Short Term Load Forecasting by using Wavelet Neural networks

Zidan Bashir El-Hawary, M.E.
Bashirza@is2.dal.ca Mohamed.El-Hawary@dal.ca
Department of Electrical & Computer Engeering
DalTeach(TUNS), Dalhousie University
Halifax, Nova Scotia, Canada B3J 2X4

Abstract :The application of the wavelet neural networks (WNNs) to short-term load forecasting is reported in this work. The wavelet neural network has much higher ability of generalization and fasts convergence for learning than a mulitlayer feedforward neural network. The Morlet wavelet has been chosen in this study as the activation function. The 3-layer backpropagation algorithm is used to train the network by learning the nonlinear relationship between input and output of the network. The input data consists of historical load and weather information, which are collected over a period of 2-years (1994-1995) to train the network and data of one year (1996) is used to test the network. The results of the network have been compared with artificial neural network and show an improved forecast with fast convergence.

1-Introduction

The quality of the short-term hourly load forecasts with lead times ranging from one hour to several days ahead has a significant impact on the efficiency of operation of any electrical utility. Since many operational decisions such as economic scheduling of the generating capacity, scheduling of fuel purchase, and system security assessment are based on such forecasts.

Many statistical methods such as multiple linear regression, stochastic time series, general exponential smoothing etc. have been used for short-term load forecasting [1]. Usually, these techniques are effective for the forecasting of normal days but fail to yield good results for those days with special events and because of their complexities, enormous computational efforts are required to produce acceptable results. For the past few years, artificial neural networks have received a great deal of attention and are now being proposed as powerful computational tools to solve the load forecasting problem [2,3,4].

Recently another technique has been used in the field of short-term load forecasting [5], which is a fruitful synthesis of ideas from neural networks and wavelet analysis called wavelet neural networks [6]. Since wavelets have shown their excellent performance in non stationary signal analysis and non linear function modeling, the neural network using wavelet basis functions may provide much higher availability of rates of convergence for approximation than the ordinary multi-layer neural network [7].

Weather components such as temperature, wind speed, cloud cover and relative humidity play a major rule in short-term load forecasting by changing daily loads. The characteristics of these components are reflected in the load requirements although some are affected more than others. In this work, temperature, wind speed, and cloud cover have been used as variables with historical load date to predict hourly load.

In this work, the back propagation algorithm is proposed as a methodology for short term electric load forecasting and utilizes the gradient descent method to obtain the optimal weight. The weight is progressively updated until the convergence criterion is satisfied. The impact of error on the pattern is expressed using the Least Mean Square (LMS) error.

$$E_p = \frac{1}{2}\sum_i (o_{pi} - t_{pi})^2 \tag{1}$$

Test results show a satisfactory use of the WNNs by increasing the convergence speed and the average percentage error was around 1.6 %.

2- Wavelet neural network

The basic idea of wavelet transform is to represent a signal function by a linear combination or an integration of the wavelet basis, which are determined by dilating and translating a signal basis function. The set basis function can be expressed as:

0-7803-5957-7/00/$10.00 © 2000 IEEE

$$\Psi_{s,t}(x)=s^{\frac{-1}{2}}\Psi\left(\frac{x-t}{s}\right)t,s\in R,t\neq 0 \quad (2)$$

Where $\Psi_{s,t}(x)$ is the wavelet, which is an element of the space $L^2(R)$. The function $f(x)$ in the space is represented by:

$$\int_{-\infty}^{+\infty}[f(x)]^2 dx \quad \pi \ \infty \quad (3)$$

The inversion of the wavelet transform can be approximated to:

$$f(x)=\sum_j\sum_k w_{j,k}\Psi_{s_j,t_k}(x) \quad (4)$$

The combination of the wavelets and the feedforward neural network gives a new type of neural network, namely the wavelet neural network. The wavelet neural network is constructed by means of replacing the ordinary basis function network (sigmoid, radial basis function and so on) with a multiscale wavelet function $\Psi_{s,t}(x)$ [7]. The connection weights are the corresponding wavelet coefficients.

2.1- Structure of the wavelet neural network

Figure 1 shows the wavelet network structure, where the wavelet neurons is responsible for pre-processing the input signal which is then passed to a multi-layer perceptron. Both the synoptic weights and wavelet parameters will be adjusted during training through back propagation of the output error.

