Artificial Neural Networks For Pattern Recognize Handwritten

Arief Kelik Nugroho\textsuperscript{1,2}, Ipung Permadi\textsuperscript{1}, Aini Hanifa\textsuperscript{1}, Dian Tri Wiyanti\textsuperscript{2}
\textsuperscript{1}Departemen of Informatics, Engineering Faculty, Universitas Jenderal Soedirman, Indonesia,
\textsuperscript{2}Departement of Mathematics, Faculty of Mathematics and Natural Science, Universitas Negeri Semarang
\textsuperscript{1}arief.nugroho@unsoed.ac.id

Abstract. Backpropagation is a supervised learning method on artificial neural networks. The process of training artificial neural networks that are built will be tested several structures of artificial neural networks, with the hope that the best structure is obtained to produce the output with the smallest error. The evaluation purposes use the same data set to form a neural model with a hidden level and the neuron model with two hidden levels then compare these three models both in terms of classification accuracy and training time. A neural network can be concluded that for each architecture, the neural network architecture used to recognize type 3 is 82.42%, type 2 77.5%, type 1 98.1%.

1. Introduction
Backpropagation is a supervised learning method on artificial neural networks. Usually used in the multilayer perceptron (MLP) to change the weights associated with neurons that exist in the hidden layer[1,2]. This algorithm tries to make a gradient graph that shows the error value during the iteration at the time of training. Then the backpropagation algorithm can be accelerated significantly after finding the gradient component, and the weight is modified along the gradient direction until the minimum value. This process can be carried out without the need for intensive gradient computation calculations at each step. The new gradient components are calculated shortly after a minimum value in the direction of the previous gradient is reached. This process is only possible for cumulative weighting. A method for finding the minimum value along the gradient direction is a three-step process of finding errors for three points along the gradient direction and then using a parabolic approach, jumping directly to the minimum[1,2].

The process of training artificial neural networks that are built will be tested several structures of artificial neural networks, with the hope that the best structure is obtained in order to produce the output with the smallest error. Some parameters considered in designing neural networks in this study include are weight ($w$), bias ($b$), learning rate ($\alpha$), MSE ($e$), momentum, number of layers, number of nodes, and speed of process[7]. Other algorithms or methods in pattern recognition are using artificial neural networks. This pattern may originate from the results of the extraction process from the pattern to be recognized. For example, the pattern of an image will be recognized, then the process of extracting the features of the image will be carried out. The extraction results then become input for artificial neural networks to be recognized, one of which is through the classification process of the pattern of input provided. Pattern recognition by artificial neural networks is done by first training it with certain patterns[3].

2. Method
The goal of this study is to create software that recognizes handwritten characters. To achieve the goal, we have developed a classification module that uses a Neural network model. Form the neural network models with a series of images of data of handwritten characters. For evaluation purposes, we also use
the same data set to form a neural model with a hidden level and the neuron model with two hidden levels. We compare these three models, both in terms of classification accuracy and training time. This section describes the research methods used in this document. The methodology consists of gathering data for characters written by hand, constructing the neural network model, modeling and testing models, and classifying handwritten characters. Each part of the methodology will be described in the following subsections.

2.1. Thinning Algorithm
To obtain an image skeleton by reducing the object pixels in the image to produce the simplest form of the object contained in the image[9]. Although the thinning method is similar to the erosion or opening method, the difference between the morphological operations methods is the result of the operation. The difference is that the thinning method maintains the continuity of the object’s pixels in the image, thus maintaining the shape and structure of the object. The thinning method used to process the images obtained is the Zhang-Suen thinning algorithm. The first check removes the pixels from the lower right corner (southeast) of the object[9]. Meanwhile, the second examination removes the pixels in the upper left corner (northwestward) of the object. The processes that have been carried out in this procedure will produce images that are ready to be converted into data sets used for procedures related to artificial neural networks (procedures for forming and training neural networks and testing procedures for artificial neural networks). However, before it is used to form, train, and test artificial neural networks, the resulting image matrix must be converted to a row matrix.

2.2. Neural Network
An artificial neural network is one of the branches of discussion in the field of Artificial Intelligence studies (Artificial Intelligence). An artificial neural network is an information processing system that has characteristics such as biological neural networks and is formed as a generalization of mathematical models of artificial neural networks[10]. An artificial neural network is built consisting of an input layer, a hidden layer, and also the output layer.

Artificial neural network model Back Propagation is one of the models of artificial neural networks that are included in supervised learning. Like other artificial neural network models, Back Propagation trains networks to strike a balance between the ability of the network to recognize patterns used during training and the ability of networks to provide correct responses to input patterns similar to those used during the training process[10].

![Figure 1. Model Back Propagation](image)

3. Results and Discussion
The images were collected in the form of two kinds, namely images from the results of computer writing and images from the results of handwriting. 20 images were collected for each character, with 5 pieces of the computer writing and 15 pieces of handwriting. The scripts produced from word processing are arranged so that the five characters formed have different thicknesses. The thickness of
the letters formed is divided into normal (thickest), brighter 5%, brighter 15%, brighter 25%, and brighter 35% (thinnest).

3.1. Data Acquisition

Each instance of the dataset is presented to the classification algorithm using a representation system. The quality of the presentation system is a key factor in most classification algorithms. The most common representation system is characteristic [6]. In this system, each case is represented by a vector form value. The dataset (containing both the exercise set and the test set) is usually represented in a tabular form, with each sequence being an example and each column representing different problem areas. The quality of the action space is defined in the problem area, which usually includes the amount of action and their importance for the task in question, which has a significant impact on the performance of the classification algorithm.

The main database of handwritten characters was manually collected by some, as shown in Figure 1. Each character class consists of 100 training data. Each data set in the dataset is an 8-bit grayscale image that is 20 x 20 pixels in size.

