

How to cite this article:

Ahmarofi, A. A., Ramli, R., Abidin, N. Z., Jamil, J. M., & Shaharane, I. N. (2020). Variations on the number of hidden nodes through multilayer perceptron networks to predict the cycle time. *Journal of Information and Communication Technology*, 19(1), 1-19.

VARIATIONS IN THE NUMBER OF HIDDEN NODES THROUGH MULTILAYER PERCEPTRON NETWORKS TO PREDICT CYCLE TIME

**¹Ahmad Afif Ahmarofi, ²Razamin Ramli, ²Norhaslinda Zainal Abidin,
²Jastini Mohd Jamil & ²Izwan Nizal Shaharane**

¹*Fakulti Pengurusan Industri, Universiti Malaysia Pahang, Malaysia*

²*School of Quantitative Sciences, Universiti Utara Malaysia, Malaysia*

*aaafif@ump.edu.my; razamin@uum.edu.my; nhaslinda@uum.edu.my;
jastini@uum.edu.my; nizal@uum.edu.my*

ABSTRACT

Multilayer Perceptron Network (MLP) has a better prediction performance compared to other networks since the structure of the MLP is suitable for training processes in solving prediction problems. However, to the best of our knowledge, there is no rule of thumb in determining the number of hidden nodes within the MLP structure. Researchers normally test with various numbers of hidden nodes to obtain the lowest square error value for optimal prediction results since none of the approaches has yet to be claimed as the best practice. Thus, the aim of this study is to determine the best MLP network by varying the number of hidden nodes of developed networks to predict cycle time for producing a new audio product on a production line. The networks were trained and validated through 100 sets of production lots from a selected audio manufacturer. As a result, the 3-2-1 MLP network was the best network based on the lowest square error value compared to the 3-1-1 and 3-3-1 networks. The 3-2-1 predicted the best cycle time of 5 seconds to produce a new audio product. Hence, the prediction result could facilitate production planners in managing assembly processes on the production line.

Keywords: Artificial neural networks, multilayer perceptron, hidden node, cycle time, production line.

INTRODUCTION

Artificial Neural Network (ANN) is a brain metaphor method (Wilson & Sharda, 1992) and a non-linear data driven to solve prediction problems through a learning process (Kumar, 2013). A learning process in ANN is conducted through a network structure to solve prediction problems. A network is an arrangement of interrelated connections in a system (Ahmarofi, Ramli, & Abidin, 2017; Al-Nuaimi & Abdullah, 2017). The network for ANN model is classified into two major structures which are feed-forward multilayer perceptron (MLP) and recurrent network (Kumar, 2013). However, the MLP network has shown superior capability in solving prediction problems compared to the recurrent network due to the MLP having well-structured layers for reaching a desired target (Esfe, Afrand, Wongwises, Naderi, Asadi, Rostami, & Akbari, 2015; Samarasinghe, 2016; Priambodo & Ahmad, 2018).

In the structure of an MLP network, input is transferred from input layer to output layer through a hidden layer (Haykin, 2009; Kumar, 2013). The transmission of input from one layer to another is through a connection which is termed as a neuron (Azadeh, Kosar, Shoushtari, & Ebrahim, 2014). The structure of an MLP network is illustrated in Figure 1 as follows.

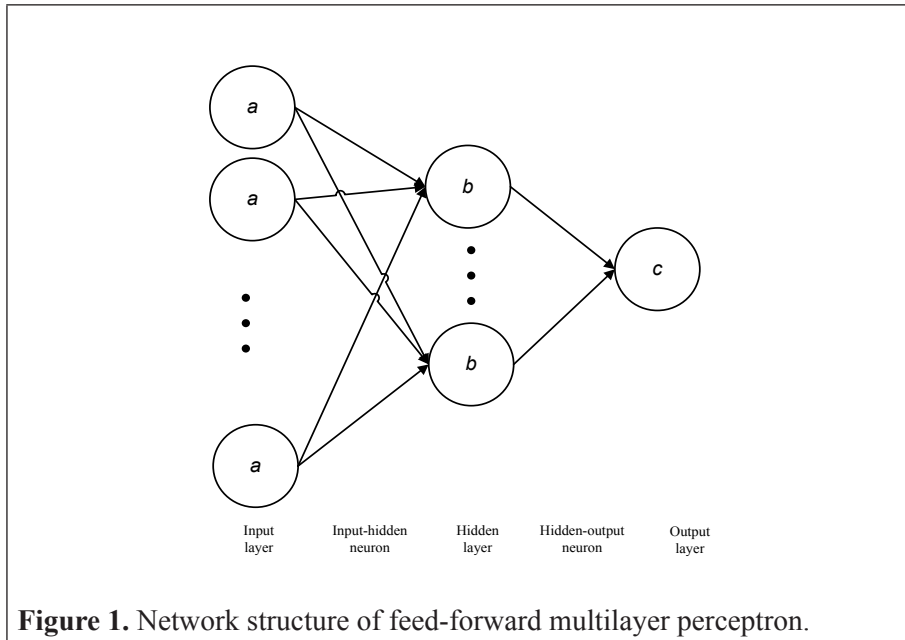


Figure 1. Network structure of feed-forward multilayer perceptron.

