American Journal of Applied Sciences 8 (8): 796-803, 2011
ISSN 1546-9239

Neural Network in Modeling Malaysian Oil Palm Yield

[1]Zuhaimy Ismail and [2]Azme Khamis
[1]Department of Mathematic, Faculty of Science, University Technology Malaysia, Malaysia
[2]Department of Mathematics, Faculty of Science, Arts and Heritage,
University Tun Hussein Onn Malaysia, Malaysia

Abstract: Problem statement: Forecasting of palm oil yield has become an important element in the management of oil palm industry for proper planning and decision making. The importance of yield forecasting has led us to explore modeling of palm oil yield for Malaysia using the most recent development of Artificial Neural Network (ANN). The main issue in yield forecasting is to predict the future value with the minimum error. **Approach:** Artificial neural networks are computing systems containing many interconnected nonlinear neurons, capable of extracting linear and nonlinear regularity in a given data set. It is an artificial intelligence model originally designed to replicate the human brain's learning process, a network with many elements or neurons that are connected by communications channels or connectors. The ANN can perform a particular function when certain values are assigned to the connections or weights between elements. In this study, a secondary data set from the Malaysian Palm Oil Board (MPOB) on the foliar nutrient composition, fertilizer trials and Fresh Fruit Bunch (FFB) yield were taken and analyzed. The foliar nutrient composition variables are the nitrogen N, phosphorus P, potassium K, calcium Ca and magnesium Mg concentration, while the fertilizer trials data are the N, P, K and Mg fertilizers and are measured in kg per palm per year. The foliar composition data was presented in the form of measured values whiles the fertilizer data in ordinal levels, from zero to three. **Results:** Two experiments were conducted to demonstrate the implementation ANN and for both experiment, the result demonstrated that the number of hidden nodes produces an effect to the overall forecast performance of the ANN architecture. From the first experiment, it shows that the number of runs does not affect the ANN performance, but changing the momentum to learning rates, due to shows a significant improvement in the forecast result. The experimental result will be in the form of statistical analysis, the best neural network performance, the residual analysis and the effect on the learning rate on the NN performance. **Conclusion:** This study showed that modeling of oil palm yield using neural network requires data to be prepared or modified to satisfy the requirement of the parameters involved. This analysis yields the conclusion that only the number of hidden nodes has a significant influence on the NN performance and there is no effect resulting from the number of runs or the momentum term value on the neural network's performance.

Key words: Palm oil yield, neural network, forecast accuracy and time series model, Artificial Neural Network (ANN), Fresh Fruit Bunch (FFB), human brain, fertilizer data

INTRODUCTION

Neural network or popularly known as Artificial Neural Networks (ANN), are computational models that consist of a number of simple processing units which communicate by sending signals to each other over a large number of weighted connections. One very important feature in ANN is in its adaptive nature where "learning by example" replaces "programming" in solving problems. This feature renders computational models very appealing in applications where one has little, or an incomplete understanding, of the problems to be solved, but where training data is available. Many different types of neural networks are being developed and used in many fields of application. New uses for ANN are being devised daily by researchers. Some of the most traditional applications include the area of Classification- to determine military operations from satellite photographs; to distinguish among different types of radar returns (weather, birds, or aircraft); to identify diseases of the heart from electrocardiograms; Noise reduction-to recognize a number of patterns (voice, images) corrupted by noise and Prediction -to predict the value of a variable, given historic values. Examples include forecasting of various types of loads, market and stock forecasting and weather forecasting (Kubde and Bansod, 2010; Adeli and Panakkat, 2009; Wang *et al*., 2009; Faraway and Chatfield, 1998).

Corresponding Author: Zuhaimy Ismail, Department of Mathematic, University Technology Malaysia, Malaysia

Neural networks, sometimes referred to as connectionist models, are parallel-distributed models that have several distinguishing features (Shabri *et al*., 2009; Ismail and Jamaluddin, 2008).

