Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 - August 4, 2005

Genetic Algorithm-Based Variable Translation Wavelet Neural Network and its Application

S.H. Ling
Centre of Multimedia Signal Processing,
Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

F.H.F. Leung
Centre of Multimedia Signal Processing,
Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract - A variable translation wavelet neural network (VTWNN) trained by genetic algorithm is presented in this paper. In the proposed wavelet neural network, the translation parameters are variables depending on the network inputs. Thanks to the variable translation parameter, the network becomes an adaptive one, providing better performance and increased learning ability than conventional wavelet neural networks. Genetic algorithm is applied to train the parameters of the proposed wavelet neural network. An application example on short-term daily electric load forecasting in Hong Kong is presented to show the merits of the proposed network.

I. INTRODUCTION

Recently, a new kind of neural networks known as the wavelet neural networks (WNNs) have been proposed [1-4], which combine feed-forward neural network with the wavelet theory [5-6]. It can provide better performance in function learning than conventional feed forward neural networks. Wavelets provide a multi-resolution approximation of discriminate functions. Researchers have successfully applied wavelet in function approximation [1], load forecasting [2], and pattern classification [3]. A typical wavelet neural network structure allows a fixed set of network parameters of which the values are learned. The network should be trained to model the input-output relationship of a given data set. However, the number of the fixed set of network parameters may not be enough for data sets that are distributed in a vast domain.

One of the important issues on neural networks is the training of the networks. The training process aims to find a set of optimal network parameters. One commonly used training method is the gradient method. However, it may only converge to a local minimum, and is sensitive to the values of the initial parameters. The function to be optimized needs to be differentiable, and the learning method may only be good for some specific network structure. Genetic algorithm (GA) [7,13] is a global search algorithm. The error functions are less likely to be trapped in a local optimum, and need not be differentiable or even continuous. Thus, GA is more suitable for searching in a large, complex, non-differentiable and multimodal domain. The same GA can be used to train many different networks, regardless of whether they are of feed-forward [14], recurrent [14], wavelet [1] or any other structure type. This generally saves a lot of efforts in developing the training algorithms for different types of networks.

An accurate load forecasting in power systems is important. Correct forecasts have significant influences to system operations such as unit commitment, load dispatch and maintenance scheduling. The forecasting information can be used to aid optimal generation scheduling, saving valuable fuel costs. The short-term load forecasting problem is non-linear, random, and time-varying. Yet, artificial neural networks have been promising and effective tools for tackling the problem [10-12]. In this paper, an industrial application on short-term load forecasting in Hong Kong is presented. The actual load data are obtained from the local utility company, the CLP Power Hong Kong Limited. By employing a proposed variable translation wavelet neural network (VTWNN) on short-term load forecasting, we can model different input load patterns with variable network parameters, which give better learning and testing results than conventional WNNs.

In the proposed VTWNN, wavelets are used as the transfer functions in the hidden layer. The network parameters, i.e. the translation parameters of the wavelets in the hidden layer, are variable depending on the network inputs. Thanks to the variable translation parameters, the proposed VTWNN has the ability to model the input-output function with variable network parameters. It works as if several individual neural networks are handling different groups of input data set. Effectively, it becomes an adaptive network capable of handling different input patterns, which exhibits a better performance. All network parameters are trained by real-coded GA with unimodal normal distribution crossover (UNDX) [8] and non-uniform mutation [7].

This paper is organized as follows. The basic theory of wavelet is discussed in Section II. In Section III, the proposed VTWNN is presented. In Section IV, the training of the parameters of the proposed network using GA will be presented. In Section V, the application example on short-term load forecasting in Hong Kong will be given to show the merits of the proposed network. A conclusion will be drawn in Section VI.

II. WAVELET THEORY

Certain seismic signals can be modelled by combining translations and dilations of an oscillatory function with finite duration called a "wavelet".

A continuous function $\psi(x)$ is a "mother wavelet" or "wavelet" if it satisfied the following properties:

Property 1:

$$\int_{-\infty}^{+\infty}\psi(x)dx = 0 \quad (1)$$

In other words, the total positive energy of $\psi(x)$ is equal to the total negative energy of $\psi(x)$.

