Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 6 (2): 391-399, 2006
ISSN 1812-5654

Neural Network Model for Oil Palm Yield Modeling

[1]Azme Khamis, [2]Zuhaimy Ismail, [3]Khalid Haron and [3]Ahmad Tarmizi Mohammed
[1]Science Studies Centre, Kolej Universiti Teknologi Tun Hussein Onn, Malaysia
[2]Department of Mathematics, Universiti Teknologi Malaysia, Malaysia
[3]Malaysian Palm Oil Board, Malaysia

Abstract: This research presents a study on the development of a model for oil palm yield using neural network approach. The structure of this neural network requires the identification of the input variables and the output. We identified that the percentages of nitrogen, phosphorus, potassium, calcium and magnesium in leave were used as input variables and fresh fruit bunch was used as the target variable. An investigation of the combinations of activation function in the input layer to the hidden layer and the hidden layer to the output layer found that each combination also affects the neural network performance. The effect of the learning rate, momentum term, number of runs and number of hidden nodes was also investigated. The number of hidden nodes was found to significantly affect the neural network performance. However, the learning rate, momentum term and number of runs were found to have an insignificant effect on the neural network performance. Using R^2 values the suitability of the models were measured. Results demonstrate that the neural network model outperformed regression analysis, which can be considered as alternative in modeling of oil palm yield.

Key words: Neural network model, modeling, oil palm yield, activation function

INTRODUCTION

The oil palm (*Eleais guineensis*. Jacq.) is a plant of African origin and is grown commercially in Africa. In the early 19th century the British brought the industry into this country. The oil palm was first planted in Bongor-Indonesia in 1848 and in Malaysia in 1870, which was the same time rubber seeds were brought[1]. The oil palm industry is developing at a fast rate and requires a lot of research. For example in 1975, oil palm plantations covered 641,791 ha but this area increased vigorously to 3.8 million ha by the year 2003. In 2003 the figure increased by around 12.5% to 12,248 million ton, from 10,886 million tons in 2002. Meanwhile, Indonesia only recorded a 7.07% increase over the same period. This study took the challenge to contribute our knowledge to the development of the oil palm industry.

Previous research such as[2-12] have contributed significantly to the oil palm industry. In most cases, the researchers focused their study on the effect of environmental factors, such as evapotranspiration, moisture and rainfall on the oil palm yield growth. The most popular method used in the oil palm industry today is multiple linear regression. This model is used to investigate the causality effect of the independent variables on the dependent variable. Literature shows that the foliar nutrient composition can be used as an indicator to estimate the oil palm yield. The foliar nutrient composition is also dependent on several factors, such as climate, soil nutrients, fertilizers, pests and diseases, but there is little research which concentration on modeling these factors. This study explores the possibility of improving the model, in particular in improving the level of accuracy that it can produce.

Multiple linear regression can be used to find the relationship between the dependent variable and the independent variable. There can be more than one independent variable, which allows for the additional relevance of the independent variables to the model. In this sense, multiple linear regression is rather flexible. Present study emphasizes the proposed new independent variables in the model, an area yet to be explored by researchers. Real life, nothing seems to work linearly all the time. Data is sometime inclusive of outlier or unusual observations. This literature suggests various new heuristic methods to improve modeling. We explore the flexibility of the neural network to improve the estimated model performance and the model's accuracy.

The neural network model is a causality model that is widely used to solve complex problems. This study discusses the implementation of the neural network approach to modeling oil palm yield. Neural networks are viable and very important models, which are used to solve a variety of problems. These include pattern classification, function approximation, image processing, clustering, forecasting and prediction. It is common practice to use