Since the invention of backpropagation algorithm to train feedforward multi-layer neural networks a decade ago, Neural Networks (NN) have been widely used for many types of problems in business, industry and science. One major use of NN is for time series forecasting. Many successful applications suggest that NN can be promising alternative tool for both forecasting researchers and practitioners. The popularity of NN is derived from the fact that they are generalized nonlinear forecasting models. Forecasting has been dominated by linear statistical methods for several decades. Although linear models possess many advantages in implementation and interpretation, they have serious limitation in that they cannot capture nonlinear relationships in the data, which are common in many complex real world problems. One of the major reasons for that problem is that there is a varying degree of nonlinearity in the data, which cannot be handled properly by linear statistical methods (Khamis *et al*., 2006, Fukushima, 2010; Cherkassky and Ma, 2009; Chow, 2007; Adeli and Panakkat, 2009).

In this study, we discuss the modeling of oil palm yield using ANN where data are required to be prepared and modified to satisfy the requirement of the parameters involved in the neural network architecture. The model developed will be used for forecasting of oil palm yield.

MATERIALS AND METHODS

ANN architecture has a set of requirements that must be satisfied. When modeling oil palm yield, the date are reorganized to fit the ANN requirement of the parameters involved. The data used in this study are secondary data set obtained from the Malaysian Palm Oil Board (MPOB). The data include the fresh fruit oil palm yield, the foliar nutrient composition, fertilizer trials. The amount of oil palm yield is measured by the amount of Fresh Fruit Bunch (FFB) yield. The foliar nutrient composition include the Nitrogen (N), Phosphorus (P), Potassium (K), Calcium (Ca) and Magnesium (Mg) concentration, while the fertilizer trials data are fertilizers with the same nutrient composition which are measured in kg per palm (palm tree) per year. The foliar composition data are in the form of measured values whiles the fertilizer data in ordinal levels, from zero to three.

In this study, the raw data used are checked, validated and then partitioned them into training, validation and testing set of data. The validity test is essential in ANN as it will indicate the presence of faulty data and once found, it pattern must be diagnosed and reason for deviances explained. In some cases, a portion of the data may be discarded. By definition, training sets are used to actually update the weights in a network, validation sets are used to decide the architecture of the network and testing sets are used to examine the final performance of the network. The primary concerns should be to ensure that (i) the training set contains enough data and a suitable data distribution to adequately demonstrate the properties we wish the network to learn; (ii) there is no unwarranted similarity between data in different data sets.

Mathematical model: ANN is an artificial intelligence model originally designed to replicate the human brain's learning process. A network consists of many elements or neurons that are connected by communications channels or connectors. It can perform a particular function when certain values are assigned to the connections or weights between elements. In a system, where there is no assumed structure but the network are trained so that a particular input leads to a specific target output (Shahrabi *et al*., 2009; Blackwell and Chen, 2009). The mathematical model for a neural network consists of a set of simple functions linked together by weights. It has a set of inputs x, output units y and hidden units z, which link the inputs to outputs (Fig. 1). The hidden units extract useful information from inputs and use them to predict the output. The type on neural network here is known the multilayer perceptron (Eskandarinia *et al*., 2010).

A network with an input vector of elements x_l (l = 1, 2,.., N_i) is transmitted through a connection that is multiplied by weight, w_{ji}, to give the hidden unit z_i (j = 1, 2, 3, …, N_k):

$$z_j = \sum_{l=1}^{N_i} w_{jl} x_l + w_{j0} \quad (1)$$

Where:
N_k = The number of hidden units
N_i = The number of input units

The hidden units consist of the weighted input and a bias (w_{j0}). A bias is simply a weight with constant input of 1 that serves as a constant added to the weight. These inputs are passed through a layer of activation function f which produces:

