic systems can be very robust when used for forecasting. Of course in many situations an exact output is needed. To produce such precise outputs, "defuzzification" can be used after the logical processing of fuzzy inputs. [19], [20] and [21] describe applications of fuzzy logic to load forecasting.

Neural Networks

The interest in applying neural networks to electric load forecasting began in 1990. Artificial Neural Networks have parallel and distributed processing structures. They can be thought of as a set of computing arrays consisting of series of repetitive uniform processors placed on a grid. Learning is achieved by changing the interconnection between the processors [15]. To date, there exist many types of ANNs which are characterized by their topology and learning rules. As for the STLF problem, the BP network is the most widely used one. With the ability to approximate any continuous nonlinear function, the BP network has extraordinary mapping (forecasting) abilities. The BP network is a kind of multilayer feed forward network, and the transfer function within the network is usually a nonlinear function such as the sigmoid function. Neural Networks are widely used for load forecasting, Fault diagnosis/Fault location, Economic load dispatch and Security assessment etc. in the field of power systems [81].

The topology of BP network can be of 3-layers or 4-layers, the transfer function can be linear, nonlinear or a combination of both. Also, the network can be either fully connected or non-fully connected. The BP network structure is problem dependent, and a structure that is suitable for a given power system is not necessarily suitable for another. The typical BP network structure for STLF is a three-layer network, with the nonlinear sigmoid function as the transfer function [16]-[24]. In addition to the typical sigmoid function, a linear transfer function from the input layer directly to the output layer was proposed in [25] to account for linear components of the load. Because fully connected BP networks need more training time a non-fully connected BP model is proposed in [26], [27]. The reported results show that although a fully connected ANN is able to capture the load characteristics, a non-fully connected ANN is more adaptive to respond to temperature changes. Moreover, [27] presents a new approach to STLF which combines several sub-ANNs together to give better forecasting results. A recurrent high order neural network (RHONN) is also proposed [28]. Due to its dynamic nature, the RHONN forecasting model is able to adapt quickly to changing conditions such as important load variations or changes of the daily load pattern. A 3-layer ANN with suitable dimension is sufficient to approximate any continuous non-linear function [28]. The 4-layer structure is implemented and a load forecaster using this structure was reported [12], [15], [22], [29].

The BP network is a kind of array which can realize nonlinear mapping from the inputs to the outputs. Therefore, the selection of input variables of a load forecasting network is of great importance. Broadly, there are two selection methods. One is based on experience [12], [15], [17], [25] and the other is based on statistical analysis such as the ARIMA [27] and correlation analysis [22]. The input variables are largely determined on engineering judgment and experience. In all, the input variables can be classified into 6 main classes:

1. Historical loads [15]-[17], [22], [23], [25] –[28], [30], [52]
2. Historical and future (forecasted) temperatures [15]-[17], [22], [25]-[27], [30],
3. Historical and future (forecasted) relative humidity [52]
4. Hour of day index [15], [17], [18], [22], [27],
5. Day of week index [15], [18], [22], [27],
6. Wind-speed and sky cover [18], [26],
7. Rainfall (Wet or dry day) [18].

The BP algorithm is widely used in STLF and has some good features such as, its ability to easily accommodate weather variables, and its implicit expressions relating inputs and outputs. However, the raining process is time consuming training process and it converges to local minima. The research work has attributed the premature saturation as the major reasons for these drawbacks [36]. A method to prevent premature saturation by the appropriate selection of the initial weights is proposed in [31]. The BP algorithm with momentum (BPM) converges much faster than the conventional BP algorithm [32]. In [17], [33], it is shown that the use of the BPM in STLF significantly improves the training process.

The authors of [24] present extensive studies on the effects of factors such as the learning step, the momentum factor to BPM. They proposed a learning algorithm for adaptive training of neural networks. A learning algorithm motivated by the principle of “forced dynamic” for the total error function is proposed in [34]. The rate of change of the network weights is chosen such that the error function to be minimized is forced to “decay” in a certain mode. An approach by updating the weights in direct proportion to total error is proposed in [35]. With this, the periods of stagnation are much shorter and the possibility of trapping in local minima is greatly reduced.

Determination of the optimal number of hidden neurons is a crucial issue. If it is too small, the network can not possess sufficient information, and thus yields inaccurate forecasting results. On the other hand, if it is too large, the training process will be very long [15]. The work in [36] discusses the number of hidden neurons in binary value cases. In order to make the mapping between the output value and input pattern for I arbitrary learning patterns, the necessary and sufficient number of hidden neurons is (I-1).

[37] highlights that a multilayer perceptron with (k-1) hidden neurons can realize arbitrary functions defined on a k-element set. Up to our knowledge, there is no absolute criteria to determine the exact number of hidden neurons that will lead to an optimal solution. Different numbers of hidden neurons are used in [12], [15], [26], [27].

ANNs can only perform what they were trained to do. As for the case of STLF, the selection of the training set is a crucial one. The criteria for selecting the training set is that the characteristics of all the training pairs in the training set must be similar to those of the day to be forecasted. To obtain good forecasting results, day type information must be taken into account. There are two ways to do this. One way is to construct the different ANNs for each day type, and feed each ANN with the corresponding day type training sets [22], [30]. The other is to use only one ANN but contain the day type information in the input variables [15], [23], [27]. The former uses a number of relatively small size networks, while the later has only one network of a relatively large size. A typical classification given in [15] categorizes the historical loads into five classes. These are Monday, Tuesday-Thursday, Friday, Saturday and Sunday/Public holiday. The work in [16], collects the data with characteristics similar to the day being forecasted, and combines these data with the data from the previous 5 days to form a training set. The conventional methods use observation and comparison [15], [16], [25] and methods based on unsupervised ANN concepts and selects the training set automatically [12], [26] are used for day type classification.

Hybrid Fuzzy Neural Approaches

Researchers have proposed several different ways to combine fuzzy logic with neural networks techniques in order to improve the overall forecasting performance. They are classified into five categories according to the method of combination:

- Fuzzy logic system at the output stage of the neural network forecaster to manipulate the output [43], [44], [45];
- Fuzzy logic at the input stage of a neural network to preprocess the inputs [46], [47], [48];

- Integrated fuzzy neural network to create a fuzzy rule base from the historical training data [49], [50];
- Separate fuzzy logic and neural network forecasters to forecast different components of the load [51];
- Fuzzy logic technique for the classification of huge training data into different classes and neural network to forecast the load according to the classified training data [52].

1.3 OBJECTIVE OF THE THESIS WORK

The objective of the work is to develop an algorithm to forecast hourly load, by incorporating weather conditions and day type information. In this work, an attempt is made to implement the above forecast using fuzzy set classified neural network approach. The work presented here is divided into three steps:

1. Fuzzy Set Based Classification: Classification of training data using Fuzzy Set Based Classification Technique and also identify classes for all the 24 hours of the days for which the load is to be forecasted.
2. Training of Neural Network: Training of the neural network for each hour of each day for which the load is to be forecasted using the training data of that particular class to which that hour belongs.
3. Short term load forecasting: Forecasting of hourly load using trained neural network.

1.4 ORGANIZATION OF THE THESIS

The thesis is organized into six chapters. The contents of these chapters are summarized as:-

Chapter 1 details the overview of the problem, summary of literature review, objectives of the work and the organization of the thesis.

Chapter 2 covers the theoretical concepts of neural networks and fuzzy logic techniques in brief.

In Chapter 3, short-term load forecasting problem is described. Classification of load forecasting methods, important factors for forecasts, characteristics of the power system load and various forecasting methods are introduced in this chapter.

Chapter 4 summarizes the proposed methodology for load forecasting. The fuzzy set based classification is described first and then neural network structure and training for short term load forecasting is explained. The algorithm of the proposed technique is also detailed.

In Chapter 5, the effectiveness of the methodology is discussed through various case studies.

In Chapter 6, the summary of major conclusions and the scope of future work is detailed.

CHAPTER-2

NEURAL NETWORKS AND FUZZY LOGIC

2.1 NEURAL NETWORKS

Neural networks (NNs) do not perform miracles. But if used sensibly they can produce some amazing results. Artificial Neural Networks (ANNs) are non linear information processing structures in which the elements called as neurons process the information. Signals are transmitted by means of connection links. The links possess an associated weight, which is multiplied along with the incoming signal (net input) for any typical Neural Network. The output signal is obtained by applying activations to the net input [53], [54]. An Artificial Neural Network is characterized by:-

- Architecture (connection between Neurons)
- Training or Learning (determining weights on connections)
- Activation functions

As Artificial Neural Networks are inspired by the way biological nervous systems, such as brain process information. Key element of ANN is the noval structure of information processing system. It is composed of a large number of processing elements called neurons working in union to solve specific problems. ANNs learn by examples as the brain does [53].

The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much. In the late 1980's interest in NN increased with algorithms like Back Propagation and Kohonen (Many of them where developed quietly during the 1970s). Progress continued during the 1990's [53], [54]. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding. NNs are used in many commercial applications like character recognition, image recognition, credit evaluation, fraud detection, insurance, load forecasting and stock forecasting.

2.1.1 BASIC NEURON STRUCTURE

The structure of biological neuron and artificial neuron is briefly reviewed.

Biological Neuron

A biological neuron or a nerve cell consists of synapses, dendrites, the cell body and the axon [53], [55]. Fig.2.1 shows a simple Biological Neuron. The various building blocks are discussed as under:-

- The synapses are elementary signal processing devices

    - A synapse is a biochemical device, which converts a pre-synaptic electrical signal into a chemical signal and then back into a post-synaptic electrical signal.

    - The input pulse train has its amplitude modified by parameters stored in the synapse. The nature of this modification depends on the type of the synapse, which can be either inhibitory or excitatory

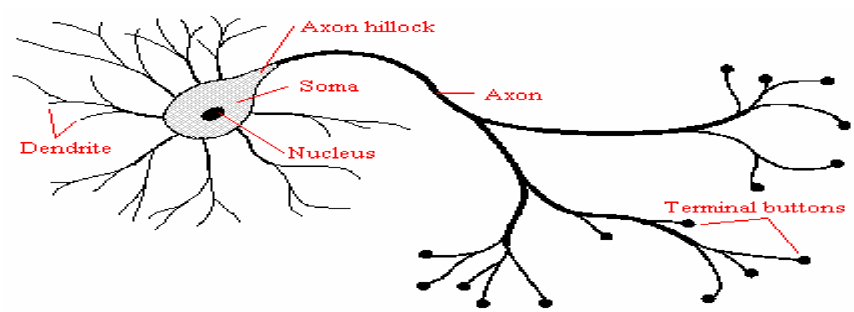

Fig.2.1: Simple Biological Neuron

- The postsynaptic signals are aggregated and transferred along the dendrites to the nerve cell body

- The cell body generates the output neuronal signal, a spike, which is transferred along the axon to the synaptic terminals of other neurons

- ❖ The frequency of firing of a neuron is proportional to the total synaptic activities and is controlled by the synaptic parameters (weights)

Artificial Neuron

The basic building block of an artificial neural network is the neuron. The basic structure of an artificial neuron is shown in Fig. 2.2. In a popular model, the connection weights between neurons are adjusted. The neuron receives inputs O_{pi} from neuron u_i while the network is exposed to input pattern p. Each input is multiplied by a connection weight w_{ij}, where w_{ij} is the connection between neurons u_i and u_j. The connection weights correspond to the strength of the influence of each of the preceding neurons. After the inputs have been multiplied by the connection weights for input pattern p, their values are summed, net_{pj}. Included in the summation is a bias value θ_j to offset the basic level of the input to the activation function, $f(net_{pj})$, which gives the output O_{pj} . In order to establish a bias value θj, the bias term can appear as an input from a separate neuron with a fixed value (a value of +1 is common). Each neuron requiring a bias value will be connected to the same bias neuron. The bias values are then self-adjusted as the other neurons learn, without the need for extra considerations [56].

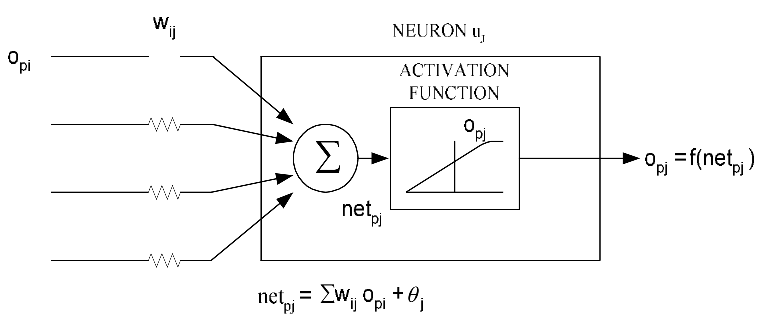

Fig.2.2: Basic Structure of an Artificial Neuron

In calculating the output of the neuron, the activation function may be in the form of a threshold function, in which the output of the neuron is +1 if a threshold level is reached and 0 otherwise. The various activation functions like squashing, hyperbolic tangent, sigmoid etc. can be used. Squashing functions limit the linear output between a maximum and minimum value. Hyperbolic tangents and the sigmoid functions are similar to real neural responses; however, the hyperbolic tangent is unbounded and hard to implement in hardware.

The artificial neural network is made up of seven major components [53],[57]. These components are summarized as:-

- Weighting Factors
- Summation Function
- Transfer Function / Activation Function
- Scaling and Limiting
- Output Function
- Error Function and Back-propagated Value
- Learning Function

Weighting Factors: A neuron usually receives many simultaneous inputs. Each input has its own relative weight which gives the input impact that it needs on the processing element's summation function. These weights perform the same type of function as do the varying synaptic strengths of biological neurons. In both cases, some inputs are made more important than others so that they have a greater effect on the processing element as they combine to produce a neural response. These strengths can be modified in response to various training sets and according to a network's specific topology or through its learning rules.

Summation Function: The first step in a processing element's operation is to compute the weighted sum of all of the inputs. Mathematically, the inputs and the corresponding weights can be represented as O_{pi} and w_{ij} respectively. The total input signal to the next layer neuron is the dot or inner product of these two vectors. This simplistic summation function is found by multiplying each component of the input by the corresponding component of the w_{ij} array and then adding up all the products. So, the input to the neuron of next layer is a single number, not a multi-element array.

Transfer Function / Activation Function: The activation function is denoted by $\Phi(.)$ [57] . The Identity function is defined by equation (2.1).

$$g(x) = x \tag{2.1}$$

It is obvious that the input units use the identity function. Sometimes a constant is multiplied by the net input to form a linear function. Fig. 2.3(a) shows identity activation function.

