Statistika, Vol. 8 No. 2, 111 – 118
Nopember 2008

An Artificial Neural Networks Forecasting for Malaysia's Load

NORIZAN MOHAMED[1)], MAIZAH HURA AHMAD[2)], ZUHAIMY ISMAIL[2)], AND KHAIRIL ANUAR ARSHAD[2)]

1) Mathematics Department, Faculty Of Science and Technology, Universiti Malaysia Terengganu (UMT), 21030 Kuala Terengganu, Terengganu, Malaysia institution

2) Department of Mathematics, Faculty Of Science, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia

Email: norizan@umt.edu.my, maizah@mel.fs.utm.my, zuhaimyi@yahoo.com, khairil@mel.fs.utm.my

ABSTRACT

In this paper, two artificial neural networks models, namely the multilayer feedforward neural network and the recurrent neural network are applied for Malaysia's load forecasting. A half hourly load data is divided equally into three distinct sets for training, validation and testing. Backpropagation is selected as the learning algorithm whereas the transfer function for both hidden layer and output layer is sigmoid the function. The forecasting performances were compared between these two models. The results show that, the sum squared error (SSE) of multilayer feedforward neural network were the lowest hence the multilayer feedforward neural network is a better model for a half hourly Malaysia's load.

Key words: *load forecasting, artificial neural networks, multilayer feedforward feural network, recurrent neural network.*

1. Introduction

Neural networks or popularly known as Artificial Neural Network (ANN) are biologically inspired forms of distributed computation. ANN are composed of many nodes that operate in parallel and communicate with each other through connecting elements. An element is a simple structure that performs three basic functions: input, processing and output. ANN can be organized into several different connection topologies and learning algorithms. The number of inputs to the network is constrained by the problem type, whereas the number of neurons in the output layer is constrained by the number of outputs required by the problem type. However, the number of hidden layers and the sizes of the layers are decided by the designer.

ANN have been extensively studied and received increasing attention in time series forecasting. The greatest advantage of a neural network is its ability to model a nonlinear process without priori assumptions of the nature of the process (BuHamra, Smaoui & Gabr, 2003; Khoa, Sakakibara & Nishikawa, 2006). Appropriate neural network architectures can be trained to predict the future values of the dependent variables (Ho, Xie & Goh, 2002). If the network paradigm and parameters are appropriately designed, these can result in satisfactory performance (Kohzadi, Boyd, Kermanshahi & Kaastra, 1996).

One the best known neural models is the multilayer feedforward neural network (MFNN) which consists of an input layer, one or several hidden layers and an output layer. The other neural model is recurrent neural network (RNN) which can learn sequences as time evolves and responds to the same input pattern differently at different time, depending on the previous input patterns as well. Both neural network models represent the performance for nonlinear prediction of time series (Connor, Martin & Atlas, 1994; Connor & Atlas, 1991). Sometimes recurrent neural network (RNN) lead to improved forecasting performance as compared to MFNN (Ho, Xie, & Goh, 2002; Connor & Atlas, 1991).

In feed-forward network, the processing elements are connected in such a way that the signals flow in one direction from input to output, where as in recurrent networks there are both feed-forward and feedback connections along which signal can propagate in opposite direction.(Pham & Liu, 1995)

The remainder of this paper is organized as follows. In Section 2, the data set, the multi-layer feed-forward neural network, the recurrent neural network and forecasting performance are described. In Section 3, details of the results are discussed. Section 4 provides concluding remarks.

2. Methodology

2.1. The Data Set

The data used in this paper is gathered from Tenaga Nasional Berhad (TNB), Malaysia. It represented a half hourly daily demand ("load") measured in Megawatts (MW) from September 01, 2005 to December 31, 2005 as illustrated in Figure 1.

