

Short-term electric load forecasting using fuzzy neural networks

Short term load forecasting for energy conservation is an ideal choice to keep load curve flat. In this paper, a novel neuro-fuzzy system is considered for short term load forecasting. The proposed neuro-fuzzy system utilizes a fuzzy expert system for the initialization of connections and link weights of the neural network. Then, a Kalman filter algorithm is used for the weight updation of the network.

Introduction

Energy is a vital input in the process of economic, social and industrial development. Energy sources are either non-renewable (replenished slowly or renewable (replenished more rapidly). In one hand, the exploitation of renewable energy sources to provide comparatively a small part of the required energy with better environment and ecology has been steadily progressing, and on the other hand, non-renewable conventional energy generation, which provides the major part of the required energy, have been facing escalating construction costs, growing environmental and siting constraints. The future energy scenario with enormously increasing number of industries can be fulfilled with non-renewable energy sources. Augmenting the generation capacity is not a viable solution to meet energy demand, therefore, researchers are constantly working to minimize losses, and maximizing the energy in industrial sectors. To face this increasing energy requirements, utilities can either invest in supply side options (new generation, transmission and distribution facilities) or in demand side options (end-use equipment providing the same or better level of services but using less energy or peak power). Load management plays a very important role in energy conservation. In countries like India, where load demand is increasing rapidly as compared to load growth, it is required to implement an efficient load management scheme to provide a secured high quality electrical energy to customer at minimum cost. This can be achieved by

- Load forecasting- appriori knowledge of the load
- An appropriate economic despatch

Messrs A. K. Swain, B. Subudhi and R. K. Swain, Department of Electrical Engineering, Indira Gandhi Institute of Technology, Sarang, Dhenkanal-759 146, Orissa.

This paper deals with the first method, i.e., power system load forecasting by using more advanced artificial intelligence technique, the neuro-fuzzy system (Spl. issue, 1992; Kosko, 1992).

Load forecasting

Power system load forecasting (Asbury, 1975; Box et al., 1976; Vemuri et al., 1981; Sharma et al., 1974) can broadly be classified into

- Long term forecasting— lead time of 5 years
- Medium term forecasting— lead time 1 month to 5 years
- Short term load forecasting— lead time 1 hour to few months
- Micro term or very short term forecasting— few minutes to one hour ahead

Long term load forecasting is necessary for power system planning like capacity expansion, price, and regulatory policy determination. Medium term load forecasting is necessary for maintenance scheduling, coordination of inter utility power transfer, and price settings. Short term load forecasting is necessary for short term unit maintenance scheduling, economic scheduling of generating capacity, scheduling of fuel purchase, security analysis, unit-commitment, and demand side management. The load models for most of these are categorized as

- Non weather model— Extrapolates the time of day, and the latest load behaviour
- Weather load model— Extrapolates past load records, and some forecasted key weather variables
- Composite or mixed model— Utilizes and stochastic nature of the load and weather variables

The conventional methods of load forecasting (Dash et al., 1992) use smoothing techniques, regression methods and statistical analysis.

Artificial intelligence based load forecasting

The different AI based techniques (Swain et al., 1993) are

- Knowledge based expert system
- Fuzzy logic method
- Neural networks, and
- Fuzzy-neural method

Knowledge based expert system (Rahman et al., 1988, 1990; Jabbour et al., 1988; Park et al., 1989) and fuzzy expert systems need the help of an "expert" capable of accurate forecast. Some forms of the neural networks and fuzzy neural networks have captured the attention of load forecasting researchers (Ho, 1992; Lee et al., 1991, 1992; Dillon et al., 1991; Sharkawi et al. 1991). Neural networks does not depend on expert rules and interrelationships between past loads and weather variables. Neural networks has its advantages like noisy or incomplete data handling, robust, high degree of fault tolerance, high parallel computation, and self learning. For all AI based techniques, first of all an appropriate load model is to be decided. The different models for the neural networks are discussed in the next section.