The Binary step function also known as threshold function or Heaviside function is shown in Fig. 2.3(b). The output of this function is limited to one of the two values as defined by equation (2.2). This kind of function is often used in single layer networks.

$$g(x) = \begin{cases} 1 & if \;\; x \geq \theta \\ 0 & if \;\; x < \theta \end{cases} \tag{2.2}$$

The sigmoid and bipolar sigmoid functions are shown in Fig.2.3(c) and 2.3(d) respectively. Equation (2.3) and (2.4) defines sigmoid and bipolar sigmoid functions respectively.

$$g(x) = \frac{1}{1+e^{-x}} \tag{2.3}$$

$$g(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{2.4}$$

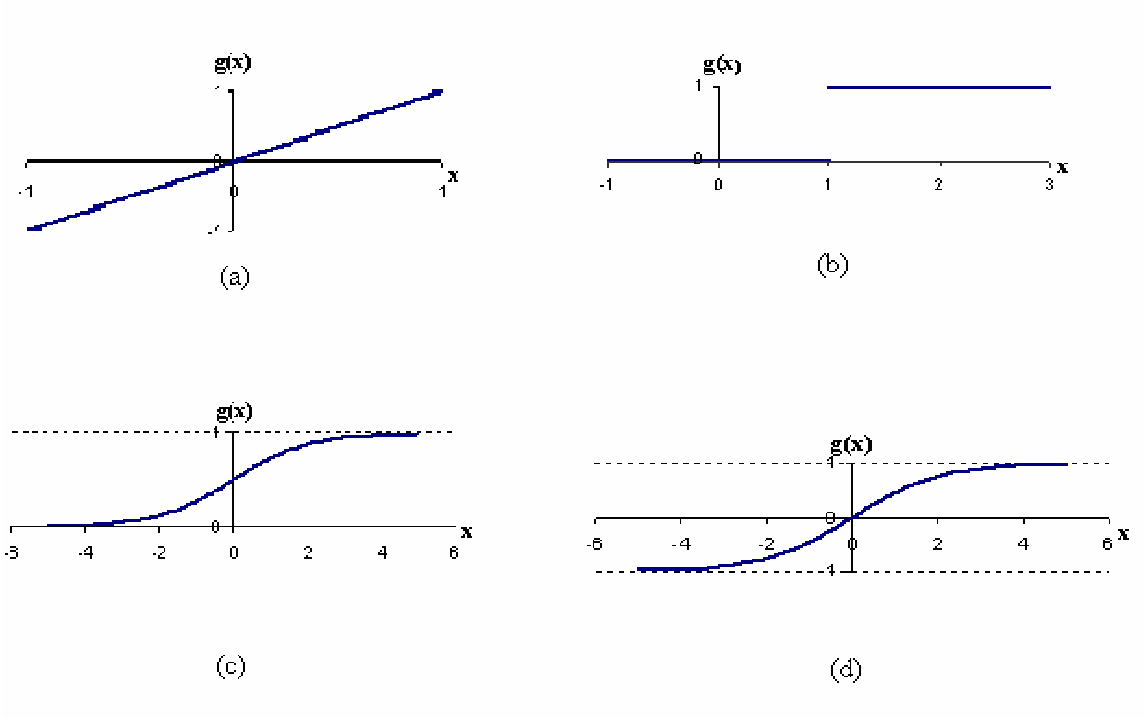

Fig.2.3: Common Activation Functions

Scaling and Limiting: After the processing element's transfer function, the result can pass through additional processes which scale and limit. This scaling simply multiplies a scale factor times the transfer value, and then adds an offset. Limiting is the mechanism which insures that the scaled result does not exceed an upper or lower bound. This limiting is in addition to the hard limits that the original transfer function may have performed.

Output Function (Competition): Each processing element is allowed one output signal which it may output to hundreds of other neurons. This is just like the biological neuron, where there are many inputs and only one output action. Normally, the output is directly equivalent to the transfer function's result. Some network topologies, however, modify the transfer result to incorporate competition among neighboring processing elements. Neurons are allowed to compete with each other, inhibiting processing elements unless they have great strength. Competition can occur at one or both of two levels. First, competition determines which artificial neuron will be active, or provides an output. Second, competitive inputs help determine which processing element will participate in the learning or adaptation process.

Error Function and Back-Propagated Value: In most learning networks the difference between the current output and the desired output is calculated. This raw error is then transformed by the error function to match particular network architecture. The most basic architectures use this error directly, but some square the error while retaining its sign, some cube the error, and other paradigms modify the raw error to fit their specific purposes. The artificial neuron's error is then typically propagated into the learning function of another processing element. This error term is sometimes called the current error.

The current error is typically propagated backwards to a previous layer. Yet, this back-propagated value can be either the current error, the current error scaled in some manner (often by the derivative of the transfer function), or some other desired output depending on the network type. Normally, this back-propagated value, after being scaled by the learning function, is multiplied against each of the incoming connection weights to modify them before the next learning cycle.

Learning Function: The purpose of the learning function is to modify the variable connection weights on the inputs of each processing element according to some neural based algorithm. This process of changing the weights of the input connections to achieve some desired result can also be called the adaption function, as well as the learning mode. There are two types of learning: supervised and unsupervised.

Advantages / Disadvantages of Artificial Neural Networks

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an expert in the category of information it has been given to analyze [53], [65], [66]. This expert can then be used to provide projections given new situations of interest and answer ‘what if’ questions.

Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

There are also some disadvantages of Neural Networks. These include:

1. The neural network needs training to operate.
2. The architecture of a neural network is different from the architecture of microprocessors therefore needs to be emulated.
3. Requires high processing time for large neural networks.

2.1.2 TEACHING AN ARTIFICIAL NEURAL NETWORK

Supervised Learning

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached [59].

With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, learning can take weeks. This training is considered complete when the neural network reaches an user defined performance level. This level signifies that the network has achieved the desired statistical accuracy as it produces the required outputs for a given sequence of inputs. When no further learning is necessary, the weights are typically frozen for the application.

After a supervised network performs well on the training data, then it is important to see what it can do with data it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network has not simply memorized a given set of data but has learned the general patterns involved within an application.

Unsupervised Learning

Unsupervised learning is the great promise of the future. Currently, this learning method is limited to networks known as self-organizing maps. These kinds of networks are not in widespread use [58], [59], [60]. This promising field of unsupervised learning is sometimes called self-supervised learning. These networks use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. Even without being told whether it's right or wrong, the network still must have some information about how to organize itself. This information is built into the network topology and learning rules.

Learning Rates

The rate at which ANNs learn depends upon several controllable factors. In selecting the approach there are many trade-offs to consider. Obviously, a slower rate means a lot more time is spent in accomplishing the off-line learning to produce an adequately trained system. With the faster learning rates, however, the network may not be able to make the fine discriminations possible with a system that learns more slowly [61].

Network complexity, size, paradigm selection, architecture, type of learning rule or rules employed and desired accuracy must all be considered before training. These factors play a significant role in determining how long it will take to train a network. Changing any one of these factors may either extend the training time to an unreasonable length or even result in an unacceptable accuracy.

Most learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one. If the learning rate is greater than one, it is easy for the learning algorithm to overshoot in correcting the weights, and the network will oscillate. Small values of the learning rate will not correct the current error as quickly, but if small steps are taken in correcting errors, there is a good chance of arriving at the best minimum convergence.

Learning Laws

Many learning laws are in common use. Most of these laws are some sort of variation of the best known and oldest learning law, Hebb's Rule. Learning is certainly more complex than the simplifications represented by the learning laws currently developed [62]. A few of the major laws are presented as examples.

Hebb's Rule: The first, and undoubtedly the best known, learning rule was introduced by Donald Hebb. The basic Hebb’s rule is: If a neuron receives an input from another neuron and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

The Delta Rule: This rule is a further variation of Hebb's Rule. It is one of the most commonly used. This rule is based on the simple idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a processing element. This rule changes the synaptic weights in the way that minimizes the mean squared error of the network. This rule is also referred to as the Widrow-Hoff Learning Rule and the Least Mean Square (LMS) Learning Rule.

The way that the Delta Rule works is that the delta error in the output layer is transformed by the derivative of the transfer function and is then used in the previous neural layer to adjust input connection weights. In other words, this error is back-propagated into previous layers one layer at a time. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feed-forward, Back-propagation derives its name from this method of computing the error term.

When using the delta rule, it is important to ensure that the input data set is well randomized. Well ordered or structured presentation of the training set can lead to a network which can not converge to the desired accuracy. If that happens, then the network is incapable of learning the problem.

The Gradient Descent Rule: This rule is similar to the Delta Rule in that the derivative of the transfer function is still used to modify the delta error before it is applied to the connection weights. Here, however, an additional proportional constant tied to the learning rate is appended to the final modifying factor acting upon the weight. This rule is commonly used, even though it converges to a point of stability very slowly.

It has been shown that different learning rates for different layers of a network help the learning process converge faster. In these tests, the learning rates for those layers close to the output were set lower than those layers near the input. This is especially important for applications where the input data is not derived from a strong underlying model.

Kohonen's Learning Law: This procedure, developed by Teuvo Kohonen, was inspired by learning in biological systems. In this procedure, the processing elements compete for the opportunity to learn, or update their weights. The processing element with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbors. Only the winner is permitted an output, and only the winner plus its neighbors are allowed to adjust their connection weights.

Further, the size of the neighborhood can vary during the training period. The usual paradigm is to start with a larger definition of the neighborhood, and narrow in as the training process proceeds. Because the winning element is defined as the one that has the closest match to the input pattern, Kohonen networks model the distribution of the inputs. This is good for statistical or topological modeling of the data and is sometimes referred to as self-organizing maps or self-organizing topologies.

2.2 FUZZY LOGIC

According to Bauer et al [70], "Fuzzy Logic is basically a multi-valued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false, black/white, etc. Notions like rather warm or pretty cold can be formulated mathematically and processed by computers."

According to Bart Kosko [57], "The facts were always fuzzy or vague or inexact... Science treated the gray or fuzzy facts as if they were the black-white facts of math. Yet no one had put forth a single fact about the world that was 100% true or 100% false."

Logic to most people relates to two state thinking, the idea that the outcome can only be either true or false, 1 or 0, right or wrong. This form of logic dates back to ancient Greece and is perfectly adequate to answer simple questions in single dimensions, for example, if A is 1 and B is 0 what is A AND B? It can be extended, as is done in Boolean algebra to more complex questions, as long as all the parts can be described using the same restricted alphabet of two symbols. Such logic is a deductive way of understanding consequences and a highly valuable intellectual technique [67].

But this sort of logic is inadequate when we need to reason about variables that have more than two values, or in cases where multiple incompatible variables are involved. Yet we still need to make decisions in these cases, so how can we proceed?

Bivalent, or two states, logic is just a sub-set of a more powerful type of logic known as fuzzy logic.

The concept of Fuzzy Logic (FL) was conceived by Zadeh and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. Zadeh reasoned that people do not require precise, numerical information input, and yet they are capable of highly adaptive control [67], [68].

2.2.1 FUZZY SETS

Fuzzy sets have membership properties defined between 0 and 1. This means that if we take an attribute say 'red' we can express the colour of any particular apple as a position in this fuzzy set. We may say for example that it is 30% red and thus has a fuzzy truth value (FTV) or membership function of 0.3. The relation of FTV to actual values depends upon the desired mapping from the real world to the normalized range 0 to 1, and this is arbitrary.

The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. There are different membership functions associated with each input and output response [69].

The commonly used shape to describe the membership function is triangular, but bell, trapezoidal and exponential can also be used. More complex functions are possible but require greater computing overhead to implement. Fig. 2.4 illustrates the different shapes of membership functions commonly in use.

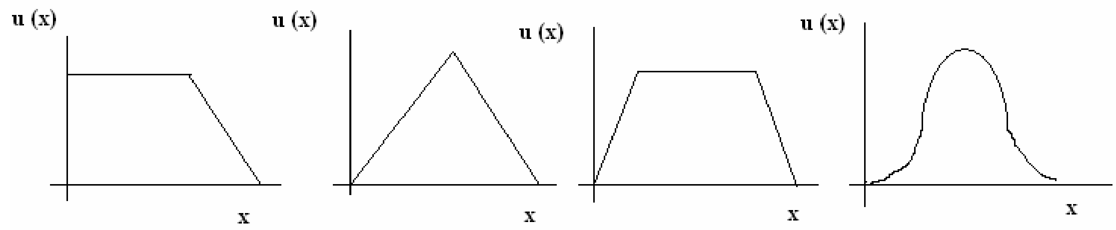

Fig.2.4: Different Shapes of Commonly Used Membership Functions

Fuzzy logic is reasoning with fuzzy sets. Operations on fuzzy sets are similar to those of standard logic but are differently defined [69]. Let us assume two FTVs to illustrate, A(0.4) and B(0.7).

Union (the joined boundarys of the values):
A OR B = Maximum of the FTVs i.e., 0.7

Intersection (the commonality between the values):
A AND B = Minimum of the FTVs i.e., 0.4
(again reducing to bivalent logic in the extremes)

Negation (the opposite of the value)
NOT A = 1 – FTV A i.e., 0.6
(Once more this is simply an extension of normal logic)

If there is just one variable then decisions are easy, the option with the best value is selected, but it is very difficult to deal with multiple variables where compromise or trade-off the values is required. In classical logic we can pick the option whose worst is the least bad (Max-min) or we could pick the option whose best is the highest (Max-max). In fuzzy logic we rate each variable as a fuzzy truth value, giving 1 to the best option, 0 to the worst and proportionate in between (we could alternatively rate them with respect to a theoretical or practical minimum and maximum for the variable in question). A motoring example is considered:

Real Values	*Consumption mpg*	*Max Speed mph*	*Acceleration s*
Car A	*30*	*120*	*9*
Car B	*40*	*110*	*11*
Car C	*45*	*100*	*12*

TABLE 2.1: A Motoring Example (Real Values)

Classical logic would set the best at 1 and rest (not-best) at 0, i.e.:

Logic Values	*Consumption*	*Max Speed*	*Acceleration*
Car A	*0*	*1*	*1*
Car B	*0*	*0*	*0*
Car C	*1*	*0*	*0*

TABLE 2.2: A Motoring Example (Logic Values)

Then Max-min would choose all of them (all are 0 minimum) and Max-max either A or C (both have 1) - not much use as a method of choice.

Fuzzifying these values instead (where for Acceleration here low is good, so the minimum gets the maximum marks) we get:

Fuzzy Values	*Consumption*	*Max Speed*	*Acceleration*
Car A	*0*	*1*	*1*
Car B	*0.66*	*0.5*	*0.33*
Car C	*1*	*0*	*0*

TABLE 2.3: A Motoring Example (Fuzzy Values)

Here Max-min would choose B (0.33 minimum satisfaction) and Max-max A (two 1s), a compromise choice is provided by fuzzy reasoning (depending on your preference for least-worst versus most-best).

Fuzzy systems being inherently nonlinear however can deal with those situations hard to formulate in traditional linear mathematical terms, and this includes complex nonlinear machines and systems with multiple interrelated variables [72].

CHAPTER-3
SHORT TERM LOAD FORECASTING

3.1 OVERVIEW

Load forecasting plays an important role in power system planning, operation and control. Forecasting is the study to estimate active loads ahead of actual load occurrence. Planning and operational applications of load forecasting requires a certain ‘lead time’ also called forecasting intervals [73]. Accurate models for electric power load forecasting are essential to the operation and planning of a utility company. Load forecasting helps an electric utility to make important decisions including decisions on purchasing and generating electric power, load switching, and infrastructure development. Load forecasts are extremely important for energy suppliers, and other participants in electric energy generation, transmission, distribution, and markets. The forecasts for different time horizons are important for different operations within a utility company.

For the next year peak forecast, it is possible to provide the probability distribution of the load based on historical weather observations. It is also possible, according to the industry practice, to predict the so-called weather normalized load, which would take place for average annual peak weather conditions or worse than average peak weather conditions for a given area. Weather normalized load is the load calculated for the so-called normal weather conditions which are the average of the weather characteristics for the peak historical loads over a certain period of time. The duration of this period varies from one utility to another. Some companies take the last 25-30 years of historical data.

3.2 CLASSIFICATION OF LOAD FORECASTING METHODS

In terms of lead time, load forecasting methods are divided into four main categories as listed below [73].

1. Very short-term load forecasting
2. Short-term load forecasting
3. Mid-term load forecasting
4. Long-term load forecasting

Short-term load forecasting can help to estimate load flows and to make decisions that can prevent overloading. Timely implementations of such decisions lead to the improvement of network reliability and to the reduced occurrences of equipment failures and blackouts. In the deregulated economy, decisions on capital expenditures based on long-term forecasting are also more important than in a non-deregulated economy when rate increases could be justified by capital expenditure projects.

