Journal of Computing and Information Technology - CIT 15, 2007, 3, 267–282
doi:10.2498/cit.1000987

Short Term Electric Load Prediction Using Fuzzy BP

Hari Seetha and R. Saravanan

School of Computing Sciences, VIT University, Vellore, India

The privatization of electricity industry in various parts of the world has increased the significance of the load prediction problem and in particular there is a need to understand and predict the demand for power with greater accuracy, even in case of imprecise input data. Prediction of power demand is essential for an efficient operation of any utility company. In this paper, a fuzzy version of neural network, namely Fuzzy back propagation network (Fuzzy BP) has been developed for short term electric load prediction. The load is predicted using fuzzy back propagation algorithm. This model is capable of handling imprecise information in input data. The proposed architecture consists of a module with 51 inputs and 24 outputs. The inputs are fuzzified and the outputs are crisp values representing the predicted load. The proposed method is implemented in MATLAB. The simulation results are presented for each day (24 hours) of the week. Besides this, a multi layer perceptron (MLP) was also implemented separately and the load was predicted using back propagation algorithm. The results obtained from Fuzzy BP were found to be satisfactory when compared to those of MLP network.

Keywords: fuzzy neuron, fuzzy back propagation, short term load prediction

1. Introduction

In the electricity supply industry, it is important to determine the future demand for power as far in advance as possible. If accurate estimates can be made for maximum and minimum load for each hour, day, month, season and year, utility companies can make significant economics in areas such as setting the operating reserve, maintenance scheduling and fuel inventory management (Hari Seetha 2005; 2006). Thus, electric load prediction is performed over long, medium and short terms, to ensure that the resources are made available to match the demand. Long term forecasting forecasts years into the future and helps in investment planning. Medium term forecasting forecasts months ahead and is needed to support energy procurement, energy marketing, tariff management, maintenance planning and network design functions. Short-term prediction is the prediction of load demand, hours or days into future and is required to support energy trading and network control functions. The factors that affect short term load forecasting were well discussed by Gross and Galiana (1987). The forecasting procedure depends on the manner in which historical data is analysed and on the type of information available at the time the forecast is prepared (Lijesen 1971). Various techniques have been applied to the problem of short term electric load prediction.

The statistical techniques that were widely used in short term electric load prediction are regression, time series analysis and general exponential smoothing. Papalexopoulos and Hesterberg (1994) described a linear regression model for short term load forecasting. Haida and Muto (1994) presented a multivariate linear regression-based peak load forecasting with a transformation technique. The transformation function is used to deal with non-linear relationship between temperature and load. Performance of the technique is verified with simulations of actual load data of Tokyo Electric Power Company. Charytonuik et al. (1998) used a novel approach to load forecasting by the application of non-parametric regression. Amjady (2001) developed a novel approach for short term load forecasting, which incorporates the time series modeling of the ARIMA (autoregressive integrated moving average) with the knowledge of experienced human operators. Although these statistical techniques are reliable, they fail to give accurate results when quick weather changes occur which form a non-linear relationship with daily load.

Hence results of statistical methods in presence of such events are not satisfactory as desired. Therefore the emphasis has shifted to the application of various artificial intelligence techniques for short term load forecasting. Among them, artificial neural networks (ANN) are widely used for short term electric load prediction. Neural networks attempt to learn by themselves the functional relationship between system inputs and outputs. They have the ability to perform non-linear modeling and adaptation. Park et al. (1991) presented an ANN approach to electric load forecasting and they used the back propagation algorithm to train the ANN. Peng et al. (1992) proposed a procedure for choosing the training cases, which are most similar to the forecasted inputs. Khotanzad et al. (1997) described a load forecasting system known as ANNSTLF. It includes two ANN forecasters. One of them predicts the base load and the other forecasts the change in load. The final forecast is computed by an adaptive combination of these two forecasts. The effect of humidity and wind speed is considered through a linear transformation of temperature.

Till date, several researchers have dealt with the application of various neural networks to short term load forecasting with varying success (Lee at al., 1992; Chen et al., 1992; Lu et al., 1993; Ranaweera, et al., 1995; Bakirtzis et al., 1996; Lamedica et al., 1996, Beccali et al., 2001; Topalli et al., 2003; Carpinteiro et al., 2004; Satish et al., 2004; Topalli et al., 2006). Although neural networks are capable of handling nonlinearity between the electric load and the weather factors that affect the load, they lack to handle unusual changes that occur in the environment. Fuzzy logic is often an effective approach to these uncertainties. Fuzzy logic-based systems were found to perform well in a dynamically changing environment. Srinivas et al. (2002) discussed various applications of Fuzzy logic. Kyung-Bin Song et al. (2005) developed a new fuzzy linear regression method for short term 24 hourly loads forecasting of the holidays. The concept of fuzzy regression analysis is employed for STLF. The fuzzy linear regression model is made from the historical data and coefficients of the model are solved by mixed linear programming problem. Tranchita et al. (2004) proposed a novel method for STLF based on similar day approach and the use of soft computing techniques. Hsu et al. (1992) used a fuzzy expert system for STLF. Experienced operators' heuristic rules are imbedded in the knowledge base. Chenthur Pandian et al. (2006) used fuzzy approach for short term load forecasting. They took time and temperature as inputs to fuzzy logic controller with forecasted load being the output and proved that the percentage error in forecasted load when compared to actual load was within 3%.

Though fuzzy logic systems were proved to be successful in handling imprecise data, they lack the ability to learn from experience. Hence combining neural networks and fuzzy logic can enhance the capability of intelligent systems to learn from experience and adapt to changes in the environment (Satish et al., 2002). Srinivasan and Lee (1995) presented a survey of hybrid fuzzy neural approaches to electric load forecasting and reported that these models showed superior performance compared to conventional methods of forecasting. They classified the published literature based on the combination of fuzzy logic and neural networks as follows:

a) Fuzzy logic system at the output stage of the neural networks forecaster to manipulate the output (Lambert Torres et al., 1990; Kwang-Ho Kim et al., 1995; Mohammad Tamimi and Robert Egbert, 2000),
b) Fuzzy logic at the input stage of a neural network to preprocess the inputs (Srinivasan et al., 1994; 1995),
c) Integrated fuzzy neural network to create a fuzzy rule base from the historical training data (Bakirtzis et al., 1995; Dash et al., 1995) and
d) Separate fuzzy logic and neural network forecasters to forecast different components of the load (Ma-Wen Xiao et al. 2002).

Thus, the application of Fuzzy BP algorithm for short term load forecasting is a novel approach. Hence in this paper, a fuzzy version of neural network, namely Fuzzy back propagation network, is applied to short term load forecasting. The proposed architecture is trained using fuzzy back propagation algorithm to predict 24 hours load ahead. Besides this, MLP was also implemented separately to predict next day's load using back propagation algorithm.

2. Fuzzy Neuron and Fuzzy Back Propagation Network

In the present work, a neural network model with fuzzy inference (Fuzzy BP network) is used to predict electric load. It performs the non-linear mapping between fuzzy inputs and crisp outputs. The fuzzy neurons in the model make use of LR-type fuzzy numbers. The triangular type of LR-type fuzzy numbers has been used for simplification of architecture and reduction of computational load. In this section, the LR-type fuzzy numbers and their operations required by Fuzzy BP architecture are first presented. The structure of a fuzzy neuron, the architecture of Fuzzy BP and its learning mechanism are presented next.

2.1. LR-type Fuzzy Number

Definition. A fuzzy number $\widetilde{m}$ is said to be a LR-type fuzzy number [Dubois and Prade, 1979] if and only if

$$\mu_{\widetilde{m}}(x)=\begin{cases} L\left(\frac{m-x}{\alpha}\right) \text{for } x\le m;\ \alpha>0 \\ R\left(\frac{x-m}{\beta}\right) \text{for } x\ge m;\ \beta>0 \end{cases} \tag{1}$$

where 'L' is a left reference and 'R is a right reference, m is a real number, α and β are left and right spreads respectively. The functions L and R are defined as follows:

$$L\left(\frac{m-x}{\alpha}\right) = \max\left(0, 1-\frac{|m-x|}{\alpha}\right)$$
$$R\left(\frac{x-m}{\beta}\right) = \max\left(0, 1-\frac{|x-m|}{\beta}\right). \tag{2}$$

LR-type fuzzy number $\widetilde{m}$ can be represented as $(m, \alpha, \beta)_{LR}$ as shown in Figure 1.