$$h_j = f\left[\sum_{l=1}^{N_i} w_{jl} x_l + w_{j0}\right] \quad (2)$$

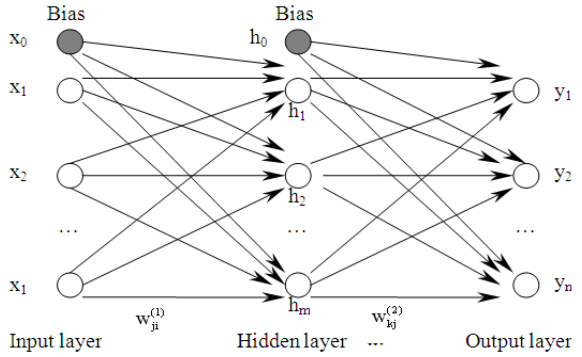

Fig. 1: Feed-forward neural network

The activation functions are designed to accommodate the nonlinearity in the input-output relationships. A common function is sigmoid or hyperbolic tangent:

$$f(z) = \tanh(z) = 1 - \frac{2}{[1+\exp(2z)]} \quad (3)$$

The outputs from hidden units pass another layer of filters:

$$v_k = \sum_{j=1}^{N_h} w_{kj} h_j + w_{k0} = \sum_{j=1}^{N_h} w_{kj} f\left[\sum_{l=1}^{N_i} w_{jl} x_l + w_{j0}\right] + w_{k0} \quad (4)$$

and fed into another activation function F to produce output y (k = 1, 2, 3, …, N_o):

$$y_k = F(v_k) = F\left[\sum_{j=1}^{N_h} w_{kj} f\left(\sum_{l=1}^{N_i} w_{jl} x_l + w_{j0}\right) + w_{k0}\right] \quad (5)$$

The weights adjustable parameters of the network and are determined from a set of data through the process of training (Cao and Zhu, 2010; Gholizadeh and Darand, 2009a). The training of a network is accomplished using an optimization procedure (such as nonlinear least squares). The objective is to minimize the Sum of Squares of Errors (SSE) between the measured and predicted output. There are no assumptions about functional form, or about the distributions of the variables and errors of the model, ANN model is more flexible than the standard statistical technique (Gholizadeh and Darand, 2009b; Chow, 2007). It allows for nonlinear relationship and complex classificatory equations. The users do not need to specify as much details about the functional form before estimating the classification equation but, instead, it lets the data determine the appropriate functional form (Shabri *et al*., 2009).

In accordance to standard analytical practice, the sample size was divided on a random basis two sets, namely the training set and the testing set. The training set and the testing set contain 80 and 20% of the total sample respectively. To evaluate the modeling accuracy the correlation coefficient, r and Mean Squares Error (MSE) were calculated. The model with a higher r and lower MSE was considered to be a relatively superior model.

A parametric analysis would be impossible without a discussion of the degrees of freedom of the network. In any parametric analysis, the number of degrees of freedom is defined as the number of observations minus the number of parameters that are free to vary (Gujarati and Porter, 2008). If N represents the number of observations and k the number of estimated parameters, then the degrees of freedom, df, can be calculated using:

$$df = N - k \quad (6)$$

This approach to the degrees of freedom can be applied directly to the feed-forward neural network if the network has only a single output. In this case, let k represent the number of estimated parameters. These estimated parameters include not only the connection weights that feed into the output and the output's intercept parameter, but also the connection weights that interconnect the hidden layers. It also calculates the bias weights that correspond to each of the hidden layers' transformation nodes. So, the numbers of parameters estimated in the feed-forward neural networks with one hidden layer are calculated as:

$$k = n_h (n_i + 2) + 1 \quad (7)$$

Where:
n_h = The number of hidden neurons
n_i = Number of input nodes

A necessary condition in any parametric model is that the number of available degrees of freedom must be positive. This constraint imposes an upper limit on the size of the network. If there are N observations, then the maximum size of the hidden nodes can be calculated using:

$$n_{h(max)} = \frac{N-1}{n_i + 2} \quad (8)$$

As shown in Fig. 1, the input nodes are N, P, K, Ca and Mg concentrations and the output node is FFB yield which is measured in tonne per hectare per year.

