Applied Mathematical Sciences, Vol. 6, 2012, no. 108, 5359 - 5368

Forecasting Short Term Load Demand Using Multilayer Feed-forward (MLFF) Neural Network Model

Norizan Mohamed[1], Maizah Hura Ahmad[2], Suhartono[3] and Wan Muhammad Amir Wan Ahmad[4]

[1,4]Mathematics Department, Faculty Of Science And Technology
Universiti Malaysia Terengganu (UMT)
21030 Kuala Terengganu, Terengganu, Malaysia

[2]Department of Mathematics, Faculty of Science
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia

[3]Department of Statistics, Institut Teknologi Sepuluh Nopember, Indonesia.

[1]e-mail: norizan@umt.edu.my

Abstract

The purpose of this study was to apply the proposed model selection strategies in order to develop the best multilayer feed-forward neural network (MLFF) model for forecasting load demand. A one year half hourly load demand of Malaysia was used with the mean absolute percentage error (MAPE) as a forecasting accuracy. The fourth model selection strategy which considers both backward procedures in the selection of hidden and input nodes was applied. These fourth model selection strategies gave the best multilayer feed-forward neural network (MLFF) model which was composed of three input nodes, three hidden nodes and one output node. The in-sample MAPE was 1.1402% and the out-sample forecasts of all selected lead time horizons were greater than 1%. Comparing the forecasting performances of current study and previous study [9] the performances were same. Therefore, to develop the best MLFF model in forecasting time series data with two seasonal cycles, especially in Malaysia load data, any one of four model selection strategies can be considered.

Keywords: Load Forecasting, Multilayer Feed-Forward Neural Network (MLFF), Model Selection Strategies

INTRODUCTION

Load demand prediction is important for electric power planning and must be assessed with the greatest precision of any model. The utility power company needs forecasts for different time horizons in order to ensure the uninterrupted energy supply of customer [3]. Load forecasting can be broadly classified into four main categories which are long term forecasts, intermediate term forecasts, short term forecasts and very short term forecasts. Several forecasting methods with varying degrees of success have been implemented for load forecasting including multiple linear regression [5] and nonlinear multivariable regression model [3]. Artificial neural network with variety of approaches such as back propagation neural network [1, 5], dynamic artificial neural network [5], Elman artificial neural network [4] and Jordan recurrent neural network [1] have also been applied. Apart from these methods, Box-Jenkins autoregressive integrated moving average (ARIMA) models [5] have also been proposed.

To obtain accurate forecasts of Malaysia load demand using artificial neural networks, specifically multilayer feed-forward neural network model with one hidden layer trained with back-propagation algorithm, appropriate network architecture is needed. Even though more hidden layers can be used, a single hidden layer is sufficient for universal approximation. The neural network architecture is composed of the number of input, hidden and output nodes. Basically the number of input and output nodes are constrained by the type of application and in fact fixed. The number of hidden nodes however is adjustable and is usually estimated by a trial and error approach or determined by the researcher [11]. In this current study the output node is fixed at one and the selection of input and hidden nodes are selected based on the model selection strategies which proposed by Norizan Mohamed [6]. There are four types of model selection strategies [6]. Norizan Mohamed et al. [9] applied the first model selection strategies where the selection of hidden and input nodes considers both forward procedures. In this study the forth model selection strategies is applied where the selection of hidden and input nodes consider both backward procedures.

This paper is organized as follows. First, the multilayer feed-forward neural network (MLFF) model is presented. This is followed by a discussion on the detail results of a MLFF model. Our findings are then concluded based on forecasting evaluation method for this study.

METERIAL AND METHOD

Multilayer Feed-forward Neural Network (MLFF)

A well known neural model, which consists of an input layer, one or several hidden layers and an output layer. The neurons in the feed-forward neural network are generally grouped into layers. Signals flow in one direction from the input layer to the next, but not within the same layer [12]. An essential factor of successes of the neural networks depends on the training network. Among the several learning algorithms available, back-propagation has been the most popular and most widely implemented [2]. Basically, the BP training algorithm with three-layer feed-forward architecture means that, the network has an input layer, one hidden layer and an output layer. In neural network time series forecasting the output node is fixed at one. Thus, for the feed-forward network with N input nodes, H hidden nodes and one output node, the values $\hat{Z}_t$ are given by:

$$\hat{Z}_t = g_2\left(\sum_{j=1}^{H} w_j h_{j,t} + w_0\right) \tag{1}$$

Here w_j is an output *weight* from hidden node j to output node, w_0 is the bias for output node, and g_1 is an activation function. The values of the hidden nodes $h_{j,t}$, $j = 1,\ldots,H$ are given by:

$$h_{j,t} = g_1\left(\sum_{i=1}^{N} v_{ji} Z_{t,i} + v_{j0}\right),\ j = 1,\ldots,H \tag{2}$$