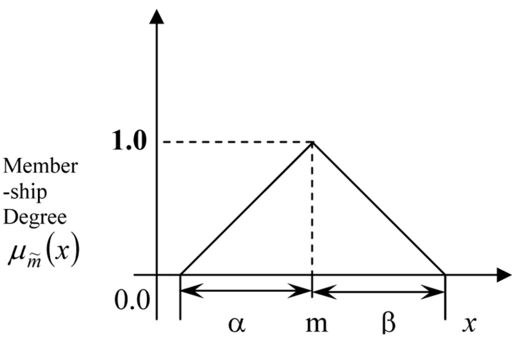


Figure 1. A triangular fuzzy number (m, α, β).

2.2. Operations on LR-type Fuzzy Numbers

Let $\widetilde{M}$ and $\widetilde{N}$ be two LR-type fuzzy numbers given by $\widetilde{M} = (m, \alpha, \beta)$ and $\widetilde{N} = (n, \gamma, \delta)$. The basic operations are

Addition:

$$(m, \alpha, \beta)_{LR}+(n, \gamma, \delta)_{LR}=(m+n, \alpha+\gamma, \beta+\delta)_{LR}$$

Subtraction:

$$(m, \alpha, \beta)_{LR}-(n, \gamma, \delta)_{LR}=(m-n, \alpha+\delta, \beta+\gamma)_{LR}$$

Multiplication:

$$(m, \alpha, \beta)_{LR}{}^{*}(n, \gamma, \delta)_{LR} = \begin{cases} (mn, m\gamma + n\alpha, m\delta + n\beta)_{LR}, \\ \text{for } m \ge 0, n \ge 0 \\ (mn, m\alpha - m\delta, n\beta - m\gamma)_{RL}, \\ \text{for } m < 0, n\ge 0 \\ (mn, -n\beta - m\delta, -n\alpha - m\gamma)_{LR}, \\ \text{for } m < 0, n < 0 \end{cases}$$

Scalar multiplication:

$$\lambda^{*}(m, \alpha, b)_{LR} = \begin{cases} (\lambda m, \lambda\alpha, \lambda\beta)_{LR}, \forall\lambda \ge 0, \lambda \in R \\ (\lambda m, -\lambda\alpha, -\lambda\beta)_{RL}, \forall\lambda < 0, \lambda \in R \end{cases}$$

2.3. Fuzzy Neuron

The fuzzy neuron is the basic element of Fuzzy BP network (Lee Hahn Ming and Lu Bing Hui, 1994), which is shown in Figure 2.

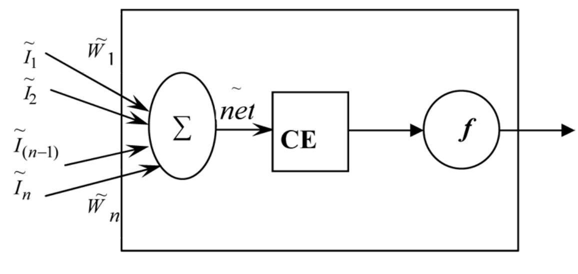


Figure 2. Fuzzy neuron.

Given the input vector $\widetilde{I} = (\widetilde{I}_0, \widetilde{I}_1, \ldots, \widetilde{I}_n)$ and weight vector $\widetilde{W}_0, \widetilde{W}_1, \ldots, \widetilde{W}_n)$, the fuzzy neuron computes the crisp output given by

$$O = f(NET) = f\left(CE\Big(\sum_{i=0}^{n} \widetilde{W}_i \widetilde{I}_i\Big)\right). \quad (3)$$

Here, the fuzzy weighted summation is given by $\widetilde{net} = \sum_{i=0}^{n} \widetilde{W}_i \widetilde{I}_i$ which is computed first and $NET = CE(\widetilde{net})$ is computed next. The function CE is the centroid of triangular fuzzy number and can be treated as defuzzification operation, which maps fuzzy weighted summation to crisp value.

Thus, if $\widetilde{net} = (net_m, net_\alpha, net_\beta)$ is the fuzzy weighted summation, then the function CE is given by

$$\begin{aligned} CE(\widetilde{net}) &= CE(net_m, net_\alpha, net_\beta) \\ &= net_m + (net_\beta - net_\alpha). \end{aligned} \quad (4)$$

The function f is a sigmoidal function that performs nonlinear mapping between the input and output. f is obtained as follows:

$$O = f(NET) = 1/(1 + \exp(-NET)). \quad (5)$$

In the fuzzy neuron, both input vector $\widetilde{I}$ and weight vector $\widetilde{W}$ are represented by triangular LR-type fuzzy numbers. Thus, for input vector $\widetilde{I} = (\widetilde{I}_0, \widetilde{I}_1, \ldots, \widetilde{I}_n)$ the input component $\widetilde{I}_i$ is represented by the LR-type fuzzy number $I_{mi}, I_{ai}, I_{\beta i}$. Similarly, for the weight vector $\widetilde{W} = (\widetilde{W}_0, \widetilde{W}_1, \ldots, \widetilde{W}_n)$, the weight vector component $\widetilde{W}_i$ is represented as $W_{mi}, W_{ai}, W_{\beta i}$.

2.4. Architecture of Fuzzy BP

Fuzzy BP is a three-layered feed forward architecture. The three layers are: input layer, hidden layer and output layer. Considering a configuration of t-input neurons, m-hidden neurons and n-output neurons, the architecture of Fuzzy BP is shown in Figure 3. Let $\widetilde{I}_p = (\widetilde{I}_{p1}, \widetilde{I}_{p2}, \ldots, \widetilde{I}_{pt})$, $p = 1, 2, \ldots, N$, be the p^{th} pattern among N input patterns that Fuzzy BP needs to be trained. Here, $\widetilde{I}_{pi}$ indicates the i^{th} component of input pattern p and is an LR-type triangular fuzzy number, i.e., $\widetilde{I}_{pi} = (\widetilde{I}_{pmi}, \widetilde{I}_{p\alpha i}, \widetilde{I}_{p\beta i})$. Let $\widetilde{O}_{pi}$

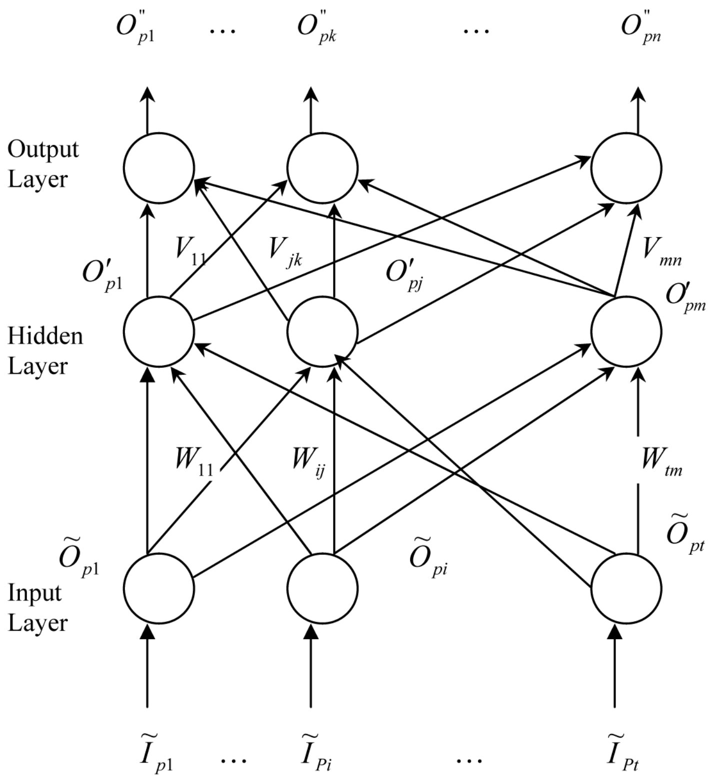


Figure 3. Three layered Fuzzy BP architecture.

be the output value of i^{th} input neuron. O'_{pj} and O''_{pk} are j^{th} and k^{th} crisp defuzzification outputs of the hidden and output layer neurons respectively. W_{ij} is the fuzzy connection weight between i^{th} input node and j^{th} hidden node. V_{jk} is the fuzzy connection weight between j^{th} hidden node and k^{th} output node. The computations carried out by each layer are as follows:

Input neurons: $\widetilde{O}_{pi} = \widetilde{I}_{pi}, \quad i = 1, 2, \ldots, t.$

Hidden neurons:

$$O'_{pj} = f\,(NET_{pj}), j = 1, 2, \ldots, m,$$

where $NET_{pj} = CE\Big(\sum_{i=0}^{t} W_{ij} O'_{pi}\Big).$

Output neurons:

$$O''_{pk} = f\,(NET'_{pk}), k = 1, 2, \ldots, n,$$

where $NET'_{pk} = CE\Big(\sum_{j=0}^{m} V_{jk} O'_{pj}\Big).$

2.5. Learning in Fuzzy BP

The learning procedure of Fuzzy BP follows the gradient descent method of minimizing error due to the learning (Lee Hahn Ming and Lu Bing Hui, 1994). Here, the mean square error function for the pattern p is defined as:

$$E_p = \sum_i \frac{1}{2}\left(D_{pi} - O''_{pi}\right)^2 \qquad (6)$$

where D_{pi} is the desired output value of i^{th} output unit, and O''_{pi} is the actual output of i^{th} output unit. The overall error of training pattern is $E = \sum_p E_p$. In the learning phase, the values of weights will be adjusted to minimize E. At time t, the weight change value is defined as:

$$\triangle\widetilde{W}(t) = -\eta\triangle E_p(t) + \alpha\triangle\widetilde{W}(t-1) \qquad (7)$$

where η is the learning rate and α is a constant value. $\alpha\triangle\widetilde{W}(t-1)$ is a momentum term, which is added for improving convergent speed.

The term $\nabla E_p(t)$ is given by

$$\nabla E_p(t) = \left(\frac{\partial E_p}{\partial W_m(t)}, \frac{\partial E_p}{\partial W_a(t)}, \frac{\partial E_p}{\partial W_\beta(t)}\right)$$

where $\widetilde{W}(t) = (W_m(t), W_\alpha(t), W_\beta(t)).$

3. Proposed Architecture

The fundamental idea behind combined fuzzy logic and neural network approaches is simple: the fuzzy logic system models the knowledge about the system and its input parameters, quantitative as well as qualitative, and the neural network captures the inexplicable relationship between fuzzy inputs and outputs. This hybrid approach exploits the inherent properties of neural networks, such as generalization, graceful degradation, retrieval from partial information, learning from well-defined patterns; and properties of fuzzy systems, such as abstract reasoning and human like responses in cases involving uncertainty and contradictory data. Thus, the motivation for the proposed architecture is to evolve a model that exploits parallel computation while demonstrating the ability to adapt to uncertainties that occur due to voltage and frequency instability, losses in transmission lines and unforeseen weather changes.

3.1. Description of Block Diagram of Proposed (Fuzzy BP) Architecture

The proposed architecture is shown in Figure 4. It consists of two stages viz., fuzzification that converts the inputs (load and temperature) into LR-type fuzzy numbers, Fuzzy BP network is then trained using Fuzzy back propagation algorithm that gives the crisp value of the forecasted load.

This is a neuro-fuzzy hybrid system in which the host is a multilayer feed forward network. The network maps fuzzy inputs to crisp outputs making use of back propagation learning.

The block diagram of MLP architecture used for short term load prediction is also shown in Appendix III (Figure 12).

The Fuzzy BP and MLP consist of three layers: input layer, hidden layer and output layer. The description of nodes in each layer is given below:

Input layer: Number of nodes = Number of inputs = 51. Each input data set consists of the following:

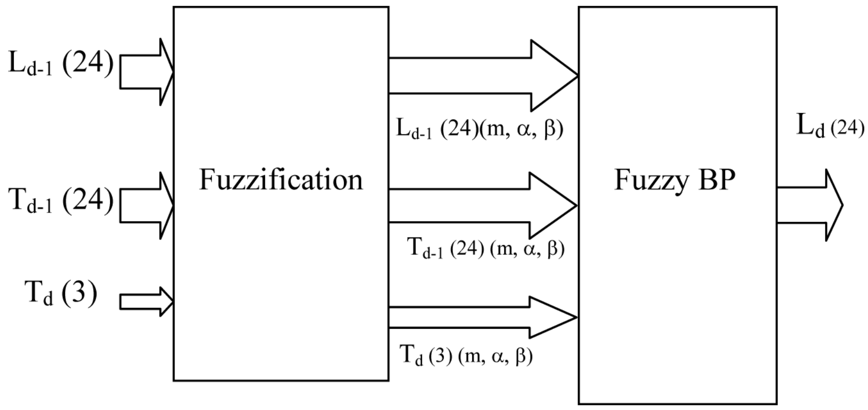


Figure 4. Block diagram of proposed architecture.

load data of previous day (d-1) L_{d-1}: 24 inputs,

temperature data of previous day (d-1) T_{d-1}: 24 inputs,

forecasted day's maximum, minimum and average temperatures T_d: 3 inputs.

Each input to Fuzzy BP network is an LR-type fuzzy number.

Hidden layer: Number of hidden layers = 1, Number of hidden neurons in hidden layer (NH) = 10.

The number of neurons in hidden layer is determined through trial and error procedure for both Fuzzy BP and MLP.

Output layer: In case of Fuzzy BP, the output layer forms the defuzzification stage, which gives the crisp output. In both Fuzzy BP and MLP, number of output neurons in the output layer is same as number of outputs. Number of outputs = 24 (predicted load for 24 hours of forecasted day L_d). Thus the number of inputs and outputs are fixed.

4. Implementation Aspects

The first step is to obtain an accurate historical data. This data is divided into two sets: training set and testing set. The accuracy of prediction is better when the correlation between training and testing data is stronger. The training data has to be chosen carefully to cover the entire range over which the different variables are expected to change. Each of the training patterns consists of an input [*load and temperature*] and a corresponding output [*load*] for 24 hours. In the load data, generally, all the Sunday's load data look alike, all the Monday's data look alike and this holds good for all the days of the week.

Hence for testing a day, the training data considered is the past data same as that of the testing day. Figure 5 shows 20 training patterns of Thursday. The first pattern represents the load of 7^{th} January 1999, the second pattern represents the load of 14^{th} January 1999 and so on, and the last pattern represents the load of 29^{th} July 1999. In all the patterns the two peaks occur at the same time and the same is true in case of valley load. The load at each hour is also approximately the same for all Thursdays. Hence for predicting the load for 21^{st} Thursday, Fuzzy BP and MLP are trained with past 20 Thursday patterns. Each training pattern consists of an *input* (each of the past 20 Wednesday's 24 hrs load (load data of previous day $(d-1)$ L_{d-1}), each of the past 20 Wednesday's 24 hrs temperature (temperature data of previous day $(d-1)$ T_{d-1}) and corresponding Thursday's minimum temperature, maximum temperature and average temperature ((forecasted day's maximum, minimum and average temperatures T_d)) and its corresponding *output* (each of the 20 corresponding Thursday's 24 hrs load L_d). In the same way, to predict the load for a particular day in a week, Fuzzy BP and MLP are trained with past 20 corresponding days' patterns.