Property 2:

$$\int_{-\infty}^{+\infty}|\psi(x)|^2 dx < \infty \quad (2)$$

where most of the energy in $\psi(x)$ is confined to a finite domain and is bounded. In order to control the magnitude and the position of $\psi(x)$, $\psi_{a,b}(x)$ is defined as:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \quad (3)$$

where a is the dilation parameter and b is the translation parameter. It should be noted that $\psi_{a,b}(x)$ is scaled down as the dilation parameter a increases, and the location of the centre of the wavelet is controlled by the translation parameter b.

III. GENETIC ALGORITM BASED VARIABLE TRANSLATION WAVELET NEURAL NETWORK

In this section, the GA-based variable translation wavelet neural network (VTWNN) will be presented. The wavelet neural network (WNN) can be considered as a particular case of the feed-forward neural network model. The difference is that the hidden layer of a WNN contains transfer functions of multi-scaled wavelet functions $\psi_{a,b}(x)$. In the proposed VTWNN, the translation parameter in the transfer function of the hidden node is variable and depends on the network inputs. With the variable translation parameters, the proposed VTWNN performs better and has higher learning ability than the conventional WNNs [1]. The tuning of the network parameters is done by GA.

The proposed VTWNN has a three-layer structure with n_{in} nodes in the input layer, n_h nodes in hidden layer, and n_{out} nodes in output layer as shown in Fig. 1. The input of the hidden layer, S_j, is given by,

$$S_j = \sum_{i=1}^{n_{in}} z_i v_{ji}, \quad j = 1, 2, \ldots, n_h \quad (4)$$

where z_i, i = 1, 2, ..., n_{in} are the input variables; v_{ji} denotes the weight of the link between the i-th input and the j-th hidden node. In order to control the magnitude and the position of the wavelet, the multi-scaled wavelet function $\psi_{a,b}(x)$ defined in (3) is used.

The dilation parameter a of the first hidden node (j=1) is set at 1, i.e. $\psi_{1,b}(x) = \psi(x-b)$. For the second hidden node (j=2), the dilation parameter a is set at 2, i.e. $\psi_{2,b}(x) = \frac{1}{\sqrt{2}}\psi\left(\frac{x-b}{2}\right)$, where the output of wavelet is scaled down by $1/\sqrt{2}$. Similarly, for the j-th hidden node,

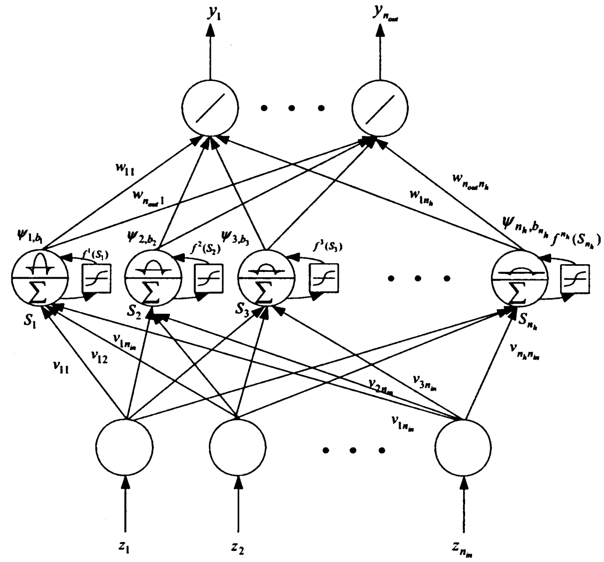

Fig.1. Proposed wavelet neural network structure

the dilation parameter a is set at j. Hence, the output of the hidden layer of the proposed VTWNN is given by,

$$\psi_{j,b_j}(S_j) = \frac{1}{\sqrt{j}}\psi\left(\frac{S_j - b_j}{j}\right) \quad (5)$$