Here, v_{ji} is the input *weight* from input node i to hidden node j, v_{j0} is the bias for hidden node j, $Z_{t,i}$ are the lag variables where $\left(Z_{t,i},\ldots,Z_{t,N}\right)$ are $\left(Z_{t-i},\ldots,Z_{t-N}\right)$ respectively, $i = 1,\ldots,N$ and g_1 is an activation function. We illustrate the architecture of the multilayer feed-forward neural network model in Figure 1.

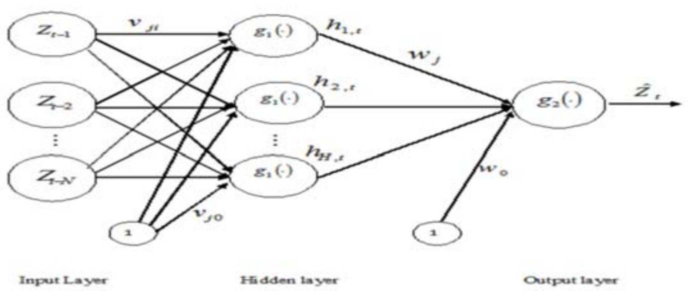

Fig. 1: The architecture of the multilayer feed-forward neural network model with one hidden layer, N input nodes, H hidden nodes and one output node.

In this study, mean absolute percentage error (MAPE) is considered as the standard measurement to examine the accuracy of the prediction model. This measure is most commonly used in the literature to evaluate forecasting performances. MAPE is defined as [10]:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Z_i - \hat{Z}_i}{Z_i}\right| \times 100 \tag{3}$$

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

The data used is one year half-hourly load demand measured in Megawatt (MW) from September 01, 2005 to August 31, 2006 is used in the current study. They are gathered from Malaysian electricity utility company, Tenaga Nasional Berhad (TNB), Malaysia. The data from September 01, 2005 to July 31, 2006 are used for training and the data from August 01, 2006 to August 31, 2006 are used for testing. The training data is illustrated in Figure 2.

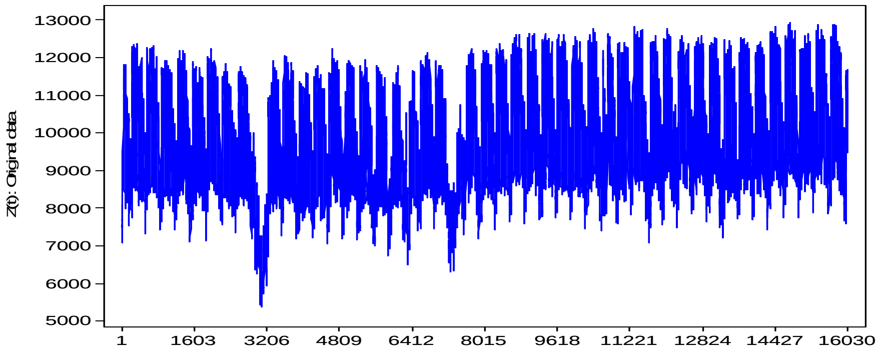

Fig. 2: A half hourly load from September 1, 2005 to July 31, 2006.

SELECTION OF HIDDEN NODES

The neural network architecture is composed of the number of input, hidden and output nodes. The main focus of the current study is to select the appropriate input variables and the appropriate number of hidden nodes since the output node is fixed at one. A common approach to select the optimal number of hidden nodes is a trial and error approach. Such approach was applied by Darbellay and Slama [2]. However, Norizan Mohamed [6] proposed four model selection strategies to find the best multilayer feed-forward neural network model. These four model selection strategies differ from each other in term of the selection of hidden and input nodes. The selections of hidden and input nodes are based on two procedures which are forward and backward. Previously study, Norizan Mohamed et al. [9] applied the first model selection strategies where it considers both forward procedures in the selection of hidden and input nodes. In this study, the fourth model selection strategy is applied where it considers both backward procedures in the selection of hidden and input nodes.