Corresponding Author: Azme Khamis, Science Studies Centre, Kolej Universiti Teknologi Tun Hussein Onn, Malaysia

the trial and error method to find suitable neural network architecture for a given problem. A number of neural network model have been successfully used[13,14]. There are several different types of neural network: perceptron network, multiple layer perceptron, radial basis function network, Kohonen network, Hopfield network etc. The multiple layer perceptron is the most reported and applied neural network in current usage. The most popular architecture, in the class of the multiple layer perceptron, is the feedforward neural network, Shearer *et al.*[15] developed a method of corn yield prediction using a neural network classifier, which was trained using soil landscape features and soil fertility data. They found that neural network classifiers can be used to predict spatial yield variability when describing growing corn crops. Comparative studied by using several methods for predicting crop yield based on soil properties was done by Drummond *et al.*[16]. They noted that the process of understanding yield variability is made extremely difficult by the number of the factors that affect the yield. They used several methods such as Multiple Linear Regression (MLR), Stepwise Linear Regression (SMLR), Partial Least Squares (PLS), Projection Pursuit Regression (PPR) and back-propagation neural network. They found that the neural network gave similar results to PPR and better results than MLR, SMLR and PLS. They also concluded that less complex statistical methods, such as standard correlation, did not seem particularly useful in understanding yield variability. The correlation matrices described each factor's linear relationship to the yield. However, when complex nonlinear relationships between factors exist, the correlation may provide inaccurate and even misleading information about these relationships.

Prediction capabilities were highest for nonlinear and non-parametric methods. One method tried to use was a feed-forward, back propagation neural network for corn and soybean yield prediction[16]. They included soil properties such as phosphorus (P), potassium (K), pH, organic matter, topsoil depth and magnesium saturation as inputs and compared the results with other statistical models. The NN showed promise as an aid in understanding yield variability, although their network model needed further improvements to increase the accuracy and they did not include weather information and other factors in their NN. Artificial neural network to build a corn yield prediction model for application in precision farming was conducted by Liu *et al.*[17]. A feedforward, completely connected, back-propagation NN was designed to approximate the nonlinear yield function relating corn yield to the factors influencing yield. After the NN was developed and trained, three aspects of the input factors were investigated; (i) yield trends with four input factors, (ii) interaction between nitrogen application rate and late July rainfall and (iii) optimization of the 15 input factors with a genetic algorithm to determine maximum yield. They concluded that the back-propagation, feed-forward neural network had an accuracy of 80% when predicting corn yields. When an example with an abnormally low yield was discarded, the accuracy rose to 83.5%. They also found that the network could capture the expected interaction between rainfall and amount of applied nitrogen fertilizer. The application of neural network modeling fore prediction of milk yield in dairy sheep and used Pearson and rank correlations between predicted and observed values was conducted by Kominakis *et al.*[18]. The results showed that the difference between observed and predicted yields was generally statistically insignificant

The Malaysian Palm Oil Board (MPOB) provided us with a data set taken from two of the estates in Peninsular Malaysia. The factors included in the data set were foliar nutrient composition and Fresh Fruit Bunches (FFB) yield. The variables in foliar nutrient composition included the percentages of nitrogen, phosphorus, potassium, calcium and the magnesium concentration. The concentrations were considered as input nodes and the FFB yield as an output variable. These factors are considered as input nodes because, as the literature shows, these factors were to significantly affect to the oil palm yield[12,19]. The number observations for station S1 recorded 243 observations and station S2 recorded 527 observations. This study was conducted to use the neural network model in the modeling of oil palm yield. The effects of the activation function combination on the neural network performance will be investigated. Following this, the neural network performance will be compared with research done by Azme *et al.*[20]. Other objectives of this study were to investigate the effects of the number of hidden nodes, the number of runs, momentum terms and learning rate to the NN performance.