Neuro-fuzzy systems, basically integrate fuzzy logic and neural networks in two ways :

- Neural network power is used within the larger framework of a preexisting fuzzy model.
- Incorporation of notion of fuzziness onto a neural network.

In the first category, neural networks are used to implement different fuzzy logic operations such as Union (max-nets), Intersection (min-nets), derivation of optimal rule sets for fuzzy controller (Berenji, 1992; Keller et al., 1992; Hall, 1991), and rule selection from a large rule set (Passino et al., 1992). Whereas, in the second category the network is used as a classifier with fuzzy target outputs, alternation of intergrator/transfer functions at each node, so that they perform some sort of fuzzy aggregation on the numerical information arriving at each node, or input data to the network may be fuzzified.

In this paper we have used a new concept of putting fuzzy knowledge for weight and connection initialization along with a very fast learning algorithm based on Kalman filtering technique.

Proposed neuro-fuzzy system

In this neuro-fuzzy system the neural network learns from the knowledge of an expert, and also this changes climatic conditions. The knowledge base of the fuzzy logic controller is used to initialize the internal connections and link weights of the network. Then, a Kalman filter algorithm along with the backpropagation algorithm is used to train the network, which results in a very fast network. This Kalman filter based backpropagation algorithm changes the symbolic representation of the network. This new symbolic representation is then

returned to a fuzzy logic representation, and thus the system acquires knowledge. The entire scheme is illustrated in Fig.1 This algorithm is given in Table 1.

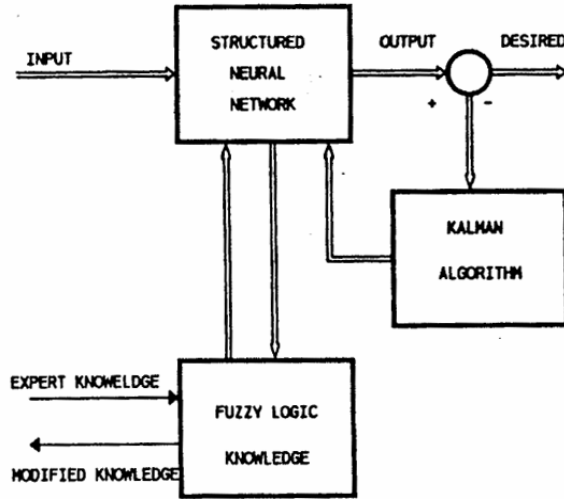


Fig.1 Block schematics of neuro-fuzzy system

TABLE 1 NEURO-FUZZY ALGORITHM

Step 1 :	Symbolic representation of expert knowledge
Step 2 :	Initialization of connections and its weights by symbolic representation
Step 3 :	Weight updation using the Kalman filter algorithm along with the backpropagation technique
Step 4 :	Distributed knowledge transformation into symbolic representation

In the Kalman filter algorithm for weight updation during the learning phase, the mean square error is minimized with respect to the summation outputs, i.e., input to nonlinearities, in contrast to the conventional backpropagation algorithm, which is with respect to the weights. This technique increases the convergence speed by 80% (Scalero et al., 1992). The following steps highlight the weight updation scheme for jth layer of the network. The Kalman gain vector is

$$K_j(t) = \left[R_j^{-1}(t-1)x_{j-1}(t) \right] / \left[b_j + x_{j-1}^T(t)R_j^{-1}(t-1)x_{j-1}(t) \right]$$

where R is the correlation matrix of the training set, x is the input vector, and b is the forgetting factor. The weight updation equation is

$$R_j^{-1}(t) = \left[R_j^{-1}(t-1) - K_j(t)x_{j-1}^T R_j^{-1}(t-1) \right] b_j^{-1}$$

The forgetting factor is updated by

$$b_j = b_0 b_j + (1-b_0)$$

Then, the weight updation equations are

Output layer :

$$W_{LK}(t-1) + K_L(t)(d_k - y_k)$$

Hidden layer :

$$W_{jk}(t) = W_{jk}(t-1) + K_j(t) e_{jk}(t) a_j$$

Where d is the desired summation output, L is the output layer, y is the calculated output, e is the error, and a_j is the backpropagation step size.

