Artificial Neural Network (ANN) to Predict Mathematics Students' Performance

Norpah Mahat [1]*, Nor Idayunie Nording [2], Jasmani Bidin [3], Suzanawati Abu Hasan[4], Teoh Yeong Kin [5]

[1,2,3,4,5] *Faculty of Computer & Mathematical Sciences, Universiti Teknologi MARA*
Perlis Branch, Arau Campus, 02600 Arau, Perlis, Malaysia

Corresponding author: * norpah020@uitm.edu.my

HIGHLIGHTS

- The Neural Network was used to develop models to predict the performance of mathematics students.
- Data sets were trained using TRAINLM, TRAINGDM and TRAINGDA functions.
- Each training function was compared based on correlation coefficient, Performance value, epoch and gradient.
- Identified significant function was referred to the higher correlation coefficient and the smallest Performance value.

ABSTRACT

Predicting students' academic performance is very essential to produce high-quality students. The main goal is to continuously help students to increase their ability in the learning process and to help educators as well in improving their teaching skills. Therefore, this study was conducted to predict mathematics students' performance using the Artificial Neural Network (ANN). The secondary data from 382 mathematics students from UCI Machine Learning Repository Data Sets used to train the neural networks. The neural network model built using nntool. Two inputs are used which are the first and the second period grade while one target output is used which is the final grade. This study also aims to identify which training function is the best among three Feed-Forward Neural Networks known as Network1, Network2 and Network3. Three types of training functions have been selected in this study, which are Levenberg-Marquardt (TRAINLM), Gradient descent with momentum (TRAINGDM) and Gradient descent with adaptive learning rate (TRAINGDA). Each training function will be compared based on Performance value, correlation coefficient, gradient, and epoch. MATLAB R2020a was used for data processing. The results show that the TRAINLM function is the most suitable function in predicting mathematics students' performance because it has a higher correlation coefficient and a lower Performance value.

Keywords: mathematics, students' performance, neural network, Levenberg Marquardt

INTRODUCTION

Predicting students' academic performance is very important for any learning institution. Early prediction will help in identifying slow learners at the early stage as they could be assisted through programs that specifically assist them in their weak subjects. Besides, the learning institution could also strategically plan to boost the teaching performance among educators in helping students to achieve excellent performance. Thus, students and educators of high quality in all aspects raise the credibility of the university internationally.

The Artificial Neural Networks (ANN) are widely used in predicting students' academic performance besides the traditional approaches. It has the ability to learn, remember and get the solution quickly. The neural networks process the information almost similar to the biological human brain. One of the studies was done by Chen, Hsieh and Quang (2014) where they applied ANN with two types of algorithms, Cuckoo Search (CS) and Cuckoo Optimization Algorithm (COA) in predicting students' academic performance. Their findings show various factors such as results of previous high school or university exams, student learning location, the form of secondary school whether private or public and students' gender that affect students' academic achievement. This result is consistent with the study done by Amirah and Nur'aini (2015). They also found that attributes that influenced the academic performance were including the student's demographic background, external assessments, extracurricular activities, high school background, social networking, psychometric considerations, method of prediction and the measurement of teaching performance among educators. Kalejaye et. al. (2015) also used a model of ANN to predict the academic performance of students. They used students' CGPA from their first year to third year as input variables to train the ANN model. The model was constructed using the ANN tool and Final Grades (CGPA) serves as the target output. The effectiveness of each Feed-forward Neural networks function was tested by comparing the simulated CGPA value with the actual final value. The results indicate an accuracy of 91.7%, it shows that the ANN model can predict the value of a student's final grade with certainty. Feng (2019) used Decision Tree and Neural Network methods to analyze the factors that influence the academic performance. The results from their study show that the Neural Network classification achieves the accuracy at 97.6% on average.

This study will use ANN to predict the mathematics students' performance. The secondary dataset was taken from 382 Mathematics students from UCI Machine Learning Repository Data Sets (Cortez and Silva, 2008). The data consists of three grades of students which are first-period grade, second-period grade and final grade from 2005 to 2006 of the school year. Hence, this study aims to develop a classification model to predict the mathematics student academic performance at university and to determine the most suitable training function among three networks.