In this proposed network, the function of Maxican Hat [9] as shown in Fig. 2 is selected as the mother wavelet $\psi(x)$, which is defined as:

$$\psi(x) = e^{-x^2/2}(1-x^2) \quad (6)$$

$\psi(x)$ meets the Property 1 in (1) and Property 2 in (2) of wavelet. Referring to (5) and (6),

$$\psi_{j,b_j}(S_j) = \frac{1}{\sqrt{j}} e^{\frac{-\left(\frac{S_j-b_j}{j}\right)^2}{2}} \left(1-\left(\frac{S_j-b_j}{j}\right)^2\right) \quad (7)$$

In this proposed network, the translation parameter b_j is variable depending on the inputs S_j, and is governed by a nonlinear function $f^j(\cdot)$,

$$b_j = f^j(S_j) \quad (8)$$

$$f^j(S_j) = 4*\left(\frac{2}{1+e^{-\kappa_j \times S_j}} - 1\right) \quad (9)$$

where κ_j is a tuned parameter which is used to control the shape of the nonlinear function $f^j(\cdot)$. In Fig. 3, the effect of the tuned parameter κ_j to b_j is shown. We see as $\kappa_j \to \infty$, the function reduces to a threshold function.

The output of the proposed VTWNN is defined as,

$$y_l = \sum_{j=1}^{n_h} \psi_{j,b_j}(S_j) \cdot w_{lj} \quad (10)$$

$$= \sum_{j=1}^{n_h} \psi_{j,b_j}\left(\sum_{i=1}^{n_{in}} z_i v_{ji}\right) \cdot w_{lj} \quad (11)$$

where w_{lj}, $j = 1, 2, \ldots, n_h$; $l = 1, 2, \ldots, n_{out}$ denotes the weight of the link between j-th hidden node and l-th output node. All the parameters of the network are tuned by the GA [7].

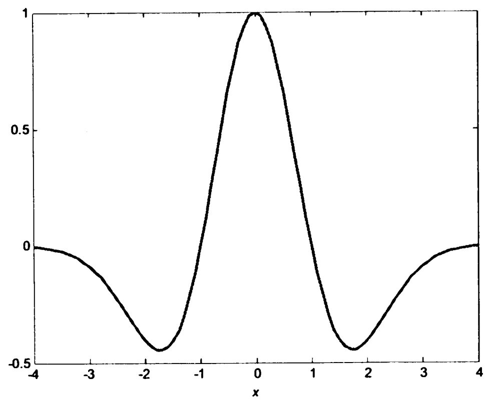

Fig.2. Maxican Hat mother wavelet

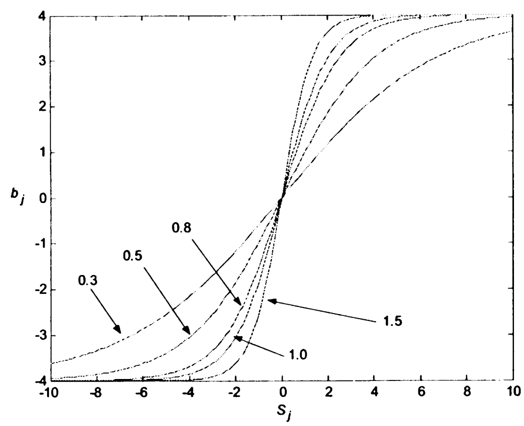

Fig. 3. Sample nonlinear functions with different values of parameter κ_j ($\kappa_j = 0.3$, $\kappa_j = 0.5$, $\kappa_j = 0.8$, $\kappa_j = 1.0$ and $\kappa_j = 1.5$)