Sr.No.	Nature of forecast	Lead time	Application
1.	Very short term	A few seconds to few minutes	Generation, distribution schedules, contingency analysis for system security
2.	Short-term	Half an hour to a few hours	Allocation of spinning reserve, operational planning and unit commitment, maintenance scheduling
3.	Mid-term	A few days to a few weeks	Planning for seasonal peak-summer, winter
4.	Long-term	A few months to a few years	Planning generation growth

TABLE 3.1: Load Forecasting Methods

Table 3.1 summarizes these categories with their applications.

Most of the forecasting methods use statistical techniques or artificial intelligence algorithms such as regression, neural networks, fuzzy logic and expert systems. Two of the methods, so-called end-use and econometric approach are broadly used for medium- and long-term forecasting. A variety of methods, which include the so-called similar day approach, various regression models, time series, neural networks, statistical learning algorithms, fuzzy logic, and expert systems, have been developed for short-term forecasting. As we see, a large variety of mathematical methods and ideas have been used for load forecasting. The development and improvements of appropriate mathematical tools will lead to the development of more accurate load forecasting techniques.

For short-term load forecasting several factors should be considered, such as time factors, weather data, and possible customers' classes. The medium and long-term forecasts take into account the historical load and weather data, the number of customers in different categories, the appliances in the area and their characteristics including age, the economic and demographic data and their forecasts, the appliance sales data, and other factors. The time factors include the time of the year, the day of the week, and the hour of the day. There are important differences in load between weekdays and weekends. The load on different weekdays also can behave differently. For example, Mondays and Fridays being adjacent to weekends, may have structurally different loads than Tuesday through Thursday. This is particularly true during the summer time. Holidays are more difficult to forecast than non-holidays because of their relative infrequent occurrence. The change of the weather causes the change of consumers' comfort feeling and in turn the usage of some appliances such as space heater, water heater and air conditioner.

3.3 FACTORS AFFECTING SYSTEM LOAD

The system load is the sum of all the consumers' load at the same time. The objective of system Load Forecasting is to forecast the future system load. Good understanding of the system characteristics helps to design reasonable forecasting models and select appropriate models in different situations. Various factors influence the system load behavior, which can be mainly classified into the following categories

- Weather
- Time
- Economy
- Random disturbance.

The effects of all these factors are introduced as follows to provide a basic understanding of the load characteristics.

Weather

Weather factors include temperature, humidity, wind speed, cloud cover, light intensity and so on. The change of the weather causes the change of consumers' comfort feeling and in turn the usage of some appliances such as space heater, water heater and air conditioner. Weather-sensitive load also includes appliance of agricultural irrigation due to the need of the cultivated plants. In the areas where summer and winter have great meteorological difference, the load patterns differ greatly. Fig.3.1 shows the typical different seasonal weekday load profiles of the year.

Normally the intraday temperatures are the most important weather variables in terms of their effects on the load; hence they are often selected as the independent variables in STLF. Temperatures of the previous days also affect the load profile. For example, continuous high temperature days might lead to heat buildup and in turn a new system peak. Humidity is also an important factor, because it affects the human being's comfort feeling greatly. People feel hotter in the environment of 35 and 70% relative humidity than in the environment of 37 and 50% relative humidity. That's why THI (temperature-humidity index) is sometimes employed as an affecting factor of load forecasting. Furthermore, WCI (wind chill index) is another factor that measures the cold feeling. It is a meaningful topic to select the appropriate weather variables as the inputs of STLF.

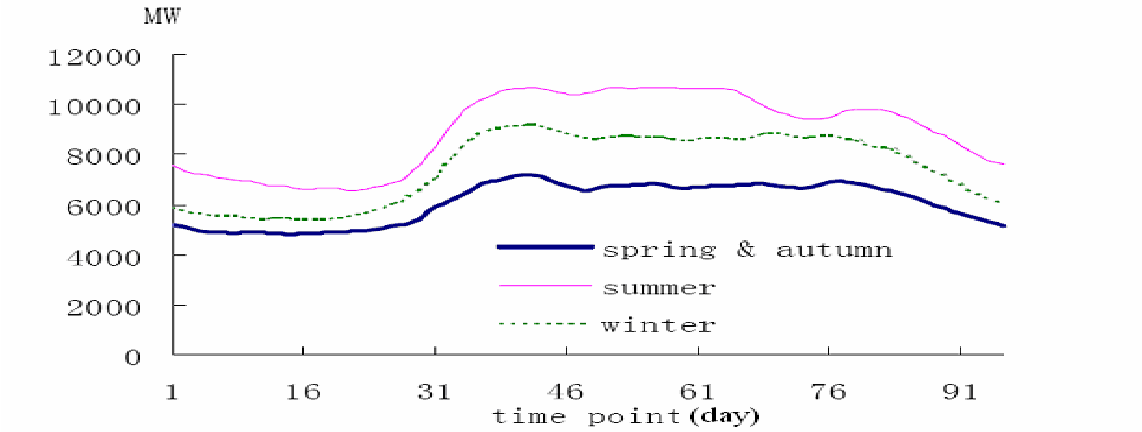

Fig.3.1: Typical Seasonal Workday

Time

Time factors influencing the load include time point of the day, holiday property, weekday/weekend property and season property. From the observation of the load curves it can be seen that there are certain rules of the load variation with the time point of the day. For example, the typical load curve of the normal winter weekdays (from Monday to Friday) is shown in Fig. 3.2, with the sample interval of 15 minutes, i.e. there are altogether 96 sample points in one day. The load is low and stable from 0:00 to 6:00; it rises from around 6:00 to 9:00 and then becomes flat again until around 12:00; then it descends gradually until 17:00; thereafter it rises again until 19:00; it descends again until the end of the day. Actually this load variation with time reflects the arrangement of people's daily life: working time, leisure time and sleeping time.

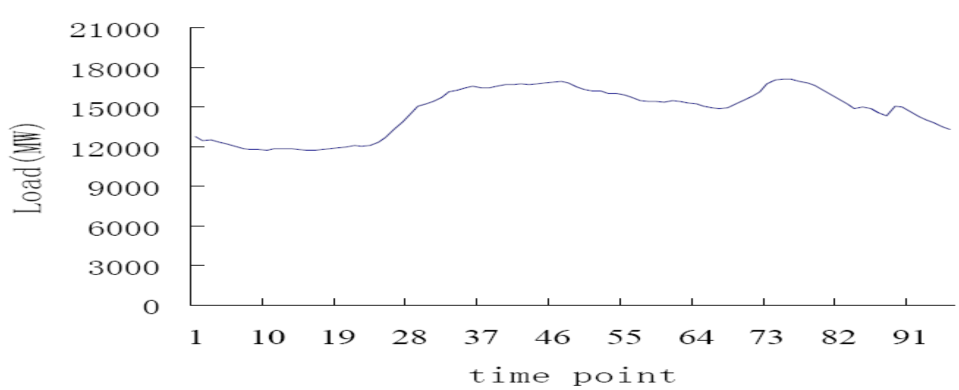

Fig.3.2: Typical Load Curve of the Normal Winter Weekdays

There are also some other rules of load variation with time. The weekend or holiday load curve is lower than the weekday curve, due to the decrease of working load. Shifts to and from daylight savings time and start of the school year also contribute to the significant change of the previous load profiles.

Periodicity is another property of the load curve. There is very strong daily, weekly, seasonal and yearly periodicity in the load data. Taking good use of this property can benefit the load forecasting result.

Economy

Electricity is a kind of commodity. The economic situation also influences the utilization of this commodity. Economic factors, such as the degree of industrialization, price of electricity and load management policy have significant impacts on the system load growth/decline trend. With the development of modern electricity markets, the relationship between electricity price and load profile is even stronger. Although time-of-use pricing and demand-side management had arrived before deregulation, the volatility of spot markets and incentives for consumers to adjust loads are potentially of a much greater magnitude. At low prices, elasticity is still negligible, but at times of extreme conditions, price-induced rationing is a much more likely scenario in a deregulated market compared to that under central planning.

Random Disturbance

The modern power system is composed of numerous electricity users. Although it is not possible to predict how each individual user consumes the energy, the amount of the total loads of all the small users shows good statistical rules and in turn, leads to smooth load curves. This is the groundwork of the load forecasting work. But the startup and shutdown of the large loads, such as steel mill, synchrotrons and wind tunnels, always lead to an obvious impulse to the load curve. This is a random disturbance, since for the dispatchers, the startup and shutdown time of these users is quite random, i.e. there is no obvious rule of when and how they get power from the grid. When the data from such a load curve are used in load forecasting training, the impulse component of the load adds to the difficulty of load forecasting. Special events, which are known in advance but whose effect on load is not quite certain, are another source of random disturbance. A typical special event is, for example, a world cup cricket match, which the dispatchers know for sure will cause increasing usage of television, but cannot best decide the amount of the usage.

3.4 SHORT TERM LOAD FORECASTING METHODS

A large variety of statistical and artificial intelligence techniques have been developed for short-term load forecasting, which include the methods employing similar day approach, regression models, time series, neural networks, expert systems and fuzzy logic approaches.

3.4.1 SIMILAR-DAY APPROACH

This approach is based on searching historical data for days with similar characteristics to the forecasted day. Similar characteristics may include day of the week, day of the year or even weather. The similar-day method may be also used for modeling the special event (e.g. holiday) component; then the search is conducted on historical data within one, two or three years. The load of a similar day is considered as a forecast. Instead of a single similar-day load, the forecast can be a linear combination or a regression procedure that can include several similar days.

A simple, yet in some cases surprisingly powerful implementation of the similar-day or naive method can be as follows: a Monday is similar to the Monday of the previous week and the same rule applies for Saturdays and Sundays; analogously, a Tuesday is similar to the previous Monday, and the same rule applies for Wednesdays, Thursdays and Fridays. This method can be used as a benchmark for more sophisticated models. Not carefully calibrated forecasting procedures surprisingly do not pass this ‘test’.

3.4.2 REGRESSION METHODS

Regression is one of the most widely used statistical techniques. The general purpose of multiple regression is to learn more about the relationship between several independent or predictor variables and a dependent or criterion variable. Multiple regression is based on least squares: the model is fit such that the sum-of-squares of differences of observed and predicted values is minimized.

For electric load forecasting regression methods are usually used to model the relationship of load and other factors such as weather, day type and customer class.

3.4.3 TIME SERIES

Time series methods are based on the assumption that the data have an internal structure, such as autocorrelation, trend, or seasonal variation. Time series forecasting methods detect and explore such a structure. Time series have been used for decades in such fields as economics, digital signal processing, as well as electric load forecasting. In particular, ARMA (autoregressive moving average), ARIMA (autoregressive integrated moving average), ARMAX (autoregressive moving average with exogenous variables), and ARIMAX (autoregressive integrated moving average with exogenous variables) are the most often used classical time series methods. ARMA models are usually used for stationary processes while ARIMA is an extension of ARMA to non-stationary processes. ARMA and ARIMA use the time and load as the only input parameters.

3.4.4 NEURAL NETWORKS

The use of artificial neural networks has been a widely studied electric load forecasting technique since 1990. Neural networks are essentially non-linear circuits its output is some linear or nonlinear mathematical function of its inputs. The inputs may be the outputs of other network elements as well as actual network inputs. In practice network elements are arranged in a relatively small number of connected layers of elements between network inputs and outputs. Feedback paths are sometimes used. In applying a neural network to electric load forecasting, one must select the architecture (e.g. Hopfield, multi layer, Boltzmann machine), the number and connectivity of layers and elements, use of bi-directional or uni-directional links and the format (e.g. binary or continuous). The most popular ANN architecture for electric load forecasting is multi layer employing back propagation algorithm. Back propagation use continuously valued functions and supervised learning. That is, under supervised learning, the actual numerical weights assigned to element inputs are determined by matching historical data (such as time and weather) to desired outputs (such as historical electric loads) in a pre-operational “training session”. Artificial neural networks with unsupervised learning do not require pre-operational training.

3.4.5 EXPERT SYSTEMS

Rule based forecasting makes use of rules, which are often heuristic in nature, to do accurate forecasting. Expert systems, incorporates rules and procedures used by human experts in the field of interest into software that is then able to automatically make forecasts without human assistance. Expert system use began in the 1960’s for such applications as geological prospecting and computer design. Expert systems work best when a human expert is available to work with software developers for a considerable amount of time in imparting the expert’s knowledge to the expert system software.

3.4.6 FUZZY LOGIC

Fuzzy logic allows one to (logically) deduce outputs from fuzzy inputs. In this sense fuzzy logic is one of a number of techniques for mapping inputs to outputs (i.e. curve fitting). Among the advantages of fuzzy logic are the absence of a need for a mathematical model mapping inputs to outputs and the absence of a need for precise (or even noise free) inputs. With such generic conditioning rules, properly designed fuzzy logic systems can be very robust when used for forecasting. After the logical processing of fuzzy inputs, a “defuzzification” process can be used to produce such precise outputs. [19], [20], [21] described applications of fuzzy logic to electric load forecasting.

CHAPTER-4

SHORT TERM LOAD FORECASTING USING FUZZY SET CLASSIFIED NEURAL NETWORK

4.1 OVERVIEW

Load forecasting is an essential part of any Utility Energy Management System. Load forecasting with lead-times, from an hour to several days, helps the power system operator in an Energy Control Center to perform necessary scheduling functions such as Hydro-Thermal Coordination, Power System Network Analysis such as Power Flow and Optimal Power Flow which provide useful information for possible energy interchange with other utilities. In addition to these reasons, load forecasting is also useful for system security assessment. The following characteristics make electric load forecasting a challenging task:

1. Actual load of the power system at any point in time depends on a number of factors, all of which cannot be accurately predicted.

2. The load demand changes cyclically in response to the seasonal variations.

3. In general, the overall load demand of the system continually increases.

4. The load follows a definite pattern for each day of the week, but changes considerably over the weekends and public holidays.

Two basic load models have been used for short term load forecasting: the peak load model and the load curve model. The peak load models are not widely used since they do not give any information about the shape of the load curve. Various techniques for power system load forecasting have been proposed in the past few decades.

The application of the artificial neural network (ANN) to short-term load forecasting has gained a great deal of interest and several researchers have reported the effectiveness of the ANN approach [75], [76], [77]. Unlike the previous techniques, the ANN learns the patterns from the inputs and outputs of the utilities' system and then, it creates its own non-linear models that are used to predict the short term loads. The ANN input data are stored in the following way, the hourly historical data are subdivided into different classes based on the weather conditions using the concept of fuzzy set theory. For each class of data, the ANN creates a non-linear model which forecasts the hourly system load.

The classifications can be done based on several characteristics such as season (spring, winter, summer, and rainy), day (Monday, Tuesday, Wednesday ...), time (1 a.m. to 5 a.m., 6 a.m. to 10 a.m. ...) [76]-[80].

4.2 THE PROPOSED TECHNIQUE

The hourly historical data of the weather conditions and the load are classified according to their characteristics and are used by the ANN to create a non-linear model for each class. These non-linear models are then used to forecast the short-term (hourly) loads. The classification of load data is accomplished by fuzzy set techniques.

The Fig. 4.1 shows the block diagram representation of the proposed method. The inputs to the fuzzy set based classifier are hourly data of weather information i.e., Temperature and Relative Humidity and day type information. Pre-Holiday and Post-Holiday categories are made keeping in view that Holiday effect on load can be seen the days before and after the holiday. This classifier converts the data into various classes. Each class uses the training data of that particular class to train the neural network and produces system load as output depending upon the input set. Neural networks are trained using Back-propagation algorithm with Delta learning rule. A sigmoid transfer function is used in this method. The fuzzy set based classification and Neural Network structure and training are elaborated in sections 4.2.1 and 4.2.2 respectively.