Artificial Neural Networks (ANN)

ANN is a network of interconnected nodes controlled by neurons in the brain. Based on Figure 1, each round node shows an artificial neuron, and the arrow illustrates the relationship between the output of the artificial neuron and the other input (Aydoğdu, 2020). The Feed-Forward Neural Network (FFNN) basically refers to artificial neural networks where the node connections do not form a cycle (Aaron, 2021). They are biologically inspired algorithms that have several neurons like units arranged in layers. FFNN are also known as multi-layered networks on neurons. The neuron network is called feedforward as the information flows only in the forward direction in the network through the input nodes to output nodes.

The number of neurons in the FFNN model can consist of one or more neurons in the hidden layer. The relationship between the neurons is referred to as the *synapses*. The objective of FFNN training is to minimize the Mean Squared Error (MSE) or Sum of Squared Errors (SSE) between actual and target outputs by changing the synaptic weight of the network and neuron parameters (Kalejaya et. al., 2015).

To improve the prediction of student performance, neural networks will compare the value of real and target value. Some researchers have measured the precision of ANN in their study and prove the success of neural networks in estimating students' performance such as Livieris, Drakopoulou and Pintelas (2012), Kalejaya et. al. (2015), Isong, Udonyah and Ansa (2018) and Pal and Bhatt (2019). Therefore, ANN was chosen to conduct this study because of the ability of neural networks to build a platform of prediction, and determine the difference of input and output.
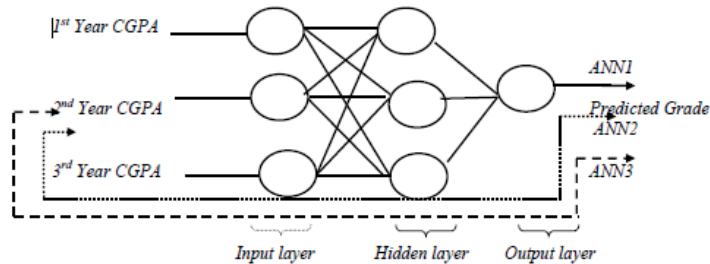

Figure 1: Artificial Neural Network (ANN) *(source: Kalejaya, 2015)*

METHODOLOGY

Firstly, students related data was collected before making student performance predictions. For this research, the secondary data was obtained from 382 mathematics students from UCI Machine Learning Repository Data Sets. The data was divided into two parts to train the neural networks. Student grade data from the first and second assessments were used as input variables and the final grade was used as output variables. The data was split into 50% of the training part and 50% of testing part.

For data processing, MATLAB R2020a was used. This study will only focus on three training functions which are Levenberg-Marquardt backpropagation (TRAINLM), Gradient descent with momentum backpropagation (TRAINGDM) and Gradient descent with adaptive learning rate backpropagation (TRAINGDA). Using *nntool* of MATLAB software, the data will be trained using Network1 which is referred to TRAINLM function, Network2 is referred to TRAINGDM function and Network3 is referred to TRAINGDA function as described in Table 1. There were seven steps in developing the Feed-Forward Neural Network (FFNN) as follows.
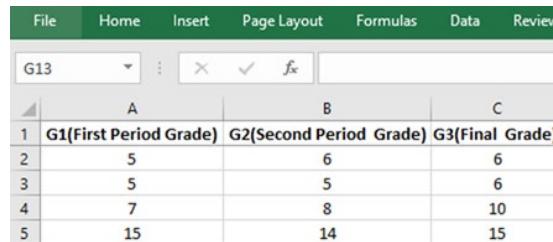
Table 1: Network configurations

Network	Training Functions
Network1	TRAINLM
Network2	TRAINGDM
Network3	TRAINGDA

Step 1: Import Data to Microsoft Excel

The data was entered into the Microsoft Excel spreadsheet before being imported into MATLAB. In Microsoft Excel menu, click on the "Data" icon and then "From Text/CSV" icon to arrange the data from text to the column spreadsheet. Next, select the data from the folder and load the dataset into Excel. The dataset was successfully imported into the Excel spreadsheet. There are a lot of attributes in dataset, but only three attributes were chosen for this research. Therefore, all unnecessary data was removed from Excel by selecting these attributes and then clicking on the "Delete" icon, and then selecting the "Delete Sheet Column". Finally, the three desired attributes were successfully imported into Excel which are G1 (First Period Grade), G2 (Second Period Grade), and G3 (Final Grade) as shown in Figure 2.