IV. Training of Network Parameters

The proposed neural network is employed to learn the input-output relationship of an application using GA with unimodal normal distribution crossover (UNDX) [8] and non-uniform mutation [7]. More details about these two genetic operations are discussed in Appendix. A population of chromosomes P is initialized and then evolves. First, two parents are selected from P by the method of spinning the roulette wheel [7]. Then a new offspring is generated from these parents using the crossover and mutation operations, which are governed by the probabilities of crossover and mutation respectively. These probabilities are chosen by trial and error through experiments for good performance. The new population thus generated replaces the current population. The above procedures are repeated until a certain termination condition is satisfied, e.g. a predefined number of generations has been reached. The input-output relationship can be described by,

$$\mathbf{y}^d(t) = \mathbf{g}\left(\mathbf{z}^d(t)\right), t = 1, 2, \ldots, n_d, \quad (12)$$

where $\mathbf{z}^d(t) = \left[z_1^d(t) \quad z_2^d(t) \quad \cdots \quad z_{n_{in}}^d(t)\right]$ and $\mathbf{y}^d(t) = \left[y_1^d(t) \quad y_2^d(t) \quad \cdots \quad y_{n_{out}}^d(t)\right]$ are the given inputs and the desired outputs of an unknown nonlinear function $\mathbf{g}(\cdot)$ respectively; n_d denotes the number of input-output data pairs. The fitness function is defined as,

$$fitness = \frac{1}{1 + err}, \quad (13)$$

$$err = \frac{\sum_{t=1}^{n_d}\sum_{l=1}^{n_{out}} \frac{\left|y_l^d(t) - y_l(t)\right|}{y_l^d(t)}}{n_d n_{out}}. \quad (14)$$

The objective is to maximize the fitness value of (13) (minimize *err*) using the GA by setting the chromosome to be $\left[v_{ji} \quad w_{lj} \quad \kappa_j\right]$ for all i, j, l, and v_{ji}, $w_{lj} \in \left[-1.5, 1.5\right]$, $\kappa_j \in \left[0.3, 2\right]$. The fitness value of (13) $\in [0, 1]$.

V. Application Example

We consider the short-term load forecasting (STLF) for the power supply system in Hong Kong. STLF is important to power system, because it plays a role in the formulation of economic, reliable, and secure operating strategies for the power system. The objectives of STLF is i) to derive the scheduling functions that determine the most economic load dispatch with operational constraints and policies, environmental and equipment limitations; ii) to assess the security of the power system at any time point; iii) to provide system dispatchers with timely information.

The proposed VSNN is applied to do STLF. The application of neural networks to STLF has been explored extensively in the literature [10-12]. The idea is to construct seven multi-input multi-output neural networks, one for each day of a week. Each neural network has 24 outputs representing the expected hourly load for a day. A diagram of one of the seven neural networks for the load forecasting is shown in Fig. 4. The network has 28 inputs and 24 outputs. Among the 28 inputs nodes, the first 24 nodes represent the previous 24 hourly loads [12] and are denoted by $z_i = L_i^d(t-1)$, $i = 1, 2, \ldots, 24$. Node 25 (z_{25}) and node 26 (z_{26}) represent the average temperatures of the previous day ($T(t-1)$) and the forecasted averaged temperatures of the present day ($T(t)$) respectively. Node 27 (z_{27}) and node 28 (z_{28}) represent the average relative humidity at the previous day ($RH(t-1)$) and the forecasted average relative humidity at the present day ($RH(t)$) respectively. The output layer consists of 24 output nodes that represent the forecasted 24 hourly loads of a day, and are denoted by $y_k(t) = L_i(t)$, $i = 1, 2, \ldots, 24$. Such a network structure is chosen based on the assumption that the consumption patterns of the seven days within a week

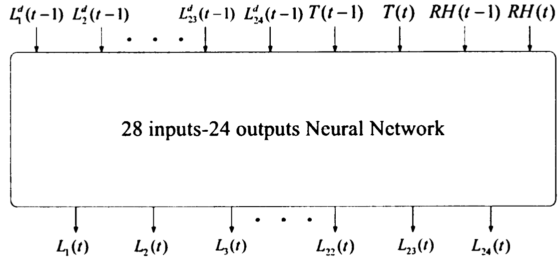

Fig.4. Proposed neural network based short-term load forecaster

would differ significantly among each other, while the patterns among the same day of weeks are similar. By using the past 24 hourly loads as the inputs, the relationship between a given hour's load and the 24 hourly loads of the pervious day can be considered. Temperature information (Node 25 and Node 26) is important inputs to the STLF. For any given day, the deviation of the temperature variable from a normal value may cause such significant load changes as to require major modifications in the unit commitment pattern. Humidity is similar to temperature on affecting the system load, especially in hot and humid areas.