where,

a is the number of input nodes at the input layer where $a = 1, 2, \dots, A$,

b is the number of hidden nodes at the hidden layer where $b = 1, 2, \dots, B$,

c is the number of output nodes at the output layer where $c = 1, 2, \dots, C$

Each of the layers comprises several nodes. A node is a processing centre of input for each layer (Samarasinghe, 2016). The input at the node is processed through Backpropagation (BP) learning algorithm. The BP algorithm is suitable to train data in the MLP network since BP is a supervised learning method that has the capability of mapping an input to an output (Dzakiyullah, 2015; Ibrahim, Shamsuddin, & Qasem, 2015; Turban, Sharda, & Delen, 2011). Moreover, various numbers of hidden nodes within an MLP network has been experimented by researchers to obtain the best network for solving prediction problems. However, none of the approaches has been claimed as the best practice.

In order to develop the MLP network, a case problem based on a real company situation was examined in this study. The company is a global business manufacturer for audio products to meet customer requirements in the latest technology. The company was facing an issue with the uncertain cycle time of new audio products. Consequently, the completion time of the new products was affected as the production site was unable to fulfil customer delivery requirements on time.

The objective of this study is to determine the best MLP network to predict the cycle time of a new audio product through the establishment of single input, hidden and output layers while the number of hidden nodes, b , varies based on small datasets, which is 100 datasets. Previous work related to cycle time and variations in the number of hidden nodes in an ANN model are discussed in the literature review section. This is followed by the methodology of the study, related results, the conclusion and future work.

LITERATURE REVIEW

In a production system, cycle time is the time needed to process a product with a specific task on a production line (Ahmarofi, Abidin, & Ramli, 2017a; Seth, Seth, & Dhariwal, 2017). Therefore, the determination of cycle time is crucial to avoid the postponement of a job's completion time, also known as tardiness (Schafer, Chankov, & Bendul, 2016).

Nevertheless, the uncertainty of cycle time which is dependent on: the number of manpower, material preparation and machine capacity are difficult to predict (Ahmarofi et al., 2017b). From previous studies, the prediction problem is solved through three main predictive methods which are regression analysis, decision trees and ANN (Jamil & Shaharane, 2015). It has been found that the regression analysis method showed low performance for data mining processes (Carbonneau, Vahidov, & Laframble, 2007; Turban et al., 2011) while decision tree is only best implemented when the number of classes are low (Wang, 2007) and for classification purposes (Chien, Wang, & Cheng, 2007). However, in many cases, ANN has proven its capability in solving prediction problems due to its capability to assimilate the relationship between input and output (Azadeh et al., 2014; Haykin, 2009; Mehrjerdi & Aliheidary, 2014). Several researchers have implemented the MLP in their studies and have successfully solved prediction problems in production operations (Karam, Centobelli, Doriana, Addona, & Teti, 2016; Poonia, Soni, & Khanam, 2016; Vrabel, Mankova, & Beno, 2016).

Moreover, the number of hidden nodes has been investigated by researchers in their studies to find the optimal MLP structure. In a study by Wanas, Auda, Kamel, and Karray (1998), they claimed that the best performance of a network was when the number of hidden nodes was equal to the logarithm of number for training sample. The results showed that the higher the number of hidden nodes, the better the performance of the network. They found that 15 hidden nodes showed almost similar performance with 11 hidden nodes for a 2000 sample data. However, the recommendation by Wanas et al. (1998) differed from Turban et al. (2011) as the latter posited that the number of hidden nodes could be set based on the result of the summation between the numbers of input with output nodes and divided by two.

Despite this, Leung, Lam, Ling and Tam (2003) highlighted that a large network may have some of its connections redundant. Besides, the authors suggested that the number of hidden nodes should be chosen manually starting from a small number and then increased gradually if the learning performance in terms of fitness value was not acceptable. Based on their experiment to predict sunspots, the optimal number of hidden nodes was six for 288 datasets. On the other hand, even though both of these studies used the same datasets as Zhang (2003), their findings contradicted. It was found that the optimal number of hidden nodes was four instead of six.

Furthermore, Huang, Chen, and Siew (2006) emphasized that the number of hidden nodes should be selected at random through trial and error. The results showed that 100 hidden nodes had better performance than 200 nodes for 15000 data without interrupting the learning process. Their finding was supported by Feng, Huang Ling, and Gay (2009). In their study based on

the same method as Huang et al. (2006), Feng et al. (2009) found that 21 nodes was the optimal hidden node number for 209 datasets.

Based on recent studies, to the best of our knowledge, no method has been claimed as the best practice in determining the number of hidden nodes as related studies determined the number of hidden nodes through trial and error. For example, in the study by Esfe et al. (2015) in predicting thermal conductivity, they found that a network with two hidden layers and five neurons in each layer had the lowest error and highest fitting coefficient for 30 datasets. Moreover, in the study by Lekamalage, Song, Huang, Cui, and Liang (2017) in experimenting with the image classification of 24300 samples, there were two hidden layers with the first hidden node consisting of 200 nodes, while the second hidden layer which consisted of 3000 nodes was found to have better testing accuracy. Besides, in a study by Zhang (2017) in recognizing and predicting mental disease based on 10000 datasets, the best mean square error value was recorded when the number of hidden nodes increased from eight to 16 nodes. Furthermore, in a study by Zhang, Zuo, Gao, and Zhao (2017) to recognize the image of 1000 digital instruments, they found that 18 hidden nodes was the optimal number of hidden nodes.