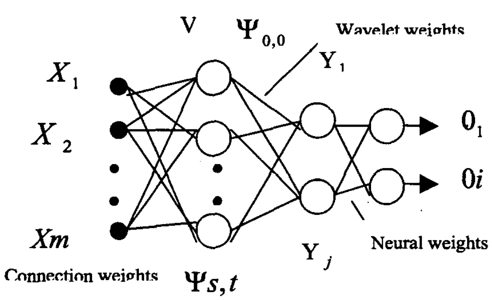

Figure 1. Structure of WNNs

$$V_{s,t}=\psi_{s,t}(\sum_{m=1}^{n}W_{(s,t),m}*X_m) \quad (5)$$

$$Y_j=g(\sum_{(s,t)}^{n}W_{j(s,t)}*V_{(s,t)}) \quad (6)$$

$$O_i=g(\sum_{j=1}^{n}W_{ij}*Y_j) \quad (7)$$

The selection of the mother wavelet is very important and depends on the particular application. There are a number of well-defined mother wavelets such as Morlet, Haar, Mexican Hat, and Meyer. Groups of them are called families, such as Daubechies, Biorthogonals, Coifets, and Symlets [8]. For this wavelet neural network, Morlet wavelet has been chosen to serve as an adaption basis function to the network's hidden layer, which has been the preferred choice in most work dealing with *WNNs*, due to its simple explicit expression.

$$\Psi_z(x)=\cos(5x_z)\exp(-\frac{x_z^2}{2}) \quad (8)$$

Where $x_z=(\frac{x-t_z}{s_z})$.

A neural learning algorithm modifies the wavelet network parameters, that is, the dilation and translation coefficients of every wavelet neuron, as well as the weights of the linear combination (network output). The approximation shown in Eq. 2 can be implemented by using a neural network architecture. The network parameters w_z, s_z, and t_z can be optimized by minimizing the Least-Mean-Square (LMS) energy E_p as in Eq. 1. Simplifying equations 5,6, and 7, the output of the wavelet neural network O_i will be represented by the following equation:

$$O_i=\sum_{j=i}^{n}\sum_{(s,t)}^{n}\sum_{(s,t)j}^{n}W_{(s,t)m}X_m*W_{(s,t)j}*Wij \quad (9)$$

Where Oi represents the i^{th} component of the output vector, and $W_{(s,t)m}$ in the first layer is the connection weight from input X_m to each wavelet neuron and $W_{(s,t)j}$ in the second layer is the corresponding weight (wavelet coefficient), and W_{ij} is the neural weights. The total error is given by:

$$E=\frac{1}{2}\sum_{j=i}^{n}\sum_{(s,t)}^{n}\sum_{(s,t)j}^{n}W_{(s,t)m}X_m*W_{(s,t)j}*Wij-t_i]^2 \quad (10)$$

Where ti denotes the target output value for i^{th} component of the output. To minimize E, The Gradient descent weight/bias learning function is used as the learning rule of the wavelet neural networks. The training process is the same training algorithm used for the normal neural network (Levenberge-Marquardit Backpropagation Algorithm) [9]. From equations (1), (2) and (4), we can obtain the negative gradient of $E_p^{'}$ using the partial derivatives of the cost function E with respect with each parameter of the WNN. Once the gradient vector is constructed from the partial derivatives, the WNN parameters can be updated by using the Gradient Descent Method.

3- Characteristics of input data

It was observed from the hourly load data of one year for Nova Scotia Power Inc. that load patterns for working days were similar while those for weekend were different from weekdays. Therefore in this study, loads and weather information for weekdays and weekends were forecasted separately. The input data for $WNNs$ are scaled such that they fall within the range (0,1), to avoid convergence problems during the training process. As shown in table 1, the network is trained with 127 neurons as input data, 24 neurons for the hidden layer, and 24 neurons as the output. There is no general rule that can be followed to determine the number of neurons in the hidden layer. In this study, the optimum number of the neurons in the hidden layer determined by trial and error (24,14 neurons for weekdays and weekend respectively). Holidays are treated as Saturday or Sunday depends on the markets are open or close during those holidays.

4- Training

The perception training algorithm is a form of supervised learning where the weights are adjusted to reduce errors whenever the network output does not match a known training target output. The training in this application was done, by using the TRAINLM Levenberg-Marquardt Backpropagation with actual data that has been recorded from the some period. TRAINLM is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization [9]. Initialization of weights, biases and the parameters of the $WNNs$ (t_z, s_z) were done by using a layer initialization function (INITNW) which initializes a layer's weights and biases according to the Nguyen-Widrow [9]. Testing of the network was done by using input data for some period that are independent from the training set and is not shown to the model during its training. It is observed that a choice of a value of 0.05 and 0.15 for η gives satisfactory $ANNs$ and $WNNs$ respectively for the given example in the training process. The increase in the number of training iterations (Epoch) may not necessarily reduce the error. It is therefore important to choose the value of η that can give an acceptable convergence speed. The number of epochs are used for training $ANNs$ were 2500 epochs to meet the performance goal when the maximum error goal (Maximum performance gradient) was 0.0003. While the number of epoches for $WNNs$ were 750 epoches for the same maximum error goal.

