IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 9, Issue 4 Ver. II (Jul – Aug. 2014), PP 01-06
www.iosrjournals.org

Load Forecasting For Weekend Load Using ANN Technique in Deregulated Environment

Anju Bala[1], N.K.Yadav[2]
[1,2] (E.E. Department, D.C.R. University of science & Technology,
Murthal (Sonepat)-Haryana, INDIA)

Abstract: *In this paper, artificial neural network technique is implemented to predict weekend load. The performance of methodology is compared with the actual data received from a power station operating in northern region of India. This technique accurately forecast the hourly load for weekend days. Results obtained from training and testing of neural network confirms the validity of proposed methodology. In this approach demand sensitivity due to change in some weather variables such as temperature and humidity is also considered. The accuracy of proposed method is given in terms of Mean Absolute Percentage Error (MAPE).*

Keywords: *Artificial Neural Network, (ANN), Back Propagation Algorithm (BPA), Short-Term Load Forecasting (STLF).*

I. Introduction

Short-term load forecasting predicts the electricity demand over an interval from one hour to one week. Due to requirement of forecasting in power industries, it becomes a major area of research not only for production side but also from financial perspective. To match the generation and demand of electricity, tracking of load is necessary for reliable operation of power system. To do this, power utility requires an adequate model for operation and control which is fulfilled by accurate load forecasting. It looks like a core process for operational decision such as economic load dispatch, hydro scheduling and optimal power flow as well as for effective economic planning [1]. The accuracy of applied technique is presented in the form of an error. Lesser the error, more accurate the forecasting results. If the error is more, it will increase the operational cost of system [2]. Some traditional economic approaches, such as interpolation and regression, may not give adequate results. Ecumenical problem with time series method includes numerical instability and insufficient accurate results. The problem of inaccurate results in this method is due to absence of weather information [3]. Because there is an inviolable correlation between power consumption behavior and weather variables such as humidity, cloud cover and wind speed, etc. This is particularly unfeigned in residential areas [4]. The time series approaches mostly employ computationally cumbersome matrix-oriented adaptive algorithms which, in certain cases, may be precarious. The Box-Jenkins method needs autocorrelation function to distinguish autoregressive moving average (ARMA) model [5]-[7]. This can be achieved by applying pattern recognition techniques. A major obstruction here is its sluggish performance [8]. Recently, with the exploitations of artificial intelligence techniques, alternative solutions to the STLF problem are proposed. Artificial neural network successfully implemented to solve short-term load forecasting. Neural network is a promising area of intelligent technique due to their learning abilities from the environment. Neural network utilizes a network algorithm which includes a combination of both linear and nonlinear terms. The network algorithm maps past load data and temperature as input to predict the load forecast output [9].

In this paper the back propagation algorithm is proposed as a methodology for electric load forecasting. A nonlinear load model is intimated and the parameters of the nonlinear load model are calculated using the back propagation algorithm. Test results show a satisfactory use of the ANN, and the percentage forecasting error is about 1.3 %. Papalexopoulos et. al. proposed a new scheme that includes the additional input variables such as cooling/heating degree, seasonal factors into a single network [10].

For radial basis function, feed forward and dynamic neural networks, the weights are adjusted using gradient-based methods. The Levenberg-Marquardt algorithm is used by default, because for many problems it is considered to be the best choice. Another feature of this algorithm is that it can take advantage of a site where a network is linear in some of its parameters. The training speed is increased considerably by using both linear and nonlinear terms [11]. When the data set is large, consequently, neural network model becomes large, while the training algorithms used for some type of networks become intensive. Using same load data, the forecasting can be done separately for weekdays and weekend days. In neural network, two approaches are commonly used. In first approach, future load is predicted using load pattern as time series. In second approach, future load is highly dependent on previous load pattern and weather variables. Weather variables play a major role in load forecasting by changing daily load pattern. Effective transaction such as Available Transfer Capability (ATC) also depends on load forecasting [12].