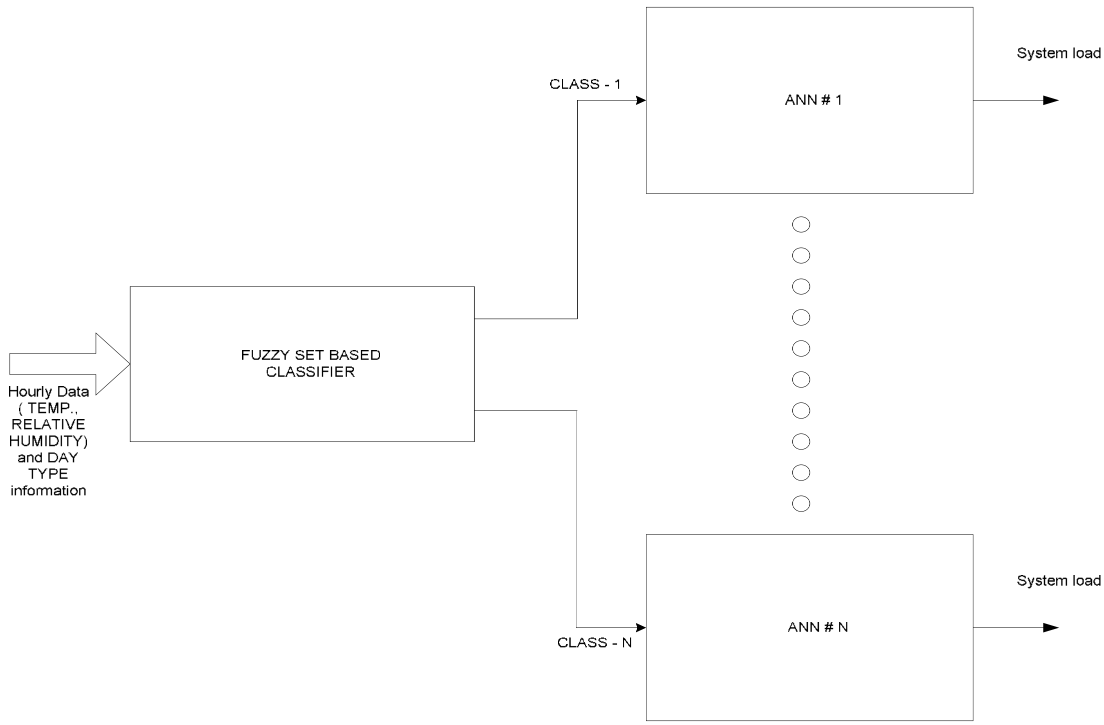

Fig. 4.1: Block Diagram Representation of the Proposed Technique

4.2.1 FUZZY SET BASED CLASSIFICATION

Fuzzy logic technique is found to be most promising technique for classification of huge data. In the proposed methodology concepts of Fuzzy Logic set theory are used to classify the huge amount of historical data for Temperature, Relative Humidity and Day type.

Temperature data is fuzzified into five main fuzzy sets described as

1. Very Cold (VC),
2. Cold (C),
3. Normal (N),
4. Hot (H) and
5. Very Hot (VH).

Fig. 4.2 shows the membership function representation for temperature using trapezoidal shaped membership functions.

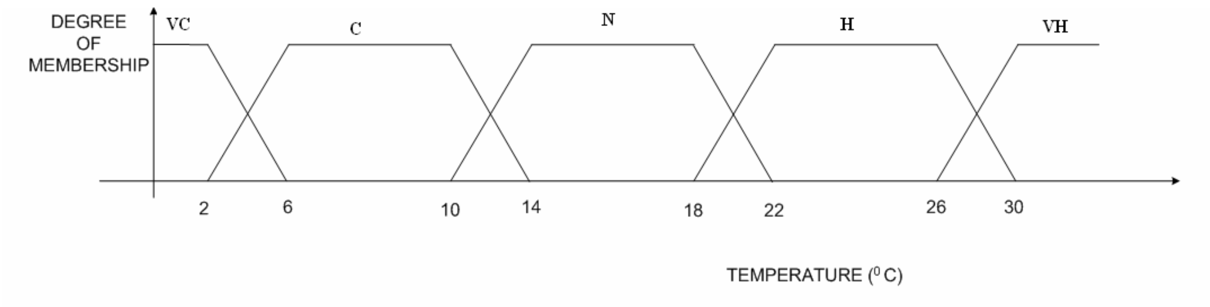

Fig. 4.2: Membership Function Representation for Temperature (^{0}C)

Relative Humidity is fuzzified into four main fuzzy sets described as

1. Very Dry (VD),
2. Dry (D),
3. Humid (H) and
4. Very Humid (VH).

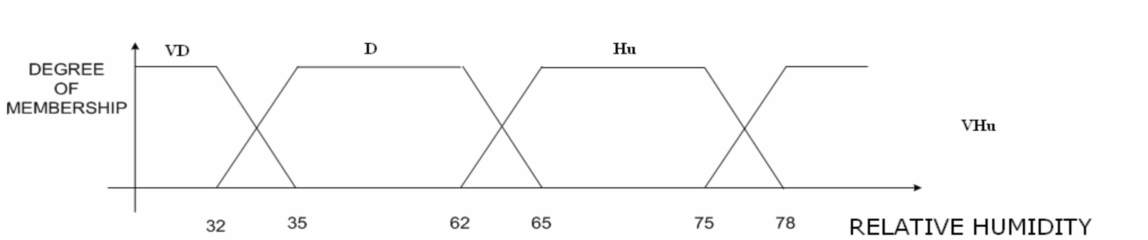

Fig. 4.3: Membership Function Representation for Relative Humidity

Fig. 4.3 shows membership function representation for Relative Humidity. Trapezoidal shaped membership functions are used to represent relative humidity.

Day Type is fuzzified into four main fuzzy sets described as

1. Post-Holiday (PostH),
2. Weekday (WD),
3. Pre-Holiday (PreH) and
4. Holiday (H).

Fig. 4.4 shows the membership function representation for Day Type.

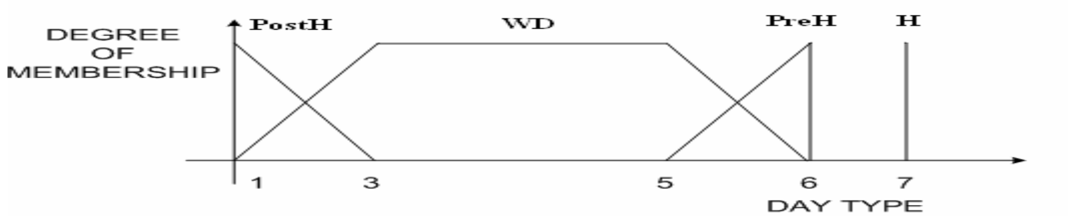

Fig. 4.4: Membership Function Representation for Day Type

DAY TYPE →		PostH	WD	PreH	H
TEMP.↓	**RH↓**				
VC	**VD**	1	2	3	4
VC	**D**	5	6	7	8
VC	**Hu**	9	10	11	12
VC	**VHu**	13	14	15	16
C	**VD**	17	18	19	20
C	**D**	21	22	23	24
C	**Hu**	25	26	27	28
C	**VHu**	29	30	31	32
N	**VD**	33	34	35	36
N	**D**	37	38	39	40
N	**Hu**	41	42	43	44
N	**VHu**	45	46	47	48
H	**VD**	49	50	51	52
H	**D**	53	54	55	56
H	**Hu**	57	58	59	60
H	**VHu**	61	62	63	64
VH	**VD**	65	66	67	68
VH	**D**	69	70	71	72
VH	**Hu**	73	74	75	76
VH	**VHu**	77	78	79	80

TABLE 4.1: Numbers Assigned to Various Classes

The categories of temperature, relative humidity and day type are then used to form eighty classes of weather conditions and day type information such as Very Cold-Very Dry-Post Holiday (VC-VD-PostH), Very Cold-Very Dry-Weekday (VC-VD-WD), Very Cold-Very Dry-Pre-Holiday (VC-VD-PreH), Very Cold-Very Dry-Holiday (VC-VD-H). The numbers from 1-80 are assigned to the eighty classes and shown in Table 4.1.

4.2.2 NEURAL NETWORK STRUCTURE AND TRAINING

The Artificial Neural Network used in this study, is a Multi-Layered Feed-forward network using a back-propagation algorithm. As shown in Fig. 4.5, the ANN architecture consists of three layers: an input layer, a hidden layer, and an output layer. Neurons in different layers are connected by the interconnecting weights W_{kj} and the output from each neuron is multiplied by its corresponding weight before reaching the inputs of the neurons in the next layer. Each neuron consists of an activation function which is used to determine the output of the neuron from its inputs. All inputs to each hidden layer neuron are summed to make an activation function for the neuron. Likewise, the sum of all inputs to each output neuron makes the neuron activation function. For each neuron k in the hidden layer and neuron l in the output layer, the net inputs are computed as the weighted sum of all the inputs of that neuron.

After the historical data is stored in classes of weather conditions and day type, the ANN is assigned to each class in order to learn and perform a forecast. The training set for the ANN of any corresponding class, i.e., the hourly data whose membership values of the corresponding temperature, humidity and day type categories are not zero. For instance, if the weather condition of interest is Cold-Dry-Weekday (C-D-WD) then any hourly data whose membership values, $\mu_C \neq 0$, $\mu_D \neq 0$ and $\mu_{WD} \neq 0$ are selected and used to produce the training set for this class. Furthermore, this process provides each ANN a chance to learn not only from the data that are most likely belong to its corresponding class, but also some data that are less likely but still have some degree of belonging in that class.

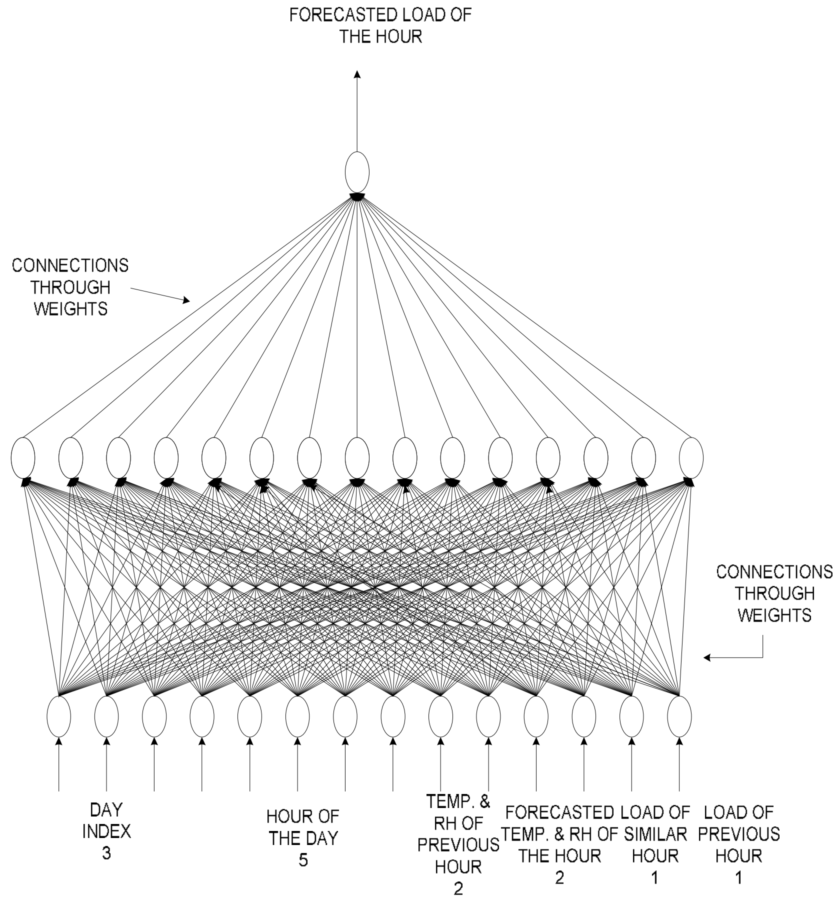

Fig. 4.5: Neural Network Structure used in the Proposed Method

In the proposed method of short term load forecasting, input variables taken are those having a strong influence on the system load. These variables include weather information, day and time information and load. These are as:

1. Day of the week. This input variable is presented as three digit binary code from 001 to 111 (001 is for Monday while 111 is for Sunday)
2. Hour of the day. This input variable is presented as five digit binary code from 00001 to 11000 (00001 is for first hour of the day i.e., 01:00 am while 11000 is for last hour of the day i.e., 12:00 midnight)

3. Temperature (normalized) and RH (normalized) of previous hour to the hour for which the load is to be forecasted
4. Forecasted Temperature (normalized) and RH (normalized) of the hour for which the load is to be forecasted
5. Load (normalized) of similar hour
6. Load (normalized) of previous hour

There are 14 input variables for the ANN and the output variable of the ANN is the hourly system load. Thus, the architecture of the ANN has 14 inputs, 1 output and 15 hidden neurons as shown in Fig. 4.5.

4.3 THE ALGORITHM

The algorithm for the proposed technique is implemented in two parts namely data classification and training and load forecasting.

4.3.1 ALGORITHM FOR DATA CLASSIFICATION

Training data preparation and classification based on fuzzy set based classification technique is the first part in the implementation of the technique.

Step 1: Enter the training set including temperature, relative humidity and day type.

Step 2: Fuzzify temperature, relative humidity and day type using their membership functions.

Step 3: Based upon the fuzzified values of temperature, relative humidity and day type, classify the data (sets).

Step 4: Tabulate the data according to their classes and then stop.

The Fig. 4.6 shows the flowchart for the data classification.

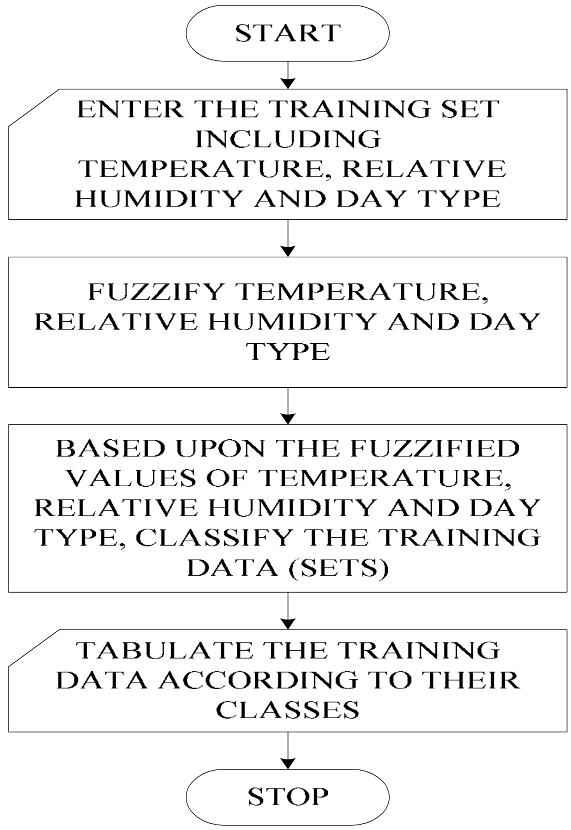

Fig. 4.6: Flowchart for Data Classification

4.3.2 ALGORITHM FOR TRAINING AND LOAD FORECASTING

The second part of the implementation of the technique includes training of Neural Network to forecast hourly loads based upon the classified data. The algorithm steps are:-

Step 1: Enter the input vector set including weather information i.e., temperature, relative humidity and day type information.

Step 2: Based upon the weather information and day type information provided select the class form the training data.

Step 3: Using the training data of that class train the Neural Network using back-propagation algorithm and display the final weights. Back-propagation algorithm is discussed in appendix.

Step 4: Forecast the hourly load using these final weight matrices and input vector provided and then stop.

The above steps are shown as flowchart in Fig. 4.7.