Figure 2: Import Data to Excel

Step 2: Define Inputs and Target Data

The dataset in the Excel was divided into two parts which are inputs and target data. Then, define these two files in the workspace. Choose a new file and name the files as input and target data. Next, double-click on input and then variables' spreadsheet will appear, head to Excel, copy all inputs and paste it on the variable spreadsheet as in Figure 3. After that, transpose input variables from column to row by right-clicking on it and choosing the transpose variable. Repeat the same steps for the target data.
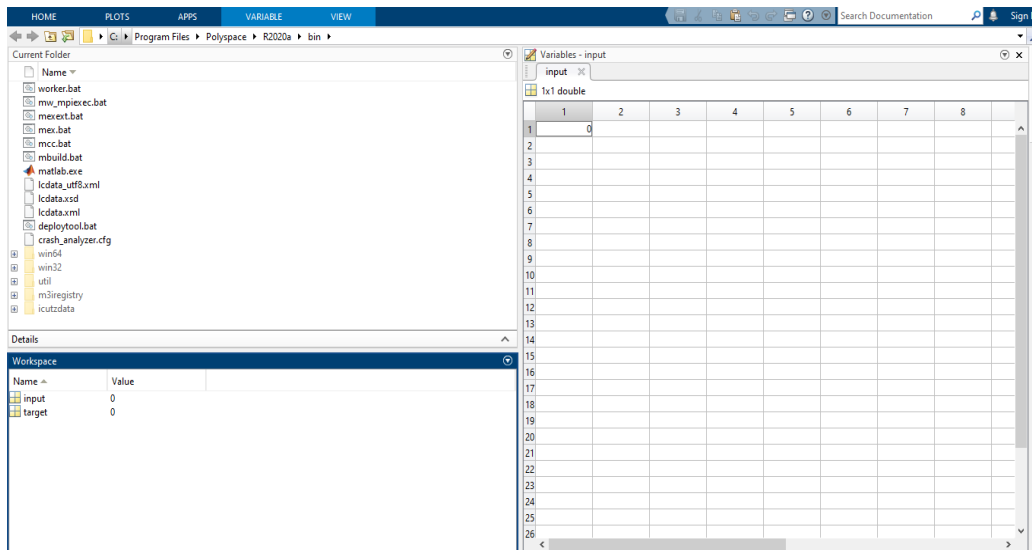

Figure 3: Paste Inputs and Target Data on Variable Spreadsheet

Step 3: Call *nntool*

Next, call the Neural Network/Data Manager *(nntool)* by typing *nntool* at the command window in the Neural Network Toolbox as shown in Figure 4.
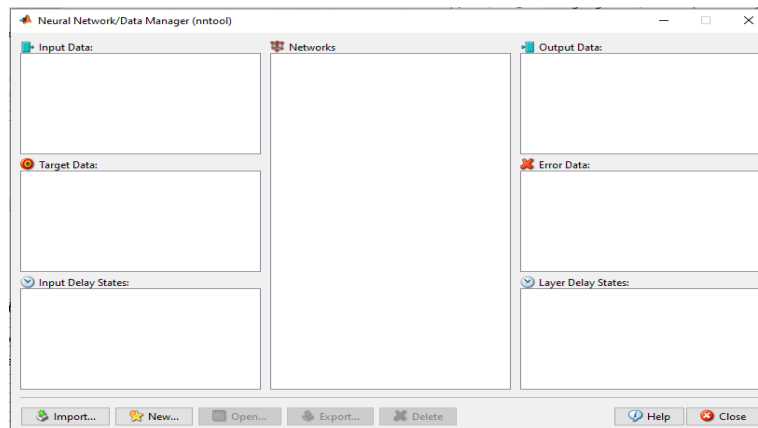