In this paper, we use a data set in year 2000 provided by CLP Power Hong Kong Ltd to illustrate the proposed VTWNN on doing STLF. The proposed VTWNN was trained using half year (22 weeks) load data for every Thursday or Sunday from Mar. 23 to Aug. 20, 2000. The load pattern for every Thursday and Sunday from Mar. 23 to Aug. 20, 2000 are shown in Fig. 5 and Fig. 6 respectively. In these two figures, we can see that the shape of every load pattern is similar but the power consumption is much different. Conventional feed-forward neural networks are only good at minimizing the average error of the system. It is difficult to model all the load patterns accurately.

In the VTWNN, each input set will be individually handled by its corresponding network parameter set. It should be able to handle the load forecasting problem better. Referring to (13), the proposed VTWNN used for doing STLF is governed by,

$$y_l = \sum_{j=1}^{n_h} \psi_{j,b_j}\left(\sum_{i=1}^{28} z_i v_{ji}\right) \cdot w_{lj},\ l = 1, 2, \ldots 23, 24. \qquad (15)$$

GA is used to tune the parameters of the proposed VTWNN of (17). The fitness function is given by,

$$fitness = \frac{1}{1+err}$$

$$\text{where } err = \frac{\sum_{t=1}^{22}\sum_{l=1}^{24} \frac{\left|y_l^d(t) - y_l(t)\right|}{y_l^d(t)}}{22 \times 24} \qquad (16)$$

The value of *err* indicates the mean absolute percentage error (MAPE) of the load forecasting system. For comparison, a wavelet neural network (WNN) [1] is also applied to do the same job. All parameters of the two networks are trained by real coded GA with unimodal normal distribution crossover (UNDX) [8] and non-uniform mutation [7]. For all cases, the initial values of the parameters of the neural networks are randomly gene-

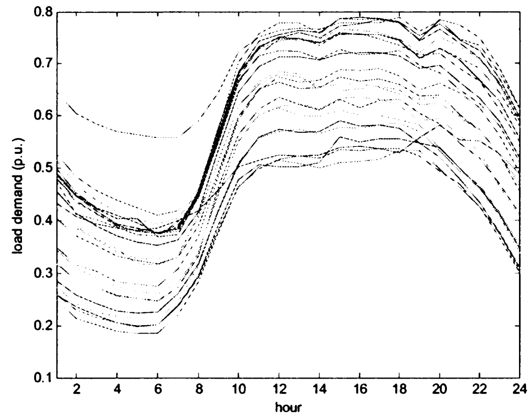

Fig.5. Load patterns for every Thursday from Mar. 23 to Aug. 20, 2000.

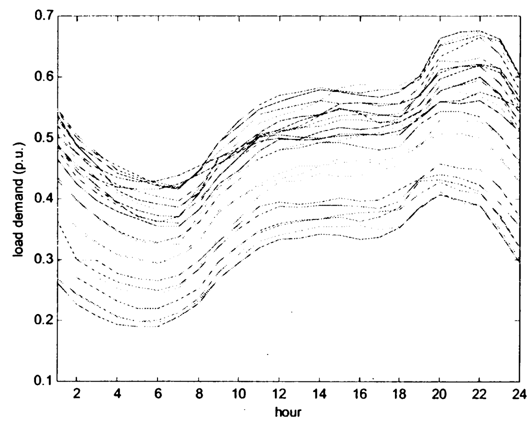

Fig.6. Load patterns for every Sunday from Mar. 23 to Aug. 20, 2000.