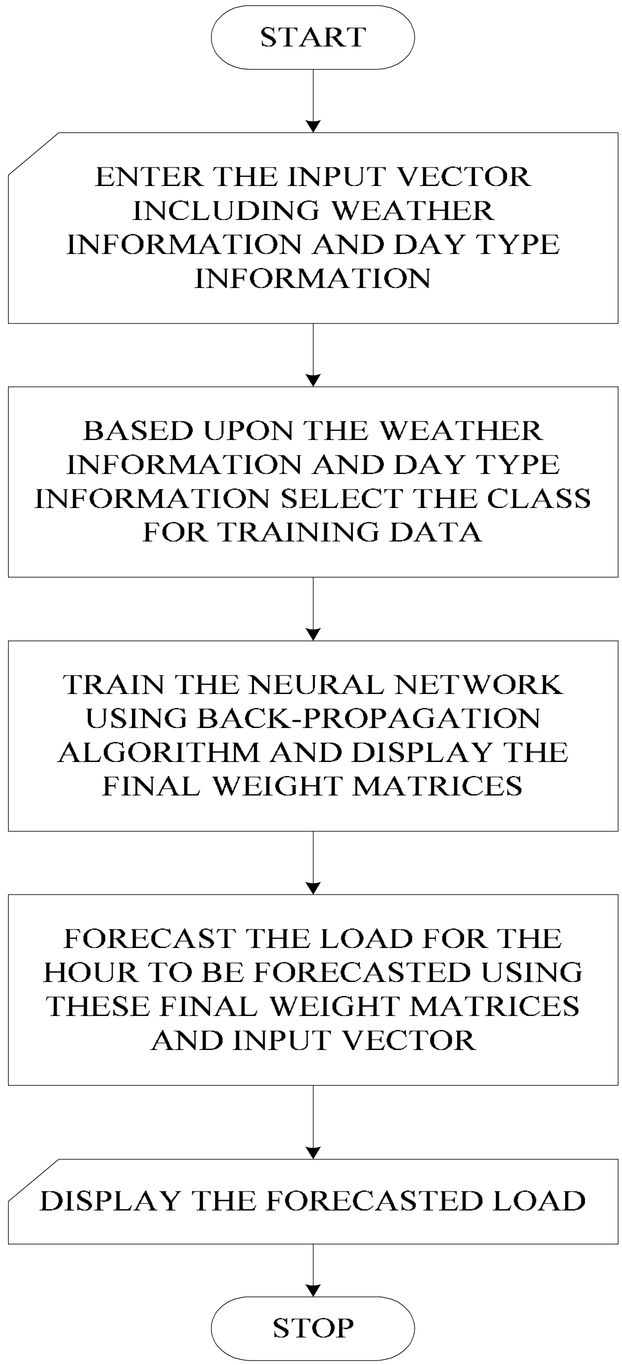

Fig. 4.7: Flowchart for Training and Load Forecasting

CHAPTER-5

RESULTS AND DISCUSSIONS

Fuzzy set classified neural network approach to short term load forecasting is proposed in this thesis work. The hourly load data from Himachal Pradesh State Electricity Board and Punjab State Electricity Board for different day types is used for training and load forecasting. HPSEB data is the complete load / demand of Himachal Pradesh taken from State Load Dispatch and Communication Center, Shimla and the PSEB data pertains to an industrial feeder taken from a sub-station at Mahilpur near Hoshiarpur. The Training data is classified with the help of fuzzy set based classification technique as discussed in Chaper-4. The forecasted load is compared with the actual load and percentage error is also calculated. Forecasted weather data and day type for the days for which the load is to be forecasted is used for classification. Training data from same class is used to train the network for forecasting load for that particular class. The following five cases are investigated to validate the proposed methodology.

5.1 CASE-I: LOAD FORECASTING FOR POST HOLIDAY & SUMMER

In this case, hourly load for Himachal Pradesh State Electricity Board (HPSEB) is forecasted for a post-holiday & summer day i.e. Monday of Summer (April).

Training data includes an input vector and corresponding target output. Input vector is of 14 bit string. It includes, first 5-bits defining hour of the day (e.g., 00001 for 1st hour, 00010 for 2nd hour of the day and so on), next 3-bits defining the day of the week (e.g., 001 for Monday, 010 for Tuesday and so on), next 2-bits defining normalized values of temperature of previous hour and forecasted temperature for the hour for which load is to be forecasted, next 2-bits defining normalized values of relative humidity of previous hour and forecasted relative humidity for the hour for which load is to be forecasted, next 2- bits defining normalized values of load of previous hour and similar hour. Target output is the normalized value of load of that hour. Training data used for this case (summer and post-holiday) is given in Table 5.1. The table includes temperature (^{0}C), relative humidity (%), class and load (MW) information.

Time (Hrs.)	7-Apr-08				21-Apr-08				28-Apr-08			
	Temp. (^{0}C)	R.H. %	Class	Load (MW)	Temp. (^{0}C)	R.H. %	Class	Load (MW)	Temp. (^{0}C)	R.H. %	Class	Load (MW)
1	21	52	53	591	22	32	49	621	22	32	49	620
2	20	53	53	578	21	36	53	610	21	36	53	611
3	19	57	37	576	20	38	53	599	20	38	53	600
4	18	62	37	585	18	59	37	603	19	43	37	598
5	16	68	41	583	17	60	37	599	19	46	37	596
6	16	70	41	657	17	62	37	666	19	45	37	662
7	15	73	41	812	22	47	53	736	22	47	53	732
8	15	74	41	884	24	42	53	757	24	42	53	750
9	19	62	37	857	28	41	69	730	28	41	69	728
10	22	53	53	814	30	46	69	722	31	36	69	719
11	25	44	53	773	33	32	65	696	33	32	65	697
12	28	35	69	747	33	29	65	691	35	28	65	692
13	31	28	65	745	33	30	65	684	37	22	65	688
14	32	27	65	708	34	25	65	667	38	16	65	669
15	32	25	65	683	35	23	65	658	38	17	65	656
16	32	25	65	687	33	23	65	668	37	16	65	670
17	32	25	65	669	31	26	65	657	37	17	65	658
18	32	26	65	662	35	18	65	628	35	18	65	625
19	30	29	65	651	30	18	65	587	30	18	65	590
20	27	36	53	729	28	27	65	643	28	27	65	640
21	24	42	53	749	23	32	49	687	27	32	49	681
22	23	46	53	685	25	33	49	632	25	33	49	633
23	22	48	53	652	26	43	53	638	25	37	53	642
24	21	49	53	666	23	37	53	626	23	37	53	631

TABLE 5.1: Training Data for Case-I (Summer and Post holiday)

For classification of training data, weather information (Temperature and Relative Humidity) and day type information is used. For above mentioned input string, Day type is Post Holiday (as day is Monday), temperature is High (as temperature is 22^0C) and on the basis of relative humidity, the weather for the hour is Very Dry (as percentage relative humidity is 32 %). So the class for the hour is Post Holiday – High – Very Dry. From the Table 4.1, the number assigned to this class of weather information and day type information is 49.

Similarly, class number for the hour for which the load is to be forecasted is found on the basis of the forecasted temperature and forecasted relative humidity and day type. For example, the above mentioned training data is to be used to train the neural network to forecast the load for first hour of 28-04-2008. Forecasted temperature and forecasted relative humidity are 22^0C (High) and 32 % (Very Dry) respectively. Day type for the day is Post-Holiday. So, the number assigned to this class is 49 as shown in Table 4.1. As the class for this hour is same as that of the training set mentioned above, so the above training set can be used to train the neural network.

The input string for training of neural network for the first hour of case-1 is as

{**0 0 0 0 1 0 0 1 0.21 0.22 0.51 0.32 0.636 0.638**}

This training data is for the first hour of Monday having previous hour temperature of 21^0C and forecasted temperature for the same hour equal to 22^0C. Relative humidity for previous hour is 51 % and forecasted relative humidity for same hour is 32 %. The load for previous hour is 636 MW and for similar hour is 638 MW. Corresponding target string for above input string is {**0.621**} i.e., the target load for the hour is 621 MW.

When the neural network is trained, its weight matrices are frozen and input string is applied to the trained neural network corresponding to the first hour of 28-04-2008 to forecast. This string is as

{**0 0 0 0 1 0 0 1 0.23 0.22 0.3 0.32 0.656 0.638**}

The corresponding output of neural network is{**0.620**}i.e., the forecasted load for the first hour of 28-04-2008 is 620 MW. Forecasted results for twenty four hours for case-I are shown in Table 5.2 for 28-04-2008.

Fig. 5.1 shows the graphical comparison of the actual and forecasted load. The forecasted load follows closely to the actual load. A maximum of -0.95% error is observed for this particular example.

28/04/08 MONDAY

TIME (Hrs.)	ACTUAL LOAD (MW)	FORECASTED LOAD (MW)	DIFFERENCE (MW)	CLASS	% ERROR
1	620	620	0	49	0.00
2	611	609	-2	53	-0.33
3	600	598	-2	53	-0.33
4	598	602	4	37	0.67
5	596	597	1	37	0.17
6	662	664	2	37	0.30
7	732	735	3	53	0.41
8	750	756	6	53	0.80
9	728	729	1	69	0.14
10	719	721	2	69	0.28
11	697	695	-2	65	-0.29
12	692	690	-2	65	-0.29
13	688	683	-5	65	-0.73
14	669	666	-3	65	-0.45
15	656	657	1	65	0.15
16	670	667	-3	65	-0.45
17	658	656	-2	65	-0.30
18	625	627	2	65	0.32
19	590	586	-4	65	-0.68
20	640	642	2	65	0.31
21	681	686	5	49	0.73
22	633	631	-2	49	-0.32
23	642	637	-5	53	-0.78
24	631	625	-6	53	-0.95

TABLE 5.2: Forecasted Load for Summer and Post-Holiday (HPSEB) for 28-04-2008 (Monday)

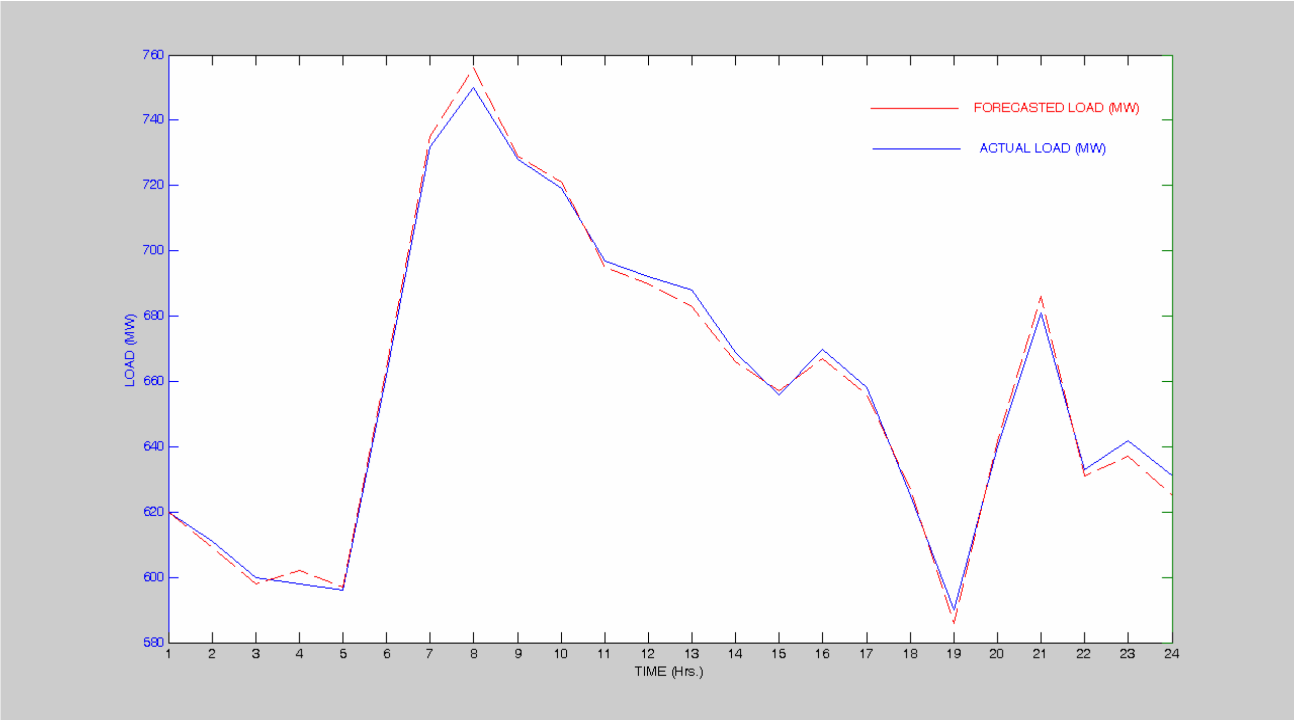

Fig. 5.1: The Actual and Forecasted Loads for Monday 28-04-2008 (Summer and Post Holiday)

5.2 CASE-II: LOAD FORECASTING FOR PRE HOLIDAY & SUMMER

In this case, hourly load for Himachal Pradesh State Electricity Board (HPSEB) is used and forecasting is done for a pre-holiday & summer day i.e., Saturday of Summer (May).

The training data used for this case (summer and pre-holiday) is given in Table 5.3. As discussed in section 5.1, the input / output strings have the similar structure and the classification is attempted in similar way. As expected, the difference is visible in the class numbers as it pertains to pre holiday and summer.

Time (Hrs.)	3-May-08 (Saturday)				10-May-08 (Saturday)				17-May-08 (Saturday)				24-May-08 (Saturday)			
	Temp. (^{0}C)	R.H. %	Class	Load (MW)	Temp. (^{0}C)	R.H. %	Class	Load (MW)	Temp. (^{0}C)	R.H. %	Class	Load (MW)	Temp. (^{0}C)	R.H. %	Class	Load (MW)
1	28	42	71	657	29	54	71	673	26	58	55	647	24	57	55	643
2	27	43	55	652	28	50	71	660	25	60	55	633	24	58	55	620
3	26	44	55	649	27	47	55	656	24	64	59	631	23	60	55	585
4	25	44	55	636	26	51	55	644	25	65	59	633	22	61	55	548
5	26	43	55	641	25	56	55	655	26	66	59	653	19	68	43	546
6	27	42	55	702	26	61	55	700	24	67	59	681	18	73	43	579
7	29	40	71	762	25	65	59	746	25	63	55	767	23	70	59	666
8	31	36	71	778	24	68	59	781	27	58	55	779	24	61	55	707
9	32	33	67	752	23	72	59	775	28	53	71	744	26	48	55	698
10	34	30	67	773	25	60	55	776	32	49	71	794	29	42	71	697
11	37	25	67	761	27	51	55	771	33	41	71	765	36	39	71	642
12	39	21	67	739	31	45	71	755	35	38	71	767	39	34	71	680
13	39	21	67	735	33	40	71	745	37	30	67	743	41	32	67	705
14	40	20	67	728	34	38	71	732	38	28	67	710	42	30	67	685
15	40	20	67	725	35	33	67	749	40	26	67	733	40	29	67	719
16	39	21	67	732	35	33	67	750	40	25	67	734	40	29	67	730
17	39	20	67	725	34	32	67	753	40	24	67	723	39	28	67	729
18	38	19	67	699	34	32	67	719	41	23	67	694	34	27	67	706
19	36	25	67	537	32	35	71	648	38	30	67	652	34	32	67	663
20	35	31	67	607	30	41	71	654	35	37	71	611	32	47	71	583
21	33	24	67	719	29	49	71	749	33	44	71	725	29	58	71	721
22	31	42	71	691	28	52	71	704	32	40	71	704	28	60	71	687
23	29	48	71	702	27	55	55	708	31	38	71	687	26	59	55	677
24	28	57	71	685	27	58	55	698	33	34	71	644	25	60	55	677

TABLE 5.3: Training Data for Case-II (Summer and Pre holiday)

The Table 5.4 gives the forecasted hourly load for Saturday 31-05-2008 (Summer & pre-holiday).