Applying the backward procedure, the number of hidden nodes started with ten nodes and reduced to the best simplest one. The results are reported based on the mean squared error (MSE) and R^2. When the hidden nodes are reduced from ten to one, the MSE in training is increased. The MSE in testing is decreased when the hidden nodes are reduced from ten to three nodes, indicating good fit in testing as tabulated in Table 1. However, beginning at three nodes, as the numbers of hidden nodes are reduced, the MSE in testing is increased. This indicates that reducing the number of hidden nodes, beginning at three nodes does not lead to further improvement. Hence, we suggest that the best number of hidden nodes in this study is three nodes.

Table 1: The hidden nodes, MSE training, MSE testing, R^2 training and R^2 testing

Hidden Nodes	MSE Training	MSE Testing	R^2 Training	R^2 Testing
10	7.0746×10^4	8.1640×10^4	0.9993	0.9993
9	7.0756×10^4	8.1588×10^4	0.9993	0.9993
8	7.0774×10^4	8.1579×10^4	0.9993	0.9993
7	7.0787×10^4	8.1571×10^4	0.9993	0.9993
6	7.0792×10^4	8.1554×10^4	0.9993	0.9993
5	7.0815×10^4	8.1535×10^4	0.9993	0.9993
4	7.0836×10^4	8.1515×10^4	0.9993	0.9993
3	$\mathbf{7.0863\times10^4}$	$\mathbf{8.1500\times10^4}$	**0.9993**	**0.9993**
2	7.0941×10^4	8.1561×10^4	0.9993	0.9993
1	7.0978×10^4	8.1602×10^4	0.9993	0.9993

SELECTION OF LAG VARIABLES

By using three nodes in hidden layer, which is concluded as the best number of hidden nodes from previous section and following the fourth model selection strategies which proposed by Norizan Mohamed [6], the backward procedure is applied to find the best number of lag variables, and consequently the combination of lag variables. There are seven lag variables in the best double seasonal ARIMA model (refer Equation 4) as proposed by Norizan Mohamed et al. [7,8]. By following the model selection strategies, the selection of lag variables is based on the best double seasonal ARIMA model [6]. Hence, the backward procedure reduced the nodes from seven to one as illustrated in Table 2. Result in Table 2 clearly shows that beginning with three input nodes, the performances of R^2 for both training and testing are reduced. These indicate that reducing input nodes do not lead to further improvement.

Therefore, the appropriate number of input nodes in this current study is three inputs. The inputs are the combination of lag variables Z_{t-1}, Z_{t-336} and Z_{t-337}.

$$Z_t = Z_{t-1} + Z_{t-48} - Z_{t-49} + Z_{t-336} - Z_{t-337} - Z_{t-384} + Z_{t-385} + a_t - 0.27184\,a_{t-1} - 0.76592\,a_{t-48} + 0.20821\,a_{t-49} - 0.85019\,a_{t-336} + 0.23112\,a_{t-337} + 0.65118\,a_{t-384} - 0.17702\,a_{t-385} \quad (4)$$

Table 2: The input variables, MSE training and MSE testing, R^2 training, R^2 testing and BIC training.

n	Input Variables	MSE Training	MSE Testing	R^2 Training	R^2 Testing	BIC Training
7	Z_{t-1}, Z_{t-48}, Z_{t-49}, Z_{t-336}, Z_{t-337}, Z_{t-384}, Z_{t-385}	1.6495×10^4	2.0145×10^4	0.9998	0.9998	9.7281
6	Z_{t-1}, Z_{t-48}, Z_{t-49}, Z_{t-336}, Z_{t-337}, Z_{t-385}	1.6991×10^4	2.0443×10^4	0.9998	0.9998	9.7559
5	Z_{t-1}, Z_{t-48}, Z_{t-49}, Z_{t-336}, Z_{t-337}	1.7086×10^4	2.0536×10^4	0.9998	0.9998	9.7596
4	Z_{t-1}, Z_{t-48}, Z_{t-336}, Z_{t-337}	2.0100×10^4	2.3661×10^4	0.9998	0.9998	9.9202
3	**Z_{t-1}, Z_{t-336}, Z_{t-337}**	**2.0357×10^4**	**2.3908×10^4**	**0.9998**	**0.9998**	**9.9311**
2	Z_{t-1}, Z_{t-336}	4.5963×10^4	4.9688×10^4	0.9995	0.9995	10.7436
1	Z_{t-1}	7.0863×10^4	8.1500×10^4	0.9993	0.9993	11.1747