rated. The number of iteration to train the neural networks is 10000. For the GA, the probability of crossover (p_c), the probability of mutation (p_m), and the shape parameter of the mutation operation are set at 0.8, 0.01, and 2 respectively for both the proposed VTWNN and WNN [1]. The population size is 100. All the results are averaged ones out of 10 runs. The training and forecasting results are tabulated in Table I and Table II, which show the simulation results for different numbers of hidden node (n_h). The forecasting results for Aug. 24 (Thursday) and Aug. 27 (Sunday) are shown in these tables. From these two tables, it can be seen that the proposed VTSNN performs better than the WNN [1] in terms of the average training and forecasting error. The best average training and forecasting error are achieved when the number of hidden node (n_h) is set at 20 for both approaches for Thursday and Sunday. With the proposed VTWNN, the average training error are 3.1266% for Thursday and 3.0213% for Sunday, which implies 12.1% and 16.7% improvements over the conventional WNN with the same number of hidden nodes. The best forecasting errors for Aug. 24 (Thursday) and Aug. 27 (Sunday) are 2.0985% and 1.8662%, which implies 16.2% and 16% improvements. Fig. 7 and Fig. 8 show the results of the load forecasting on Aug. 24 (Thursday) and Aug. 27 (Sunday). In this figure, the dashed line represents the best forecasted result using the proposed network, and the dotted line is the best forecasted result using the conventional network. The actual load is represented by the solid line. We can see that the forecasting results using the proposed neural network are better.

VI. CONCLUSION

A GA based variable translation wavelet neural network has been presented in this paper. All network parameters are tuned by GA. Thanks to the variable translation parameters in the network, the proposed VTWNN can have a higher learning ability. The application on short-term load forecasting in Hong Kong using the proposed neural networks has been discussed. Experimental results have been given to show the merits of the proposed network.

ACKNOWLEDGMENT

The work described in this paper was substantially supported by a grant from the Hong Kong Polytechnic University (Project No. G-YX31). The data in the application example are offered by CLP Power Hong Kong Limited for illustration purpose only.

TABLE I. RESULTS OF THE PROPOSED VTWNN AND THE CONVENTIONAL WNN FOR THURSDAY.

n_h	VTWNN		WNN [1]	
	Average training error (MAPE)	Average forecasting error (MAPE)	Average training error (MAPE)	Average forecasting error (MAPE)
16	3.2898%	2.4490%	3.66795	2.8088%
18	3.2237%	2.3511%	3.5823%	2.7154%
20	**3.1266%**	**2.2988%**	3.5034%	2.6033%

n_h	VTWNN		WNN [1]	
	Best Training error (MAPE)	Best Forecasting error (MAPE)	Best Training error (MAPE)	Best Forecasting error (MAPE)
16	3.0502%	2.1199%	3.3912%	2.4891%
18	3.0478%	2.1097%	3.3779%	2.4801%
20	**3.0211%**	**2.0985%**	3.3238%	2.4395%

TABLE II. RESULTS OF THE PROPOSED VTWNN AND THE CONVENTIONAL WNN FOR SUNDAY.

n_h	VTWNN		WNN [1]	
	Average training error (MAPE)	Average forecasting error (MAPE)	Average training error (MAPE)	Average forecasting error (MAPE)
16	3.1932%	2.1848%	3.6078%	2.3889%
18	3.0881%	2.0743%	3.5517%	2.3438%
20	**3.0213%**	**1.9952%**	3.5266%	2.2887%

n_h	VTWNN		WNN [1]	
	Best Training error (MAPE)	Best Forecasting error (MAPE)	Best Training error (MAPE)	Best Forecasting error (MAPE)
16	3.0691%	1.9947%	3.4391%	2.2130%
18	2.9900%	1.9334%	3.4157%	2.1881%
20	**2.9303%**	**1.8662%**	3.4121%	2.1655%