5- Simulation results

The network model is written in the MATLAB. The numerical simulations were run on a PC (Pentium, 160 MHz, 128 MB). The training time for the network was relatively short and done by using fast trained Levenberg-Marquardi Algorithm. Forecasting was performed with $WNNs$ using modified network architecture described in table 1. The results of a simulation of some arbitrary daily load are shown in figure 2. Where the actual daily load curve is compared with forecast daily load by using both $WNNs$ and $ANNs$. The average percentage error (APE) is shown in table 2.

$$APE = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}[\frac{A_i - P_i}{A_i}](i,j)}{n*24} \quad (11)$$

Table 1.Modified network architectures for WNN and ANN

Layer	Data Type		No. of neurons
Input	Day of the week	Binary	5
Input	Saturday	Binary	1
Input	Sunday	Binary	1
Input	Load of forecast day	Con.	24
Input	Load of 1 past day	Con.	24
Input	Load of 2 past day	Con.	24
Input	T. of forecast day	Con.	24
Input	W. S. of forecast day	Con.	24
Hidden	N/A	N/A	24, 14
Output	Load	Con.	24

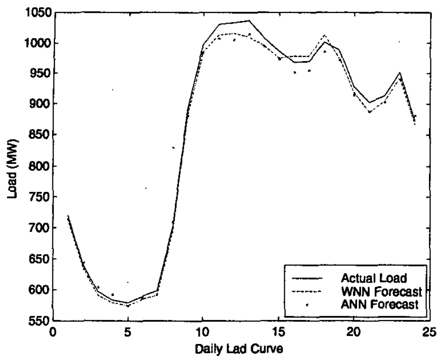

Figure 2.Forecast daily load for June 3, 1996

Table 2 Percentage error for WNN and ANN

	Average percentage error %	
Hour	ANNs	WNNs
0	1.13	0.97
1	-0.84	0.72
2	-1.2	0.8
3	-1.42	0.7
4	0.98	0.74
5	0.72	0.89
6	1.1	1.16
7	-1.21	1.05
8	1.07	1.03
9	1.39	1.22
10	2.23	1.74
11	2.72	1.61
12	2.05	2.46
13	1.28	1.26
14	1.39	1.27
15	1.62	-1.16
16	1.46	-1.3
17	1.54	-1.25
18	1.51	1.31
19	1.46	1.03
20	1.61	1.78
21	1.2	1.03
22	1.01	0.92
23	-0.95	0.63

6- Conclusion

This work investigates the ability of neural and wavelet neural networks by using the backpropagation algorithm for solving short-term load forecasting. In this study, it was found that selecting a proper network structure and input data to the network are major factors that affect the performance accuracy of the network. It shows also that using weather information data can positively affect the training and testing results. The CPU time required for testing is below ten seconds for both WNNs and ANNs. However, By comparison with neural network model, WNN shows more flexibility by updating its parameters at the same time. Hence WNNs takes considerably less CPU time for training (Less epoches).

Reference

[1]-Ibrahim Moghram and Saifur Rahman, "Analysis and Evaluation of Five Short-Term Load Forecasting Techniques", Proceedings of the IEEE Transaction on Power Systems, pp. 1484-1491, Vol. 4, No. 4, October 1989

[2]-T.W.S. Chow, C.T. Leung " Neural Networks Based Short Term Load Forecasting using Weather Compensation", Proceedings of the IEEE, pp. 1736-1742, 1996

[3]-Kun-long Ho, Yuan Yih, Chien-Cluen Yang, " Short Term Load Forecasting using a Multi Layer Neural Network with An adaptive Learning Algorithm", Proceedings of the IEEE, pp. 141-149, 1992

[4]-B.S. Kermanshahi, C.H.Poskar, G. Swift, W. Buhr and A. Silik"Artificial Neural Network for Forecasting Daily Loads of A Canadian Electric Utility", Proceedings of the IEEE, pp. 302-307, 1993

[5]- Anant Oonsivilai and M.E. El-Hawary, "Wavelet Neural Network Based Short Term Load Forecasting of Electric Power System Commercial Load", proceeding of the IEEE Canadian Conference on Electrical and Computer Engineering, pp. 1223-1228, May 1999

[6]-Qinghua Zhang, Albert Benveniste "Wavelet Networks", Proceeding of the IEEE, pp. 889-898, 1992

[7]- Yao, C.J. Wei and Z Y. He, "Evolving Wavelet Neural Networks for Function Approximation", Proceeding of IEEE Electronic Letters, pp. 360-361 Vol. 32 No. 4, Feb 1996

[8]- Misiti M, Misiti Y, Oppenheim G. Poggi JM, "Wavelet Toolbox User's Guide", The Math Works Inc, 1996

[9]- Howard Demuth and Mark Beal, Neural Network Toolbox User's Guide, The Math Works Inc.1998