THE BEST MLFF NEURAL NETWORK MODEL

The best training algorithm is the Levenberg-Marquardt back-propagation. Two comparable combinations of transfer function for hidden and output layers are the combination between log-sigmoid and linear functions and the combination between hyperbolic tangent and linear functions [6]. Previous study Norizan Mohamed et al. [9] applied the log-sigmoid function in the hidden layer and the linear transfer function in the output layer . In this study, the log-sigmoid function in the hidden layer and the linear transfer function in the output layer are also applied. In the earlier section, three nodes is proposed as the appropriate number for both input and hidden nodes. The input lag variables are Z_{i-1}, Z_{i-336} and Z_{i-337}. Hence, the appropriate neural network architecture which results the best multilayer feed-forward neural network model composed of three input nodes, three hidden nodes and one output node. This neural network architecture is same with previous study [9] and it can be represented as follows:

$$\hat{Z}_t = g_2\left(\sum_{j=1}^{3} w_j h_{j,t} + w_0\right) \tag{5}$$

where w_j is an output *weight* from hidden node j to output node, w_0 is the bias for output node, and g_2 is the linear function. $h_{j,t}$ are the values of the hidden layer nodes which can be represented as:

$$h_{j,t} = g_1\left(\sum_{i=1}^{3} v_{ji} Z_{t,i} + v_{j0}\right),\ j = 1,\ldots,3 \tag{6}$$

where v_{ji} is the input weight from input node i to hidden node j, v_{j0} is the bias for hidden node j, and g_1 is the log sigmoid function. $Z_{t,i}$ are the lag variables where $Z_{t,1}$, $Z_{t,2}$ and $Z_{t,3}$ are Z_{t-1}, Z_{t-336} and Z_{t-337} respectively. Equations 5 and 6 can also be represented as follows:

$$\hat{Z}_t = w_0 + w_1 h_{1,t} + w_2 h_{2,t} + w_3 h_{3,t} \tag{7}$$

$$h_{j,t} = \left[1 + \exp\left[-\left(v_{j0} + v_{j1} Z_{t-1} + v_{j2} Z_{t-336} + v_{j3} Z_{t-337}\right)\right]\right]^{-1},\ j = 1,\ldots,3 \tag{8}$$

The architecture of the best multilayer feed-forward neural network model is illustrated in Figure 3.

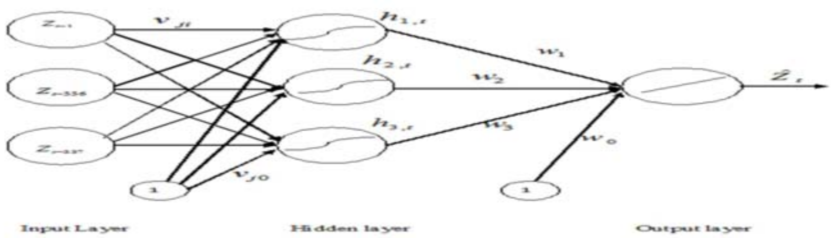

Fig. 3: The architecture of the best multilayer feed-forward neural network model with one hidden layer, 3 input nodes, 3 hidden nodes, one output node, log-sigmoid transfer function in the hidden layer and linear transfer function in the output layer

RESULTS

Since the architecture of the best multilayer feed-forward neural network model is same with previous study [9], hence the forecasting result of this current study also same. Basically, there are two types of forecasting which are one step ahead forecasts and k-step ahead forecasts. Here, both of them are presented. The results showed that the MAPE for all lead time horizons for the one-step ahead forecasts are less than the ones obtained using the k-step ahead forecasts as illustrated in Table 3. A reduction in MAPE percentages are clearly shown when one-step ahead out-sample forecasts are calculated and compared to k-step ahead out-sample forecasts. The MAPE are reduced with the reduction percentages of 60.6649%, 69.0467%, 76.4684% and 82.1030% at one-week forecast, two-week forecasts, three-week forecasts and one month forecast respectively. These reduction percentages are tabulated in Table 3. The out-samples of actual load data, k-step ahead and one-step ahead out-sample forecasts are illustrate in Figure 4.

