Prediction of courses score using Artificial Neural Network with Backpropagation algorithm

D Kurniadi*, A Mulyani, Y Septiana and I M Yusuf
Department of Informatics, Sekolah Tinggi Teknologi Garut, Jalan Mayor Syamsu 1, Garut, Indonesia

*dede.kurniadi@sttgarut.ac.id

Abstract. This study aims to predict the final score of courses of students who are undergoing education in higher education. In the future, this work will be used to inform students that there must be early anticipation in implementing the study of courses in order to get a satisfying final grade. The Artificial Neural Network with Backpropagation algorithm is used to solve the prediction problem for the final grade of this course. The dataset collected was 337 students who will be used for training and testing needs. Prediction variables used consisted of the value of attendance, assignments, midterms, and final exams. The network architecture using a multi-layer perceptron with three layers, namely the input layer, hidden layer, and output layers, with a 4-3-1 neuron pattern. The results showed that using a learning rate of 0.2 and repetition Epoch 1000 times, resulting in an RMSE value of 0.040929 and MSE of 0.001675 with an accuracy rate of 93.43%.

1. Introduction

Predicting or forecasting is a way to estimate or draw conclusions from what will happen by utilizing relevant data from previous times [1]. In doing prediction work, one of them can use a data mining approach, which is a process carried out to find the relationship between one data and other data so that it can be used as a basis for decision making [2]. In the process of data mining using statistical techniques, mathematics, artificial intelligence, and machine learning to extract and identify useful information [3].

In the education domain, data mining to support decision making in predicting the success of student performance has been carried out by researchers [4]. One indicator of the success of student performance can be seen from the achievement score of each course taken. Therefore, work of predicting the final score of each subject needs to be known early to avoid the occurrence of students who have to take back the course. Previous studies on predicting using educational data have been carried out, including predicting the value of final student projects with the results obtained showing the use of the Naïve Bayes algorithm has a higher level of accuracy compared to the ID3 and CHAID algorithms [5]. Subsequent studies on the prediction of course scores, the results obtained indicate the use of the K-Apriori algorithm has an error rate of 11%, or equal to an accuracy of 89% [6]. Then prediction of the National Examination Score, the results obtained indicate the value of Root Mean Square Error (RMSE) obtained by 0.169 +/- 0.016 [7].

The purpose of writing this article based on the background and description of previous research studies is to predict the final score of student courses using a different approach and algorithm, namely Artificial Neural Network (ANN) with Backpropagation. The dataset uses student score data on the Algorithm and Programming subjects at Sekolah Tinggi Teknologi Garut. The expected results of this
research are obtained prediction models of course scores with better accuracy, which can be implemented to inform students that there must be early anticipation of the learning process of related subjects in order to get a satisfactory final score.

2. Methods

2.1. Research workflow
The following research workflow is shown in Figure 1.

![Figure 1. Research workflow.](image)

The research workflow consists of 4 interrelated stages, namely data collection, data pre-processing, model training, testing, and evaluation. The following detailed explanation:

- Data collection, this stage is the process of collecting data on previous student subject scores from Sekolah Tinggi Teknologi Garut.
- Data pre-processing is the process of data cleansing and the process of selecting attributes to be used in training models using Artificial Neural Network (ANN) with Backpropagation.
- Model training, this stage is the process of training an ANN with Backpropagation model using pre-processed data, the purpose of finding the best rules by changing the variables used.
- Testing and evaluation, testing is done in two stages, namely testing of trained data and testing of new data that has never been trained, after which an evaluation was informed. The results of implementing the model into Artificial Neural Network with backpropagation algorithm will produce a rule. The rule is used as the basis for predicting student courses score.

2.2. Artificial neural network with backpropagation
Artificial Neural Network is a technique or approach to information processing by imitating the workings of the human brain and having the ability to learn from the sample data provided [8]. Backpropagation is a machine-learning algorithm to reduce the error rate by adjusting the weight based on the desired output and target differences [9]. The network architecture to be used is as shown in Figure 2.

![Figure 2. Artificial neural network architecture.](image)

Figure 2 is an example of artificial neural network architecture with used Multi-Layer Perceptron architecture with neurons in each layer, which consists of input, hidden, and output layers [10].

3. Result and discussion
To simplify the research process, an experiment will be conducted based on a predetermined research workflow. For training, testing, and evaluation, we use the Python Programming Language as a tool to
analyse the results of experiments in applying artificial neural network models with backpropagation algorithms [11].