Therefore, it has been noted that there is no rule of thumb in determining the number of hidden nodes based on the researchers' experiments with various numbers of hidden nodes for optimal results. This finding is aligned with Turban et al. (2011) as they observed that the determination of the number of hidden nodes was rather tricky. Despite this, the suggestion by Turban et al. (2011) is more practical for small datasets as considered in this research to obtain the best MLP structure for predicting the best cycle time and this is elaborated in the following section.

METHODOLOGY

This section describes the case study, data collection and development of the MLP.

Case Study

A manufacturer of audio products was selected as a case company. The cycle time of a new product on the production line of the company was difficult to determine as it was dependent on: the number of manpower, material preparation time and machine breakdown rate. Thus, the best cycle time was predicted based on 100 datasets of pre-production test run.

Data Collection

The data recorded in the production daily report was collected from the manufacturer. The data was recorded at the production site from production lot $n = 1$ until $n = 100$ based on the 100 lots of pre-production test of a new audio product. The number of manpower ($manpower_n$), waiting time material ($material_n$), and machine breakdown rate ($machine_n$) were considered as input to predict cycle time ($cycle_n$).

Development of MLP Network Structure

The development of the MLP network was based on the number of input node-the number of hidden node-the number of output node, $a-b-c$. Subsequently, the connection weights were varied for each of the MLP network to find the best network based on the smallest square error value. Thus, the development of the MLP network based on the ANN method is elaborated in Figure 2 as follows.

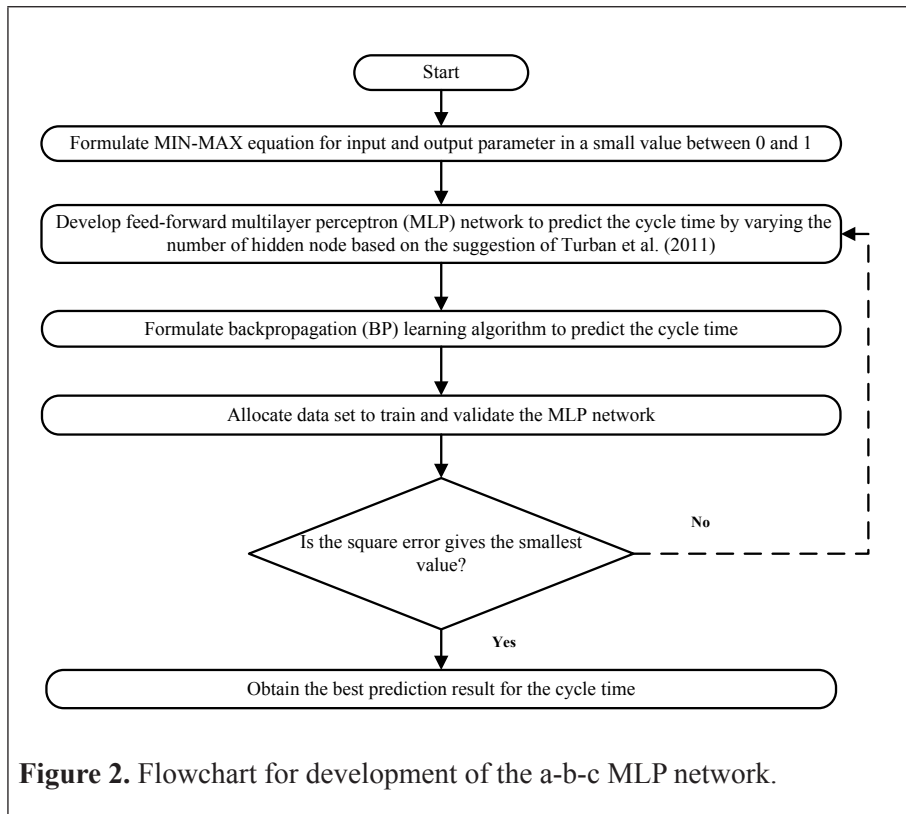


Figure 2. Flowchart for development of the a-b-c MLP network.