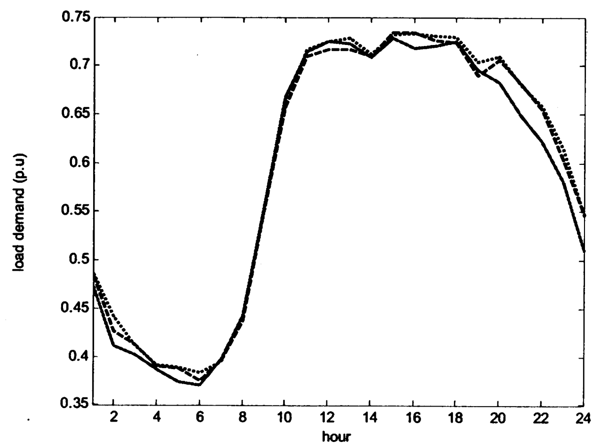

Fig. 7. Best load forecasting result for Aug. 24, 2000 (Thursday) using the proposed VTWNN (dashed line) and the conventional WNN (dotted line), as compared with the actual load (solid line)

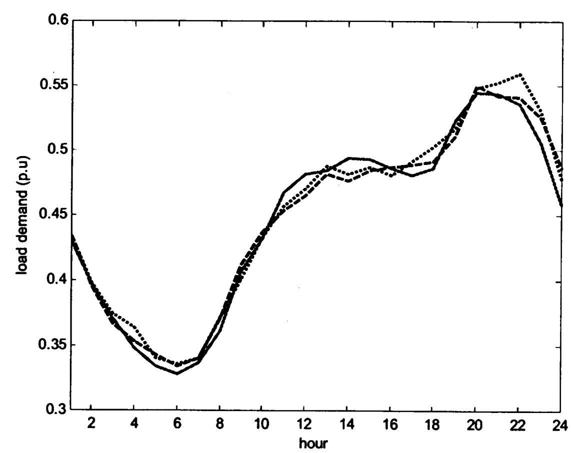

Fig. 8. Best load forecasting result for Aug. 27, 2000 (Sunday) using the proposed VTWNN (dashed line) and the conventional WNN (dotted line), as compared with the actual load (solid line)

REFERENCES

[1] S. Yao, C.J. Wei, and Z.Y. He, "Evolving wavelet neural networks for function approximation," *Electron. Lett.*, vol.32, no.4, pp. 360-361, Feb. 1996.

[2] O. Anant and M.E. El-Hawary, "Wavelet neural network based short term load forecasting of electric power system commercial load," in *Proc. 1999 IEEE Canadian Conf. Electrical and Computer*

Engineering, Alberta, Canada, May 9-12 1999, pp. 1223-1228.

[3] H. Szu, "Neural network adaptive wavelets for signal representation and classification," *Optical Engineering*, vol. 32, no. 9, pp. 1907-1916, 1992.

[4] Y.C. Huang and C.M. Huang, "Evolving wavelet networks for power transformer condition monitoring," *IEEE Trans. Power Delivery*, vol. 17, no. 2, pp. 412-416, Apr. 2002.

[5] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Information Theory*, vol. 36, no.5, pp. 961-1005, Sep. 1990.

[6] S.G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no.7, pp. 674-693, Jul. 1989.

[7] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*, 2nd extended ed. Springer-Verlag, 1994.

[8] I. Ono and S. Kobayashi, "A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover," in *Proc. 7th ICGA*, 1997, pp. 246-253.

[9] I. Daubechies, *Ten lectures on wavelets*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.

[10] I. Drezga and S. Rahman, "Short-term load forecasting with local ANN predictors," *IEEE Trans. Power System*, vol. 14, no. 3, pp. 844-850, Aug. 1999.

[11] A.G. Bakirtzls, V. Petridls, S.J. Klartzis, M.C. Alexladls, and A.H. Malssls, "A neural network short term load forecasting model for Greed power system," *IEEE Trans. Power Systems*, vol. 11, no. 2, pp. 858-863, May 1996.

[12] J.A. Momoh, Y. Wang, and M. Elfayoumy, "Artificial neural network based load forecasting," in *Proc. IEEE Int. Conf. System, Man, and Cybernetics: Computational Cybernetics and Simulation*, vol. 4, 1997, pp. 3443-3451.

[13] M. Srinivas and L.M. Patnaik, "Genetic algorithms: a survey," *IEEE Computer*, vol. 27, issue 6, pp. 17-26, June 1994.

[14] F.M. Ham and I. Kostanic, *Principles of Neurocomputing for Science & Engineering*. McGraw Hill, 2001.