Load models

In neural network based short term load forecasting [12-16] apart from the tuning parameter selection, input output data selection and scaling is very important. Of course, the output in most of the cases is a load. The possible models are

MODEL I

$$y(i,t) = f[w(i,t), y(i,t-1), \dots, y(i,t-m), \\ y(i-1,t), y(i-1,t-1), \dots, y(i-1,t-m), \\ \vdots \\ y(i-n,t), y(i-n,t-1), \dots, y(i-n,t-m)]$$

where y stands for the load, $f(\cdot)$ is some unknown function, w is the neural networks connection strength, m and n indicate the data length, i and t stands for day and instant, respectively.

MODEL II

For no number of outputs, where n varies between 1 and m

$$y(i, n) = f[w(i,t), y(i-1,t), \dots, y(i-1,t-m), \\ y(i-2,t), y(i-2,t-1), \dots, y(i-2,t-m), \\ \vdots \\ y(i-n,t), y(i-n,t-1), \dots, y(i-n,t-m)]$$

MODEL III

The load is

$$y(i, t) = f[w(i,t), y(i,t-1), T(i,t-1), H(i,t-1), T(i,t), H(i,t), \\ y(i-1,t), T(i-1,t-1), H(i-1,t-1), T(i-1,t), H(i-1,t), \\ T(i-1)_{max}, T(i-1)_{min}, H(i-1)_{max}, H(i-1)_{min}, Y(i-1), \\ \vdots \\ y(i-n,t), T(i-n,t-1), H(i-n,t-1), T(i-n,t), H(i-n,t), \\ T(i-n)_{max}, T(i-n)_{min}, H(i-n)_{max}, H(i-n)_{min}, Y(i-n)]$$

where H is the humidity, T is the temperature, and Y is

the total load.

MODEL IV

Load on i -th day and t -th instant is

$$y(i,t) = f(\cdot) + f \left[w_j \left(\sum_j I_j + \prod_j I_j \right) \right]$$

Where $f(\cdot)$ is the output as calculated in Model I to III, I_j is the j -th input variable, and w_j is the multiplier for j -th input.

Similarly, other models can be formed with addition of other environmental parameters but the accuracy will remain more or less same as those calculated in the above models. These four models serve to be the basic load models. This prediction is based on the fact that the auto correlation function of the hourly load variation shows peaks at the multiple of 24 hour lags, i.e., the load variables at the same hours of different days have very strong correlations.

Scaling of input variables to the neural network has the major role on the convergence and accuracy of the results. We have used the following scaling :

$$I_j^k = (I_j^k - I_j) / \sigma_j$$

$$O^k = (O^k - O) / \sigma_o$$

($J = 1, \dots$ number of inputs)

($k = 1, \dots$ number of patterns)

where I_j is the average value for J -th component of the input vector, σ_j is the standard deviation of I_j^k , O^k is the output tangent value, and σ_o is an estimate of the standard deviation of O^k .

Neuro-fuzzy implementation and results

The entire program has been developed in an Intel 80486 based system in C++ programming language under DOS. All the four models for one hour ahead prediction and the results so obtained are compared with the Kalman filter based backpropagation algorithm, which is illustrated in Table-2. For all the four models the common parameters on which the comparison has been made are

Number of iterations	= 400
Sigmoid slope	= 0.08
Forgetting factors	= 0.2 and 0.98
Backpropagation step size	= 40
Kalman initialization constant	= 1
Initial weights	= -10 to +10