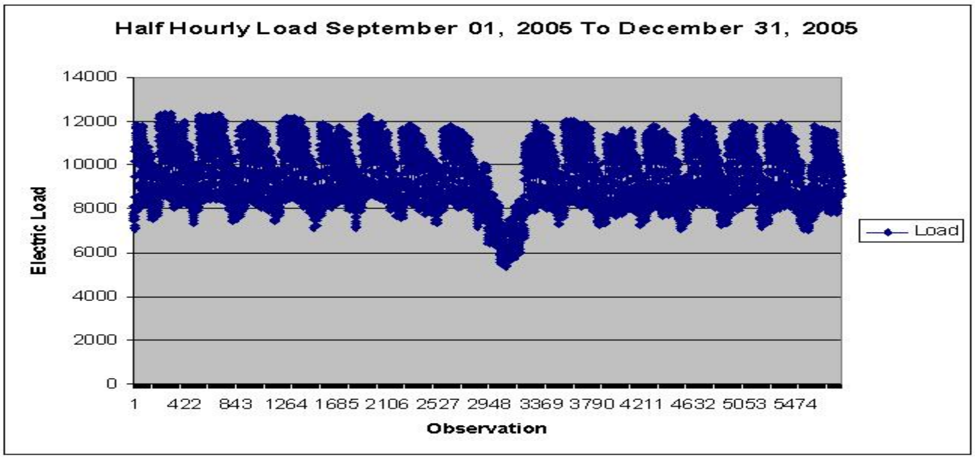


Figure 1. A Half Hourly Load from September 01, 2005 to December 31, 2005

2.2. Multilayer Feedforward Neural network

An essential factor of successes of the neural networks depends on training network. Among the several learning algorithms available, backpropagation has been the most popular and most widely implemented learning algorithm of all neural networks paradigms (BuHamra, Smaoui & Gabr, 2003).

Back Propagation (BP) trains multilayer feedforward networks with differentiable transfer functions to perform function approximation, pattern association and pattern classification. It is the process by which the derivatives of network error, with respect to network weights and biases, are computed to perform computations backwards through the network.

There are several different BP training algorithms with a variety of different computation and storage requirements. No single algorithm is best suited to all the problems. All the algorithms use the gradient of the performance function to determine how to adjust the weights to minimize the performance.

The basic BP algorithm adjusts the weights in the steepest descent direction (negative of the gradient); that is, the direction in which the performance function decreases most rapidly. The training process requires a set of examples of proper network inputs and target outputs. During the training, the weights and biases of the network are iteratively adjusted to minimize the network performance function.

Basically the BP training algorithm with three-layer feed-forward architecture which mean that, the network has an input layer, one hidden layer and an output layer. More hidden layers can be used but three layers are sufficient to enable this type of network to model any deterministic process within reasonable limit (Welstead, 1994).

In the feed-forward network, the neurons are generally grouped into layers. Signals flow from the input layer h to the output layer via unidirectional connections, the neurons being connected from one layer to the next, but not within the same layer(Pham & Liu, 1995).

Feedforward neural networks can be represented as follows, where each incoming signal or input y_{i} is multiplied by weight w_{ij} and then summed up. A hidden layer and output layer node perform a nonlinear transformation of its inputs

$$z_j = g\left(\sum_i y_i w_{ij} - \beta_j\right) \tag{1}$$

where y_i is the input, β_j is the bias of the j th node, w_{ij} is the weight which connection between two nodes i and j and $g(o)$ is a sigmoid function given by (BuHamra, Smaoui & Gabr, 2003).

$$g(x) = \frac{1}{(1+e^{-x})} \tag{2}$$

2.3. Recurrent Neural Network

Recurrent networks are quantitatively different from feedforward ones, because their structure incorporates feedback (Irwin, Warwick, & Hunt,1995; Pham & Liu, 1995; Connor & Atlas, 1991) and can be used in either continuous or discrete time (Connor & Atlas, 1991).

Generally, there are two types of recurrent networks: relaxation and standard. Relaxation networks start from a given state and settle to a fixed point, typically denoting a class. By imposing constraints on the feedback connections, the convergence can be guaranteed. It is possible for relaxation networks to never reach a fixed point, but such behavior is generally considered undesirable because it does not aid in classification (Connor & Atlas, 1991).