II. Artificial Neural Network

Artificial neural networks are used in many applications of power system and also play a major role in short-term load forecasting. Neural networks are like physical cellular networks that have ability to store and utilize experimental knowledge. Artificial neural networks are interconnection of neurons in which both lag-free and delay connections are permitted. In biological neurons, three major parts: axon, cell body and dendrites are presented. Axons are long cylindrical connections used to carry impulse from one neuron to other. Dendrites are thin fiber bush around neuron's body which receives information from other neurons. The signals received at synaptic junctions are electrical impulses. The neuron's response in network is generated only if the membrane potential reaches a threshold value. Similarly, if the incoming impulse causes firing then it is said to be excitatory. And if the incoming impulse response cause hinders the firing, it is said to be inhibitory.

Artificial neural network trained to execute a desired function by adjusting the values of the links (weights) between elements. Commonly ANN's are adjusted, or trained, so that a specific input leads to a particular target output. Therefore, the network is aligned, based on a comparison of the output and the target, until the network output equalizes the target. Generally many input/target pairs are required to train a network. A simple ANN concept is shown in Fig. 1 [13].

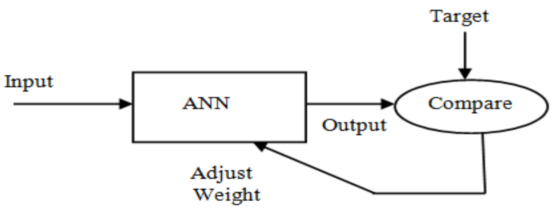

Fig.1: Simple ANN Network.

The neuron obtains (numerical) information by a number of input nodes, operates it internally, and puts out a response. The processing is commonly done in two stages: first, the input values are linearly combined, and then the result is used as the argument of a nonlinear activation function.

The combination uses the weights attributed to each connection, and a constant bias term. Fig. 2 shows one of the most commonly used schemes for a neuron.

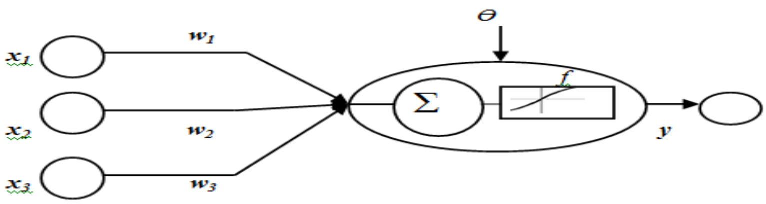

Fig.2: An Artificial Neuron.

The neuron output is given by:

$$y = f\left[\left(\sum w_i - x_i\right) - \theta\right]$$

Where x_i is the neuron input multiplied by weight link w_i, Θ is the neuron offset (bias), and f is the activation function. In multilayered feed-forward network, the most commonly used activation function is tan-sigmoid. The tan-sigmoid activation function is shown in Fig. 3 and the output is limited between [–1, 1].

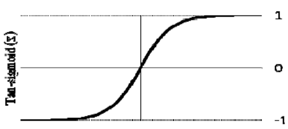

Fig.3: Tan-sigmoid function.

The output of the function is formulated as:

$$f(x) = \frac{2}{1 + \exp(-2x)} - 1$$

The neurons are organized in such a way that describes the network structure. In artificial neural network, three layered network is used to solve complex problems. Multilayer Perceptron model is used in the proposed scheme. In this model, the neurons in each layer are not linked to each other but may share the same input. The layer between the input neurons and the output layer may be one or more and called the hidden layer.

III. Implementation Of Back Propagation Algorithm

In the multi-layered feed-forward Artificial Neural Network (ANN), the back propagation algorithm is used. This means that the artificial neurons are organized in layers, and transmit their signals forward, and then the errors are propagated backwards. It may be viewed as a generalization of an equally popular adaptive filtering algorithm, the least mean square (LMS) algorithm. Error back-propagation learning consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an input vector is applied to the nodes of the network, and its effect propagates through the network layer by layer [14]. Finally, a set of outputs is produced as the actual response of the network. During the forward pass the weights of the networks are all fixed. During the backward pass, the weights are all adjusted in accordance with an error correction rule. The actual response of the network is subtracted from a desired response to produce an error signal. The network exhibits a high degree of connectivity. A change in the connectivity of the network requires a change in the population of their weights. To illustrate the process a three layer neural network with two inputs and one output, is shown in the Fig.4.