MATERIALS AND METHODS

Feedforward neural network: Neural network modeling becomes more popular technique in recent years and much research has been conducted on the application of artificial neural network in solving agriculture problems. NNs are cited as one of the most powerful computational tools ever developed. NN models operate like a black box, requiring no detailed information about the system. Instead, they assimilate the relationship between the input parameters and the controlled and uncontrolled variables by studying previous data or information. NNs can handle large and complex systems with many interrelated

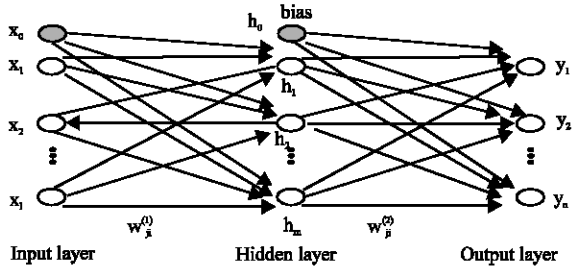

Fig. 1: Feed-forward neural network

parameters. They simply ignore any excess data of minimal significance and concentrate instead on the more important inputs. The detail discussion of the uses of neural network application for forecasting was carried out by Zuhaimy and Azme[13,14].

Several types of neural architectures are available; the most widely used being the multi-layer feed-forward back-propagation neural network. Figure 1 reveals a feed-forward network typically employs three or more layers for the architecture: an input layer, an output layer and at least one hidden layer. The computational procedure of this network is described below:

$$Y_j = f\left(\sum_{\forall i} w_{ij} X_{ij}\right), \quad (1)$$

where, Y_j is the output of node j, f(.) is the transfer function, w_{ij} is the connection weight between node $_j$and node i in the lower layer and X_i is the input signal from the node i in the lower layer.

The most popular and widely used algorithm is back propagation, popularised by Rumelhart and his group[21]. The back propagation algorithm is a gradient descent algorithm. The aim of this algorithm is to improve the performance of the neural network by changing the weights along its gradient, therefore reducing the total error. The back-propagation algorithm minimizes the square errors, which can be obtained by:

$$E = \frac{1}{2}\sum_{p}\sum_{j}\left(O_j^p - Y_j^p\right)^2 \quad (2)$$

where, E is the square errors, p is the index of pattern, O is the actual output and Y is the network output.

The back-propagation algorithm is based on a 'steepest descent' technique with a momentum weight/bias function, which calculates the weight change for a given neuron. It is expressed as follows: let $\Delta w^p_{ij}(n)$ denote the synaptic weight connecting the output of neuron i to the input of neuron j in the pth layer at iteration n. The adjustment $\Delta w^p_{ij}(n)$ to $w^p_{ij}(n)$ is given by:

$$\Delta w_{ij}^p(n) = -\eta(n)\frac{\partial E(n)}{\partial w_{ij}^p}, \quad (3)$$

where, $\eta(n)$ is the learning rate parameter. By using the chain rule of differentiation, the weight of the network with the back-propagation learning rule is updated using the following formulae:

$$\Delta w_{ij}^p(n) = \eta(n)\delta_j^p(n)X_i^{p-1}(n) + m(n)\Delta w_{ij}^p(n-1), \quad (4)$$

$$\Delta w_{ij}^p(n-1) = w_{ij}^p(n) + \Delta w_{ij}^p(n), \quad (5)$$

where, $\delta^p_{ij}(n)$ is the nth error signal at the jth neuron in the pth layer, $X^{p-1}_i(n)$ is the output signal of neuron i at the layer below and m is the momentum factor. The constant terms of η and m are specified at the start of the training cycle and determine the speed and stability of the network

Neural network procedure: The procedure to develop NN modeling requires two stages. These are important to ensure that the results and outcomes are valid and relevant. The two stages are data preparation and the calculation of the degree of freedom.

Data preparation: When the raw data has been collected, it may need converting into a more suitable format. At this stage, we should follow two stages; data validity checks and data partitioning.

Data validity checks: Data validity checks will reveal any patently unacceptable data that, if retained, would produce poor results. A simple data range check is an example of validity checking. For example, if we collected fresh fruit bunch data in tonnes per hectare per year, we would expect values above zero that do not exceed 40 ton/ha/year. A value of -5 or 100, for instance, is clearly wrong. If there is a pattern to the distribution of faulty data, then the pattern's caused is required to be diagnosed. Depending on the nature of the fault, we may need to discard the data altogether.