The outputs of some neurons are feedback to the same neurons or to neurons in preceding layers (Irwin, Warwick & Hunt,1995; Pham & Liu, 1995) with varying weight. (Irwin, Warwick & Hunt, 1995). Thus, the signal can flow in both forward and backward directions (Pham & Liu, 1995).

Recurrent neural network can be represented as follows:

$$T_i \dot{y}_i = -y_i + \sigma(s_i) + \mu_i, \qquad s_i = \sum_1^N w_{ij} y_j \tag{3}$$

with $y_i(t_o) = y_i^o$. Here y_i is the feedback isignals from all neurons and N is the number of neurons in the networks (Irwin, Warwick, & Hunt, 1995).

2.4. Forecasting Performance

In this study the criteria chosen to measure performance is the sum squared error (SSE) which is given by the following equation,

$$MSE = \frac{\sum_i^n (x_i - \hat{x}_i)^2}{n} \tag{4}$$

where x_i and $\hat{x}_i$ are the actual values and the predicted values respectively while n is the number of predicted values.

To measure the correlation between the actual values and the predicted values for training, validation and testing, the correlation coefficient denoted by *r* is calculated using the following formulae determined as

$$r = \frac{\sum_{i}^{n}(x_i - \bar{x}_i)(\hat{x}_i - \bar{\hat{x}})}{\sqrt{(s_x^2)(s_{\hat{x}}^2)}} \tag{5}$$

where x_i is the actual values, $\hat{x}_i$ is the predicted values, $\bar{x}_i$ is the mean of the actual values, $\bar{\hat{x}}_i$ is the mean of the predicted values, s_x^2 is the variance of the actual values, $s_{\hat{x}}^2$ is the variance of the predicted values and n is the number of the predicted values.

3. Results

Results for both feedforward and recurrent neural networks will be discussed based on selected optimized neural network parameters. The selected parameters for both neural networks are shown in Table 1 as follows.

Table 1. Selected Neural Networks Parameters

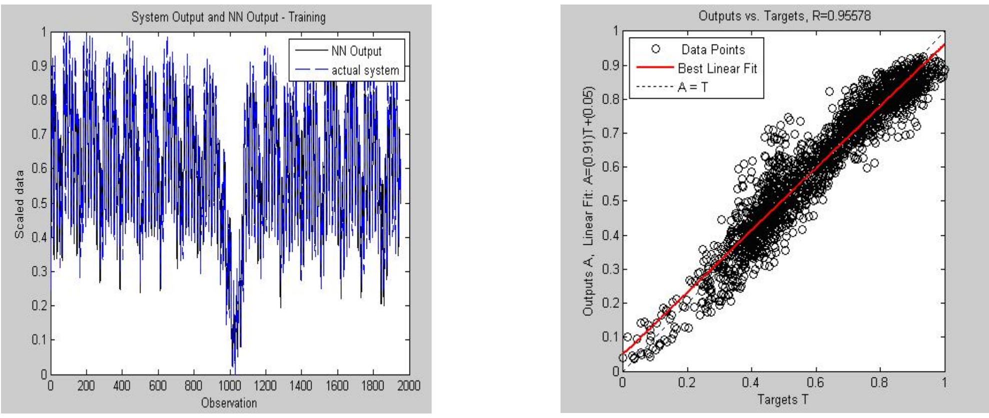
Input layer/ Hidden layer/ Output layer	One/ one/ one
Learning algorithm / Training algorithm	Back-propagation/ Leverberg Marquardt
Transfer function (hidden layer/output layer)	Log Sigmoid /Log Sigmoid
Input nodes/ Hidden nodes/ Output node	Three/ 32/ One
Epoch	300
Input layer	One

Table 2 shows *r*, the correlation coefficient, *m*, the slope of the linear regression, *b*, the intercept of the linear regression and the sum squared error SSE for both networks. Table 2 shows that, the values of *r*, correlation coefficient for both networks are greater than 0.9 which concluded highly accuracy. Also the sum squared error SSE for both networks is not much difference. In forecasting an accuracy play an important roles, hence the multilayer feedforward neural network is better for this study.