Training of a network is an iterative process. From training data set, weight coefficients are modified at each node during iteration process. Symbols w_{mn} represent weights of connections between output of neuron m and input of neuron n in the next layer. Output signal of neuron n is represented by y_n.

$$Y_1 = f_1(w_{11}x_1 + w_{21}x_2)$$
$$Y_2 = f_2(w_{12}x_1 + w_{22}x_2)$$
$$Y_3 = f_3(w_{13}x_1 + w_{23}x_2)$$
$$Y_4 = f_4(w_{14}y_1 + w_{24}y_2 + w_{34}y_3)$$
$$Y_5 = f_5(w_{15}y_1 + w_{25}y_2 + w_{35}y_3)$$
$$Y_6 = f_6(w_{46}y_4 + w_{56}y_5)$$

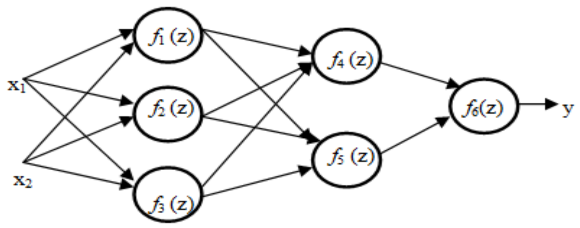

Fig.4: Three layer neural network with two inputs and single output.

Where

- z – Adder output signal.
- y – f (z) output of nonlinear element.
- x_1, x_2 - input signal.

The difference in target and output value is called error signal and denoted by δ.

$$\delta = y' - y$$
$$\delta_4 = w_{46}\,\delta$$
$$\delta_5 = w_{56}\,\delta$$
$$\delta_3 = w_{34}\delta_4 + w_{35}\delta_5$$
$$\delta_2 = w_{24}\delta_4 + w_{25}\delta_5$$
$$\delta_1 = w_{14}\delta_4 + w_{15}\delta_5$$

Weight correction = learning rate parameter*local gradient*i/p signal of neuron i.

$$\Delta W_{ij}(n) = \eta.\ \delta_i(n).y_i(n)$$

The network repeats the process until the error reaches an acceptable value which means that the neural network was trained successfully. If a maximum count of iterations is reached without attaining the goal, this implies that the ANN training was not successful. Flow chart for back-propagation algorithm is shown in Fig. 5.

IV. Training Parameters And Simulation Results

A system is developed in the paper that predicts 24 hour at a time load demand. As the input is past 24 hours load, a back propagation neural network is designed to train input data according to target data. Although in MATLAB® 2010a, neural network toolbox training parameters are predefined but these can be changed for better performance.

The parameters considered are:

No. of Epochs = 1000
Learning rate parameter = 0.4
Momentum = 0.03

The Levenberg-Marquardt algorithm is designed without computing the Hessian matrix [12]. When the performance function has the form of a sum of squares (as is typical in training feed- forward networks), then the Hessian matrix can be approximated as:

$$H = J^T J$$

and the gradient can be computed as

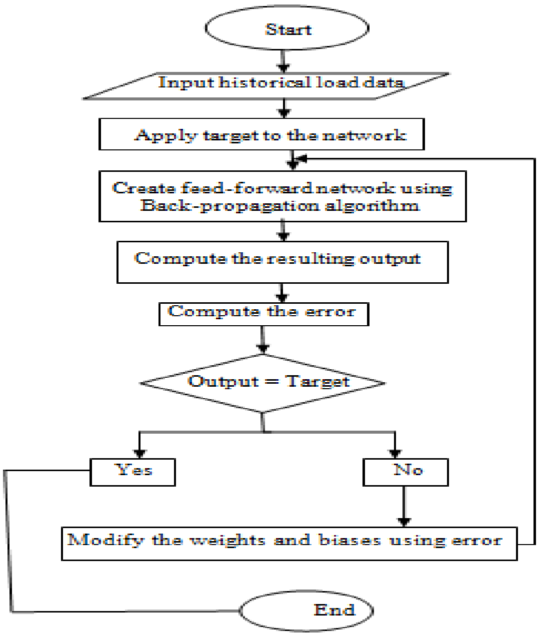

Fig.5: Flow Chart of proposed scheme.

$$g = J^T e$$

The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$X_{k+1} = X_k - [J^T J + \mu I]^{-1} J^T e$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift toward Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. The input load data is given in table 1.