Data partitioning: Partitioning is the process of dividing the data into training sets, validation sets and testing sets. 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; and that (ii) there is no unwarranted similarity between data in different data sets.

Calculating the degree of freedom: A parametric analysis would be impossible without a discussion of the degree of freedom of the network. Because networks have historically been limited to nonparametric modeling, this topic has been conspicuously absent from the literature[22]. 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 [23,24]. 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 is the number of hidden neurons and n_i is number of input nodes. It is a necessary condition in any parametric model 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. 2, the input nodes are nitrogen concentration N, phosphorus concentration P, potassium concentration K, calcium concentration [Ca] and magnesium concentration [Mg]. The output node is FFB yield which is measured in ton per hectare per year.

We were required to transform the actual data into the range [0, 1] when the activation function was applied.

Thus we use formula $Z_j = \frac{(x_i - x_{min}) + 0.01}{(x_{max} - x_{min}) + 0.01}$ to transform the input and output value. The actual value can easily be transformed by manipulating the above equation to:

$$x_i = x_{min} + Z_i((x_{max} - x_{min}) + 0.01) - 0.01 \quad (9)$$

In this study, we used a randomized procedure to avoid the site plot effects.

Computer application: In this study we ran neural networks using the neural networks toolbox in Matlab package 6.15. Matlab's built in procedure provides a very simple neural networks program. The user only has to write a simple program to use the built-in neural network procedure[25]. Each procedure has its own specific name. 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 this case, we consider the fully connected feed-forward neural network and supervised neural networks because we have input and target data sets. We also assume that all the inputs had a significant influence on oil palm yield. We start the network with a small number of hidden nodes, which are increased one by one until the maximum number of hidden nodes, which is defined from Eq. 8, is reached. Thus, the maximum number of hidden nodes for stations S1 and S2 were found to be 34 and 75, respectively.

The first step in training a feed-forward network is to create the network object. The function newff creates a trainable feed-forward network[26]. The user should determine the transfer function in the first and second layers. 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 initialize. For feed-forward networks, the initialization of the weights is usually set as random (rands), this sets weights to random values between –1 and 1.

Once the network weights and biases have been initialized, the network is ready for training. The network can be trained for function approximation, pattern association or pattern classification. This study considers the training networks for function approximation. The training process requires networks input and outputs target. During training, the network's weights and biases are iteratively adjusted to minimize the network's performance function using mean squares error, MSE. The training algorithm that we used in this study is Leverberg-Marquardt (trainlm) because this algorithm

appears to be the fastest method for training moderately-sized feed-forward neural networks, in addition to being very efficient[26].

An early stopping technique used to terminate and avoid overtraining the neural network, therefore improving network generalization. This technique requires the data to be divided into three sets. 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 errors in the validation set are 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 in the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped and the weights and biases are returned to their values at the minimum of the validation error. We divided the data into, the training set, validation set and testing set using ratios of 70, 15 and 15%, respectively.

In brief, the procedures to set up a back-propagation network[27-29] are given below:

- Select input and define output variable.
- Determine layer(s) and the number of neurons in hidden layers: No hard rule is available for determining them and therefore this may depend on trial and error. The number of hidden nodes must satisfy that the degree of freedom must always be positive.
- Learning or training from historical data. Learning is the process by which a neural network modifies its weights in response to external inputs in order to minimize the global error. The equation that specifies this change is called the learning rule.
- Testing: When a neural network is well trained after learning, the neural networks are processed via a test set containing historical data that the network has never seen. If the testing results are in an acceptable range, the network can be considered as fully trained.

Experimental design: This study was conducted to investigate the influenced of the number of hidden nodes, the number of runs, the momentum term and the learning rate to the NN performance simultaneously using the same data that was analyzed before. We must first clarify how the experiment was designed. In the first stage, we considered three factors, which according to the literature should produce a significant effect on the NN performance, i.e. the number of hidden nodes HN, the number of runs NR and the 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 (0.25, 0.5, 0.75 and 0.95).