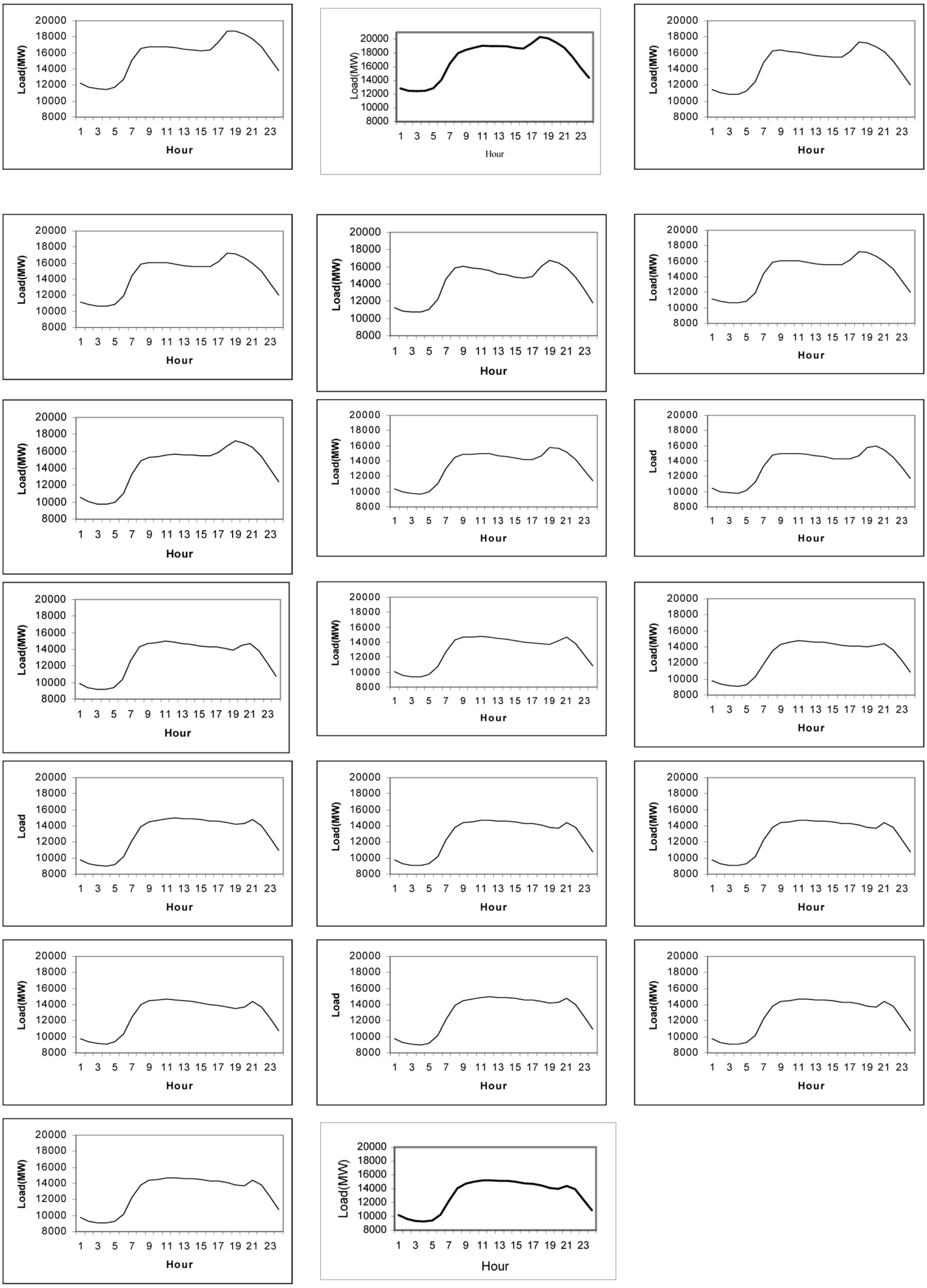


Figure 5. Actual load patterns of Thursday.

4.1. Data Preparation

In this stage, the raw input data has to be arranged as input and output pattern pairs for training the neural network. Seven months of past data (from 1^{st} January 1999 to 31^{st} July 1999, as shown in Appendix-I) is used for training and testing both Fuzzy BP and MLP. The 24 outputs are crisp values and the 51 inputs to the Fuzzy BP network are to be arranged as LR type fuzzy numbers after normalization. Normalization is an important stage as it reduces the learning time for both networks. Both load and temperature must be normalized to the same range of values.

4.2. Training

The proposed architecture is trained using Fuzzy BP algorithm, as shown in the flowchart in Figure 6. The terms used in the following flowchart are explained in Appendix-II. MLP is trained using back propagation algorithm, as shown in the flowchart in Figure 13. The following training parameters are determined on a trial and error basis to reduce the forecasting error for both Fuzzy BP and MLP (refer to Table A3).

Initial weights between different layers

$$= (0.5, 0.5, 0.5),$$

learning rate $(\eta) = 0.79$ and

momentum $(\alpha) = 0.59$

4.3. Optimum Learning Sequence

In the training stage, each of the Fuzzy BP and MLP networks are presented with pairs of several input and corresponding desired output data. The output always depends on the amount of data used for training. If insufficient data is used, then the output will not be a representative to true input. On the other hand, if network is trained with more data inputs, then the training time becomes longer. So, in order to determine the optimum number of training patterns, the proposed architecture was trained using Fuzzy BP algorithm and MLP was trained using back propagation algorithm, with first 15,

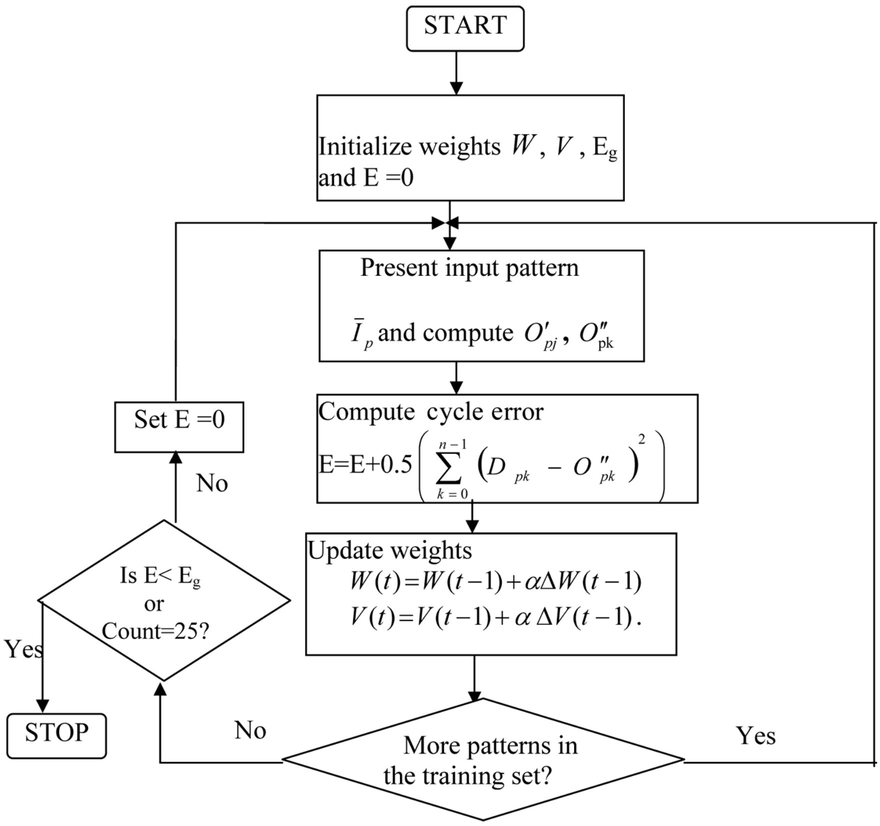


Figure 6. Fuzzv BP flow chart.

20 and 25 patterns. This was done for one week. Among all these cases the results for 20 training patterns are better than those with 15 and 25 training patterns for both Fuzzy BP and MLP. Hence the number of patterns (optimum learning sequence) required to train each of these networks efficiently for short term load prediction is 20. Table 1 and Table 3 (refer to Appendix III) show the comparison for optimum learning sequence. It can be noticed that the mean absolute percentage error (MAPE) and maximum percentage error for 20 training patterns is less for most of the days in a week. The patterns in the training set are recursively applied until either Sum Squared Error (E) is minimum or number of epochs is 25. Weights are updated after each pattern presentation.

4.4. Testing

In the testing phase, the Fuzzy BP network and MLP are presented with an unknown input pattern. Using the final weights obtained after the training process, the output (predicted load) is calculated.

5. Results and Discussions

This section presents the results for the following case studies.

Case Study 1: Here, the load forecasting for all the days in a week is performed with Fuzzy back propagation algorithm by considering

(a) 1-15 patterns for training the network and the 16^{th} pattern for testing.

(b) 1-20 patterns for training the network and the 21^{st} pattern for testing.

(c) 1-25 patterns for training the network and the 26^{th} pattern for testing.

From the above results, the optimum number of patterns required for training the Fuzzy BP network and MLP, is obtained, as shown in Table 1 and Table 3.

Case Study 2: This compares the results of short term load prediction using Fuzzy BP network with the results obtained from MLP for optimum learning sequence.

Observations from the above results are shown in Table 2 which depicts that, for most of the days in a week, the results obtained from Fuzzy BP network are more accurate than those obtained from MLP.