Table 2. The SSE and r for Multilayer Feed-Forward Neural Network and Recurrent Neural Network

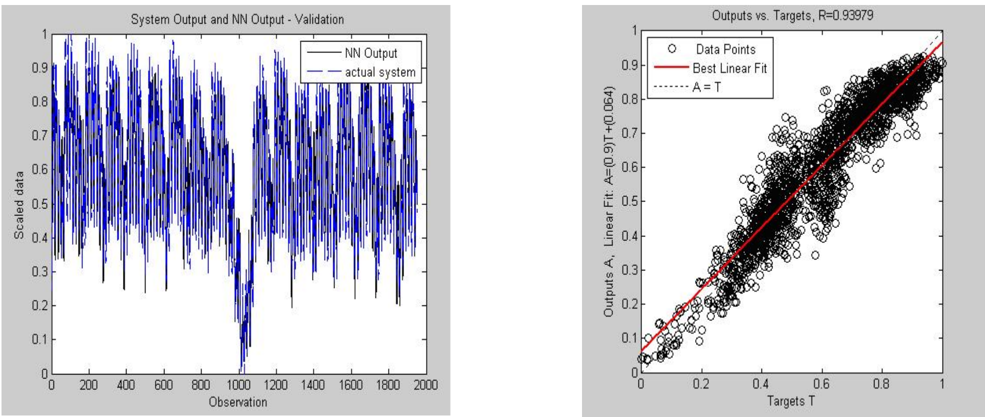
	Multi-layer Feed-forward Neural Network			Recurrent Neural Network		
	Training	Validation	Testing	Training	Validation	Testing
r	0.9558	0.9398	0.9447	0.9319	0.9137	0.9213
m, slope	0.9113	0.9024	0.8885	0.8630	0.8521	0.8416
b, intercept	0.0502	0.0641	0.0626	0.0782	0.0929	0.0899
SSE	7.1508	9.5982	9.1548	10.8806	13.5021	12.8829

Figures 2a - 4a show the actual and predicted values for training, validation and testing respectively of multilayer feed-forward neural network (MFNN) while Figures 2b - 4b show the values of *r*, the correlation coefficient between actual and predicted values for training, validation and testing respectively.



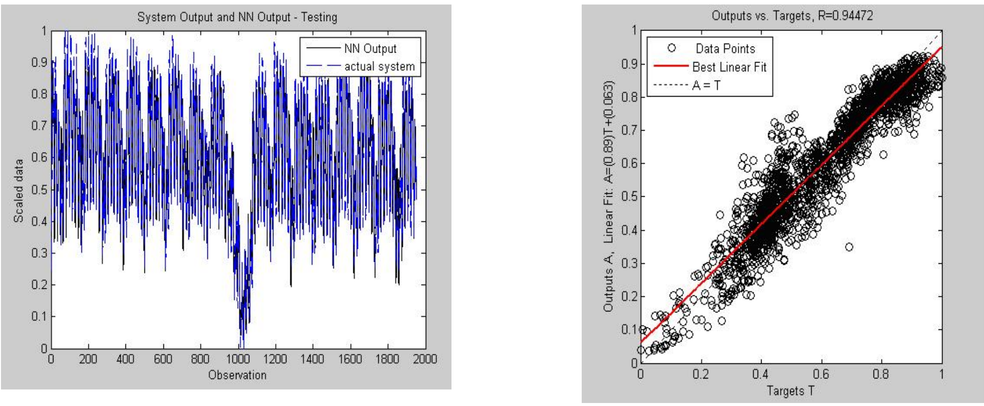
(a) (b)

Figure 2. The Actual and Predicted Values for Training of MFNN



(a) (b)