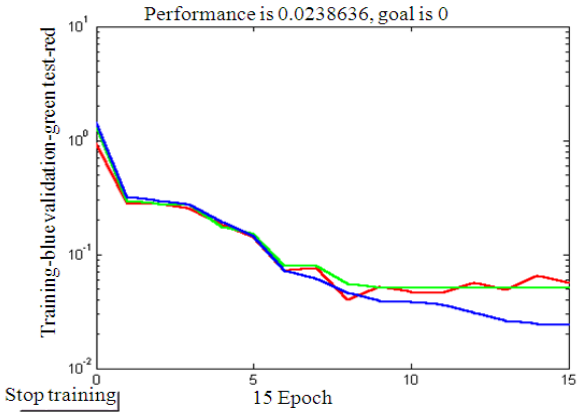

Fig. 2: The early stopping procedure for the feedforward neural networks

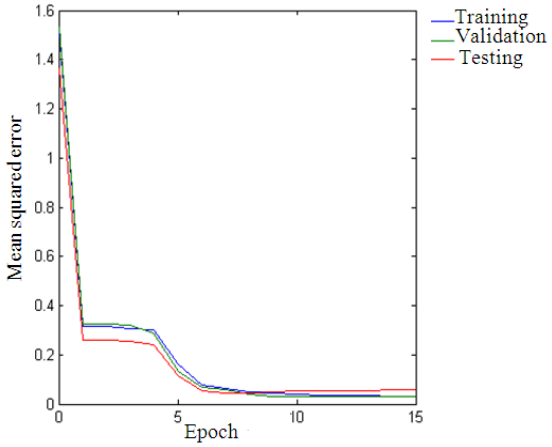

Fig. 3: The mean squares error for training, validation and testing

ANN application: In modeling oil palm yield we ran neural networks using the neural networks toolbox in MATLAB package with a built-in procedure for a simple neural networks program. Nevertheless we are required to develop a simple algorithm to call the neural networks built-in procedure. Each procedure has its own specific name. In the first part we only considered N, P, K, Ca and Mg concentration from foliar analysis as input nodes and FFB as the output node (Fig. 2). The number of hidden nodes varies from one station to another because of the different number of observations. The maximum number of hidden nodes is obtained from Eq. 8 to ensure that the degree of freedom of the model is always positive.

In our case we consider the fully connected feed-forward neural network and supervised neural networks because we have input and target data set as shown in Fig. 2. We also assume that all the inputs have a significant influence on the production of oil palm yield. We start the network with a small number of hidden nodes, which are added one by one until the maximum number of hidden nodes, which is defined from Eq. 8, is reached.

The first step in training a feed-forward network is to create the network object. The function *newff* creates a trainable feed-forward network (Al-Zubi *et al*., 2010; Wang *et al*., 2009). The user should determine the transfer function in the first and second layers and when the transfer function was obtained and the command *newff* used, the network was ready for training. Before training a feed-forward network, the weights and biases must be initialized. The initial weights and biases are created with the command *init*. This function takes a network object as input and returns a network object with all weights and biases initialized. For feed-forward networks, the weights' initialization is usually set to random (*rands*), which sets weights to random values between -1 and 1.

This study considers the training networks for function approximation. The training process requires networks inputs *input* and target outputs *target*. During training, network's weights and biases are iteratively adjusted to minimize the network performance function using mean squares error, *mse*. The training algorithm used in our study is Leverberg-Marquardt (*trainlm*) because this algorithm appears to be the fastest method for training moderate-sized feed-forward neural networks and it is also very efficient (Fahimifard *et al*., 2009). We then apply an early stopping technique to avoid overtraining the neural networks and to improve generalization of the networks. This technique requires the data to be divided into three sets of data. The first set is the training set. This is used to compute the gradient and update the network weights and biases. The second set of data is the validation set. The error on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training, as does the training set error. However, when the network begins to over fit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iteration, the training is stopped and the weights and biases at the minimum of the validation error are returned as shown in Fig. 2-3 presents the MSE value for each phase. The MSE value decreases when the number of epoch is less than five. It remains consistent until epoch fifteen, when the training stops. We divided the data into 3 sets, the training set, validation set and testing set of data with ratio 70, 15 and 15% respectively.