With reference to Figure 2, the parameters of $manpower_n$, $material_n$, $machine_n$, and $cycle_n$ are transformed to a small value in the interval of [0, 1] through MIN-MAX equation, hence the ANN learning process is smooth. The formulation of MIN-MAX equation to transform input and output parameters is expressed in Equations 1, 2 and 3:

$$input_{transformed} = \frac{input_{ori} - input_{min}}{input_{max} - input_{min}} \quad (1)$$

where,

$input_{transformed}$ = the transformation value for input, i.e., *manpower, material and machine,*
 $input_{ori}$ = the original value for *input,*
 $input_{min}$ = the minimum value for *input,*
 $input_{max}$ = the maximum value for *input.*

$$output_{transformed} = \frac{output_{ori} - output_{min}}{output_{max} - output_{min}} \quad (2)$$

where,

$output_{transformed}$ = the transformation value for *cycle,*
 $output_{ori}$ = the original value for *output,*
 $output_{min}$ = the minimum value for *output,*
 $output_{max}$ = the maximum value for *output.*

Subsequently, the feed-forward MLP network is developed for predicting the cycle time of a new audio product. In this research, the structures of the MLP network are established by varying the number of hidden nodes based on the suggestion by Turban et al. (2011) that the number of hidden nodes could be set as follows:

$$h_{hidden} = \frac{i_{in} + j_{out}}{2} \quad (3)$$

where,

- h_{hidden} = the total number of hidden nodes where $h = 1, 2, \dots, H$,
- i_{in} = the total number of input nodes where $i = 1, 2, \dots, I$,
- j_{out} = the total number of output nodes where $j = 1, 2, \dots, J$.

Since the total number of input nodes is three, i.e., $manpower_n$, $material_n$, $machine_n$, and the total number of output node is one, i.e., $cycle_n$, the total number of hidden nodes is set to two nodes, i.e., 3-2-1. However, two network structures are also developed at this stage, which are the 3-1-1 and 3-3-1 networks to explore the best MLP network in predicting the cycle time.

Furthermore, BP learning algorithm is established to guide the networks during the learning process in predicting cycle time. The learning process of BP is initialized with connection weight, w , as a relative strength. The relative strength of input-hidden neuron and hidden-output neuron (Figure 1) is represented by w_{ab} and w_{bc} , respectively. However, the value of connection weights is initially set to a random value as recommended by Samarasinghe (2016) since there is no restriction of formulating connection weights. The initialized connection weights are subsequently inserted into the summation function (Equation 4) and sigmoid function (Equation 5). Summation function is a weighted sum of connection weights and transformed parameters while sigmoid function is a non-linear function to convert the summation function value into an S-shaped interval $[0, 1]$. The formulations of summation function and sigmoid function for hidden nodes and output nodes are expressed as follows:

$$sum_b = \sum_{a=1}^A w_{ab} input_{transformed} \quad (4)$$

where,

- sum_b = the weighted sum of b th hidden nodes of connection weights and transformed values,
- w_{bc} = the connection weights for the a th input nodes and b th hidden nodes,
- sig_b = the transformed value of input.

$$sig_b = \frac{1}{(1 + e^{sum_b})} \quad (5)$$

where,

sig_b = the sigmoid value of b th hidden nodes,
 sum_b = the weighted sum of b th hidden nodes,
 e = the base of natural logarithm, i.e., 2.71828.

$$sum_c = \sum_{b=1}^B w_{bc} sig_b \quad (6)$$

where,

sum_c = the weighted sum of c th output nodes of connection weight and transformed value,
 w_{bc} = the connection weight for the b th hidden nodes and c th output nodes,
 sig_b = the sigmoid value of b th hidden nodes.

$$sig_c = \frac{1}{(1 + e^{sum_c})} \quad (7)$$

where,

sig_c = the sigmoid value of c th output nodes,
 sum_c = the weighted sum of c th output nodes,
 e = the base of natural logarithm, i.e., 2.71828.

Moreover, the learning rate and momentum rate are randomly set to the value within the interval $[0,1]$ as recommended by Shiang (2009) and Turban et al. (2011). There is no restriction on the selection of suitable values for both the learning rates as it is commonly based on experiments with various different values.

Subsequently, related data is allocated between training and validation sets to train and validate the three structures of the MLP network, respectively. At this stage, 80 percent of the data are allocated for training set while 20 percent of the remaining data are allocated for validation set. By assigning more percentage of data for the training set, the MLP network would provide better prediction results since the more the data is being trained, the stronger its predictive relationship during the learning process as recommended by Ahmarofi et al., (2017b).

Consequently, the learning process is run through BP learning algorithm within the MLP network to predict cycle time. The best predicted cycle time in producing the new audio speaker is obtained based on the smallest square error. The square error is calculated based on Equation 8:

$$E_o = \frac{1}{2} ([sig_c]_o - output_{transformed})^2 \quad o=1,2,\dots,O, \quad (8)$$

where,

- E_c = the square error of o th iteration,
- sig_c = the sigmoid value of c th output node i.e. the MLP network output value,
- $output_{transformed}$ = the transformed value of $output$ i.e. desired output value.

Consequently, the development of the MLP networks, i.e., 3-1-1, 3-2-1 and 3-3-1, with the established BP learning algorithm are presented in the following: Figure 3, Figure 4 and Figure 5, respectively, by varying the number of hidden nodes based on Equation 3 to explore the best MLP network in predicting cycle time.