The mean absolute percentage error is measured by:

$$MAPE = [\{\sum(|A_L - P_L| / A_L)\}/N]*100$$

Where-

A_L: Actual Load.
P_L: Predicted Load
N: No. Of Hours for Which Load is predicted.

MAPE of

Friday $= [\{|(286.7-382)|/286.7\}*100]/24\%$

= 1.38%

Table 1: Load profile data set for 1 March, 2013, DHBVN Ltd; Haryana (India)

Time	Voltage	Current	Temperature	Humidity
1:00	22.8	560	17	20.8
2:00	22.8	560	17	20.65
3:00	22.8	560	16	20.5
4:00	22.6	560	15	20.2
5:00	21.6	700	14	19.9
6:00	21.2	860	14	19.55
7:00	22.2	280	15	19.2
8:00	22.8	280	17	18.85
9:00	22.8	250	18	18.5
10:00	23.2	250	20	18.3
11:00	22.6	250	21	18.1
12:00	28.2	274	25	18.1
13:00	22.8	270	24	18.1
14:00	20.4	870	24	18.2
15:00	20.8	840	24	18.3
16:00	22.8	190	24	18.65
17:00	23	270	23	19
18:00	23	310	23	19.45
19:00	19.2	730	23	19.9
20:00	21.6	710	22	19.85
21:00	20.2	610	20	19.8
22:00	23	510	19	19.15
23:00	22.9	460	18	18.5
0:00	23.6	460	17	18.25

Forecasting results for weekend days are shown below:

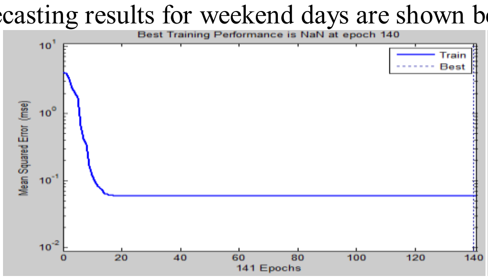

Fig.6: Mean Square Error Vs Epoch.

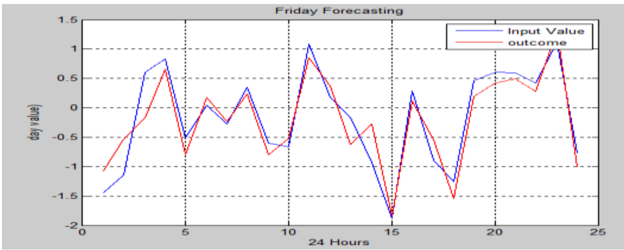

Fig.7: Comparison Plot between actual and forecast load for Friday.

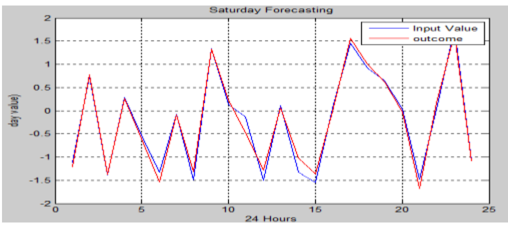

Fig. 8: Comparison Plot between actual and forecast load for Saturday.

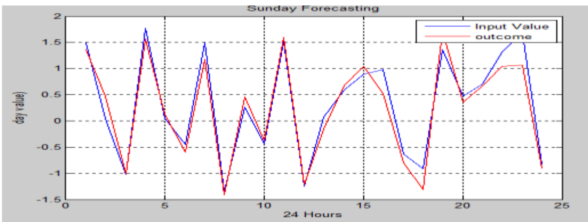

Fig.9: Comparison Plot between actual and forecast load for Sunday.