3.1. Data collection
The data collected for the study used real data scores on the Algorithm and Programming course at Informatics Engineering study program Sekolah Tinggi Teknologi Garut from the year 2015 to 2016. The data obtained still consists of a collection of raw student data, such as profile data, course data, attendance, assignments, midterms, and final exams. Before being used, the data collected will go through the stages of pre-processing data.

3.2. Data pre-processing.
To simplify the training process, the sample data will be used specifically for algorithmic and programming courses in semester 1. Samples were randomly selected, and 337 people were selected as data samples. The selection of attributes used in this study consisted of student number, attendance, assignments, midterm, final exam, final scores, and letter grades. The following sample data will be used as in Table 1.

| No. | Student ID Number | Attendance | Assignment | Midterm | Final Exam | Final Score | Letter Grade |
|-----|-------------------|------------|------------|---------|------------|-------------|--------------|
| 1   | 1106015           | 42.3       | 70         | 45      | 80         | 62.3        | C            |
| 2   | 1206065           | 7          | 0          | 0       | 0          | 1           | E            |
| 3   | 1306003           | 100        | 80         | 45      | 70         | 68.5        | C            |
| 4   | 1306025           | 71.4       | 75         | 80      | 80         | 77.9        | B            |
| 5   | 1606001           | 100        | 80         | 95      | 90         | 91.5        | A            |
| 6   | 1606004           | 100        | 80         | 90      | 90         | 90          | A            |
| 7   | 1506001           | 92.9       | 80         | 50      | 68         | 72.73       | B            |
| 8   | 1506011           | 100        | 80         | 85      | 90         | 88.75       | A            |
| 9   | 1506102           | 57         | 50         | 60      | 60         | 56.75       | D            |
| ... | ...               | ...        | ...        | ...     | ...        | ...         | ...          |
| 337 | 1706124           | 21         | 0          | 8       | 20         | 13          | E            |

Furthermore, to normalize the data, normalization is carried out so that the attribute scale of each variable is the same using the following formula [12].

$$y' = \frac{0.8(x - x_{min})}{(x_{max} - x_{min})} + 0.1 \quad (1)$$

Where: $y'$: normalized value, $x_{min}$: smallest value of all available data, $x_{max}$: highest value of all available data.

3.3. Model training
After the data collection and data cleaning stages are completed, the next step is training using Artificial Neural Network (ANN) with Backpropagation. Before conducting training, the first thing to do is to determine the ANN network architecture that will be used. The architecture used in this study is the perceptron multi-layer architecture with a 4-3-1 neuron pattern, as shown in Figure 3.
Figure 3. ANN with backpropagation architecture used in research.

Based on Figure 3, the data pattern design is following the ANN Back-propagation architecture that has been determined, namely the input layer with four neurons (attendance, assignment, midterm, final exam), hidden layer with three neurons, and one neuron as the output target. After determining the pattern followed by training using the Backpropagation algorithm, which consists of 3 phases, namely forward, backward, and updates. The steps used are as follows.

- Initial weight initialization at bias, weight, and learning rate. For each bias given a value of 1, this is done so as not to affect the weight value between the bias with the hidden layer and the bias with the output layer while the initial value for each weight can be determined freely.
- As long as the values in the output layer are wrong, repeat steps 3 through 4 again.
- For each training data, do the forward, backward, and update process.
  - Forward
  - Calculate all output in the hidden layer \( Z_j \) (\( j = 1, 2, \ldots, p \))
  - Calculate each neuron input value on layer \( Z \) with the formula (2):
  \[
  z_{inj} = W_{j0} + \sum_{i=1}^{n} x_i W_{ij}
  \]
  where \( X_i \) input value at the input layer, \( W_{ij} \) weight value from the input layer to the hidden layer, and \( W_{bji} \) weight value from the bias to the hidden layer. Furthermore, the activation results of the calculation of the value of each neuron using the sigmoid activation function with the formula (3).
  \[
  z_{outj} = f\left(z_{inj}\right) = \frac{1}{1 + e^{-z_{inj}_j}}
  \]
  Calculate all networks in the output layer \( y_k \) (\( k = 1, 2, \ldots, m \))
  - Calculate each neuron input value at the \( Y \) output layer with the formula (4):
  \[
  y_{in_k} = W_{k0} + \sum_{j=1}^{p} z_j W_{kj}
  \]
  Where \( Z_j \) output value from hidden layer, \( W_{kj} \) weight value from hidden layer to output layer, and \( W_{k0} \) weight value from bias to the output layer. The activation of the calculation of the value of each neuron using the sigmoid activation function with the formula (5).
  \[
  y_{out_k} = f\left(y_{in_k}\right) = \frac{1}{1 + e^{-y_{in_k}}}\]
  - Backward
  - Calculate the number of factors (\( \delta \)) layer output based on errors on each output layer \( y_{in_k} \) (\( k = 1, 2, \ldots, m \))
  \[
  \delta_k = (t_k - y_k) f'(y_{in_k}) = (t_k - y_{out_k}) y_{out_k}(1 - y_{out_k})
  \]
  where \( T_k \) target value, and \( Y_{out_k} \) activation value of \( Y_{in_k} \)
  - Calculate the weight change rate \( W_{kj} \) with velocity \( \alpha \)
\[ \Delta w_{kj} = \alpha \delta_k z_{outj} \]  
(7)