Table 3: The MAPE of k-step and one-step ahead out-sample forecasts of the best MLFF neural network model.

	k-step ahead out-sample forecasts	One-step ahead out-sample forecasts	Reducing
In-sample forecast	1.1418	1.1418	
Out-sample one week forecast	3.1476	1.2381	60.6649%
Out-sample two-week forecasts	3.9598	1.2257	69.0467%
Out-sample three-week forecasts	5.0059	1.1780	76.4684%
Out-sample one month forecast	6.6790	1.1953	82.1030%

CONCLUSION

Comparing the forecasting performances of current study and previous study [9] the performances were same for both one step ahead and k-step ahead out-sample forecasts, where the one step ahead out-sample forecasts outperforms k-step ahead out-sample forecasts. Therefore, to develop the best MLFF model in forecasting time series data with two seasonal cycles, especially in Malaysia load data, any one of four model selection strategies can be considered.

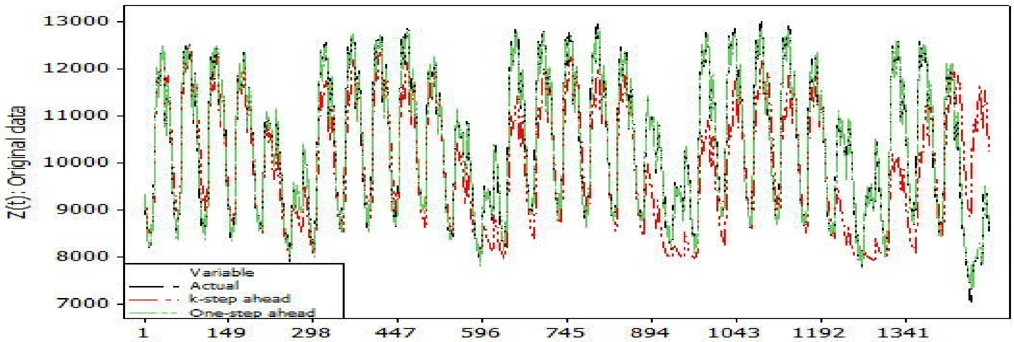

Fig. 4: The out-samples of actual data, k-step ahead and one-step ahead out-sample forecasts

ACKNOWLEDGEMENT

The authors would like to thank Universiti Malaysia Terengganu (UMT) for the support of this study. They are grateful to Tenaga Nasional Berhad (TNB), Malaysia for providing the load data. They also wish to express their sincere appreciation to all their friends who have helped them directly and indirectly.

REFERENCES

[1] B. Kermanshahi and H. Iwamiya, Up to year 2020 load forecasting using neural nets, Electric Power and Energy Systems, 24 (2002), 789-797.

[2] G.A. Darbellay and M. Slama, Forecasting the short-term demand for electricity, Do neural networks stand a better chance? International Journal of Forecasting, 16 (2000), 71-83.

[3] G.J. Tsekouras, E.N. Dialynas, N.D. Hatziargyriou and S. Kavatza, A non-linear multivariable regression model for midterm energy of power systems, Electric Power Systems Research, 77 (2007), 1560-1568.

[4] M. Beccali, M. Cellura, V. Lo Brano and A. Marvuglia, Short-tem prediction of household electricity consumption: Assessing weather sensitivity in a Mediterranean are, Renewable and Sustainable Energy Reviews, 12 (2007), 2040-2065.

[5] M.Ghiassi, D.K. Zimbra and H. Saidane, Medium term system load forecasting with a dynamic artificial neural network model, Electric Power System Research, 76 (2006), 302-316.

[6] Norizan Mohamed, Parametric and artificial intelligence based methods for forecasting short term electricity load demand, Unpublished PhD Thesis, 2011.

[7] Norizan Mohamed, Maizah Hura Ahmad and Suhartono, Short-term electricity load forecasting, Journal of Sustainability Science and Management, 6 (2011), 257-266.

[8] Norizan Mohamed, Maizah Hura Ahmad and Suhartono, Forecasting short term load demand using double seasonal ARIMA model, World Applied Sciences Journal, 13(1) (2011), 27-35.

[9] Norizan Mohamed, Maizah Hura Ahmad, Zuhaimy Ismail and Suhartono, Short term forecasting using multilayer feed-forward (MLFF) neural network model, Proceedings of Universiti Malaysia Terengganu 10th International Annual Symposium, UMTAS 2011 (2011), 40-46.

[10] R. Dong and W. Pedrycz, A granular time series approach to long-term forecasting and trend forecasting, Physica A, 387 (2008), 3253-3270.

[11] S.T. Welstead, Neural Network and Fuzzy Logic Applications in C/C++, John Willey and Sons, Canada, 1994.

[12] T.D. Pham and X. Liu, Neural Networks for Identification, Prediction and Control, Great Britain, 1995.

Received: May, 2012