Experiment 1: This experiment considers three factors, namely the number of hidden nodes, the number of runs and the momentum term. The investigation was carried out by changing the level of one factor and assumed that the other two factors are fixed to run the networks. We then changed the factors from one level to next, recorded the error in estimation for each phase (training, validation and testing) and then calculated the average of the error. We used the correlation to measure the model's performance. As an example, suppose that the number of hidden nodes was set to three, the number of runs was also set at three and that the momentum term was 0.25. Now, we can write the experiment as [3, 3, 0.25]. The first column represents the number of hidden nodes (HN), the second column represents the number of runs (NR) and the last column 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.

Experiment 2: In the second experiment, we exchanged 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 the learning rate LR and could be written as [HN, NR, LR]. We considered the number of hidden nodes as having seven levels (2, 4, 6, 8, 10, 12 and 14) and levels of runs (3, 5, 7, 9, 11, 15 and 20). The learning rate had five levels (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 AND DISCUSSION

Statistical analysis: Here, we will discuss the effect of the combination activation function in the hidden layer and output layer on the NN performance. Six combination activation functions were considered, namely Logsigmoid and Logsigmoid (LL), Logsigmoid and Purelin (LP), Logsigmoid and Tansigmoid (LT), tansigmoid and purelin (TP), Tansigmoid and Logsigmoid (TL) and Tansigmoid and Tansigmoid (TT). We ran all the combinations and recorded the mean squares error for each number of hidden nodes. As mentioned before, the NN was divided

Table 1: The MSE, RMSE and MAPE values for each combination activation function

		The activation function combination (input to hidden layer and hidden layer to output layer)					
Station		LL	LP	LT	TP	TL	TT
S1:	MSE	13.1073	16.1588	18.3724	17.3731	16.4503	16.9258
	RMSE	3.6204	4.0198	4.2863	4.1681	4.0559	4.1141
	MAPE	0.1109	0.1318	0.1460	0.1443	0.1429	0.1456
S2:	MSE	10.9375	13.0501	12.5833	13.1762	12.7735	12.6515
	RMSE	3.3072	3.6125	3.5473	3.6299	3.5740	3.5569
	MAPE	0.1162	0.1357	0.1344	0.1306	0.1263	0.1223

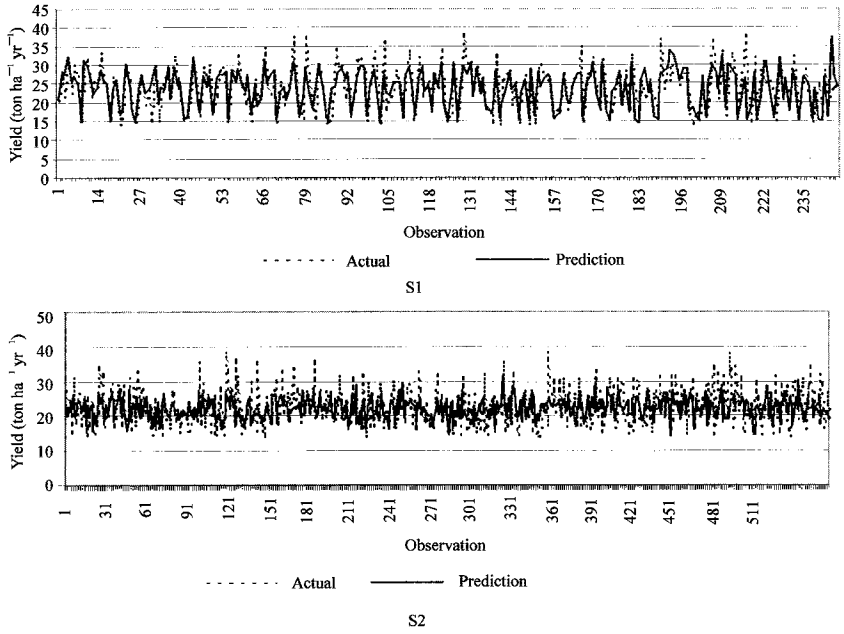

Fig. 2: The actual and predicted values for stations S1 and S2 using NN model

into three phases; training, validation and testing. We took the average of the MSE to study the overall performance of NN architecture. For the purpose of this study, we considered the MSE to be the testing variable of each phase.