\( k = 1, 2, ..., m, j = 0, 1, ..., p \)

Where \( \alpha \) learning rate, \( \delta_k \) output layer factor, \( z_{outj} \) output value of hidden layer

Calculate \( \delta \) hidden layer based on the error for each hidden layer \( z_j (j = 1, 2, ..., p) \)

\[ \delta_{netj} = \sum_{k=1}^{m} \delta_k W_{kj} \]  
(8)

Where \( \delta_k \) output layer factor, \( W_{kj} \) weight value between the hidden layer and output layer.

\[ \delta_j = \delta_{netj} = f'(z_{netj}) = \delta_{netj} z_j (1 - z_j) \]  
(9)

Calculate the rate of change in weight \( \nu_{ji} \) with velocity \( \alpha \)

\[ \Delta w_{ji} = \alpha \delta_j x_i \]  
(10)

\( j = 1, 2, ..., p, i = 0, 1, ..., n \)

- Update
  
  Calculate the change in weights that lead to the output layer line

\[ w_{kj}(new) = w_{kj}(now) + \Delta w_{kj} \]  
(11)

\( k = 1, 2, ..., m, j = 0, 1, ..., p \)

Where \( w_{kj}(new) \) is the new weight value for the next training, \( w_{kj}(now) \) is the weight value used previously, and \( \Delta w_{kj} \) is the value of the weight change term. Change weight of line leading to hidden layer with formula (12).

\[ w_{ji}(new) = w_{ji}(now) + \Delta w_{ji} \]  
(12)

\( j = 1, 2, ..., p, i = 0, 1, ..., m \)

Where \( w_{ji}(baru) \) is the new weight value for the next training, \( w_{ji}(now) \) is the weight value used previously, and \( \Delta w_{ji} \) is the value of the weight change term.

- Update the epoch value with the formula (13)

\[ epoch = epoch + 1 \]  
(13)

To get a minimal error value, this training needs to determine how many epochs to do.

3.4. Testing and evaluation

In this research, testing and evaluation are conducted based on the stages of the training model using the Backpropagation algorithm. The initial scenario is done by setting the value of bias, learning rate, and weight as follows: Learning rate = 0.2; Bias = 1; Wxz: W01 = -0.3, W11 = 0.2, W21 = 0.3, W31 = 0.2, W41 = 0.3, W02 = -0.3, W12 = 0.2, W22 = 0.3, W32 = 0.2, W42 = 0.3, W03 = -0.3, W13 = 0.2, W23 = 0.3, W33 = 0.2, W43 = 0.3, Wyz: W01 = -0.1, W11 = 0.5, W21 = -0.3, W31 = -0.4. To get a very small error value, this training needs to determine how many epochs to do. Following are the results of the training comparing 150 targets to be achieved and the predicted results by setting the epoch value 1000 times, as in Figure 4. The next step to measure the accuracy of our prediction results is to use two ways, namely by finding the value of Root Mean Square Error (RMSE) \([13]\) and Mean Square Error (MSE) \([14]\) with formula (14).

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}} \\
MSE = \frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}
\]  
(14)

Where \( x \) is the target value, and \( y \) is the predictive value, \( n \) is the amount of data that is used.
Furthermore, to see the decrease in RMSE and MSE values in each epoch, the RMSE and MSE calculations are performed every 5 epochs. The results obtained can be seen in Figure 5.