APPENDIX

A. Unimodal normal distribution crossover (UNDX)

Unimodal normal distribution crossover is defined as a mixture of three selected parents $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$. The resulting offspring $\mathbf{o}_{\mathbf{s}_c}$ is defined as,

$$\mathbf{o}_{\mathbf{s}_c^1} = \begin{bmatrix} o_{s_1^1} & o_{s_2^1} & \cdots & o_{s_{no\_vars}^1} \end{bmatrix} = \mathbf{m} + z_1\mathbf{e}_1 + \sum_{i=2}^{no\_vars} z_i\mathbf{e}_i, \text{(A.1)}$$

$$\mathbf{o}_{\mathbf{s}_c^2} = \begin{bmatrix} o_{s_1^1} & o_{s_2^1} & \cdots & o_{s_{no\_vars}^1} \end{bmatrix} = \mathbf{m} - z_1\mathbf{e}_1 - \sum_{i=2}^{no\_vars} z_i\mathbf{e}_i, \text{(A.2)}$$

where

$$\mathbf{m} = \frac{(\mathbf{p}_1 + \mathbf{p}_2)}{2}, \quad \text{(A.3)}$$

$$z_1 = N(0,\sigma_1^2), \quad z_i = N(0,\sigma_2^2), \quad \text{(A.4)}$$

$$\sigma_1 = \beta d_1, \quad \sigma_2 = \frac{\mu d_2}{\sqrt{\text{no\_vars}}}, \quad \text{(A.5)}$$

$$\mathbf{e}_1 = \frac{(\mathbf{p}_2 - \mathbf{p}_1)}{|\mathbf{p}_2 - \mathbf{p}_1|}, \quad \text{(A.6)}$$

$$\mathbf{e}_m \perp \mathbf{e}_n \quad (m \neq n), \quad m,n = 1,\ldots, no\_vars, \quad \text{(A.7)}$$

where no_vars is the number of genes, $N(\cdot)$ is a normal distributed random number, d_1 is the distance between the parents $\mathbf{p}_1$ and $\mathbf{p}_2$, d_2 is the distance of $\mathbf{p}_3$ from the line connecting $\mathbf{p}_1$ and $\mathbf{p}_2$, β and μ are constants.

B. Non-uniform mutation (NUM)

Non-uniform mutation is an operation with a fine-tuning capability. Its action depends on the generation number of the population. The operation takes place as follows. If $\mathbf{o}_\mathbf{s} = \begin{bmatrix} o_{s_1}, & o_{s_2}, & \ldots, & o_{s_{no\_vars}} \end{bmatrix}$ is a chromosome and the element o_{s_k} is randomly selected for mutation (the value of o_{s_k} is inside [$para_{\min}^k, para_{\max}^k$]), the resulting chromosome is then given by $\hat{\mathbf{o}}_\mathbf{s} = \begin{bmatrix} o_{s_1}, & \cdots & ,\hat{o}_{s_k}, & \cdots & ,o_{s_{no\_vars}} \end{bmatrix}$, $k \in 1, 2, \ldots no\_vars$, and

$$\hat{o}_{s_k} = \begin{cases} o_{s_k} + \Delta\left(\tau, para_{\max}^k - o_{s_k}\right) \text{ if } r_d = 0 \\ o_{s_k} - \Delta\left(\tau, o_{s_k} - para_{\min}^k\right) \text{ if } r_d = 1 \end{cases}, \quad \text{(A.8)}$$

where r_d is a random number equal to 0 or 1 only. The function $\Delta(\tau, y)$ returns a value in the range $[0, y]$ such that $\Delta(\tau, y)$ approaches 0 as τ increases. The function is defined as follows,

$$\Delta(\tau, y) = y\left(1 - r^{\left(1-\frac{\tau}{T}\right)^b}\right), \quad \text{(A.9)}$$

where r is a random number in [0, 1], τ is the present generation number of the population, T is the maximum generation number of the population, and b is a system parameter that determines the degree of non-uniformity.