Figure 4: Neural Network/Data Manager (nntool)

Step 4: Import Inputs and Target Data from the Workspace With [IMPORT]

Next step, import the inputs and target data from MATLAB Workspace into *nntool* by clicking on the "Import" button as shown in Figure 5. Then, click the "New" button to define our Neural Network structure.
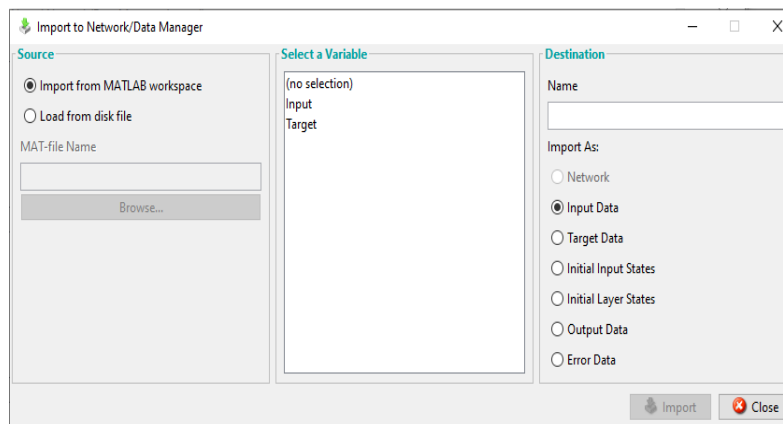

Figure 5: Import Input and Target Data

Step 5: Create the Network

The network created for this research is the Feed-Forward Neural Network (FFNN). There are three different names of FFNN that have been developed namely Network1, Network2, and Network3 as shown in Figure 6. These networks were implemented to predict the final grade of students by approaching the function that maps the grades from their first two sessions to their final grade.
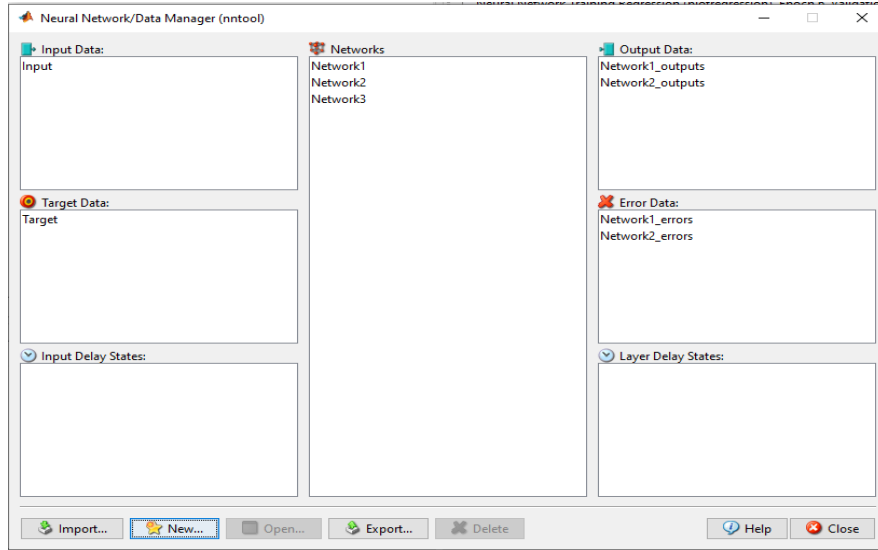
Figure 2: The Networks

Step 6: Determination of the Networks and Training Functions

Now the networks were created known as Network1, Network2, and Network3. The network will be displayed by using the "view" button as shown in Figure 7. All of these Networks have similarities in terms of inputs, hidden layers, and output.
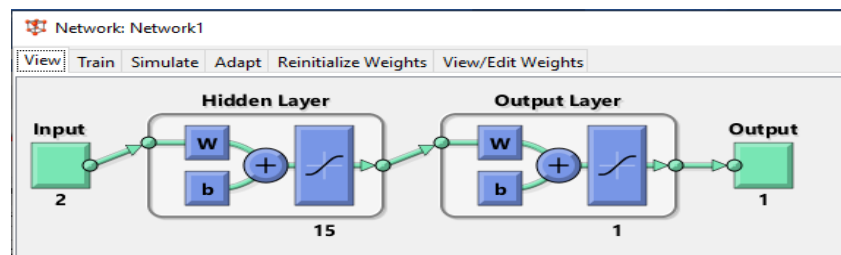