The mean absolute percentage error is calculated as follows:
mean absolute percentage

$$\text{error} = \left(\frac{100}{n}\right) \sum_{i=1}^{n} \left(\frac{|L_{acti} - L_{predi}|}{L_{acti}}\right),$$

where L_{acti} is the actual load at i^{th} hour, L_{predi} is the predicted load at i^{th} hour and $n = 24$ (since load for each of 24 hrs is considered).

The peak load percentage error is calculated as follows:
peak load percentage error

$$= (100)\left(\frac{|PL_{acti} - PL_{predi}|}{PL_{acti}}\right),$$

where PL_{acti} is the actual peak load at i^{th} hour and PL_{predi} is the predicted load at i^{th} hour.

Table 1 compares the maximum and average percentage errors obtained when Fuzzy BP was trained with the first 15, 20, and 25 training patterns for all the days in a week. It shows that errors are less in the case of 20 training patterns for most of the days in a week. Hence the

Day	Percentage Error	**Optimum Learning Sequence**		
		Number of training patterns 1-15	Number of training patterns 1-20	Number of training patterns 1-25
Sunday	Max.%Error	5.3924	5.3648	18.6049
	MAPE	2.2480	2.2310	15.4218
Monday	Max.%Error	7.3276	4.5079	14.2035
	MAPE	2.0583	1.7124	12.3420
Tuesday	Max.%Error	4.1420	4.5130	8.2608
	MAPE	2.0660	1.6452	8.3298
Wednesday	Max.% Error	3.9375	5.4105	7.7308
	MAPE	1.8551	2.5848	4.7926
Thursday	Max.%Error	8.0489	1.9961	6.0966
	MAPE	2.6598	0.7620	3.5665
Friday	Max.%Error	8.1904	3.4021	12.6674
	MAPE	5.8418	1.7090	10.2127
Saturday	Max.%Error	6.9615	2.2114	19.8496
	MAPE	3.5699	1.1981	15.2810

Table 1. Comparison table for optimum learning sequence.

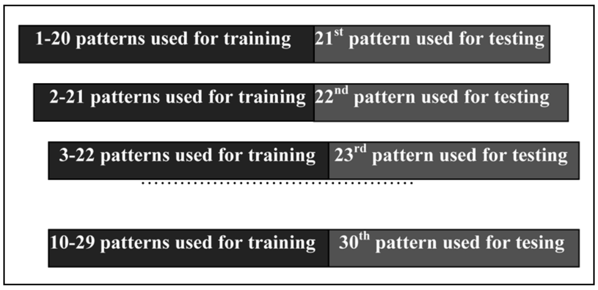


Figure 7. Optimum learning sequence.

optimum learning sequence is determined to be 20 (Figure 7). The Fuzzy BP was then trained with a different set of 20 training patterns i.e. (1-20, 2-21, 5-24, 10-29 training patterns and so on) and tested for 21^{st}, 22^{nd}, 25^{th}, 30^{th} corresponding day respectively, as shown in the results depicted by Figures 8, 9, 10 and 11 respectively.

Table 2 shows the observations from the comparison of the results obtained from Fuzzy BP and MLP for the case of optimum learning sequence. It also shows the comparison of MAPE and peak load percentage errors for all the days in a week. For most of the days in a week, the errors obtained from Fuzzy BP are less compared to those of MLP.

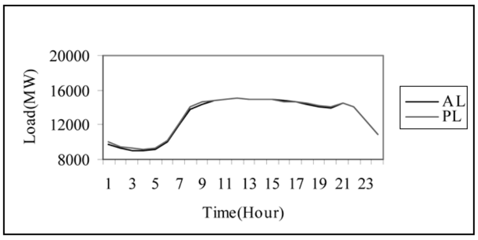


Figure 8. Comparison of actual and predicted loads of 21^{st} Thursday (using 1-20 Thursday patterns for training). Maximum % error=2%, MAPE=0.76% (AL – actual load; PL – predicted load).

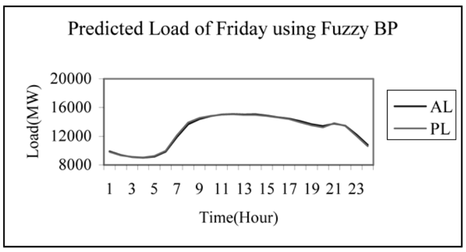


Figure 9. Comparison of actual and predicted loads of 22^{nd} Friday (using 2-21 Friday patterns for training). Maximum % error=1.61%, MAPE=0.78%.

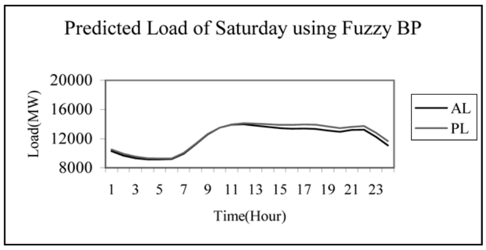


Figure 10. Comparison of actual and predicted loads of 25^{th} Saturday (using 5-24 Saturday patterns for training). Maximum % error=4.2%, MAPE=2.47%.

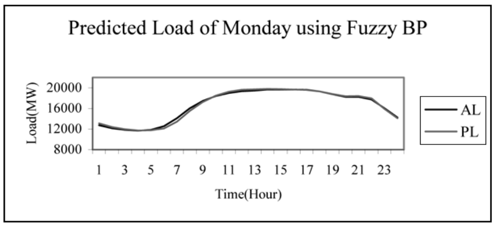


Figure 11. Comparison of actual and predicted loads of 30^{th} Monday (using 10-29 Monday patterns for training). Maximum % error =3.3%, MAPE=1.23%.

Day	Mean Absolute Percentage Error (MAPE)		Peak Load Percentage Error	
	Fuzzy BP	MLP	Fuzzy BP	MLP
Sunday	2.2310	2.5118	0.3379	1.1341
Monday	1.7124	2.2282	4.5084	5.3692
Tuesday	1.6452	1.2081	0.5222	0.2111
Wednesday	2.5848	2.2134	2.066	1.033
Thursday	0.7620	0.8766	0.0199	0.1665
Friday	1.7090	1.9353	0.2406	0.4768
Saturday	1.1981	1.8541	1.0942	1.6331

Table 2. Comparison of average and peak load percentage errors of Fuzzy BP and MLP.

5.1. Advantages of the Proposed Architecture

- The proposed architecture mainly works with fuzzy inputs.
- The fuzzy number being represented in LR-type reduces network complexity.
- The architecture is simple.
- The operator's knowledge in the form of "if. . . then" rules can also be directly applied as inputs without any preprocessing of the rules (Srinivasan et al., 1994).

5.2. Disadvantages of Proposed Architecture

- A little more calculations are required than for an ordinary MLP, as shown in Section 2.2. Hence the time taken to train the Fuzzy BP network is longer than that of ordinary MLP.

6. Conclusions

The following are the conclusions drawn from the present work:

- The optimum learning sequence for short term load forecasting is 20 training patterns.
- The proposed method does not require any heavy computational burden and can be easily implemented.
- Patterns to be considered for training have an impact on forecasting the load. High error is obtained in the case when the test data is not close enough to any one of the training data. The size of error increases when the forecasted day falls in the days interspersed between seasons, as well as if there is seasonal variation between the training patterns and the testing pattern.
- The hour at which forecasted load is obtained using MLP is also the corresponding hour of forecasted peak load of Fuzzy BP.
- The results show that peak load percentage error and average percentage error are considerably less for Fuzzy BP, compared to those of MLP.

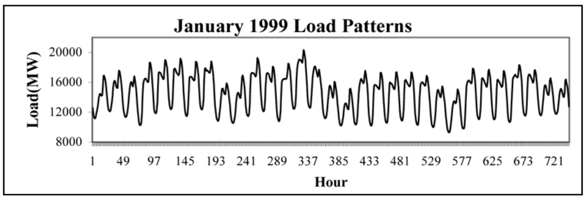
7. Scope for Future Work

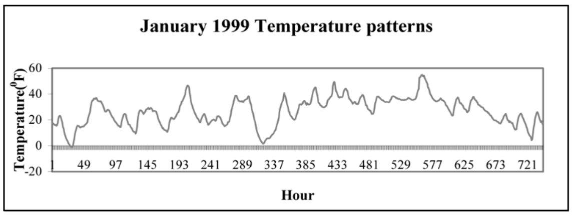
By using 'Classification techniques' or 'Clustering techniques' the patterns similar to that of the testing pattern can be selected (instead of choosing the patterns corresponding to the day to be forecasted) from the past data to train the network. This may further reduce forecasting error.