31-05-08 (SATURDAY)

TIME (HRS.)	ACTUAL LOAD (MW)	FORECASTED LOAD (MW)	DIFFERENCE (MW)	CLASS	% ERROR
1	638	641	3	55	0.47
2	624	619	-5	55	-0.80
3	580	584	4	55	0.69
4	572	579	7	55	1.22
5	547	545	-2	43	-0.37
6	579	578	-1	43	-0.17
7	658	665	7	59	1.06
8	700	706	6	55	0.86
9	692	697	5	55	0.72
10	690	696	6	71	0.87
11	645	641	-4	71	-0.62
12	673	679	6	71	0.89
13	700	704	4	67	0.57
14	680	684	4	67	0.59
15	712	717	5	67	0.70
16	720	728	8	67	1.11
17	723	729	6	67	0.83
18	701	703	2	67	0.29
19	659	662	3	67	0.46
20	582	580	-2	71	-0.34
21	716	720	4	71	0.56
22	681	687	6	71	0.88
23	673	676	3	55	0.45
24	673	676	3	55	0.45

TABLE 5.4: Forecasted Load for Summer and Pre Holiday (HPSEB) for 31-05-2008 (Saturday)

The Fig. 5.2 shows the graphical comparison of the actual and forecasted load. The forecasted load closely matches the actual load. A maximum of 1.22% error is observed for this particular example.

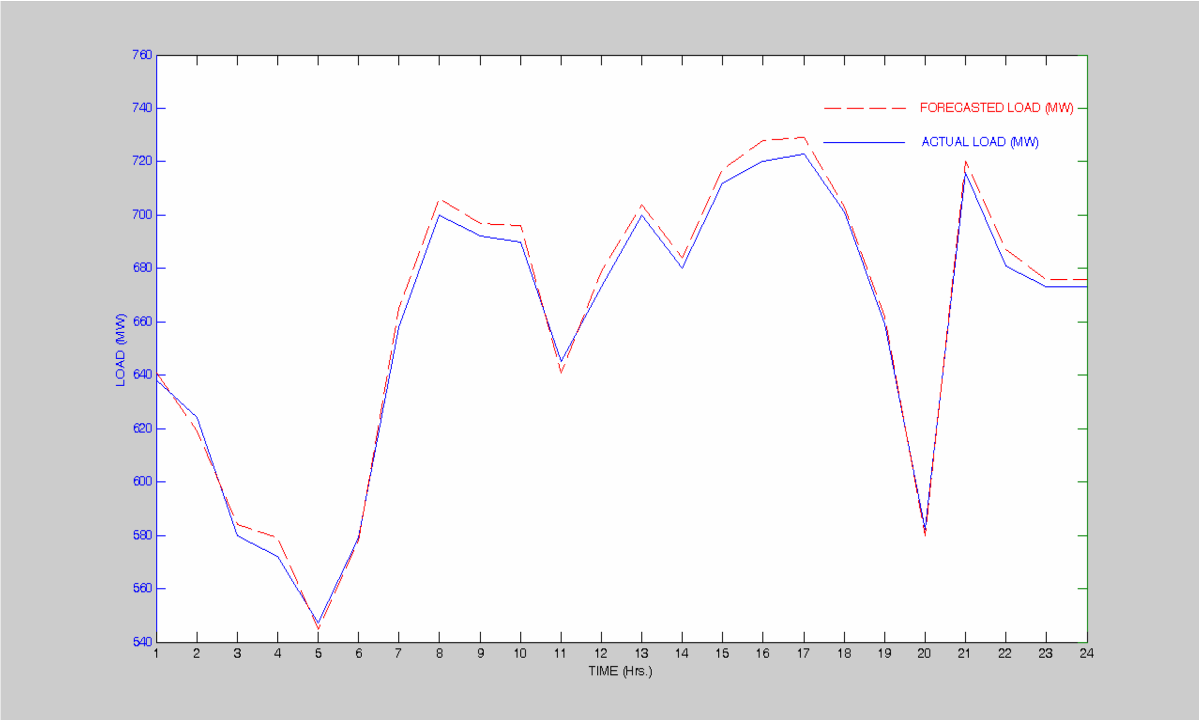

Fig. 5.2: The Actual and Forecasted Loads for Saturday 31-05-2008 (Summer & Pre holiday)

5.3 CASE -III: LOAD FORECASTING FOR WEEKDAY AND WINTER

In this case, hourly load for Punjab State Electricity Board (PSEB) is forecasted for Wednesday, 28-02-2007 a weekday and winter day.

As mentioned in section 5.1, the input has similar 14-bit string and similar output string. Similarly the classification of training data is done. Because of change in temperature and relative humidity etc. the class numbers are different.

The training data used for this case (summer and pre-holiday) and the corresponding classes obtained is summarized in Table 5.5.

Time (Hrs.)	6/2/2007 (TUESDAY)				7/2/2007 (WEDNESDAY)				9/2/2007 (FRIDAY)				13/2/2007 (TUESDAY)			
	Temp. (^{0} C)	R.H. %	CLASS	LOAD (MW)	Temp. (^{0} C)	R.H. %	CLASS	LOAD (MW)	Temp. (^{0} C)	R.H. %	CLASS	LOAD (MW)	Temp. (^{0} C)	R.H. %	CLASS	LOAD (MW)
1	16	82	46	92	16	81	46	96	14	90	46	90	12	93	46	68
2	17	78	46	91	15	82	46	95	14	91	46	89	11	93	30	67
3	16	79	46	90	14	85	46	94	14	92	46	88	11	94	30	66
4	17	75	42	90	14	87	46	93	14	92	46	92	11	94	30	66
5	16	78	46	90	13	87	46	94	14	92	46	92	11	95	30	66
6	16	80	46	97	13	88	46	101	13	93	46	99	12	95	46	71
7	15	87	46	114	13	89	46	119	13	94	46	116	11	94	30	84
8	15	84	46	137	13	90	46	142	13	94	46	139	11	95	30	86
9	16	78	46	142	14	89	46	154	14	95	46	148	11	95	30	88
10	18	74	42	145	14	92	46	151	15	88	46	143	11	96	30	90
11	19	70	42	141	15	93	46	147	17	82	46	139	11	97	30	94
12	22	60	54	136	17	84	46	141	19	76	42	134	13	94	46	96
13	23	58	54	134	18	78	46	140	19	78	46	132	12	92	46	98
14	24	56	54	129	21	66	58	134	19	79	46	131	12	94	46	100
15	24	54	54	127	21	59	54	132	20	76	58	129	13	91	46	110
16	26	49	54	127	22	59	54	132	21	72	58	129	14	86	46	106
17	25	49	54	128	22	59	54	133	20	74	58	130	14	85	46	104
18	23	57	54	121	21	64	58	126	19	80	46	119	15	82	46	100
19	21	66	58	121	19	71	42	126	19	82	46	119	14	84	46	89
20	19	73	42	150	17	77	46	156	18	83	46	150	13	82	46	89
21	18	74	42	118	15	84	46	122	18	83	46	120	13	85	46	86
22	18	75	42	112	15	88	46	117	17	84	46	114	13	87	46	82
23	17	79	46	114	14	89	46	119	17	86	46	112	12	89	46	84
24	16	80	46	99	14	89	46	103	17	85	46	97	11	90	30	73

TABLE 5.5: Training Data for Case-III (Winter and Weekday)

Table 5.6 gives the forecasted hourly load for Wednesday 28-02-2007 (Weekday and winter). The Fig. 5.3 shows the graphical comparison of the actual and forecasted load. A maximum of 4.08 % error is observed for this particular example. The higher error in this case is due to the fact that this load data is for an industrial feeder having fluctuations in load

28-02-2007

TIME (HRS.)	ACTUAL LOAD (MW)	FORECASTED LOAD (MW)	DIFFERENCE (MW)	CLASS	% ERROR
1	93	94	1	46	1.08
2	92	93.5	1.5	46	1.63
3	92	92.75	0.75	46	0.82
4	90	89.47	-0.53	42	-0.59
5	88	89.46	1.46	42	1.66
6	94	96.22	2.22	46	2.36
7	99	98.38	-0.62	46	-0.63
8	100	99.22	-0.78	46	-0.78
9	150	153.16	3.16	46	2.11
10	150	150	0	46	0.00
11	144	144.6	0.6	42	0.42
12	146	146.2	0.2	46	0.14
13	138	140.8	2.8	46	2.03
14	132	128.8	-3.2	46	-2.42
15	100	100	0	46	0.00
16	110	109.82	-0.18	46	-0.16
17	102	100.4	-1.6	46	-1.57
18	126	128	2	46	1.59
19	120	118	-2	46	-1.67
20	150	155.6	5.6	46	3.73
21	120	122	2	46	1.67
22	114	116	2	46	1.75
23	116	118	2	46	1.72
24	98	102	4	46	4.08

TABLE 5.6: Forecasted Load for Winter and Weekday (PSEB) for Dated 28-02-2007 (Wednesday)

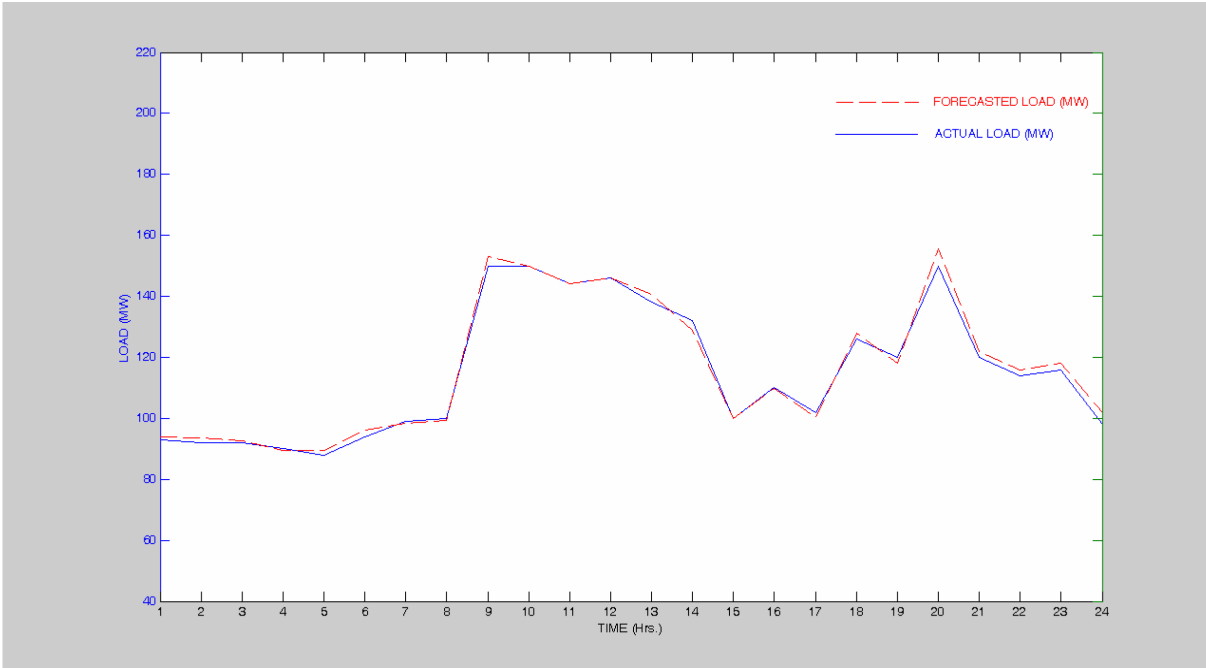

Fig. 5.3: The Actual and Forecasted Load for Wednesday 28-02-2007

5.4 CASE –IV: LOAD FORECASTING FOR HOLIDAY AND SUMMER

In this case, hourly load for Punjab State Electricity Board (PSEB) is forecasted for a holiday and summer day i.e., Sunday, 15-04-2007.

The classification of training, as given in Table 5.7, is done through the procedure outlined in Chapter-4. The forecasted hourly load for Sunday, 15-04-2007 is summarized in Table 5.8. The graphical comparison between actual and forecasted loads is shown in Fig. 5.4. A maximum of 4.86 % error is resulted. The error is higher in this case because of the reason that industrial load is fluctuating in nature and this load pertains to an industrial feeder.

Time (Hrs.)	1/4/2007 (SUNDAY)				8/4/2007 (SUNDAY)			
	Temp. (^{0} C)	R.H. %	CLASS	LOAD (MW)	Temp. (^{0} C)	R.H. %	CLASS	LOAD (MW)
1	24	75	60	62	25	53	56	64
2	24	73	60	60	23	58	56	62
3	25	67	60	60	22	61	56	62
4	25	64	60	62	23	60	56	64
5	25	65	60	62	22	62	56	64
6	25	64	60	72	21	64	60	70
7	24	67	60	80	20	65	60	78
8	24	65	60	80	20	68	60	78
9	25	64	60	77	23	61	56	78
10	29	52	72	77	28	50	72	78
11	33	34	72	75	32	37	72	76
12	35	34	72	75	35	29	68	76
13	36	33	68	75	36	26	68	74
14	37	32	68	71	36	24	68	70
15	37	29	68	69	37	25	68	68
16	37	27	68	69	36	22	68	68
17	37	27	68	67	36	20	68	66
18	36	30	68	67	35	23	68	66
19	34	39	72	62	33	27	68	60
20	30	47	72	82	30	34	72	80
21	28	52	72	70	28	39	72	68
22	27	55	56	63	26	42	56	64
23	25	64	60	67	25	46	56	68
24	23	76	60	59	14	89	46	60

TABLE 5.7: Training Data for Case-IV (Summer and Holiday)

15-04-2007 (SUNDAY)

TIME (HRS.)	ACTUAL LOAD (MW)	FORECASTED LOAD (MW)	DIFFERENCE (MW)	CLASS	% ERROR
1	62	63.6	1.6	56	2.58
2	60	62	2	56	3.33
3	60	62	2	56	3.33
4	62	64	2	56	3.23
5	65	64	-1	56	-1.54
6	68	70	2	60	2.94
7	73	75	2	60	2.74
8	74	77.6	3.6	60	4.86
9	74	77.6	3.6	56	4.86
10	74	77.6	3.6	72	4.86
11	72	73.6	1.6	72	2.22
12	71	73.6	2.6	68	3.66
13	70	71.7	1.7	68	2.43
14	68	70	2	68	2.94
15	66	68	2	68	3.03
16	66	68	2	68	3.03
17	64	66	2	68	3.13
18	63	66	3	68	4.76
19	59	60	1	68	1.69
20	78	79.7	1.7	72	2.18
21	70	68	-2	72	-2.86
22	66	64	-2	56	-3.03
23	67	68	1	56	1.49
24	60	60	0	56	0.00

TABLE 5.8: Forecasted Load for Summer and Holiday (PSEB) for Dated 15-04-2007 (Sunday)

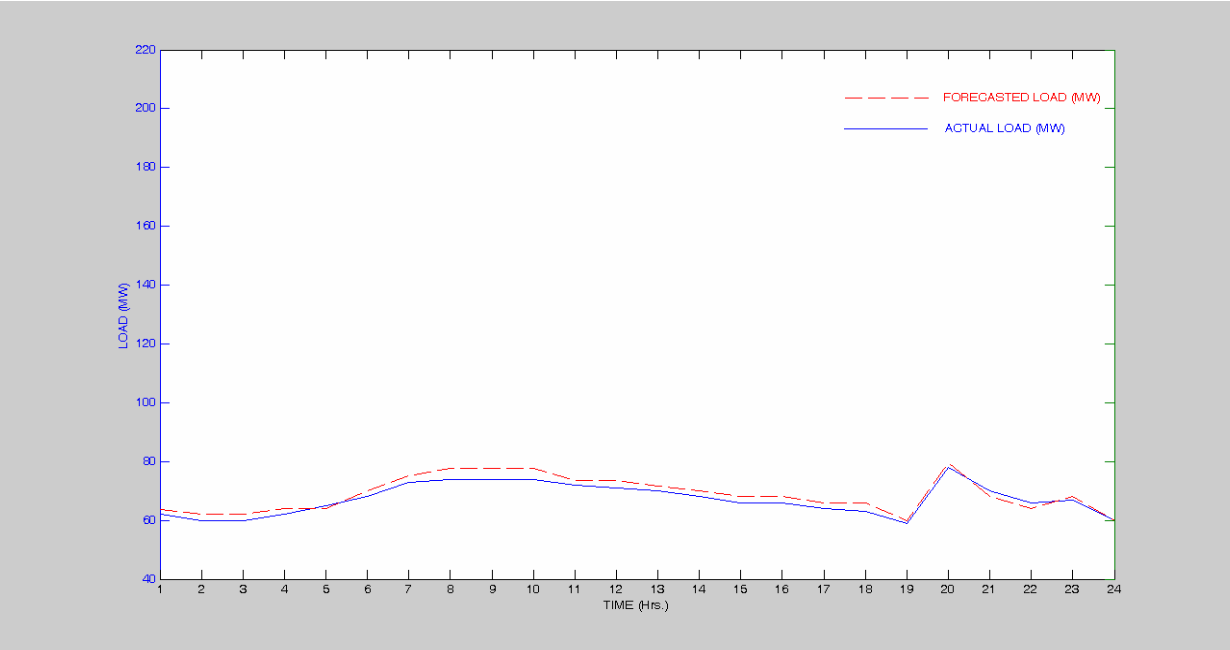

Fig. 5.4: The Actual and Forecasted Loads for Sunday 15-04-2007

5.5 CASE –V: LOAD FORECASTING FOR WEEKDAY AND RAINY SEASON

In case, hourly load for Punjab State Electricity Board (PSEB) is forecasted for a weekday and rainy season day.