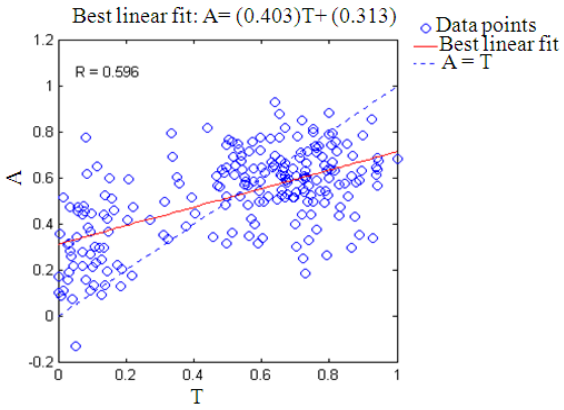

Fig. 4: The correlation coefficient between the actual (A) and predicted (T) values

In our study we use the correlation coefficient as a method of measuring the correlation between the actual value and the predicted value. When the correlation value approaches one, it shows that the actual and predicted values are close and that the model fits the data well. Let X and $\hat{X}$ be the actual value and predicted value from the specified model, σ_X^2 and $\sigma_{\hat{X}}^2$ be the variance of the actual and predicted observation and $\overline{X}$ and $\overline{\hat{X}}$ are the mean actual and mean predicted observation. So, the correlation coefficient between the actual and predicted values is defined as:

$$r = \sum_{i=1}^{n} \frac{(X_i - \overline{X})(\hat{X} - \overline{\hat{X}})}{\sqrt{\sigma_X^2 . \sigma_{\hat{X}}^2}} \quad (9)$$

MATLAB also provides the graphically best fitted line between the actual and target values.

We also carry out test to determine if the number of hidden nodes, the number of runs, momentum terms and learning rate does affect the performance of the NN model. In doing this we first clarify how the experiment was designed. In the first stage, three factors were considered namely the number of hidden nodes HN, number of runs NR and momentum rate MR. The number of hidden nodes has eight levels, between 3 and 10, there are six levels for the number of runs, (3, 5, 7, 10, 15 and 20) and four levels for the momentum terms, which are 0.25, 0.5, 0.75 and 0.95. We combined the information from fertilizer trials (N, P, K and Mg fertilisers) and information from foliar analysis the N, P, K, Ca and Mg concentration. Therefore, this neural network architecture involves nine inputs and one output as shown in Fig. 4.

Experiment 1: This experiment considers three factors, namely hidden nodes, number of runs and momentum term. This was carried out by changing the level of one factor and assuming that two factors are fixed to run the networks. We then changed them from one level to next level, recorded the error in estimation for each phase (training, validation and testing) then calculate the average of the error. As an example, suppose that the hidden node was set to three and number of runs was also three and that the momentum term is 0.25. Now we can write the experiment as [3, 3, 0.25].

The first value represents the number of Hidden Nodes, (HN), the second value represents the Number of Runs, (NR) and the last value represents the Momentum Term, (MT). In general the experiment can be written as [HN, NR, MT]. The momentum term level will increase and the process is repeated for each factor until the maximum value at [10, 20, 0.95] is reached. The three layers of fully connected neural networks with nine input nodes and one output node.

Experiment 2: In the second experiment, we changed the momentum term with the learning rate and we set the momentum term at random. We still used the two factors of the number of hidden nodes and the number of runs. The experiment then included the number of hidden nodes HN, number of runs NR and learning rate, LR and could be written as [HN, NR, LR]. We considered the number of hidden nodes as having seven levels, i.e. 2, 4, 6, 8, 10, 12 and 14 and the number of runs as also having seven levels i.e. 3, 5, 7, 9, 11, 15 and 20. The learning rate had five levels, i.e. 0.05, 0.15, 0.25, 0.45, 0.65 and 0.95. We repeated the process as in the first experiment until the maximum levels for each factor were obtained.