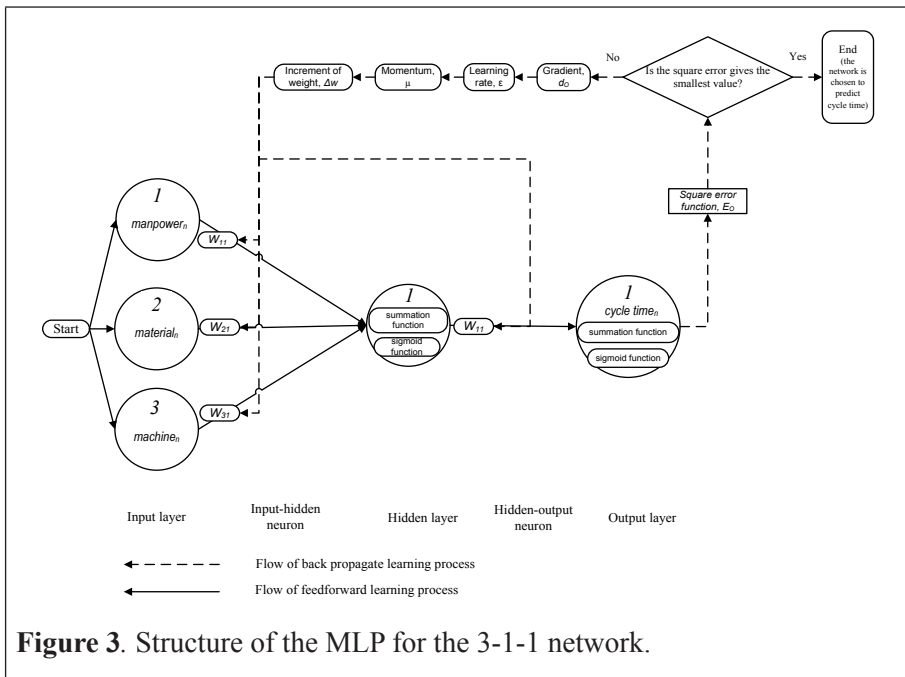


Figure 3. Structure of the MLP for the 3-1-1 network.

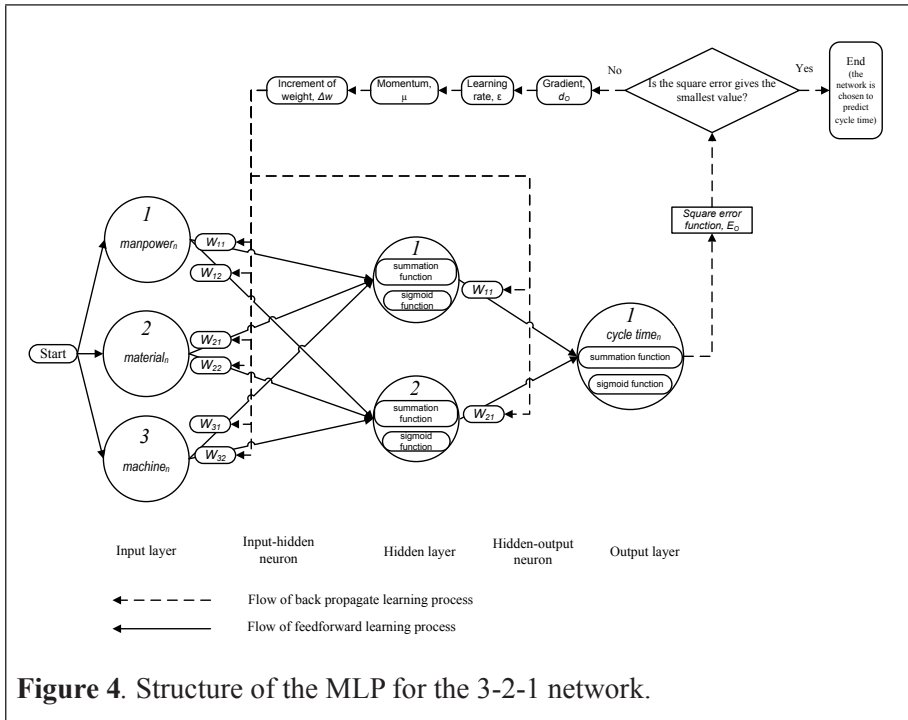


Figure 4. Structure of the MLP for the 3-2-1 network.