The training data is classified, as given in Table 5.9, through the procedure outlined in Chapter-4. The forecasted hourly load for Wednesday, 08-08-2007 is summarized in Table 5.10. The graphical comparison between actual and forecasted loads is shown in Fig. 5.5. A maximum of 3.41 % error is resulted. The error is higher in this case because of the reason that this load pertains to an industrial feeder and industrial load is fluctuating in nature.

Time (Hrs.)	25/7/2007 (WEDNESDAY)				1/8/2007 (WEDNESDAY)			
	Temp. (^{0}C)	R.H. %	CLASS	LOAD (MW)	Temp. (^{0}C)	R.H. %	CLASS	LOAD (MW)
1	31	82	78	94	31	79	78	96
2	31	84	78	92	31	82	78	94
3	30	84	78	92	31	83	78	94
4	29	88	78	94	30	83	78	96
5	29	89	78	94	29	82	78	96
6	29	89	78	103	29	82	78	102
7	29	88	78	117	29	84	78	116
8	29	86	78	119	30	85	78	118
9	30	83	78	115	31	83	78	118
10	32	78	78	115	33	77	78	118
11	33	73	74	111	35	68	74	114
12	33	69	74	109	36	61	70	112
13	32	69	74	107	37	61	70	110
14	34	63	70	107	37	59	70	106
15	34	62	70	103	38	57	70	102
16	34	65	70	103	35	64	74	102
17	35	65	70	101	27	94	62	100
18	36	65	70	99	28	95	78	98
19	34	71	70	91	28	91	78	90
20	31	75	70	116	29	87	78	120
21	31	75	70	98	28	90	78	102
22	31	77	78	94	29	90	78	98
23	30	80	78	98	29	91	78	102
24	30	82	78	86	29	86	78	90

TABLE 5.9: Training Data for Case-V (Rainy Season and Weekday)

8/08/2007 (WEDNESDAY)

TIME (HRS.)	ACTUAL LOAD (MW)	FORECASTED LOAD (MW)	DIFFERENCE (MW)	CLASS	% ERROR
1	95	93	-2	78	-2.11
2	94	95	1	78	1.06
3	92	93	1	78	1.09
4	94	95	1	78	1.06
5	95	95	0	78	0.00
6	102	101	-1	78	-0.98
7	114	111	-3	78	-2.63
8	115	112	-3	78	-2.61
9	116	118	2	78	1.72
10	116	118	2	78	1.72
11	111	113	2	74	1.80
12	111	113	2	70	1.80
13	110	108	-2	70	-1.82
14	104	102	-2	70	-1.92
15	102	100	-2	70	-1.96
16	99	96	-3	74	-3.03
17	98	95	-3	62	-3.06
18	97	95	-2	78	-2.06
19	92	90	-2	78	-2.17
20	118	114	-4	78	-3.39
21	100	102	2	78	2.00
22	96	98	2	78	2.08
23	100	103	3	78	3.00
24	88	91	3	78	3.41

TABLE 5.10: Forecasted Load for Rainy Season and Weekday (PSEB) for 08-08-2007 (Wednesday)

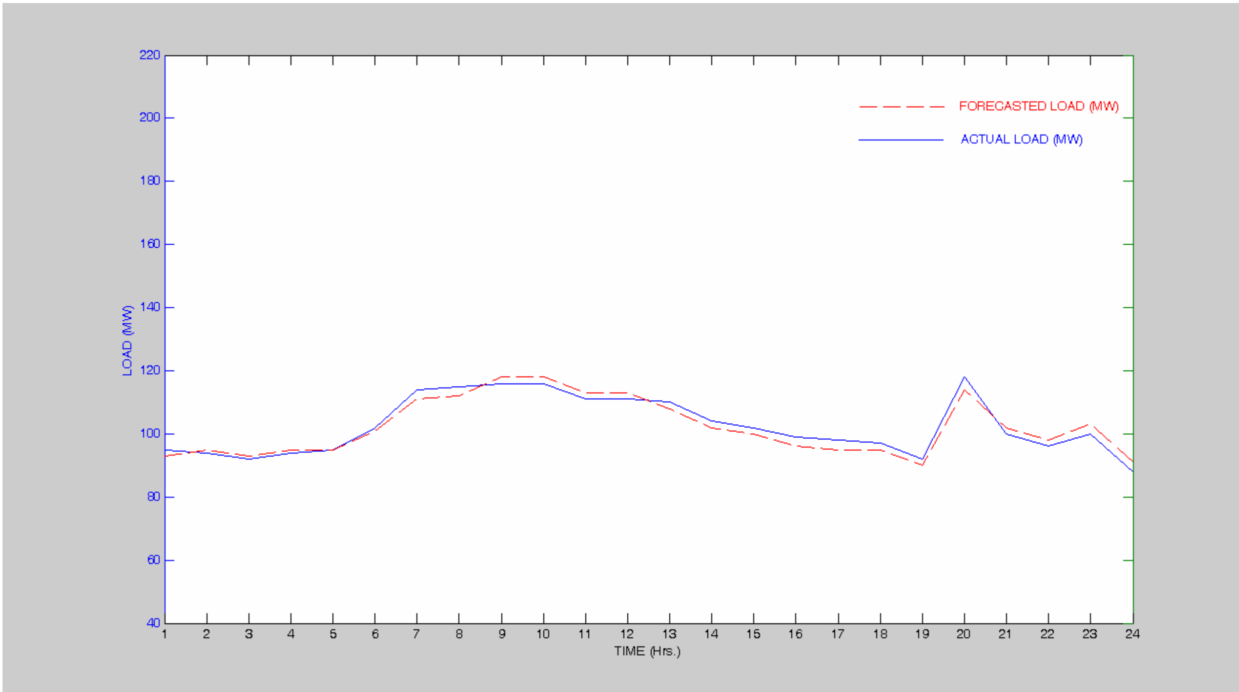

Fig. 5.5: Graphical Comparison of the Actual and Forecasted Load for Dated 08-08-2007

From the above discussed cases, it is observed that the error in case of HPSEB is around 1-1.2 % while for PSEB it is up to a maximum of 4.86%. This is because of the reason that HPSEB data is the complete load for Himachal Pradesh and the PSEB data pertains to an industrial feeder. Industrial load is fluctuating in nature and resulted in higher value of error.

CHAPTER-6

CONCLUSIONS AND FUTURE SCOPE OF WORK

6.1 CONCLUSIONS

A multi-layered feed-forward ANN combined with the fuzzy set-based classification technique for short-term electric load forecasting has been proposed in this thesis work. The hourly data was classified into classes based on the fuzzy set representation of two weather variables; temperature and relative humidity and day type information. The classification is based on the fact that the power system load is heavily influenced by the weather condition. The fuzzy set was used to assist the classification process in order to achieve the smooth transition between the classes of weather condition. After the classification, the neural network is trained for various classes by using the actual load data. For training, the supervised learning using BP algorithm is used for the training. The effectiveness is tested for the following five cases:-

- Case-1: Summer and Post-Holiday
- Case-2: Summer and Pre-Holiday
- Case-3: Winter and Weekday
- Case-4: Summer and Holiday
- Case-5: Rainy season and Weekday

The forecasted load follows the actual load. Among all the cases a maximum of 4.86 % of error is observed, which can even be lowered if a large training data is used to train the neural network and industrial fluctuations are also considered while forecasting load for industrial feeders.

6.2 FUTURE SCOPE OF WORK

The error in the actual and forecasted load can even be lowered if a large and accurate training data is used for classification and to train the neural network. Due to various reasons, the historical database might have some bad data, which is away from their actual values. The existence of bad data in historical load curve affects the precision of load forecasting results. So, while carrying out studies on STLF using neural networks, bad data should be detected and eliminated. The load forecasting may be extended to include the industrial parameters like industrial growth, GDP, urbanization, per capita income etc. After carrying thesis work in Short Term Load Forecasting, the following guidelines seem to be worth pursuing in this area:

- Detection and removal of bad data from the historical data.
- Using industrial parameters like industrial growth, GDP, urbanization, per capita income etc.
- Using other weather parameters like wind speed, sky cover and rainfall etc.

REFERENCES

[1] R.F. Engle, C. Mustafa and J. Rice, “Modeling Peak Electricity Demand”, Journal of Forecasting, vol. 11, no. 3, pp. 241–251, 1992.

[2] O. Hyde and P.F. Hodnett, “An Adaptable Automated Procedure for Short-Term Electricity Load Forecasting” IEEE Transactions on Power Systems, vol. 12, no. 1, pp. 84–93, 1997.

[3] S. Ruzic, A. Vuckovic and N. Nikolic, “Weather Sensitive Method for Short-Term Load Forecasting in Electric Power Utility of Serbia”, IEEE Transactions on Power Systems, vol.18, no. 4, pp.1581–1586, 2003.

[4] T. Haida and S. Muto, “Regression Based Peak Load Forecasting using a Transformation Technique”, IEEE Transactions on Power Systems, vol. 9, no. 4, pp. 1788–1794, 1994.

[5] W. Charytoniuk, M.S. Chen and P. Van Olinda, “Nonparametric Regression Based Short-Term Load Forecasting”, IEEE Transactions on Power Systems, vol. 13, no. 1, pp. 725–730, 1998.

[6] J.Y. Fan and J.D. McDonald, “A Real-Time Implementation of Short-Term Load Forecasting for Distribution Power Systems”, IEEE Transactions on Power Systems, 9:988–994, 1994.

[7] M.Y. Cho, J.C. Hwang and C.S. Chen, “Customer Short-Term Load Forecasting by using ARIMA Transfer Function Model”, Proceedings of the International Conference on Energy Management and Power Delivery, vol. 1, no. 1, pp.317–322, 1995.

[8] D.B. Fogel, “An Introduction to Simulated Evolutionary Optimization”, IEEE Transactions on Neural Networks, vol. 5, no. 1, pp.3–14, 1994.

[9] H.T. Yang and C.M. Huang, “A New Short-Term Load Forecasting Approach using Self-Organizing Fuzzy ARMAX Models”, IEEE Transactions on Power Systems, vol. 13, no. 1, pp. 217–225, 1998.

[10] K.L. Ho, “Short-Term Load Forecasting of Taiwan Power System using A Knowledge Based Expert System”, IEEE Transactions on Power Systems, vol. 5, no. 1, pp. 1214 – 1221, 1990.

[11] S. Rahman and O. Hazim, “Load Forecasting for Multiple Sites: Development of an Expert System-Based Technique”, Electric Power Systems Research, vol. 39, no. 1, pp. 161 – 169, 1996.

[12] M. Peng, N.F. Hubele and G.G. Karady, “Advancement in the Application of Neural Networks for Short-Term Load Forecasting”, IEEE Transactions on Power Systems, vol. 7, no. 1, pp. 250–257, 1992.

[13] A.G. Bakirtzis, V. Petridis, S.J. Kiartzis, M.C. Alexiadis and A.H. Maissis, “A Neural Network Short-Term Load Forecasting Model for the Greek Power System”, IEEE Transactions on Power Systems, vol. 11, no. 1, pp. 858–863, 1996.

[14] A.D. Papalexopoulos, S. Hao and T.M. Peng, “An Implementation of a Neural Network Based Load Forecasting Model for the EMS”, IEEE Transactions on Power Systems, vol. 9, no. 1, pp. 1956–1962, 1994.

[15] A. Khotanzad, R.A. Rohani, T.L. Lu, A. Abaye, M. Davis and D.J. Maratukulam, “ANNSTLF–A Neural-Network-Based Electric Load Forecasting System”, IEEE Transactions on Neural Networks, vol. 8, no. 1, pp. 835–846, 1997.

[16] A. Khotanzad, R.A. Rohani and D. Maratukulam, “ANNSTLF– Artificial Neural Network Short-Term Load Forecaster–Generation Three”, IEEE Transactions on Neural Networks, vol. 13, no. 2, pp. 1413–1422, 1998.

[17] H. Mori and N. Kosemura, “Optimal Regression Tree Based Rule Discovery for Short-Term Load Forecasting”, Proceedings of IEEE Power Engineering Society Transmission and Distribution Conference, vol. 2, no.1, pp. 421–426, 2001.

[18] T.W.S. Chow and C.T. Leung, “Nonlinear Autoregressive Integrated Neural Network Model for Short-Term Load Forecasting”, IEE Proceedings on Generation, Transmission and Distribution, vol. 143, no. 3, pp. 500– 506, 1996.

[19] S.E. Skarman and M. Georgiopoulous, “Short-Term Electrical Load Forecasting using a Fuzzy ARTMAP Neural Network”, Proceedings of SPIE, vol. 2, no. 1, pp.181–191, 1998.

[20] S.J. Kiartzis and A.G. Bakirtzis, “A Fuzzy Expert System for Peak Load Forecasting: Application to the Greek Power System”, Proceedings of the 10th Mediterranean Electrotechnical Conference, vol. 3, no.1, pp. 1097– 1100, 2000.

[21] V. Miranda and C. Monteiro, “Fuzzy Inference in Spatial Load Forecasting”, Proceedings of IEEE Power Engineering Winter Meeting, vol. 2, no. 1, pp. 1063–1068, 2000.

[22] D. Srinivasan, “A Neural Network Short-Term Load Forecaster”, Electric Power Research, vol. 28, no. 2, pp. 227-234, 1994.

[23] O. Mohammed, D. Park, R. Merchant, T. Dinh, C. Tong, A. Nazeem, J. Farah and C. Draks “Practical Experiences with an Adaptive Neural Network Short-Term Load Forecasting System”, IEEE Trans. on Power Systems, vol. 10, no. 2, pp. 254-265, 1995.