RESULTS

The performance of NN was due to the effect of the combination activation function in the hidden layer and output layer. Six combination activation functions were considered, namely log-sigmoid and log-sigmoid (LL), log-sigmoid and pure-lin (LP), log-sigmoid and tan-sigmoid (LT), tan-sigmoid and pure-lin (TP), tan-sigmoid and log-sigmoid (TL) and tan-sigmoid and tan-sigmoid (TT). All the combinations of activation functions were run and the mean squares error for each number of hidden nodes were recorded. The three phases in NN modeling are the training, validation and testing. Using the average MSE, the performance of NN architecture is determined. Here we consider the MSE as the testing variable of each phase. We also tested the correlation between the predicted and observed values.

Table 1: The F statistics value for different combination of activation functions used in coastal and inland areas

	The F-value				
Station	Training	Validation	Testing	Average	Correlation
Inland stations					
ILD1	3.368*	17.997*	12.055*	10.729*	3.062*
ILD2	7.850*	15.516*	14.949*	10.091*	1.601
ILD3	2.736*	12.899*	1.924	7.951*	4.431*
ILD4	2.291*	15.055*	10.452*	13.700*	3.058*
ILD5	1.859	2.306*	42.523*	16.715*	2.519*
ILD6	3.132*	22.047*	4.755*	0.927	4.606*
ILD7	1.766	13.455*	7.028*	5.812*	0.916
ILDT	0.853	11.736*	12.742*	3.732*	0.613
Coastal stations					
CLD1	0.847	11.091*	18.495*	15.103*	6.454*
CLD2	3.664*	7.724*	9.166*	10.197*	2.403*
CLD3	3.265*	3.762*	1.918	2.905*	4.017*
CLD4	1.295	12.413*	6.006*	8.957*	5.272*
CLD5	1.524	10.218*	7.112*	1.615	2.139
CLD6	3.232*	6.523*	10.145*	7.518*	6.146*
CLD7	1.366	5.092*	3.354*	2.865*	5.911*
CLDT	2.794*	8.083*	35.022*	13.037*	2.047

Note: * significant at 5% level

Table 2: The MAPE values of the neural network model

Station	LL	LP	LT	TP	TL	TT
Inland area						
ILD1	0.116	0.132	0.146	0.144	0.143	0.146
ILD2	0.116	0.126	0.124	0.131	0.126	0.122
ILD3	0.119	0.119	0.113	0.109	0.114	0.117
ILD4	0.136	0.128	0.130	0.129	0.135	0.132
ILD5	0.094	0.113	0.107	0.096	0.096	0.097
ILD6	0.099	0.101	0.099	0.102	0.105	0.072
ILD7	0.167	0.158	0.169	0.164	0.161	0.175
ILDT	0.057	0.056	0.056	0.058	0.057	0.057
Coastal						
CLD1	0.146	0.134	0.139	0.149	0.122	0.146
CLD2	0.073	0.064	0.072	0.066	0.078	0.074
CLD3	0.076	0.066	0.075	0.069	0.084	0.079
CLD4	0.113	0.129	0.112	0.128	0.116	0.113
CLD5	0.149	0.149	0.158	0.143	0.153	0.151
CLD6	0.128	0.124	0.123	0.124	0.133	0.130
CLD7	0.107	0.108	0.108	0.103	0.106	0.110
CLDT	0.150	0.146	0.151	0.157	0.153	0.152

The activation function combination (input layer to hidden layer and hidden layer to output layer)

Results of experiment 1: After completing the experiment, we rearranged the data for variance analysis and response surface analysis. The F value of hidden nodes, 8.7759 ($p = 0.0000$ and $df = (7, 1912)$), indicated that the performance of the neural networks model was statistically affected by the number of hidden nodes. The F value for number of runs, 1.6950 ($p = 0.1330$ and $df = (5, 1914)$) and the momentum term, 1.3300 ($p = 0.2630$ and $df = (3, 1916)$), show that both factors did not influence the overall performance of the neural networks. This analysis yields the conclusion that only the number of hidden nodes has a significant influence on the NN performance and there is no effect resulting from the number of runs or the momentum term value on the neural network's performance.