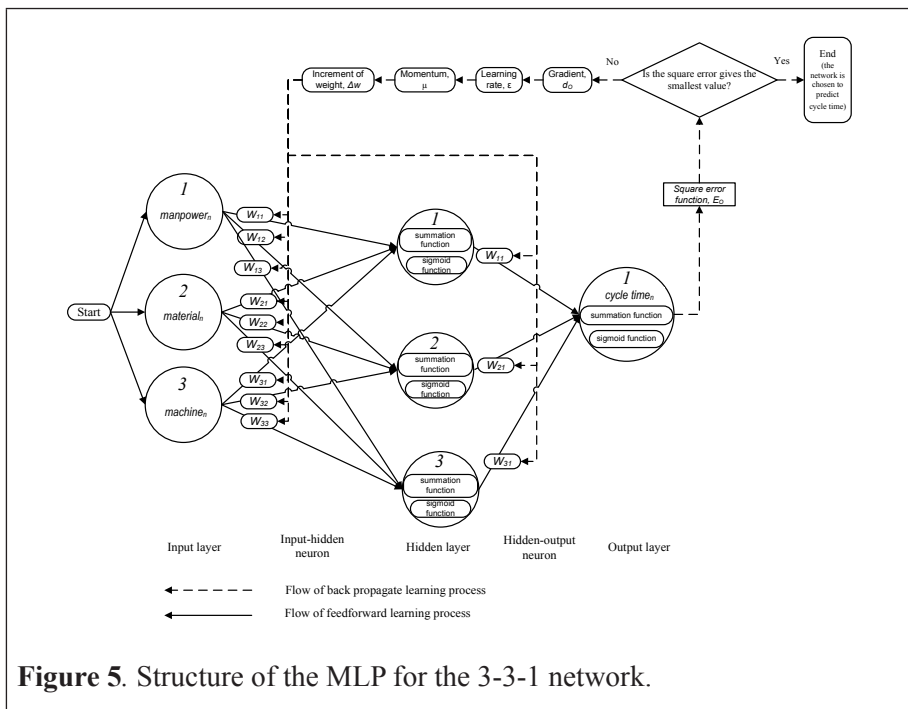


Figure 5. Structure of the MLP for the 3-3-1 network.

As a result, the equation of the predicted cycle time is expressed in Equation 9:

$$y_{predict} = \sum_{a=1}^A [w_{ab}]_o \mathit{input}_{transformed} + \sum_{b=1}^B [w_{bc}]_o \mathit{sig}_b \quad (9)$$

where,

- $y_{predict}$ = the predicted output value,
- $[w_{ab}]_o$ = the final connection weight for a th input node and b th hidden node of o th iteration,
- $\mathit{input}_{transformed}$ = the transformed value of input ,
- $[w_{bc}]_o$ = the final connection weight for b th hidden node and c th output node of o th iteration,
- sig_b = the sigmoid function value of b th hidden node.

RESULTS AND DISCUSSION

Three sets of experiments were conducted for the developed MLP networks, i.e., 3-1-1, 3-2-1, and 3-3-1. Besides, the established BP learning algorithm was implemented to investigate the three different MLP networks on their error values in terms of square error function, E_o , i.e., between the output value of the network and the desired output of cycle time. The value for connection weights between a th input nodes and b th hidden nodes, w_{ab} , and b th hidden nodes to c th output nodes, w_{bc} , were set to random values since the initial value of connection weights have no restriction in a learning process. Therefore, 0.1, 0.3, 0.5, 0.7, 0.9, 1 and 1.5 were randomly selected as the values of connection weights, i.e., w_{ab} and w_{bc} in this research.

In addition, the value of learning rate, ϵ , and momentum rate, μ , were set within the range of 0.1 to 1.0 during the learning process since both parameters were commonly based on iterative experiments and no heuristic method had been claimed as the best one. Thus, the value of ϵ and μ were set to 0.2 and 0.5, respectively, for all experiments. Furthermore, the data from

production lot $n = 1$ until $n = 100$ was separated into 80 % (80 production lots) and 20 % (20 production lots) between the training set, *train*, and validation set, *valid*, for each of the MLP network based on the work by Ahmarofi et al. (2017b) as the paper successfully predicted the completion time of producing a new audio product. The more the data was assigned for training processes, the stronger its predictive link as recommended by Turban et al. (2011) and Jamil and Shaharane (2015). Finally, the iteration of learning process, o , for each of the developed MLP network was terminated once the result of square error, E_o , provided the smallest error value.

The values of final E_o for the first experiment of the 3-1-1 network are presented in Table 1 as follows:

Table 1

The E_o of the 3-1-1 Network

MLP network	Connection weight		Learning rate	Momentum rate	Separation of data from production lot number		Iteration	Square error
	w_{ab}	w_{bc}	ϵ	μ	<i>train</i>	<i>valid</i>	o	E_o
3-1-1 network	0.1	0.1	0.2	0.5	80%	20%	58	0.0164
	0.3	0.3					60	0.0373
	0.5	0.5					61	0.0671
	0.7	0.7					59	0.0934
	0.9	0.9					63	0.1898
	1	1					67	0.0676
	1.5	1.5					74	0.0786

From Table 1, the smallest E_o for the 3-1-1 MLP network is 0.0164 during the 58th iteration of the learning process which is obtained through the input-hidden node connection weight, $w_{ab} = 0.1$ and the hidden-output node connection weight, $w_{bc} = 0.1$.

The values of final E_o for the second experiment of the 3-2-1 network are presented in Table 2 as follows:

Table 2

The E_o of the 3-2-1 Network

MLP network	Connection weight		Learning rate	Momentum rate	Separation of data from production lot number		Iteration	Square error
<i>a-b-c</i>	w_{ab}	w_{bc}	ε	μ	<i>train</i>	<i>valid</i>	<i>o</i>	E_o
	0.1	0.1					65	0.0367
	0.3	0.3					65	0.0354
	0.5	0.5					66	0.0589
3-2-1 network	0.7	0.7	0.2	0.5	80%	20%	78	0.0743
	0.9	0.9					56	0.0945
	1	1					45	0.0234
	1.5	1.5					78	0.0156

Based on the results in Table 2, the smallest E_r_o for the 3-2-1 MLP network is 0.0156 during the 78th iteration of the learning process as obtained with $w_{ab} = 1.5$ and $w_{bc} = 1.5$.