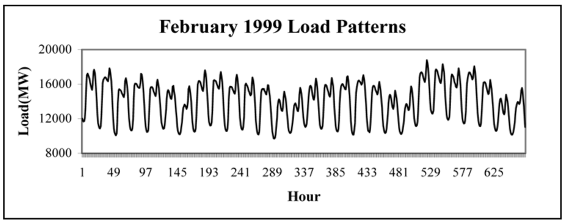
8. Acknowledgments

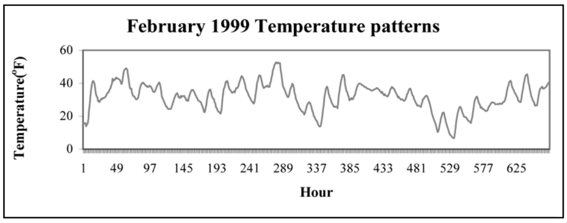
The authors gratefully acknowledge Dr. K. Shanti Swarup, Asst. Professor, IIT, Chennai, for giving all the necessary data and information and also thank Prof. M. Narasimha Murthy, CSA, IISc, Bangalore for all his valuable suggestions. The authors sincerely thank the reviewers for their valuable suggestions as well.

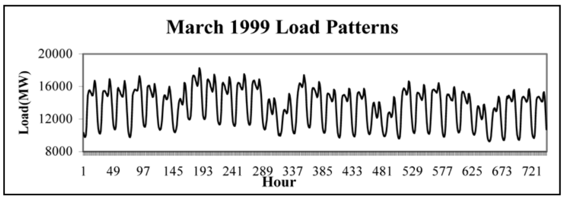
Appendix I: Figures of monthly load and corresponding temperature patterns

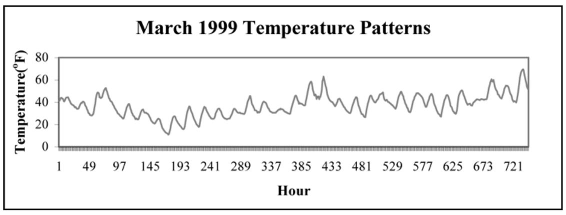


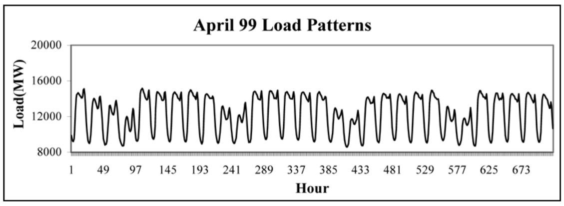


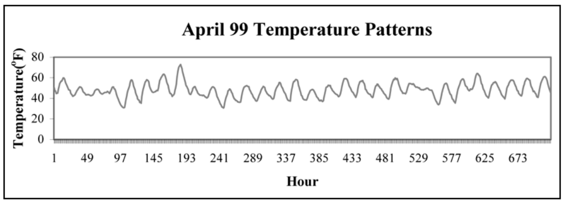


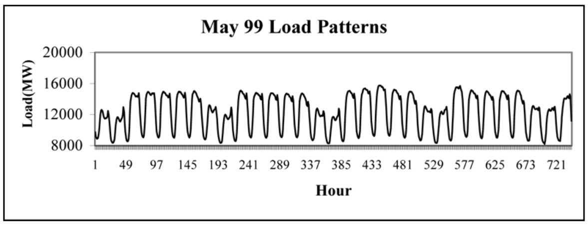


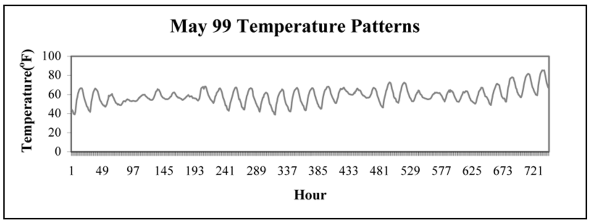


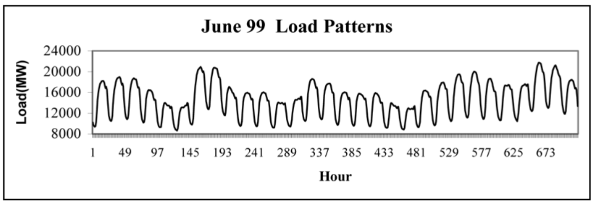


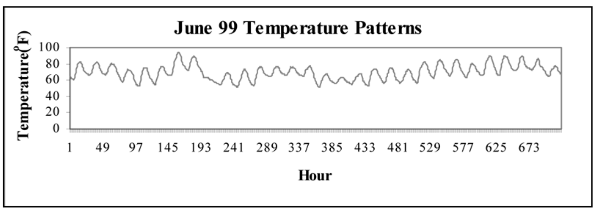


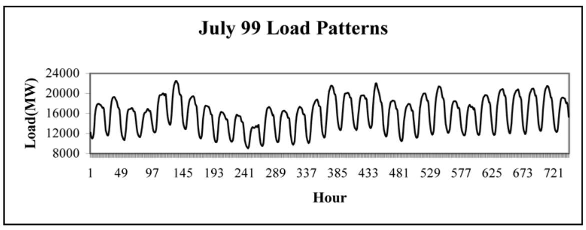


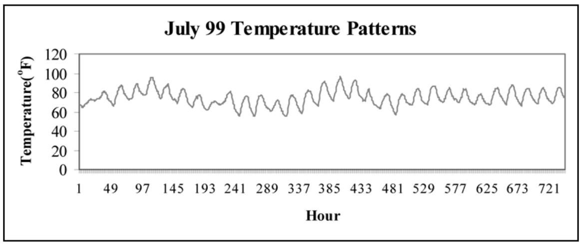












Appendix II: Terms used in Fuzzy BP Flowchart:

The terms used in the Fuzzy BP flowchart, as shown in Figure 6. are described as follows: W represents the weight matrix between input and hidden layer, V represents the weight matrix between hidden and output layer, $\overline{I}_p$ represents the training vector for pattern p, O'_{pj} represents the calculated output from j^{th} hidden node for pattern p, O''_{pk} represents the calculated output from k^{th} output node for pattern p, E represents sum squared error for all patterns, D_{pk} represents target output on k^{th} node for pattern p, η is the learning rate and E_g is the maximum error tolerance.

Appendix III: Description of Block Diagram of MLP Architecture:

The following figure shows the block diagram of MLP that uses back propagation learning to predict the 24 hours load ahead.

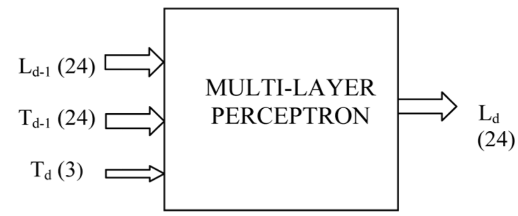


Figure 12. Block diagram of MLP architecture.

The optimal training parameters for MLP are also determined by trial and error approach. As an example, the forecasting error (MAPE) for three different sets of training parameters with varying number of training patterns is shown in Table 3. From the table, it can be observed that the forecasting error is less for most of the days when the MLP is trained using first 20 training patterns with training parameters being NH=10, $\eta = 0.79$, $\alpha = 0.59$.