MODEL I

Here, $m = n = 2$

TABLE 2 COMPARISON OF RESULTS OF KALMAN FILTER BASED ALGORITHM AND NEURO-FUZZY SYSTEM

Sl. no.	Model error in %age							
	Kalman filter based net				Neuro-fuzzy system			
	1	2	3	4	1	2	3	4
1	0.013	0.0135	0.012	0.011	0.011	0.0095	0.009	0.0082
2	0.014	0.0138	0.015	0.0123	0.012	0.0128	0.010	0.0101
3	0.014	0.0152	0.032	0.0131	0.013	0.0141	0.029	0.0129
4	0.012	0.0131	0.078	0.0102	0.011	0.0123	0.069	0.0100
5	0.015	0.0172	0.090	0.0135	0.012	0.0151	0.081	0.0128
6	0.015	0.0178	0.120	0.0139	0.013	0.0168	0.098	0.0121

Neural network structure has 2 layers with 8 inputs, 14 hidden layer neurons, and 1 output neuron.

MODEL II

$n = 2$ and $m = 24$

The neural network structure has 2 layers with 48 inputs, 76 hidden neurons, and 24 output neurons.

MODEL III

$n = 0$, T_{max} , T_{min} , H_{max} , H_{min} , Y are not considered for prediction purpose.

The neural network structure is a 2 layer network with 5 inputs, 7 hidden layer neurons, and 1 output neuron.

MODEL IV

Tested for Model I output and the additional weights, ignoring product terms, are 0.001, 0.1, 0.001, 0.015 and 0.001, respectively.

$m = n = 2$

The two layer neural network with 8 inputs, 14 hidden layer neurons, and 1 output.

Conclusion

This paper deals with the conservation of energy with proper load management via load forecasting. Energy saving is possible with proper implementation of a good innovative load forecasting technique. We are now constantly working for further development of this research area.

References

1. Special issue on Fuzzy Neural Networks (1992) : *IEEE Trans on Neural Networks*, Vol.3, No.5.
2. Kosko, B. (1992) : *Neural Networks and Fuzzy Systems*. Prentice Hall Inc.
3. Asbury, C. E. (1975) : Weather Load Model for electric Demand and energy Forecasting. *IEEE Trans on*

Power Appr and Sys., Vol. PAS-94, 1111-1116.

4. Box, G.E.P. and Jenkins, G.M. (1976) : *Time Series Analysis, Forecasting and Control*. Holden Day.
5. Vemuri, S., Huang, W.L. and Nelson, D.J. (1981) : On-line Algorithms for Forecasting Hourly Loads of an Electric Utility. *IEEE Trans on Power Appr. and Sys.*, Vol. PAS-100, 3775-3784.
6. Sharma, K. L. and Mahalanabis, A.K. (1974) : recursive Short term Load Forecasting Algorithm. *Proc. IEE*, Vol. 121 59-62.
7. Dash, P.K., Swain, A.K., Mohapatra, T. and Nanda, P.K. (1992) : A Study on Time Series Load Forecasting. 37th Annual Session of Orissa Congress, BBSR.
8. Swain, A. K., Patra, K. C. and Swain, R. K. (1993) : Neural Network Based Short term Load Forecasting for energy Conservation. *Technical Annual of IE (I) Orissa State Center*, Thirty Fifth Session, 102-106.
9. Rahman, S. and Bhatnagar, R. (1988) : "An Expert System Based Algorithm for Short Term Load Forecast" *IEEE Trans on Power Systems*, Vol. 3, No. 2, 392-399.
10. Rahman, S. (1990) : Formulations and Analysis of a Rule Based Short-term Load Forecasting Algorithm. *Proc. IEEE*, Vol. 78, No. 5, 805-816.
11. Jabbour, K., Rivcros, J., Landbergen, D. and Meyer, W. (1988) : ALFA : Automated Load Forecasting Assistant. *IEEE Trans. on Power Sys*, Vol.3, No. 11, Aug. 1988, 908-914.
12. Park, Y. and Park, J. K. (1989) : An Expert System for Short Term Load Forecasting by Fuzzy Decision. *Proceedings of ESAPS*, 244-250.
13. Ho, K. L. (1992) : "Short term Load Forecasting using a Multilayer Neural Network with an Adaptive Learning Algorithm", *IEEE Trans. on Power Sys.*, Vol.7, No.1, 141-149

(Continued on page 10)