The values of the final E_o for the third experiment of the 3-3-1 MLP network are presented in Table 3 as follows:

Table 3

The E_o of the 3-3-1 Network

MLP network	Connection weight		Learning rate	Momentum rate	Separation of data from production lot number		Iteration	Square error
<i>a-b-c</i>	w_{ab}	w_{bc}	ε	μ	<i>train</i>	<i>valid</i>	<i>o</i>	E_o
	0.1	0.1					49	0.0290
	0.3	0.3					58	0.0458
	0.5	0.5					66	0.0786
3-3-1 network	0.7	0.7	0.2	0.5	80%	20%	74	0.0456
	0.9	0.9					50	0.0234
	1	1					48	0.0453
	1.5	1.5					84	0.0734

From Table 3, the smallest E_o for the 3-3-1 MLP network is 0.0234 during the 50th iteration of the learning process which resulted from $w_{ab} = 0.9$ and $w_{bc} = 0.9$.

Based on the results in Table 1, Table 2 and Table 3, the smallest value of E_o is 0.0156 which is obtained from the 3-2-1 network with $w_{ij} = 100$, $w_{jk} = 100$ during the iteration, $o = 78$ as described in Table 2. Thus, the 3-2-1 network is the best MLP network based on the smallest E_o value, i.e., 0.0156 for predicting cycle time compared to the 3-1-1 and 3-3-1 networks.

Therefore, the 3-2-1 MLP network is selected for predicting the cycle time of the new audio product on the semiautomatic production line. Thus, the equation of predicted product cycle time during the 121st production lot based on Equation (9) with the respective final connection weights during the 78th iteration is expressed as follows:

$$y_{predicted} = [0.161]_{78}input_{manpower_{121}} + [0.654]_{78}input_{material_{121}} + [0.246]_{78}input_{machine_{121}} + [0.201]_{78}sig_1 + [0.0146]_{78}sig_2$$

Since the company would like to predict the cycle time of a new audio product for the next production lot, $n = 121$ with available number of manpower, $manpower_{121} = 30$ persons, waiting time of material, $material_{121} = 1$ hour and machine breakdown rate, $machine_{121} = 0.0013$, the best predicted cycle time of the new audio product, $cycle_{121}$ is 5 seconds based on Equation (9). Therefore, the best predicted cycle time for the new audio product during the 121st production lot, based on the transformed input values for the corresponding $manpower_{121}$, $material_{121}$ and $machine_{121}$, are presented in Table 4 as follows:

Table 4

The best predicted cycle time with respective input parameters for production lot, $n = 121$

Input variable	Original input parameter	Transformed input value	Transformed output value for predicted cycle time based on Equation (2)	The best predicted cycle time in original output parameter
			$y_{predicted}$	$cycle_{121}$
$manpower_{121}$	30 persons	$input_{manpower_{121}}$	1	5 seconds
$material_{121}$	1 hour	$input_{material_{121}}$	0.333	
$machine_{121}$	0.0013	$input_{machine_{121}}$	0	

CONCLUSION AND FUTURE WORK

The aim of this study was to find the best MLP network by varying the number of hidden nodes. Based on previous studies, it has been found that there is no rule of thumb in determining the number of hidden nodes. However, the best MLP structure was determined through variations in the number of hidden nodes as suggested by Turban et al. (2011) as it was more practical for small datasets as was considered in this research. Hence, three MLP networks were established with different numbers of hidden nodes, i.e., 3-1-1, 3-2-1, 3-3-1 and subsequently experimented through a developed BP learning algorithm. The 3-2-1 network was found to be the best network based on the smallest value of square error, i.e., 0.0156 with a connection weight of 100 during the 78th iteration. The 3-2-1 as the best MLP network is aligned with the suggestion by Turban et al. (2011). Therefore, the 3-2-1 network was selected with a cycle time of 5 seconds for producing a new audio product. For future work, the number of hidden nodes for the MLP structure could be determined by integrating ANN with other artificial intelligence (AI) methods such as Evolutionary Algorithm and Ant Colony Algorithm instead of at random since AI is currently gaining much attention from researchers.

ACKNOWLEDGEMENT

This work was supported by MyBrain15 (MyPhD) postgraduate scholarship programme, Ministry of Education, Malaysia.

REFERENCES

- Ahmarofi, A. A., Abidin, N. Z., & Ramli, R. (2017a). Effect of manpower factor on semiautomatic production line completion time: A system dynamics approach. *Journal of Mechanical Engineering and Sciences*, *11*(2), 2567–2580.
- Ahmarofi, A. A., Ramli, R., & Zainal Abidin, N. (2017b). Predicting completion time for production line in a supply chain system through artificial neural networks. *International Journal of Supply Chain Management*, *6*(3), 82–90.
- Al Nuaimi, Z. N. A. M., & Abdullah, R. (2017). Neural network training using hybrid particle-move artificial bee colony algorithm for pattern classification. *Journal of Information & Communication Technology*, *16*(2), 314–334.