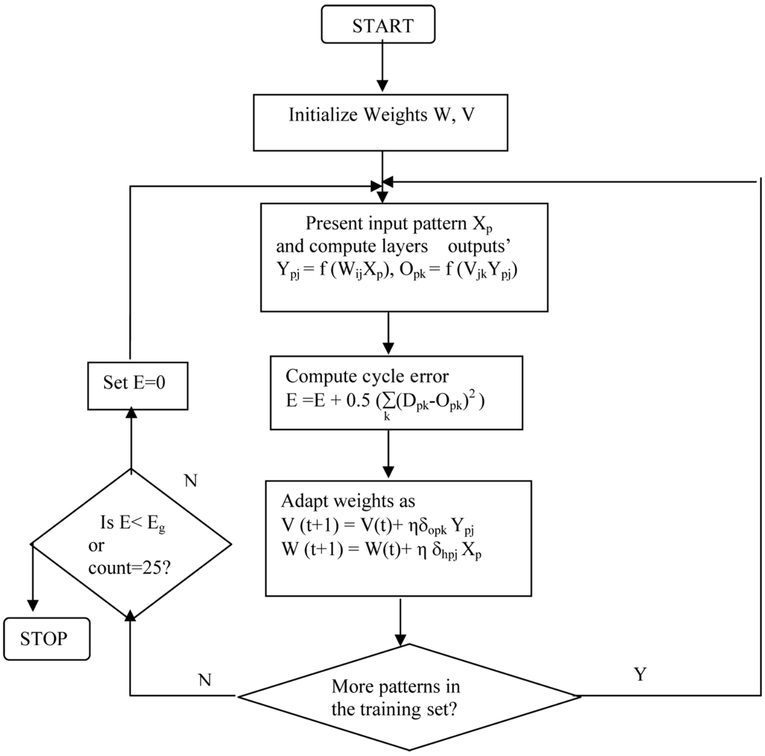


Figure 13. MLP flowchart.

Day	Training Parameters	MAPE		
		Training Patterns 1-15	Training Patterns 1-20	Training Patterns 1-25
Sunday	NH=8, $\eta = 0.5$, $\alpha = 0.7$	5.3173	5.6667	17.8925
	NH=10, $\eta = 0.79$, $\alpha = 0.59$	2.1275	2.5118	16.1565
	NH=12, $\eta = 0.9$, $\alpha = 0.1$	4.3836	2.9355	19.4294
Monday	NH=8, $\eta = 0.5$, $\alpha = 0.7$	6.4213	5.9944	16.6522
	NH=10, $\eta = 0.79$, $\alpha = 0.59$	2.1332	2.2282	13.6286
	NH=12, $\eta = 0.9$, $\alpha = 0.1$	6.0627	2.8552	18.3541
Tuesday	NH=8, $\eta = 0.5$, $\alpha = 0.7$	2.3890	2.3313	10.7516
	NH=10, $\eta = 0.79$, $\alpha = 0.59$	2.1033	1.2081	8.8998
	NH=12, $\eta = 0.9$, $\alpha = 0.1$	2.7166	1.1442	15.2960
Wednesday	NH=8, $\eta = 0.5$, $\alpha = 0.7$	3.5541	1.4265	5.7858
	NH=10, $\eta = 0.79$, $\alpha = 0.59$	1.9478	2.2134	4.8418
	NH=12, $\eta = 0.9$, $\alpha = 0.1$	2.9047	2.3717	6.9753
Thursday	NH=8, $\eta = 0.5$, $\alpha = 0.7$	2.2803	2.3522	5.4404
	NH=10, $\eta = 0.79$, $\alpha = 0.59$	2.6973	0.8766	4.0902
	NH=12, $\eta = 0.9$, $\alpha = 0.1$	1.9938	1.1101	4.0275
Friday	NH=8, $\eta = 0.5$, $\alpha = 0.7$	2.0755	3.0115	11.1599
	NH=10, $\eta = 0.79$, $\alpha = 0.59$	5.6754	1.9353	10.7262
	NH=12, $\eta = 0.9$, $\alpha = 0.1$	1.2903	1.3215	12.7522
Saturday	NH=8, $\eta = 0.5$, $\alpha = 0.7$	2.7219	3.1452	17.2967
	NH=10, $\eta = 0.79$, $\alpha = 0.59$	3.4889	1.8541	15.9874
	NH=12, $\eta = 0.9$, $\alpha = 0.1$	2.0256	1.7680	17.8278

Table 3. Comparison of average percentage error of MLP for different training parameters and training patterns.

Terms used in MLP flowchart

W – weight matrix between input layer and hidden layer.
V – weight matrix between hidden layer and output layer.
X_p – training vector for pattern p.
Y_{pj} – the output of j^{th} node in hidden layer for pattern p.
O_{pk} – calculated output from k^{th} output node for pattern p.
E – sum squared error (SSE) of all the patterns.
D_{pk} – actual output on k^{th} node for pattern p.
η – learning rate.
E_g – maximum error tolerance.

References

[1] A. M. AL-KANDARI, S. A. SOLIMAN, M. E. EL-HAWARY, Fuzzy Short-term Electric Load Forecasting. *International Journal of Electrical Power & Energy Systems*, (2004), Vol. 26, No. 2, pp. 111–122.

[2] N. AMJADY, Short term hourly load forecasting using time-series modeling with peak load estimation capability. *IEEE Transactions on Power Systems*, (2001), Vol. 16, No. 3, pp. 498–505.

[3] A. G. BAKIRTZIS, V. PETRIDIS, S. J. KIARTZIS, M. C. ALEXIADIS, A. H. MAISSIS, A neural network short-term load forecasting model for the Greek power system. *IEEE Transactions on Power Systems*, (1996), Vol. 11, No. 2, pp. 858–863.

[4] A. G. BAKIRTZIS, J. B. THEOCHARIS, S. J. KIARTZIS, K. J. SATSIOS, Short-term load forecasting using fuzzy neural networks. *IEEE Transactions on Power Systems*, (1995), Vol. 10, No. 3, pp. 1518–1524.

[5] M. BECCALI, M. CELLURA, V. LO BRANO, A. MARVUGLIA, Forecasting daily urban electric load profiles using artificial neural networks. *Energy Conversion and Management*, (2004), Vol. 45, pp. 2879–2900.

[6] O. A. S. CARPINTEIRO, A. J. R. REIS, A. P. A. DA SILVA, A hierarchical neural model in short-term load forecasting. *Applied Soft Computing*, (2004), Vol. 4, No. 4, pp. 405–412.

[7] W. CHARYTONUIK, M. S. CHEN, P. VAN OLINDA, Non-parametric Regression-based Short-term Load Forecasting. *IEEE Transactions on Power Systems*, (1998), Vol. 13, No. 3, pp. 725–730.

[8] S. T. CHEN, D. C. YU, A. R. MOGHADDAMJO, Weather sensitive short-term load forecasting using nonfully connected artificial neural network. *IEEE Transactions on Power Systems*, (1992), Vol. 7, No. 3, pp. 1098–1105.

[9] S. CHENTHUR PANDIAN, K. DURAISWAMY, C. CHRISTOBER ASIR RAJAN, N. KANAGARAJ, Fuzzy approach for short-term load forecasting. *Electric Power Systems Research*, (2006), Vol. 76, pp. 541–548.

[10] W. R. CHRISTIAANSE, Short term load forecasting using general exponential smoothing. *IEEE Transactions on Power Apparatus and Systems*, (1971), Vol. 90, No. 2, pp. 900–910.

[11] P. K. DASH, A. C. LIEW, S. RAHMAN, Peak load forecasting using a fuzzy neural network. *Electric Power Systems Research*, (1995), Vol. 32, No. 1, pp. 19–23.

[12] G. GROSS, F. GALIANA, Short-term Load Forecasting, *Proc. IEEE*, (1987), Vol. 75, No. 12, pp. 1558–1573.

[13] T. HAIDA, S. MUTO, Regression-based peak load forecasting using a transformation technique. *IEEE Transactions on Power Systems*, (1994), Vol. 9, No. 4, pp. 1788–1794.

[14] H. SEETHA, B. SATISH, K. S. SWARUP, R. SARAVANAN, Application of Data Mining in Electric Load Prediction Using Fuzzy BP. *Proceedings of International Conference on Systemics, Cybernetics and Informatics*, (2005), ICSCI-2005, Vol. 1, pp. 247–250.

[15] H. SEETHA, Data mining for short-term electric load prediction. *M. Ph. dissertation*, Vellore Institute of Technology (Deemed University), (2006).

[16] K. L. HO, Y. Y. HSU, C. C. YANG, Short-term load forecasting using a multilayer neural network with an adaptive learning algorithm. *IEEE Transactions on Power Systems*, (1992), Vol. 7, No. 1, pp. 141–149.

[17] Y. Y. HSU, K. L. HO, Fuzzy expert systems: an application to short-term load forecasting. *IEEE Proceedings on Generation, Transmission and Distribution*, (1992), Vol. 139, No. 6, pp. 471–477.

[18] J. C. LU, D. X. NIU, Z. Y. JIA, A study of short term load forecasting based on ARIMA-ANN. *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics*, (2004), ICMLC 2004, Vol. 5, pp. 3183–3187.