Figure 3: Network1

This study was selecting three different training functions which were TRAINLM, TRAINGDM, and TRAINGDA. These three training functions update weight and bias values of FFNN according to the Levenberg-Marquardt Optimization and Gradient Descent Optimization. TRAINGDM is a gradient descent with momentum while TRAINGDA is a gradient descent with an adaptive learning rate. Figure 8 shows the training functions of Network3 that have been chosen and used for this research. Each network has two neurons in the input layer, fifteen neurons in the hidden layer and only one neuron in the output layer.
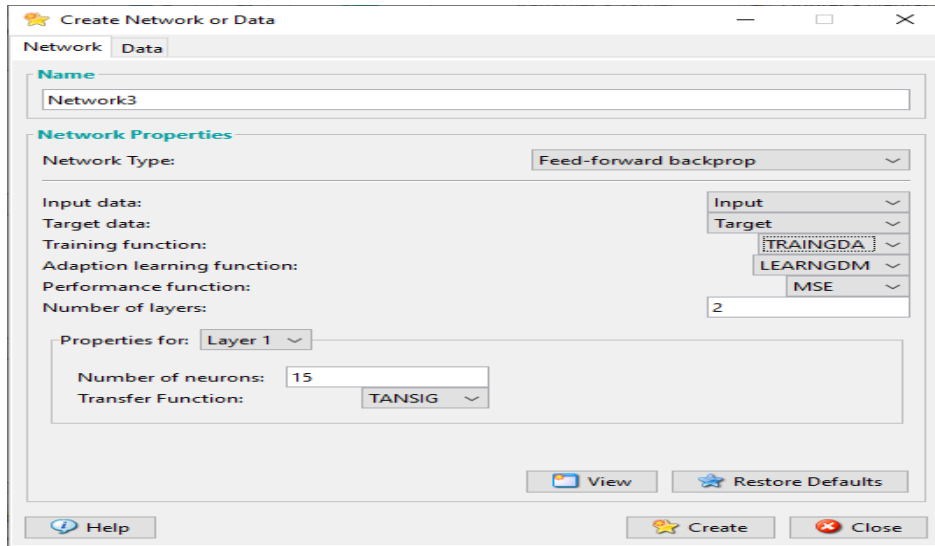
Figure 8: Training Function for Network3

STEP 7: Select the Training Function in the Network Window [TRAIN]

The next step, choose the "Create" button. The new network called Network1, Network2, and Network3 were added to the Network/Data Manager. Then, click on these networks one by one and click the "Open" button to begin training our network by choosing the "Train" function. After that, insert inputs and target data at the training data. To train the network, just press the "Train Network" button. Throughout the training process, 1000 training iterations were maintained for each of the three experimental series and the time of measurement for each network shall not exceed 25 minutes. Figure 9 shows the characteristics of the ANN model before the training session.
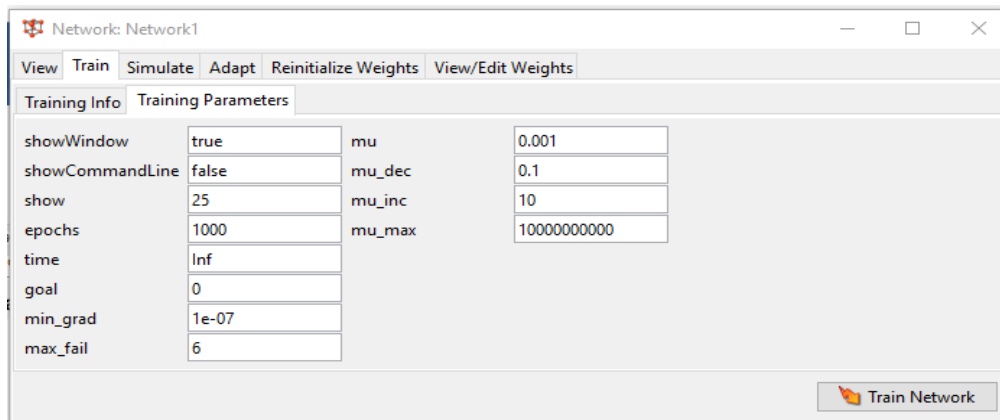