Results of experiment 2: After running the analysis of variance, we found that the F value for the hidden nodes is 8.0480 ($p = 0.0000$ and $df = (6, 2932)$) and the F value for the number of runs is 2.8840 ($p = 0.0080$ and $df = (6, 2932)$). This indicates that both factors affect the neural network's performance. However, the F value for the learning rate is 1.6090 ($p = 0.1540$ and $df = (5, 2933)$), which means that the null hypothesis cannot be rejected and we conclude that the learning rate does not influence the neural network's performance.

DISCUSSION

We are interested to test whether all combination of activation functions will produce equal MSE values. So that, two hypotheses were tested; the null hypothesis, H_0: all the MSE forming for each combination activation functions are equal and the alternative hypothesis, H_a: at least one combination activation function is not equal. In this case, the dependent variable was MSE value and the independent variables were the combinations of activation function. For further explanation, station ILD1was considered as an example. For each combination activation function, the NN was running using 2 hidden nodes to 30 hidden nodes and the MSE values and correlation coefficient for each phase were recorded. The numbers of hidden node depended on the maximum number of hidden nodes which was obtained from equation (6.3). Then, rearranged the MSE values and performed the analysis of variance.

As a standard procedure, the F statistic was used to test the null hypothesis. The results of the test for the inland area are presented in Table 1. The F values for training, validate, testing and average are 3.368, 17.997, 12.055 and 10.729, respectively and found statistically significant at 5% level at (5, 198) degrees of freedom. Then, by using the same procedure, others station were obtained. As we can see, almost all the stations showed the p-value less than 5% corresponded with their degrees of freedom, which signified rejection of the null hypothesis, except for stations ILD6, ILD7 and the ILDT. At station ILD6 the average MSE for different combination activation functions can be assumed equal as it was also found in the training and correlation results at station ILD7.

Our analysis shows that the result is the same for coastal area, where all the tests are significant at the 5 percent level (Table 2). In the training phase, the null hypothesis is not rejected at three stations, for one

station at the testing phase and for the others in the average of MSE. This study shows that the combination of activation functions has significant influence on the overall performance of the NN model.

The Duncan test was performed using the average of MSE. For each station, the test shows a different result. This means that the NN performance depends on the data site characteristic. Station CLD2 stated that the LL activation function gives the smallest average of MSE, compared to the LP and others. If we look at the value itself, the differences between combination functions are rather small. In ANOVA testing, variability of the variance plays a big role. The fluctuation of MSE occurs when the number of hidden nodes is added. Station CLD7 gave an interesting situation, where 5 combination activation functions performed more comparably than with the TP combination. This also happened at station ILD3 in the inland area. According to the findings, no generalization can be made on the selection of the combination activation function and we suggest that the trial and error method is an alternative way in selecting the best combination.

The best models in determining the hidden nodes and best NN architecture were selected based on the minimum value of MAPE and are represented in Table 2.

CONCLUSION

For both experiments, the result demonstrated that the number of hidden nodes produces an effect in the overall performance of the NN architecture. The experiment also found that the momentum term and learning rate do not reflect their influence in the NN's performance. The first experiment shows that the number of runs did not affect the NN performance, but changing the momentum term to learning rate, due to shows a significant effect on NN performance. The number of runs too influences NN performance.

ACKNOWLEDGEMENT

This research was supported by the Ministry of Science, Technology and Innovation, Malaysia (MOSTI) under IRPA Grant Vot No. 74285 and the Department of Mathematics, Faculty of Science, University Technology Malaysia. These supports are gratefully acknowledged. The authors would like to thank lecturers and friends for the helpful ideas and discussion. We also wish to thank Malaysian Palm Oil Board (MPOB) for providing necessary data in this study.