![Figure 4. True value and predicted results.](image1)

![Figure 5. Comparison graph of RMSE and MSE.](image2)

In Figure 5, the RMSE values from epoch 1 to epoch 200 have a drastic decrease, but from epoch 200 to epoch 1000 the decrease is not too significant. At the 1000th epoch, RMSE values of 0.040929 and MSE of 0.001675 were obtained, changes in RMSE and MSE in epoch 0, 100, 200, 400, 800 and 1000, a comparison as in table 2.

| Epoch | RMSE   | MSE    |
|-------|--------|--------|
| 0     | 0.174140 | 0.030324 |
| 100   | 0.049186 | 0.002419 |
| 200   | 0.051694 | 0.002672 |
| 400   | 0.044574 | 0.001986 |
| 800   | 0.042440 | 0.001801 |
| 1000  | 0.040929 | 0.001675 |

Table 2. Changes in value on Epoch.

To find out the level of accuracy obtained, this study uses the Mean Absolute Percentage Error (MAPE) formula (15) [12,15].

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%
\]

(15)

Where \( y_i \) predicted value, \( \hat{y}_i \) target value from data sample, \( |x_i - y_i| \) shows the value of the reduction results must always be absolute and \( n \) is the sum of all sample data.

The level of prediction accuracy based on formula 15 obtained an accuracy value of 93.43%.

4. Conclusion

Based on research that has been done using Artificial Neural Network with Backpropagation algorithm and Multi-Layer Perceptron architecture using three layers with a Learning Rate value of 0.2, with repetition of epochs as much as 1000 times, the results obtained RMSE value of 0.040929 and MSE of 0.001675 and an accuracy rate of 93.43%. Expectations from resulting research can be implemented to help manage and predict the final grades of courses for lecturers and students to improve learning.

References

[1] Kurniadi D, Abdurachman E, Warnars H L H S and Suparta W 2018 The prediction of scholarship recipients in higher education using k-Nearest neighbor algorithm IOP Conf. Ser. Mater. Sci. Eng. 434 012039

[2] Roiger R J 2017 Data Mining: A Tutorial-Based Primer (CRC Press)

[3] Kurniadi D, Abdurachman E, Warnars H L H S and Suparta W 2019 A proposed framework in an intelligent recommender system for the college student J. Phys. Conf. Ser. 1402 066100
[4] Badr G, Algobail A, Almutairi H and Almutery M 2016 Predicting Students’ Performance in University Courses: A Case Study and Tool in KSU Mathematics Department Procedia Computer Science vol 82

[5] Mayadewi P and Rosely E 2015 Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining SESINDO 2015 2015

[6] Yunita W L, Muflikah L and Furqon M T 2017 Prediksi Nilai Mata Kuliah Mahasiswa Menggunakan Algoritma K-Apriori Isisfo 06 157–72

[7] Utomo W 2015 Prediksi Nilai Ujian Nasional Produktif Sekolah Menengah Kejuruan Menggunakan Metode Neural Network Techno.COM 14 33–41

[8] Da Silva I N, Spatti D H, Flauzino R A, Liboni L H B and dos Reis Alves S F 2017 Artificial neural networks Cham Springer Int. Publ. 39

[9] Nielsen M 2015 How the backpropagation algorithm works Neural networks Deep Learn. Determ. Press

[10] Ramchoun H, Idrissi M A J, Ghanou Y and Ettaouil M 2016 Multilayer Perceptron: Architecture Optimization and Training IJIMA! 4 26–3

[11] Nelli F 2018 Python Data Analytics (Berkeley, CA: Apress)

[12] Widodo A P, Suhartono E A S, Sarwoko E A and Firdaus Z 2017 Akurasi Model Prediksi Metode Backpropagation Menggunakan Kombinasi Hidden Neuron Dengan Alpha MATEMATIKA 20

[13] Wang W and Lu Y 2018 Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model IOP Conference Series: Materials Science and Engineering vol 324 p 12049

[14] Kanwal S, Khan B, Ali S M, Mehmood C A and Rauf M Q 2018 Support vector machine and gaussian process regression based modeling for photovoltaic power prediction 2018 International Conference on Frontiers of Information Technology (FIT) pp 117–22

[15] Baswardono W, Kurniadi D, Mulyani A and Arifin D M 2019 Comparative analysis of decision tree algorithms: Random forest and C4.5 for airlines customer satisfaction classification J. Phys. Conf. Ser. 1402 066055