Figure 9: Training Parameters of Network1

After the training process is finished, the results of the neural network training will come out for each network in the Neural Network Training (*nntrain tool*) as can be seen in Figure 10. The results will display the Neural Networks model, Algorithms, Progress and graph representation for each of the networks involved.
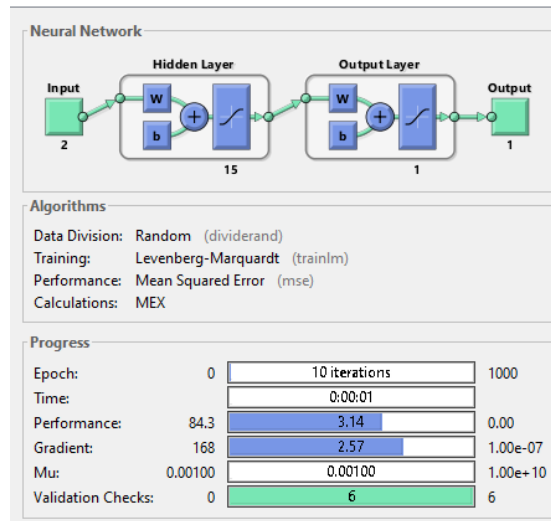
Figure 10: Neural Network Training

RESULTS AND DISCUSSIONS

The neural networks were trained with different models known as Network1, Network2 and Network3 which referred to TRAINLM, TRAINGDM, and TRAINGDA functions respectively using MATLAB software. After that, these training functions were compared based on correlation coefficient, performance value, epoch and gradient as shown in Table 2. The results from the training process will be used for choosing the best training function to train FFNN models more effectively and efficiently. The Performance values obtained after the training process as shown in Figure 10 refers to the Mean Square Error (MSE) function that is comparing the observed and inferred outputs for a data sample. The smallest Performance value shows the success of the training function because it indicates the closer the data mapping to an accurate model. The regression R-value measures the correlation between output and target. The highest R-value implies a strong positive relationship between output and the corresponding target values.

Table 1 shows that TRAINLM function is the most stable training function since it has the highest correlation coefficient between output and target. The correlation coefficient, R-value between target (Final Grade) and outputs for TRAINLM function was 0.92064 which is close to 1. This indicates that there is the strongest correlation between the target and the output value. TRAINLM success in training since it give the lowest Performance value (3.14) compared to TRAINGDM and TRAINGDA functions. TRAINLM function also requires only 10 iterations and a gradient of 2.57 indicates a very fast training rate.

Table 2: Comparison of Three Different Training Functions of FFNN

Training Functions	Correlation Coefficient	Epoch	Performance	Gradient
TRAINLM	0.92064	10 iterations	3.14	2.57
TRAINGDM	0.90941	39 iterations	3.78	37.2
TRAINGDA	0.86827	23 iterations	7.40	82.2

Figure 11 until Figure 16 represents the performance of TRAINLM, TRAINGDM and TRAINGDA functions according to their validation performance and training regression. The performance of TRAINGDM function is 3.78 with 39 iterations and R value is 0.90941. While TRAINGDA function gives 7.40 for performance value with 23 iterations and R value is 0.86827. The Mean-Square Error for