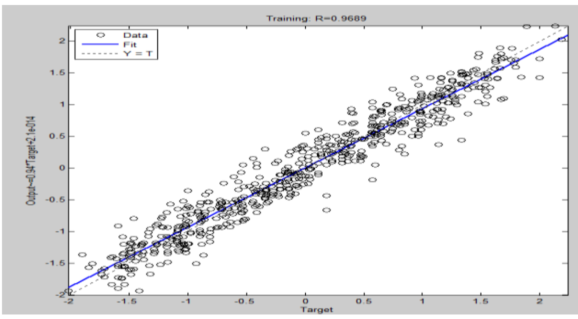

Fig.10: Regression Analysis.

V. Conclusion

An artificial neural network model is designed using Matlab®2010a for short-term load forecasting for weekend load. The implementation of model, training process and test results are all sucessful with a high degree of accuracy. The load data is obtained from a power station of India. The obtained results are compared with the actual data. Several networks are trained and a numbers of algoritms are proposed before arriving at the best Mean Absolute Percentange Error (MAPE) of 1.38%.

References

[1] S. Ramos, J. Soares, M. Silva, Z. Vale, Short-term Wind Forecasting for Energy Resources Scheduling, European Wind Energy Association (EWEA 2012), April 2012.
[2] W. Charytoniuk , E. Don Box , W. J. Lee , M. S. Chen , P. Kotas and P. V. Olinda Neural Network Based Demand Forecasting in a Deregulated Environment, IEEE Transaction Industry Application, 36 (3), 2000, 893 -898.
[3] K. Methaprayoon, L. Wei-Jen, S. Rasmiddatta, et al, Multistage Artificial Neural Network Short-Term Load Forecasting Engine with Front-End Weather Forecast, IEEE Transaction Industry Application., 43 (6), Nov. 2007, 1410-1416.
[4] M. Aydinalp-Koksal, A New Approach for Modeling of Residential Energy Consumption, VDM Verlag Dr. Muller Aktiengesselschaft & Co.Kg, 2008, 9-19.
[5] Shyh-Jier Huang, Kuang-Rong Shih, Short-TermLload Forecasting via ARMA Model Identification Including Non-Gaussian Process Considerations, IEEE Transactions on Power Systems, 18 (2), May 2003, 673-679.
[6] S. Ramos, J. Duarte, J. Soares, Z. Vale, F. J. Duarte, Typical Load Profiles in the Smart Grid Context - A Clustering Methods Comparison, IEEE Power and Energy Society General Meeting, San Diego CA, USA, July 2012, 22-26.
[7] George Gross, Francisco D. Galiana, Short-Term Load Forecasting, Proceeding of IEEE, 75 (12), Dec. 1987.
[8] Samsher Kadir Sheikh, M.G Unde, Short Term Load Forecasting using ANN Technique, International Journal of Emerging Sciences & Emerging Technologies, 1 (2), 2012, 91-107.
[9] S. Ruzic, A. Vuckovic, N. Nikolic, Weather Sensitive Method for Short-Term Load Forecasting in Electric Power Utility of Serbia, IEEE Transactions on Power Systems, 18, 2003, 1581 – 1586.
[10] Muhammad Buhari, Sanusi Sani Adamu, Short-Term Load Forecasting Using Artificial Neural Network, Proceedings of the International Multi Conference of Engineers and Computer scientist, 1, 2012.
[11] Ibreheem, Naresh Kumar Yadav, Implementation of FACTS Device for Enhancement of ATC Using PTDF, International Journal of Computer and Electrical Engineering, 3 (3), 2011.
[12] Jacek M. Zurada, Introduction to Neural Network, Jaico Publishing House, Delhi-110 002 (India), 25-235.
[13] B. Yegnanarryana, "Artificial Neural Network", Prentice-Hall of India Pvt. Ltd., New Delhi, 110 001,2006.
[14] P. Webros, "Generalization of Back propagation with Application of recurrent gas market model", Neural Network, 1, 1988 339-356.