- Azadeh, A., Kosar, D., Shoushtari, M. S., & Ebrahim, T. (2014). An integrated artificial neural network and system dynamics approach in support of the viable system model to enhance industrial intelligence: The case of a large broiler industry. *Systems Research and Behavioral Science*, 31(2), 236–257.
- Carbonneau, R., Vahidov, R., & Laframboise, K. (2007). Machine learning-based demand forecasting in supply chains. *International Journal of Intelligent Information Technologies*, 3(4), 40–57.
- Chien, C. F., Wang W. C., & Cheng, J. C. (2007). Data mining for yield enhancement in semiconductor manufacturing and an empirical study. *Expert Systems with Applications*, 33, 192–198.
- Dzakiyullah, R. N. R. (2015). *Production quantity estimation using improved artificial neural networks* (Unpublished dissertation). Universiti Teknikal Malaysia, Melaka.
- Esfe, M. H., Afrand, M., Wongwises, S., Naderi, A., Asadi, A., Rostami, S., & Akbari, M. (2015). Applications of feedforward multilayer perceptron artificial neural networks and empirical correlation for prediction of thermal conductivity of Mg (OH) 2–EG using experimental data. *International Communications in Heat and Mass Transfer*, 67, 46–50.
- Feng, G., Huang, G. B., Lin, Q., & Gay, R. (2009). Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Transactions on Neural Networks*, 20(8), 1352–1357.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. New Jersey: Prentice Hall.
- Huang, G. B., Chen, L., & Siew, C. K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4), 879–892.
- Ibrahim, A. O., Shamsuddin, S. M., & Qasem, S. N. (2015). Hybrid NSGA-II optimization for improving the three-term bp network for multiclass classification problems. *Journal of Information & Communication Technology*, 14.
- Jamil, J. M., & Shaharane, I. N. M. (2015). *A Quick Guide for Data Mining: Predictive and Descriptive Modeling*. Sintok: UUM Press.
- Karam, S., Centobelli, P., Doriana, M., Addona, D., & Teti, R. (2016). Online prediction of cutting tool life in turning via cognitive decision making. *Procedia CIRP*, 41, 927–932.

- Kumar, S. (2013). *Neural Networks: A Classroom Approach*. New Delhi: McGraw Hill.
- Lekamalage, C. K. L., Song, K., Huang, G. B., Cui, D., & Liang, K. (2017). Multilayer multi objective extreme learning machine. In *Image Processing IEEE International Conference*, 1297–1301.
- Leung, F. H. F., Lam, H. K., Ling, S. H., & Tam, P. K. S. (2003). Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural networks*, 14(1), 79–88.
- Mehrjerdi, Y. Z., & Aliheidary, T. (2014). System dynamics and artificial neural networks integration: A tool to evaluate the level of job satisfaction in services. *International Journal of Industrial Engineering and Production Research*, 25(1), 13–26.
- Poonia, A. K., Soni, A. B., & Khanam, S. (2016). Optimization of operating parameters for sponge iron production process using neural network. *International Journal of ChemTech Research*, 9(2), 20–34.
- Priambodo, B., & Ahmad, A. (2018). Traffic flow prediction model on neighbouring roads using neural networks and multiple regression. *Journal of Information & Communication Technology*, 17(4), 513–535.
- Samarasinghe, S. (2016). *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*. New York: Auerbach Publication.
- Schäfer, R., Chankov, S., & Bendul, J. (2016). What is really “on-time”? A comparison of due date performance indicators in production. *Procedia CIRP*, 2, 124–129.
- Seth, D., Seth, N., & Dhariwal, P. (2017). Application of value stream mapping (VSM) for lean and cycle time reduction in complex production environments: A case study. *Production Planning & Control*, 28(5), 398–419.
- Shiang, T. S. (2009). *Predicting employment condition of TARC's ICT graduates using backpropagation neural network* (Unpublished dissertation). Universiti Utara Malaysia, Kedah.
- Turban, E., Sharda, R., & Delen, D. (2011). *Decision Support and Business Intelligence System*. New Jersey: Pearson Education Inc.
- Vrabel, M., Mankova, I., & Beno, J. (2016). Monitoring and control of manufacturing process to assist the surface workpiece when drilling. *Procedia CIRP*, 41, 735–739.
- Wanas, N., Auda, G., Kamel, M. S., & Karray, F. A. K. F. (1998). On the optimal number of hidden nodes in a neural network. In *Electrical and Computer Engineering IEEE Canadian Conference*, 2, 918–921.

- Wang, K. (2007). Applying data mining to manufacturing: The nature and implications. *Springer Science and Business, 18*, 487–495. doi:10.1007/s10845-007-0053-5.
- Wilson, R., & Sharda, R. (1992). Neural networks. *Operation Research and Management Sciences Today*, 36–42.
- Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159–175.
- Zhang, J., Zuo, L., Gao, J., & Zhao, S. (2017). Digital instruments recognition based on PCA-BP neural network. In *Technology, Networking, Electronic and Automation Control Conference IEEE 2nd Information*, 928–932.
- Zhang, L. (2017). Artificial neural networks model design of Lorenz chaotic system for EEG pattern recognition and prediction. In *Life Sciences Conference (LSC)*, 9-42.