TRAINLM Network1 showed that it is very effective in minimizing the error between the output values and the target values, which can be seen from the performance plot in Figure 11. After comparing these training functions, this study exhibited the TRAINLM function as the most suitable function in the application of student academic prediction due to the strong positive relationship and the lowest of Performance value between target and output.
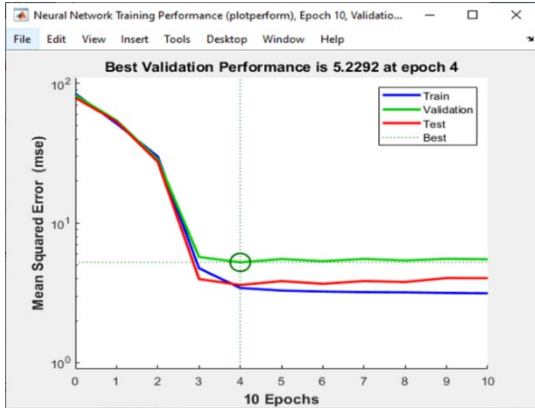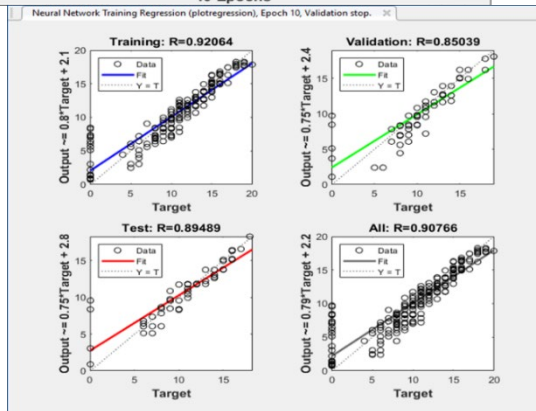

Figure 11: Validation Performance (TRAINLM)   **Figure 12**: Training Regression (TRAINLM)
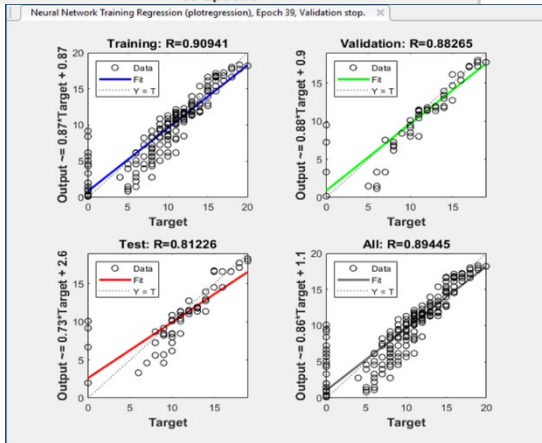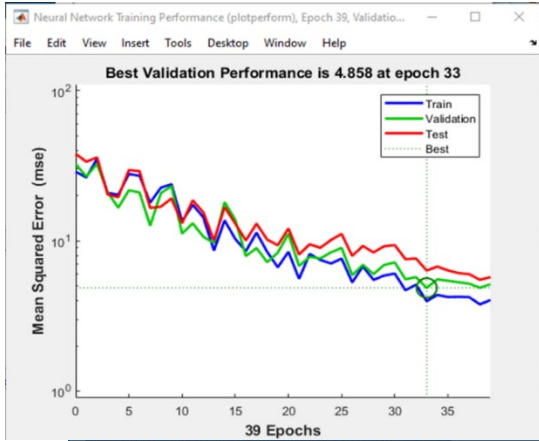
Figure 13: Validation Performance (TRAINGDM)   **Figure 14**: Training Regression (TRAINGDM)
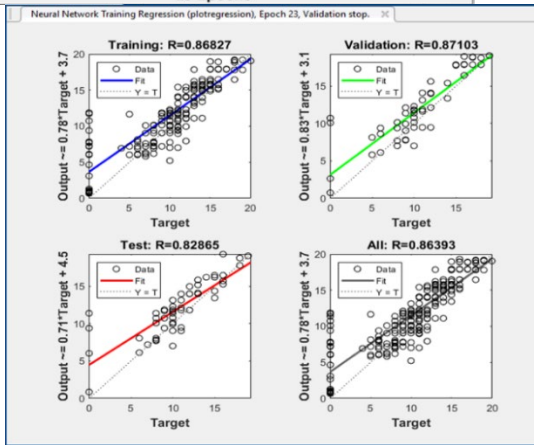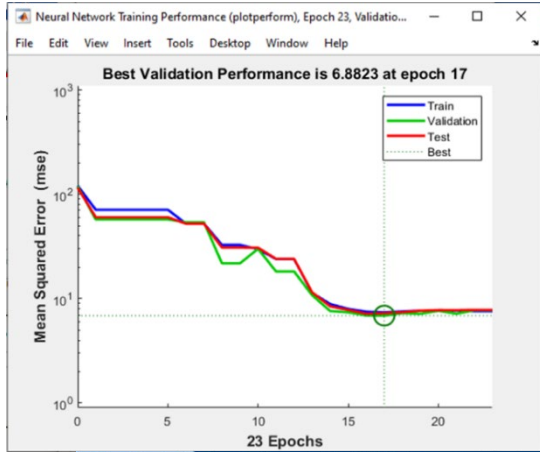
Figure 15: Validation Performance (TRAINGDA)   **Figure 16**: Training Regression (TRAINGDA)