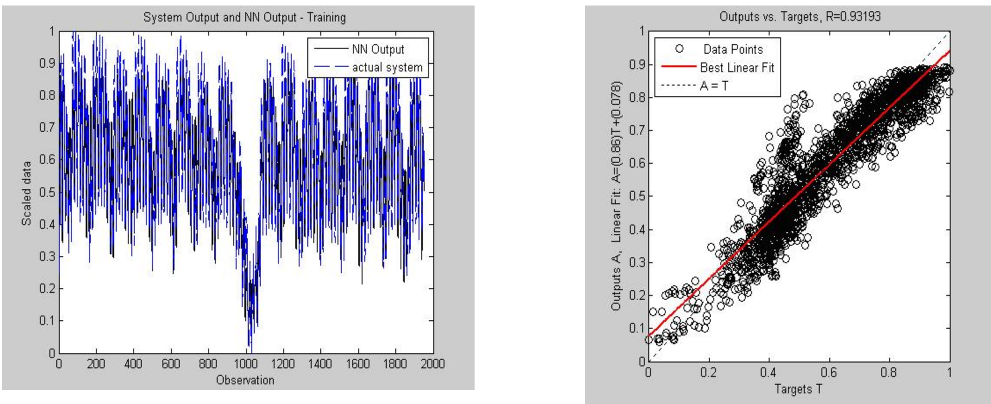
Figure 3. The Actual and Predicted Values for Validation of MFNN



(a) (b)

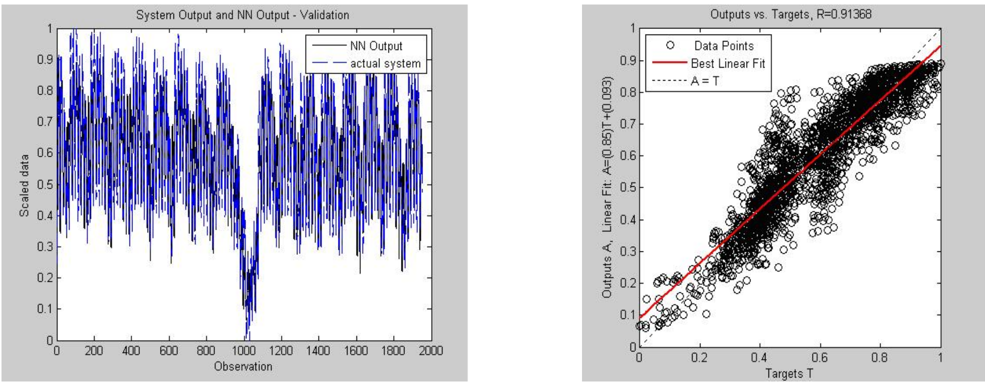
Figure 4. The Actual and Predicted Values for Testing of MFNN

Figures 5a - 7a show the actual and predicted values for training, validation and testing respectively of recurrent neural network (RNN) while Figures 5b - 7b show the values of *r*, the correlation coefficient between actual and predicted values for training, validation and testing respectively.



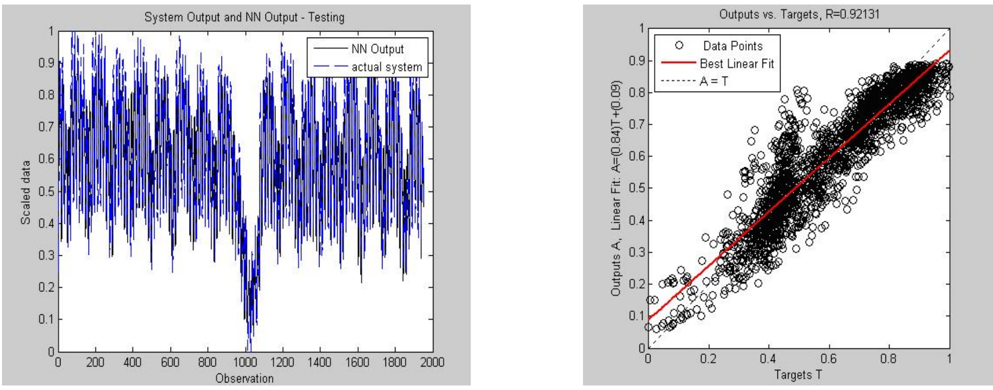
(a) (b)