[19] A. KHOTANZAD, R. AFKHAMI-ROHANI, T. L. LU, A. ABAYE, M. DAVIS, D. J. MARATUKULAM, ANNSTLF-a neural-network-based electric load forecasting system. *IEEE Transactions on Neural Networks*, (1997), Vol. 8, No. 4, pp. 835–846.

[20] K. H. KIM, J. K. PARK, K. J. HWANG, S. H. KIM, Implementation of hybrid short-term load forecasting system using Artificial Neural Networks and Fuzzy Expert Systems. *IEEE Transactions on Power Systems*, (1995), Vol. 10, No. 3, pp. 1534–1539.

[21] K. B. SONG, Y. S. BAEK, D. H. HONG, G. JANG, Short-term load forecasting for the holidays using fuzzy linear regression method. *IEEE Transactions on Power Systems*, (2005), Vol. 20, No. 1, pp. 96– 101.

[22] L. TORRES, C. O. TRAORE, P. J. LAGACE, D. MUKHEDKAR, A knowledge engineering tool for load forecasting. *Proc. of 33rd Midwest Symposium on Circuits and Systems*, (1990), Vol. 1, pp. 144–147.

[23] R. LAMEDICA, A. PRUDENZI, M. SFORNA, M. CACIOTTA, V. O. CENCELLLI, A neural network-based technique for short-term forecasting of anomalous load periods. *IEEE Transactions on Power Systems*, (1996), Vol. 17, No. 3, pp. 1749–1756.

[24] L. H. MING, L. B. HUI, Fuzzy BP: A Neural Network Model with Fuzzy Inference. *Proc. ICANN*, (1994), pp. 1583–1588.

[25] K. Y. LEE, Y. T. CHA, J. H. PARK, Short-term load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*, (1992), Vol. 7, No. 1, pp. 124–132.

[26] D. P. LIJESEN, J. ROSING, Adaptive forecasting of hourly load based on load measurements and weather information. *IEEE Trans. PowerApp. Syst.*, (1971), Vol. PAS-90, No. 4, pp. 1757–1767.

[27] M. L. M. LOPES, C. R. MINUSSI, A. D. P. LOTUFO, Electric load forecasting using a fuzzy ART & ARTMAP neural network. *Applied Soft Computing*, (2005), Vol. 5, No. 2, pp. 235–244.

[28] C. N. LU, H. T. WU, S. VEMURI, Neural network-based short term load forecasting. *IEEE Ttransactions on Power Systems*, (1993), Vol. 8, No. 1, pp. 336–342.

[29] M. TAMIMI, R. EGBERT, Short-term electric load forecasting via fuzzy neural collaboration. *Electric Power Systems Research*, (2000), Vol. 56, pp. 243–248.

[30] S. E. PAPADAKIS, J. B. THEOCHARIS, S. J. KIARTZIS, A. G. BAKIRTZIS, A novel approach to short-term load forecasting using fuzzy neural networks. *IEEE Transactions on Power Systems*, (1998), Vol. 13, No. 2, pp. 480–492.

[31] A. D. PAPALEXOPOULOS, S. HAO, T. M. PENG, An implementation of a neural network-based load forecasting model for the EMS. *IEEE Transactions on Power Systems*, (1994), Vol. 9, No. 4, pp. 1956–1962.

[32] A. D. PAPPALEXOPOULUS, T. C. HESTERBERG, A regression-based approach to short-term load forecasting. *IEEE Transactions on Power Systems*, (1994), Vol. 5, No. 4, pp. 1535–1547.

[33] D. C. PARK, M. A. EL-SHARKAWI, R. J. MARKS II, L. E. ATLAS, M. J. DAMBORG, Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*, (1991), Vol. 6, No. 2, pp. 442–449.

[34] T. M. PENG, N. F. HUBELE, G. G. KARADY, Advancement in the application of neural networks for short-term load forecasting. *IEEE Transactions on Power Systems*, (1992), Vol. 7, No. 1, pp. 250–257.

[35] D. K. RANAWEERA, N. F. HUBELE, A. D. PAPALEXOPOULOS, Application of radial basis function neural network model for short-term load forecasting. *IEEE Proceedings on Generation, Transmission and Distribution*, (1995), Vol. 142, No. 1, pp. 45–50.

[36] B. SATISH, S. SRINIVAS, H. SEETHA, A study of neuro fuzzy systems and applications. *Bulletin of Pure and Applied Sciences*, (2002), Vol. 21E, No. 1, pp. 227–239.

[37] B. SATISH, K. S. SWARUP, S. SRINIVAS, A. HANUMANTHA RAO, Effect of temperature on short term load forecasting using an integrated ANN. *Electric Power Systems Research*, (2004), Vol. 72, No. 1, pp. 95–101.

[38] S. SRINIVAS, H. SEETHA, B. SATISH, A review on fuzzy logic and applications. *Engineering Today*, (2002), Vol. IV, No. 3, pp. 6–13.

[39] D. SRINIVASAN, M. A. LEE, Survey of Hybrid Fuzzy Neural Approaches to Electric Load Forecasting. *Proc. of the IEEE Conference on System, Man and Cybernetics*, (1995), Vol. 5, pp. 4004–4008.

[40] D. SRINIVASAN, A. C. LIEW, C. S. CHANG, Forecasting Daily Load Curves Using a Hybrid Fuzzy-Neural Approach. *IEEE Proceedings on Generation, Transmission & Distribution*, (1994), Vol. 141, No. 6, pp. 561–567.

[41] D. SRINIVASAN, A. C. LIEW, C. S. CHANG, Demand Forecasting Using Fuzzy Neural Computation, with Special emphasis on Weekend and Public Holiday Forecasting. *IEEE Transactions on Power Systems*, (1995), Vol. 10, No. 4, pp. 1897–1903.

[42] D. SRINIVASAN, S. S. TAN, C. S. CHANG, E. K. CHAN, Practical implementation of a hybrid fuzzy neural network for one-day-ahead load forecasting. *IEEE Proceedings on Generation, Transmission and Distribution*, (1998), Vol. 145, No. 6, pp. 687–692.

[43] J. W. TAYLOR, R. BUIZZA, Neural network load forecasting with weather ensemble predictions. *IEEE Transactions on Power Systems*, (2002), Vol. 17, No. 3, pp. 626–632.

[44] A. K. TOPALLI, I. ERKMEN, A hybrid learning for neural networks applied to short term load forecasting. *Neurocomputing*, (2003), Vol. 51, pp. 495–500.

[45] A. K. TOPALLI, I. ERKMEN, I. TOPALLI, Intelligent short-term load forecasting in Turkey. *International Journal of Electrical Power and Energy Systems*, (2006), Vol. 28, No. 7, pp. 437–447.

[46] C. TRANCHITA, A. TORRES , Soft computing techniques for short term load forecasting. *IEEE Power Systems Conference and Exposition*, (2004), Vol. 1, pp. 497–502.

[47] H. C. WU, C. LU, Automatic fuzzy model identification for short-term load forecast. *IEEE Proceedings on Generation, Transmission and Distribution*, (1999), Vol. 146, No. 5, pp. 477–482.

Received: December, 2006
Revised: April, 2007
Accepted: June, 2007

Contact addresses:

Hari Seetha
School of Computing Sciences
VIT University
Vellore – 632014
India
e-mail: `hariseetha@gmail.com`

R. Saravanan
School of Computing Sciences
VIT University
Vellore – 632014
India

HARI SEETHA obtained her Master's degree in engineering physics with specialization in electronics from National Institute of Technology (formerly R. E. C.) Warangal, Post Graduate Diploma in computer applications from Pondicherry Central University and M.Phil degree from V. I. T. University, India and is presently pursuing Ph. D. She has research interests in the fields of data mining, text mining and machine learning. She has about 5 years of research experience and 13 years of teaching experience. She has published a few research papers in national and international journals and conferences. She is also a co-investigator to a major research project sponsored by the Department of Science and Technology, Government of India.

DR. R. SARAVANAN is an Associate Professor in the School of Computing Sciences at V. I. T. University, Vellore, India. He received Ph. D. from Ramanujan Institute for Advanced Study in Mathematics, University of Madras in 1998. He has about fourteen years of teaching and research experience. His areas of specialization include digital image processing, mobile computing and computer networks and security. He is a life-time member of the Computer Society of India and Ramanujan Mathematical Society. He has published ten papers in various international and national journals and conferences.