[24] D.C. Park, “Electric Load Forecasting using an Artificial Neural Network”, IEEE Transactions on Power Systems, vol. 6, no. 2, pp. 412-449, 1991.

[25] T.S. Dillon, “Short-Term Load Forecasting Using an Adaptive Neural Network”, Electrical Power & Energy Systems, vol. 13, no. 1, pp.186-191, 1991.

[26] M. Djukanvic, “Unsupervised/Supervised Learning Concept for 24-hour Load Forecasting”, IEE Proceedings -C, vol. 140, no. 2, pp. 311-318, 1993.

[27] K.Y. Lee and J. H. Park, “Short-Term Load Forecasting Using an Artificial neural Network”, IEEE Transactions on Power Systems, vol. 7, no. 1, pp. 124-132, 1992.

[28] C.N. Lu, “Neural Network Based Short Term Load Forecasting”, IEEE Trans. on Power Systems, vol. 8, no. 2, pp. 336-341, 1993.

[29] K.L. Ho, “Short Term Load Forecasting using a Multilayer Neural Network with an Adaptive Learning Algorithm”, IEEE Transactions on Power Systems, vol. 7, no. 2, pp. 141-149, 1992.

[30] B.S. Kermanshahi, “Load Forecasting under Extreme Climatic Conditions”, Proceedings, IEEE Second International Forum on the Applications of Neural Networks to Power Systems, vol. 5, no. 1, pp. 213-218, 1993.

[31] S.T. Chen, “Weather Sensitive Short-Term Load Forecasting using Non Fully Connected Artificial Neural Networks”, IEEE Tranactions on Power Systems, 7: 1098-1105, 1992.

[32] G.N. Kariniotakis, “Load Forecasting using Dynamic High-Order Neural Networks”, Proceedings, IEEE Second International Forum on the Applications of Neural Networks to Power Systems, vol. 5, no. 1, pp. 801-805,1993.

[33] J. Villiers, “Back-Propagation Neural Nets with One and Two Hidden Layers”, IEEE Trans. on Neural Networks, vol. 4, no. 1, pp. 136-146, 1992.

[34] Y.Y. Hsu, “Design of Artificial Neural Networks for Short-Term Load Forecasting”, IEE Proc. C, vol. 138, no. 1, pp. 407-418, 1991.

[35] A.D. Papalexopoulos, “Application of Neural Network Technology to Short-Term System Load Forecasting”, Proceedings, IEEE Second International Forum on the Applications of Neural Networks to Power Systems, vol. 5, no. 1, pp. 796-800, 1993.

[36] Y. Lee, “An Analysis of Premature Saturation in Back Propagation Learning”, Neural Networks, vol. 6, no. 1, pp. 719-728, 1993.

[37] V.V. Phansalkar, “Analysis of the Back-Propagation Algorithm with Momentum”, IEEE Transactions on Neural Networks, vol. 5, no. 1, pp. 505-506, 1994.

[38] Y. Rui and P. Jin, “The Modelling Method for ANN-Based Forecaster”, CDC' 94, China, 1994.

[39] A. G. Parlos, “An Accelerated Learning Algorithm for Multilayer Perceptron Networks”, IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 493-497, 1994.

[40] A.V. Ooyen, “Improving The Convergence of The Back-Propagation Algorithm”, Neural Network, vol. 5, no. 2, pp. 465-471, 1992.

[41] M. Arai, “Bounds on the Number of Hidden Units in Binary-Valued Three-Layer Neural Networks”, Neural Networks, vol. 6, no. 2, pp. 855-860, 1993.

[42] S.C. Huang, “Bounds on the Number of Hidden Neurons in Multilayer Perceptrons”, IEEE Transactions on Neural Networks, 2: 47-55, 1991.

[43] G. L.Torres, C.O. Traore, P.J. Lagace and D. Mukhedkar, “A Knowledge Engineering Tool for Load Forecasting”, Proc. of the 33rd Midwest Symposium on Circuits and Systems, vol. 1, no. 2, pp. 14-147, 1990.

[44] K H. Kim, J.K. Park, K.J. Hwang and S.H. Kim, “Implementation of Hybrid Short-term Load Forecasting System Using Artificial Neural Networks and Fuzzy Expert Systems”, IEEE Transactions on Power Systems, vol. 10, no. 3, pp. 1534-1539, 1995.

[45] P.K. Dash, S. Dash, G. R. Krishna and S. Rahman, “Forecasting of a Load Time Series Using a Fuzzy Expert System and Fuzzy Neural Networks”, International Journal of Engineering Intelligent Systems, vol. 1, no. 1, pp. 103-118, 1993.

[46] D. Srinivasan, A.C. Liew and C.S. Chang, "Forecasting Daily Load Curves Using A Hybrid Fuzzy-Neural Approach", IEE Proceedings-C, vol. 141, no. 2, pp. 561-567, 1994.

[47] D. Srinivasan, C.S. Chang and AC. Liew, "Demand Forecasting Using Fuzzy Neural Computation, With Special Emphasis on Weekend and Public Holiday Forecasting", IEEE Transactions on Power Systems, vol. 9, no. 2, pp. 1780-1787, 1994.

[48] Y. Qiu, "A Fuzzy Neural Network for Short-term Load Forecasting", Proceedings of the IEEE Transactions on Power systems, vol. 9, no. 2, pp. 1772-1780, 1994.

[49] P.K. Dash, A.C. Liew and S. Rahman, "Peak Load Forecasting using a Fuzzy Neural Network", Electric Power Systems Research, vol. 32, no. 1, pp. 19-23, 1995.

[50] A.G. Bakirtzis, J.B. Theocharis, S.J. Kiatzis and K.J. Satios, "Short Term Load Forecasting Using Fuzzy Neural Networks", IEEE Transactions on Power Systems, vol. 9, no. 2, pp. 1760-1772, 1994.

[51] H. Gottschalk, S. Heine, B. Fox and I. Neumann, ''Economic Operation of a Power System with a Significant Amount of Controllable Load", Proceedings of the 29th Universities Power Engineering Conference, vol. 2, no. 2, pp. 673-675, 1994.

[52] M. Daneshdoost, M. Lotfalian, G. Bumroonggit and J.P. Ngoy, " Neural Network with Fuzzy Set-Based Classification for Short Term Load Forecasting ", IEEE Transactions on Power Systems, vol.13, no. 4, pp. 1386-1391, 1998,.

[53] J.A. Anderson, "Cognitive and Psychological Computation with Neural Models", IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-13, No. 5, pp. 799-815, 1983.

[54] A. C. Gail and G. Stephen, “A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine”, Computer Vision, Graphics and Image Processing, vol. 37, no. 1, pp. 54-115, 1987.

[55] A. C. Gail and G. Stephen, “The ART of Adaptive Pattern Recognition by a Self Organizing Neural Network”, Computer, vol. 21, no. 1, pp. 77-88, 1988.

[56] D. O. Hebb, “The Organization of Behaviour”, Wiley, NY, 1949.

[57] B. Kosko, “Bidirectional Associative Memories”, IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-18, no. 1, pp.49-60, 1988.

[58] L. V. Fausett, “Fundamentals of Neural Networks, Architectures, Algorithms and Applications”, Prentice Hall, Englewood Cliffs, 1994.

[59] R. Linsker, “Self-Organization in a Perceptual Network”, IEEE Computer Society Press, vol. 21, no. 3, pp. 105-117, 1988.

[60] R.P. Lippman, “Introduction to Computing with Neural Nets”, IEEE ASSP Magazine, vol. 4, no. 6, pp. 4-22, 1987.

[61] R.J. Macgregor, “Neural and Brain Modelling”, Academic Press London, 1987.

[62] W.S. McCulloch and W. Pitts, “A Logical Calculus of the Ideas Imminent in Nervous Activity”, Bulletin of Mathematical Biophysics, vol. 5, no. 1, pp. 115-153, 1943.

[63] D. W. Patterson, “Artificial Neural Networks: Theory and Applications”, Prentice Hall, 1998.

[64] F. Rosenblatt, “The Perceptrons: A Probabilistic Model for Information Storage and Organization in the Brain”, Psychological Review, vol. 65, no. 1, 386-408, 1958.

[65] M. McInerney and A.P. Dhawan, “Use of Genetic Algorithm with Backpropagation in Training of Feedforward Neural Networks”, IEEE

Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, Innsbruck, Springer-Verlag, Wein, vol. 3, no. 1, pp. 203-208, 1993.

[66] S. Rajasekaran and G.A. Vijayalakshmi Pai, “Genetic Algorithm Based Weight Determination for Backpropagation Networks”, Proc. Trends in Computing, Tata McGraw-Hill, pp. 73-80, 1996.

[67] L. A. Zadeh, “Fuzzy Sets”, Inf. Control, vol. 8, no. 1, pp. 338-353, 1965.

[68] C.L. Chang and R.C. Lee, “Symbolic Logic and Mechanical Theorem Proving”, Academic Press, NY, 1973.

[69] E. Cox, “The Fuzzy Systems Handbook”, Morgan Kaufmann Publishers, 1998.

[70] C. Bauer and G. Viot, “Fuzzy Logic Concepts and Constructs”, AI Expert, pp. 26-33, 1993.

[71] H. Hellendoorn and C. Thomas, “Defuzzification in Fuzzy Controllers”, Intelligent and Fuzzy Systems, vol. 1, no. 1, pp. 109-123, 1993.

[72] T. N. Hung and A. W. Elbert, “A First Course in Fuzzy Systems and Control”, Prentice Hall, 1999.

[73] D.P. Kothari and I.J.Nagrath, ‘Modern Power System Analysis’, Tata McGraw Hill.

[74] H.S. Hippert, C.E. Pedreira, and R.C. Souza, “Neural Networks for Short-Term Load Forecasting: A Review and Evaluation”, IEEE Transactions on Power Systems, vol. 16, no. 1, pp. 44–55, 2001.

[75] D.C. Park, M.A. El-Sharkawi, R. J. Marks II, , L.E. Atlas and M.J. Damberg, “ Electric Load Forecasting Using an Artificial Neural Network”, IEEE Transactions on Power Systems, vol. 6, no. 2, pp. 442-449, 1991.

[76] A. Khotanzad, R. C. Hwang and D. Maratukulam, "Hourly Load Forecasting by Neural Networks", Panel Session on Application of Neural Networks to Short-Term Load Forecasting, IEEE PES Winter Meeting, Columbus, Ohio, 1993.

[77] K. H. A. Rahman and S. M. Shahidepour, "Static Security in Power System Operation with Fuzzy Real Load Conditions", IEEE PES Winter Meeting, New York, NY, paper No. 94 WM 219-6 PWRS, 1994.

[78] K. H. A. Rahman, S. M. Shahidepour and M. Daneshdoost, "AI Approach to Optimal Var Control with Fuzzy Reactive Loads", IEEE PES Winter Meeting, New York, NY, paper No. 94 WM 193-3 PWRS, 1994.

[79] P. K. Dash, S. Dash and S. Rahman, "A Hybrid Artificial Neural Network-Fuzzy Expert System for Short Term Load Forecasting", Presented at ESAP93, pp. 175-180, 1993.

[80] G. L.Torres, C. O. Traore, F. G. Mandolesi and D. K. Mukhedkhar, "Short Term Forecasting Using a Fuzzy Engineering", Presented at ANNPS, vol. 91, no. 1, 37-40, 1991.

[81] Y. H. Song, A. Johns and R. Aggarwal, "Computational Intellgence Applications to Power System", Kluwer Academic Publishers, London.

APPENDIX

BACK PROPAGATION LEARNING ALGORITHM

Back propagation learning algorithm is used at the learning stage. Steps for Back-propagation learning algorithm are as:

Step 1: Normalize the inputs and outputs with respect to their maximum values. For each training pair, assume there are l inputs given by $\{I\}_{l \times 1}$ and n outputs $\{O\}_{n \times 1}$ in a normalized form.

Step 2: Assume the number of neurons in the hidden layer to lie between l <m<2l.

Step 3: [V] Represents the weights of synapses connecting input neurons to hidden neurons and [W] represents the weights of synapses connecting hidden neurons and output neurons. Initialize the weights to small random values.

$$[V]^0 = [\text{ Random Weights}]$$

$$[W]^0 = [\text{Random Weights}]$$

$$[\Delta V] = [\Delta W] = [0]$$

Step 4: For the training data, present one set of inputs and outputs. Present the pattern to the input layer $\{I\}_I$ as inputs to the input layer. By using linear activation function, the output of the input layer may be evaluated as

$$\underset{l \times 1}{\{O\}_I} = \underset{l \times 1}{\{I\}_I}$$

Step 5: Compute the inputs to the hidden layer by multiplying corresponding weights of synapses as

$$\underset{m \times 1}{\{I\}_H} = \underset{m \times l}{[V]^T} \underset{l \times 1}{\{O\}_I}$$

Step 6: Let the hidden layer units evaluate the output using the Sigmoidal function as

$$\{O\}_H = \begin{pmatrix} \dots\dots\dots\dots \\ \dots\dots\dots\dots \\ \frac{1}{(1+e^{-I_{Hi}})} \\ \dots\dots\dots\dots \\ \dots\dots\dots\dots \end{pmatrix}$$

Step 7: Compute the inputs to the output layer by multiplying corresponding weights of synopses as

$$\underset{n \times 1}{\{I\}_O} = \underset{n \times m}{[W]^T} \underset{m \times 1}{\{O\}_H}$$

Step 8: Let the output layer units evaluate the output using sigmoidal function as

$$\{O\}_O = \begin{pmatrix} \dots\dots\dots\dots \\ \dots\dots\dots\dots \\ \frac{1}{(1+e^{-I_{Oj}})} \\ \dots\dots\dots\dots \\ \dots\dots\dots\dots \end{pmatrix}$$

Step 9: Calculate the error and the difference between the network output and the desired output as for the ith training set as

$$E^P = \frac{\sqrt{\sum (T_j - O_{Ok})^2}}{n}$$

Step 10: Find {d} as

$$\{d\} = \begin{pmatrix} \dots\dots\dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ (T_k - O_{Ok})O_{Ok}(1 - O_{Ok}) \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots\dots\dots \end{pmatrix}$$

Step 11: Find [Y] matrix as

$$[Y] = \{O\}_H \quad \{d\}$$

$$\underset{m \times n}{} \quad \underset{m \times 1}{} \ \underset{1 \times n}{}$$

Step 12: Find $[\Delta W]^{t+1} = \alpha\, [\Delta W]^{t} + \eta[Y]$

$$\underset{m \times 1}{} \quad \underset{m \times n}{} \quad \underset{n \times 1}{}$$

Step 13: Find $\underset{m \times 1}{\{e\}} = \underset{m \times n}{[W]} \ \underset{n \times 1}{\{d\}}$

$$\underset{m \times 1}{\{d^*\}} = \underset{m \times 1}{\begin{pmatrix} \cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots \\ e_i (O_{Hi})(1 - O_{Hi}) \\ \cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots \end{pmatrix}}$$

Find [X] matrix as

$$\underset{1 \times m}{[X]} = \underset{1 \times 1}{\{O\}_I} \underset{1 \times m}{\{d^*\}} = \underset{1 \times 1}{\{I\}_I} \underset{1 \times m}{\{d^*\}}$$

Step 14: Find $[\Delta V]^{t+1} = \alpha\, [\Delta V]^{t} + \eta[X]$

Step 15: Find

$$[V]^{t+1} = [V]^{t} + [\Delta V]^{t+1}$$

$$[W]^{t+1} = [W]^{t} + [\Delta W]^{t+1}$$

Step 16: Find the error rate as

$$\text{Error rate} = \frac{\sum E_P}{nset}$$

Step 17: Repeat steps 4-16 until the convergence in the error rate is less than the tolerance value.