Two hypotheses will be tested; the null hypothesis H_0 - all the MSE forming each combination activation function are equal versus the alternative hypothesis, H_a - at least one combination is not equal. As standard procedure, the F statistic was used to test the null hypothesis. The F statistic values for training, validation, training and average were recorded 3.36, 17.99, 12.05 and 10.73, respectively at station S1. At station S2, the F statistic values for training, validation, training and average were recorded 7.85, 15.51, 14.95 and 10.09, respectively. The tests show a significant value lower than the 5% level, which signifies the rejection of the null hypothesis. This study shows that the combination of activation functions has a significant influence on the overall performance of the NN model. 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 a good alternative in selecting the best combination

Neural network performance: The best models were selected based on the minimum average of MSE. For station S1, the LL combination activation function with 24 hidden nodes is the best model which recorded the MSE value as 0.0252 and the correlation coefficient was recorded as 0.7984 ($R^2 = 0.6374$). Whereas, the average of the MSE value as 0.0200 was recorded at station S2, when the LL activation combination function and 27 hidden nodes was considered. The LL combination at station S2 also produced a highest value (0.7840 or $R^2 = 0.6146$) The MSE values in the training phase are slightly lower compared to validation and testing phases. After the NN architecture was selected, the mean squares error MSE, root mean squares error RMSE, the mean absolute percentage error MAPE and each combination were calculated for stations S1 and S2. Table 1 shows the MSE, RMSE and MAPE values for station S1 and S2 for each combination activation function. Normally, looking at the model, which produces the minimum error, can make

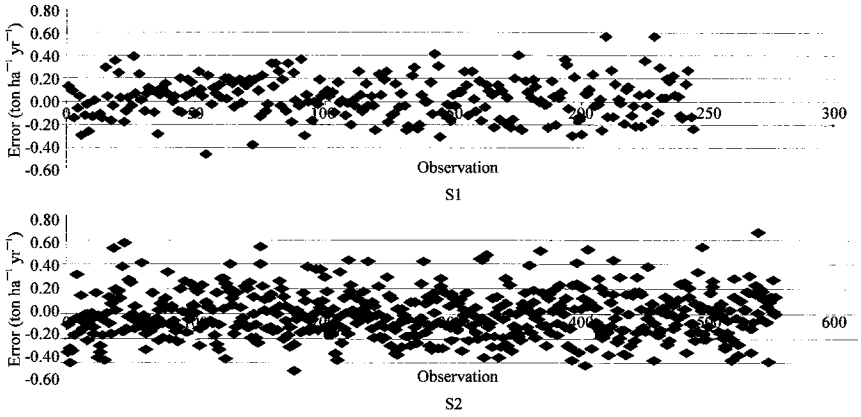

Fig. 3: The error distribution plots of the neural network model for stations S1 and S2

decisions. For both stations, LL combination activation function was recorded the minimum values of MSE, RMSE and MAPE. Figure 2 found that the NN model has the ability to fit the oil palm data quite well, as the predicted values are not too different from the actual values.

Error analysis: Next, the models were diagnosed using error analysis. The error analysis is performed to analyze the difference between the error values and the estimated values of observation. The scattered errors plot is important in deciding whether the error values are uniformly distributed, whether there is no systematic trend of the residual values, or whether the variance is consistent or not (Fig. 3). If the error plot shows that the errors have a homogeneous variance then the model is adequate to model the data. Figure 3 shows that the errors are scattered uniformly and so that the residual of variance is homogeneous. It is proven that the neural network model used in this study has the ability to examine the behaviors of the causality effect of foliar nutrient composition and FFB data.