Figure 5. The Actual and Predicted Values for Training of RNN



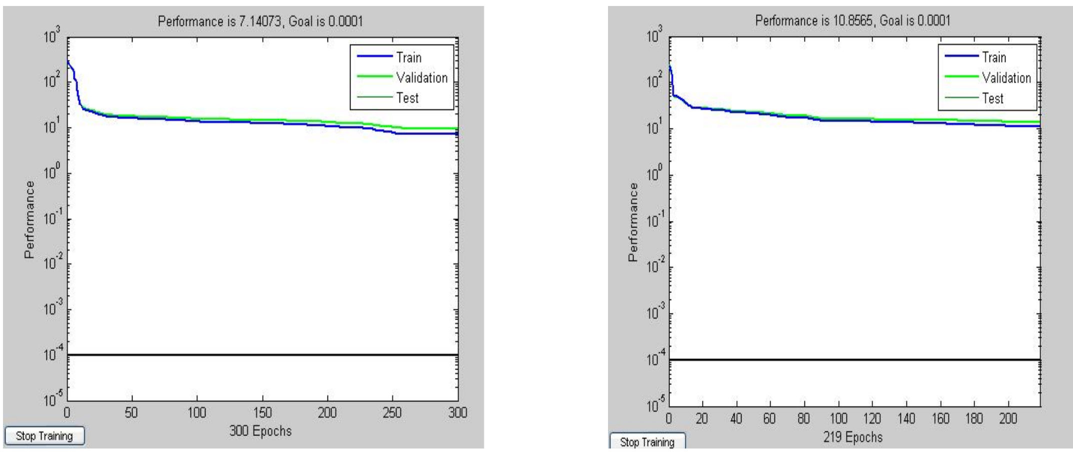
(a) (b)

Figure 6. The Actual and Predicted Values for Validation of RNN



(a) (b)

Figure 7. The Actual and Predicted Values for Testing of RNN



(a) (b)

Figure 8. Performance of Training for MFNN and RNN

Figure 8a and 8b show the performance (SSE) of training for multilayer feedforward neural network and recurrent neural network (RNN) respectively.

4. Conclusions

Two neural networks models have been used to forecast Malaysia's load. Comparing the results, we conclude that that, the multilayer feed-forward neural network is a better model for a half hourly Malaysia's load.

References

[1]. BuHamra, S., Smaoui, N., and Gabr, M. (2003). The Box-Jenkins analysis and neural networks: prediction and time series modeling. *Applied Mathematical Modelling*, 27, 805-815.

[2]. Connor, J., and Atlas, L. (1991). Recurrent Neural Networks and Time Series Prediction, *Proceeding IJCNN-91-Seattle, International Joint Conference on Neural Networks*, 301-306.

[3]. Connor, J. T., Martin, R. D., & Atlas, L. E.(1994). Recurrent Neural Networks and Robust Time Series Prediction. *IEEE Transactions on neural networks*, Vol. 5, No. 2, 240-254.

[4]. Pham, T. D., and Liu, X. (1995). *Neural Networks for Identification, Prediction and Control.* Great Britain.

[5]. Irwin, G. W., Warwick, K. and Hunt, K. J. (1995). *Neural Network Application in Control.* The Institution of Electrical Engineers, London.

[6]. Ho, S. L., Xie, M., and Goh, T. N. (2002). A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction. *Computer & Industrial Engineering*, 42, 371-375.

[7]. Khoa, N. L. D., Sakakibara, K. and Nishikawa, I. (2006). Stock Price Forecasting using Back Propagation Neural Networks with Time and Profit Based Adjusted Weight Factors. *SICE-ICASE International Joint Conference*, 5484-5488.

[8]. Kohzadi, N., Boyd, M. S., Kermanshi, B., and Kaastra, I. (1996). A comparison of artificial neural network and time series model for forecasting commodity prices. *Neurocomputing*, 10, 169-181.

[9]. Welstead, S. T. (1994). *Neural Network and Fuzzy logic applications in C/C++*, John Willey and Sons, Canada.