REFERENCES

Adeli, H. and A. Panakkat, 2009. A probabilistic neural network for earthquake magnitude prediction. Neural Netw., 22: 1018-1024. PMID: 19502005

Al-Zubi, Y., A. Sheta and J. Al-Zubi, 2010. Nile river flow forecasting based takagi-sugeno fuzzy model. J. Applied Sci., 10: 284-290. DOI: 10.3923/jas.2010.284.290

Blackwell, W.J. and F.W. Chen, 2009. Neural networks in atmospheric remote sensing. 1st Edn., Artech House, ISBN-10: 1596933720, pp: 234.

Cao, Y. and Q.X. Zhu, 2010. The software reliability forecasting method using fractals. Inform. Technol. J., 9: 331-336. http://docsdrive.com/pdfs/ansinet/itj/2010/331-336.pdf

Cherkassky, V. and Y. Ma, 2009. Another look at statistical learning theory and regularization. Neural Netw., 22: 958-969. PMID: 19443179

Chow, T.W.S., 2007. Neural networks and computing: Learning algorithms and applications. 1st Edn., World Scientific Publishing Company, ISBN-10: 9781860947582, pp: 324.

Eskandarinia, A., H. Nazarpour, M. Teimouri and M.Z. Ahmadi, 2010. Comparison of neural network and K-nearest neighbor methods in daily flow forecasting. J. Applied Sci., 10: 1006-1010. DOI: 10.3923/jas.2010.1006.1010

Fahimifard, S.M., M. Homayounifar, M. Sabouhi and A.R. Moghaddamnia, 2009. Comparison of ANFIS, ANN, GARCH and ARIMA techniques to exchange rate forecasting. J. Applied Sci., 9: 3641-3651. http://www.irinaclimbs.com/Papers/GARCH_vs_MachineLearning.pdf

Faraway, J. and C. Chatfield, 1998. Time series forecasting with neural networks: A case study. J. Royal Stat.: Series C., 47: 231-250. DOI: 10.1111/1467-9876.00109

Fukushima, K., 2010. Neural network model for completing occluded contours. Neural Netw., 23: 528-540. DOI: 10.1016/J.NEUNET.2009.10.002

Gholizadeh, M.H. and M. Darand, 2009a. Forecasting precipitation with artificial neural networks (case study: Tehran). J. Applied Sci., 9: 1786-1790. http://www.cabdirect.org/abstracts/20093120615.html

Gholizadeh, M.H. and M. Darand, 2009b. Forecasting the air pollution with using artificial neural networks: The case study: Tehran City. J. Applied Sci., 9: 3882-3887. DOI: 10.3923/jas.2009.3882.3887

Gujarati, D. and D. Porter, 2008. Basic Econometrics. 5th Edn., McGraw-Hill/Irwin, New York, ISBN-10: 9780073375779, pp: 944.

Ismail, Z. and F.A. Jamaluddin, 2008. A backpropagation method for forecasting electricity load demand. J. Applied Sci., 8: 2428-2434. DOI: 10.3923/ jas.2008.2428.2434

Khamis, A., Z. Ismail, K. Haron and A.T. Mohammed, 2006. Neural network model for oil palm yield modeling. J. Applied Sci., 6: 391-399. DOI: 10.3923/jas.2006.391.399

Kubde, R.A. and S.V. Bansod, 2010. Collaborative planning forecasting and replenishment initiatives: A state of art. Asian J. Industrial Eng., 2: 89-104. http://docsdrive.com/pdfs/knowledgia/ajie/2010/89-104.pdf

Shabri, A., R. Samsudin and Z. Ismail, 2009. Forecasting of the rice yields time series forecasting using artificial neural network and statistical model. J. Applied Sciences, 9: 4168-4173. DOI: 10.3923/jas.2009.4168.4173

Shahrabi, J., S.S. Mousavi and M. Heydar, 2009. Supply chain demand forecasting: A comparison of machine learning techniques and traditional methods. J. Applied Sci., 9: 521-527. http://docsdrive.com/pdfs/ansinet/jas/2009/521-527.pdf

Wang, J., L. Huang and Z. Guo, 2009. Global asymptotic stability of neural networks with discontinuous activations. Neural Netw., 22: 931-937. PMID: 19443177