Experiment 1 and 2: Experiment 1 found that the F value of hidden nodes is 8.7759 ($p = 0.0000$), indicated that the number of hidden nodes statistically affected the performance of the neural networks model. The F-value for the number of runs, 1.6950 ($p = 0.1330$) and the F-value for momentum term is 1.3300 ($p = 0.2630$). These values show that both factors did not influence the overall performance of the neural network. 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 on the neural network's performance.

In the experiment 2, we found that the F-value for the hidden nodes is 8.0480 ($p = 0.0000$) and the F-value for the number of runs is 2.8840 ($p = 0.0080$). This indicates that both factors affect the neural network's performance. However, the F value for the learning rate is 1.6090 ($p = 0.1540$), which means that the null hypothesis cannot be rejected. We therefore conclude that the learning rate does not influence the neural network's performance.

CONCLUSIONS

This experimental design was conducted to determine the effects of the neural network parameters (such as learning rate, momentum terms, number of hidden nodes and number of runs) on the neural network's performance. By using the analysis of variance test, it was found that the number of runs and the momentum term did not influence neural network performance in the first experiment. In the second experiment, the learning rate did not exert a statistically influence on the neural network's performance. However, the number of runs and the hidden nodes influenced the neural network's performance significantly. In both experiments, the influence of the number of hidden nodes on the neural network's performance was statistically significant at the 0.05 level.

Compared to the MLR model, the R^2 values produced from the NN model were increased by about 62.60% and 45.63%, from 0.392 to 0.6374 and from 0.422 to 0.6146, for Stations S1 and S2, respectively. This shows that the neural network model performs better than the regression model. The comparative study showed that the neural network model provided more reliable results when compared to both the MLR and RMR models. Hence, the neural network's performance is robust to the impact of outlier observations when compared to the linear regression model. The neural network model was introduced in this study and has previously never been used to model oil palm yield. We found that the modeling accuracy of the neural network is much better than that of

the multiple linear regression approach. The neural network model proved to be an efficient and reliable model.