CONCLUSIONS

The main objective of this study is to predict mathematics students' performance using the Artificial Neural Network (ANN). The sub-objectives of this study were to develop a classification model for predicting students' academic performance using MATLAB R2020a and to determine which of the three network training functions, which are TRAINLM, TRAINGDM and TRAINGDA, that produces the best results. The results were compared by looking at the highest R-value and the Performance value of each function. It is found that TRAINLM of Network1 gives the highest correlation coefficient which is 0.92064 and the smallest Performance value which is 3.14 compared to others. The Performance measures how close the mean difference of the predicted data is to the actual data. The less performance value indicates a more accurate model. Besides that, the highest value of correlation coefficient indicates that there is a strong positive relationship between the target and the output. Therefore, it can be concluded that the TRAINLM function is the most suitable training function that can be used to train FFNN models as well as to predict the mathematics student performance more effectively and accurately. For future research, it is recommended to use various neural networks training functions. From this study, it is found that each training function that has been selected will give different results for each network.

ACKNOWLEDGEMENT

CONFLICT OF INTEREST DISCLOSURE

All authors declare that they have no conflicts of interest to disclose.

REFERENCES

Aaron, E. (2021). Feedforward Neural Networks. *Data Science Central.*

Aydoğdu, S. (2020). Predicting student final performance using artificial neural networks in online learning environments. Education and Information Technologies, 25(3), 1913-1927.

Amirah, M. S., & Nur'aini, H. R. (2015). A Review on Predicting Student's Performance Using Data Mining Techniques. *Procedia Computer Science, 72*, 414-422.

Chen, J.F., Hsieh, H.N. & Quang, H.D. (2014). Predicting Student Academic Performance: A Comparison of Two Meta-Heuristic Algorithms Inspired by Cuckoo Birds for Training Neural Networks. *Algorithms, 7*(4), 538-553.

Cortez, P. & Silva, A. (2008). Using Data Mining to Predict Secondary School Student Performances. *15th European Concurrent Engineering Conference 2008, ECEC 2008 - 5th Future Business Technology Conference, FUBUTEC 2008*, 5-12.

Feng, J. (2019). Predicting Students' Academic Performance with Decision Tree and Neural Network. *Electronic Theses and Dissertations, 2004-2019*, 1-35.

Isong, E., Udonyah, K. & Ansa, G. (2018). Cognitive Factors in Students' Academic Performance Evaluation using Artificial Neural Networks. *Information and Knowledge Management, 8*(5), 57-71.

Kalejaye, B.A, Folorunso, O. & Usman, O.L. (2015). Predicting Students' Grade Scores Using Taining Functions of Artificial Neural Network. *Journal of Natural Sciences, Engineering and Technology, 14*, 1-23.

Livieris, I.E., Drakopoulou, K. & Pintelas, P. (2012). Predicting students' performance using artificial neural networks. *In Proceedings of the 8th Pan-Hellenic Conference "Information and Communication Technology in Education.*, 28-30.

Pal, V.K. & Bhatt, K.K. (May, 2019). Performance Prediction for Post Graduate Students using Artificial Neural Network. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8*(7), 446-454.