REFERENCES

1. Hartley, C.W.S. 1977. The Oil Palm. Longman, London.
2. Green, A.H., 1976. Field Experiments as a Guide to Fertilizer Practice. In Corley, R.H.V., J.J. Hardon and B.J. Wood, (Eds.). Oil Palm Research: Developments in Crop Science (1). Elsevier Scientific Publishing Company, Netherlands.
3. Tarmizi, M.A., H.L. Foster, Z. Zin Zawawi and C.S. Chow, 1986. Statistical and economic analysis of oil palm fertilizer trials in Peninsular Malaysia between 1970-1981. PORIM Occasional Paper. No. 22.
4. Tarmizi, M.A., Z. Zin Zawawi, D. Mohd Tayeb, H.L. Foster, A.B. Hamdan and H. Khalid, 1991. Relative efficiency of urea to sulphate of ammonia in oil palm: Yield response and environmental factors. Proceedings of the 1991 PORIM International Palm Oil Conference-Agriculture, pp: 340-348.
5. Tarmizi, M.A., H. Abu Bakar, M.T. Dolmat and K.W. Chan, 1999. Development and validation of PORIM fertilizer recommendation system in Malaysian oil palm cultivation. Proceedings of the 1999 PORIM International Palm Oil Congress (Agriculture), pp: 203-217.
6. Foster, H.L., K.C. Chang, D.M. Tayeb, M.A. Tarmizi and Z. Zin Zakaria, 1985. Oil palm yield responses to N and K fertilizers in different environments in Peninsular Malaysia (PORIM occasional Paper No. 16). Palm Oil Research Institute of Malaysia Kuala Lumpur.
7. Foster, H.L., M. Ahmad Tarmizi and Z. Zin Zakaria, 1987. Foliar diagnosis of oil palm in Peninsular Malaysia. Proceedings of 1987 International Palm Oil Conference-Agriculture, pp: 249-261.
8. Chan, K.W., 1999. System approach to fertilizer management in oil palm. Proceedings of the 1999 PORIM International Palm Oil Congress - Agriculture, pp: 171-187.
9. Oboh, B.O. and M.A. B.Fakorede, 1999. Effects of weather on yield components of the oil palm in forest location in Nigeria. J. Oil Palm Res., 11: 79-89.
10. Henson, I.E., 2000. Modeling the effects of 'haze' on oil palm productivity and yield. J. Oil Palm Res., 12: 123-134.
11. Soon, B.B.F. and H.W. Hong, 2001. Oil palm responses to N, P, K and Mg fertilizers on two major soil types in Sabah. Proceedings of the 2001 PIPOC International Palm Oil Congress (Agriculture), pp: 318-334.
12. Foster, H., 2003. Assessment of oil palm fertilizer requirements. In Thomas Fairhust and Rolf Hardter, Oil Palm Management for Large and Sustainable Yields. PPI, PPIC and IPI.
13. Zuhaimy, I. and K. Azme, 2001. A review on combining neural network and genetic algorithm. Laporan Teknik/M. 5. Jabatan Matematik, Universiti Teknologi, Malaysia.
14. Zuhaimy, I. and K. Azme, 2002. A review on neural network and its application in forecasting. Laporan Teknik/M. 3. Jabatan Matematik, Universiti Teknologi Malaysia 12. Foster, H. 2003. Assessment of oil palm fertilizer requirements. . In Thomas, F. and R. Hardter, (Eds.). Oil Palm Management for Large and Sustainable Yields. PPI, PPIC and IPI.
15. Shearer, S.A., T.F. Burks, J.P. Fulton, S.F. Higgins, J.A. Thomasson, T.G. Mueller and S. Samson, 1994. Yield prediction using a neural network classifier trained using soil landscape features and soil fertility data. Agron. J., 89: 54-59.
16. Drummond, S.T., K.A. Sudduth and S.J. Birrell, 1995. Analysis and correlation methods for spatial data. ASAE Paper No. 95-1335. St. Joseph, Mich.: ASAE
17. Liu, J., C.E. Goering and L. Tian, 2001. A neural network for setting target corn yields. Trans. Am. Soc. Agric. Eng., 44: 705-713.
18. Kominakis, A.P., Z. Abas, I, Maltaris and E. Rogdakis, 2002. A preliminary study of the application of artificial neural networks to prediction of milk yield in dairy sheep. Computers and Electronics in Agriculture, 35: 35-48.
19. Fairhurst, T.H. and E. Mutert, 1999. Interpretation and management of oil palm leaf analysis data. Better Crops Intl., 13: 48-51.
20. Azme, K., I. Zuhaimy, H. Khalid and M.A. Tarmizi, 2005. Modeling oil palm yield using multiple linear regression and robust M-regression, (In Press).
21. Rumelhart, D.E., G.E. Hinton and R.J. Williams, 1986. Learning Internal Representation by Back-propagating Errors. In: Rumelhart, D.E., J.L. McCleland and the PDP Group, (Eds.). Parallel Processing: Explorations in the Microstructure of Cognation. MA. MIT Press.
22. Ripley, B.D., 1994. Neural networks and flexible regression and discrimination. Advances in Applied Statistics, pp: 39-57.
23. Gujarati, D.N., 1988. Basic Econometrics. New York, McGraw-Hill.
24. Adam, J.B., 1999. Predicting pickle harvest using a parametric feedforward neural network. J. Applied Stat., 26: 165-176.

25. Demuth, H. and M.H. Beale, 1998. Neural Network Toolbox: User's Guide. The Mathwork Inc.
26. Hagan, M.T., H.B. Demuth and M.H Beale, 1996. Neural Network Design. PWS Publishing Company, Boston.
27. Patterson, D.W., 1996. Artificial Neural Networks: Theory and Applications. Prentice Hall, Singapore.
28. Lai, L.L., 1998. Intelligent System Applications in Power Engineering: Evolutionary Programming and Neural Networks. John Wiley and Sons. West Sussex, England.
29. Hykin, S., 1999. Neural Networks: A Comprehensive Foundation. 2nd Edn., Prentice Hall, New Jersey.