![Figure 2. Type 1](image1.png)

![Figure 3. Type 2](image2.png)

![Figure 4. Type 3](image3.png)

3.2. Preprocessing Data

The scripted image data that has been collected will go through sequential processes so that the data obtained will be used in the next procedure. These processes aim to make image data that will be used in neural network training to be clean from noise or interference so that the results obtained are accurate. This normalization must be carried out to uniform the size of the training image. When this
process is not carried out, subsequent processes will be hampered due to the unequal size of the characters used.

The next process is extracting the shape characteristics of the image. This process is intended to take the characteristics possessed by the image that will be used as training data in artificial neural networks. This characteristic is used as a differentiator between images. The characteristic shape that can be extracted from the image is by using the skeleton method. Skeleton is a unique form of an object that resembles the framework of an object and is used for handwriting recognition and recognition, fingerprint patterns, biological cell structures, and other such things. This skeleton has three characteristics, its thickness is 1 mm; passes through the center of the object; and states the object's topology.

3.3. Formation and Learning of Artificial Neural Networks

This procedure is carried out to form and train artificial neural networks. This procedure will produce an artificial neural network that can recognize the types. This procedure begins by compiling data sets (datasets) that will be used, determining the stages that need to be done in learning in artificial neural networks, determining the value of the parameters needed to build an artificial neural network, and conducting neural network learning, imitation. The learning data used is divided into two parts. The two parts are training data and testing data. Training data is used in the training stage of artificial neural networks. Meanwhile, the test data is the data used in the testing stage of artificial neural networks. Of the 20 literacy image data that have been collected and preprocessed, 80% of the illiterate sample data are used as training data. Meanwhile, the remaining 20% of sample image data is used as test data.

Much of the training data and testing data used in each artificial neural network architecture are listed in table 1

| Type | Data(Learning) | Data (Training) | Data (Testing) |
|------|----------------|----------------|----------------|
| Type 1 | 920            | 736            | 184            |
| Type 2 | 400            | 320            | 80             |
| Type 3 | 100            | 80             | 20             |

3.4. Learning ANN

The first step taken in the procedure this time is to compile data sets (datasets). The data sets that will be used in the learning process of artificial neural networks are taken from the pixel values of images that have been processed in such a way as to produce the shape. Then, the image matrices will be converted into a row matrix (matrix with dimensions of 1 x dimensional image of the beginning of [the number of rows multiplied by the number of columns]).

The parameters used are the number of layers/layer on each architecture, the number of neurons in each layer/layers neural network, the pace of understanding/learning rate, momentum, value fault tolerance (in this case is the value of the mean square error ) learning, and coefficients maximum iteration. These parameters determine the results of the artificial neural network training process.

3.5. Number of Layers

The number of layers/layers used in artificial neural network architecture is 3 layers, namely 1 input layer, 1 hidden layer, and 1 output layer. The number of neurons for each layer/layers of artificial neural networks is determined as follows. Neurons in the lining of the input/input layer amounted to 400 neurons. This number adjusts the size of the input image 20 x 20 pixels or in other words 400 pixels. Neurons in the hidden layer / hidden layers totaling 50 pieces of neurons. Neurons in the output layer vary in number. The number of neurons in this layer is determined based on the type of character. The number of neurons in the output layer of artificial neural networks to recognize the type
of goujon script is the number of neurons. Meanwhile, the number of neurons in the output layer of artificial neural networks to recognize the type of dotaon is 20 neurons and the number of neurons in the layer of an artificial neural network to recognize the type 1 type of alphabet is 5 neurons. The value of the coefficient of the learning rate, momentum, errors tolerance, and maximum iteration is shown in the following table.

| (learning rate) | 0.005; 0.003; 0.001 |
|-----------------|---------------------|
| Momentum        | 0.1; 0.05; 0.01     |
| Error tolerance| \(1 \times 10^{-20}\) |
| Learning        |                     |
| Maks iteration  | 10000               |

The training phase is carried out using predetermined learning parameters. Training of the training data sets results in the data listed in table 3 and table 4.

| Type | A     | B     | Hidden layer | Iteration | Average (Training) | Avg (Testing) |
|------|-------|-------|--------------|-----------|--------------------|---------------|
| Type 1 | 0.005 | 0.1   | 50           | 10000     | 0.8673             | 0.7656        |
|       | 0.003 | 0.05  |              |           | 0.9214             | 0.7703        |
|       | 0.001 | 0.01  |              |           | 0.8322             | 0.7311        |
| Type 2 | 0.005 | 0.1   |              |           | 0.7063             | 0.6733        |
|       | 0.003 | 0.05  |              |           | 0.6368             | 0.5651        |
|       | 0.001 | 0.01  |              |           | 0.6918             | 0.6458        |
| Type 3 | 0.005 | 0.1   |              |           | 0.7060             | 0.6882        |
|       | 0.003 | 0.05  |              |           | 0.8360             | 0.8429        |
|       | 0.001 | 0.01  |              |           | 0.7658             | 0.7586        |

| Architecture | Hidden Layer | A     | B     | Error tolerance | Max iteration |
|--------------|--------------|-------|-------|-----------------|--------------|
| Type 1       |              | 0.003 | 0.05  |                 |              |
| Type 2       | 50           | 0.005 | 0.1   | \(1 \times 10^{-20}\) | 25000        |
| Type 3       |              | 0.003 | 0.05  |                 |              |
The second learning parameters have been set. The second stage of learning about neural network architecture is ready to do. After the second stage of learning, the learning score is shown in Table 5.

| Architecture | AVG training Score | AVG testing score |
|--------------|--------------------|------------------|
| Type 1       | 0.8242             | 0.8252           |
| Type 2       | 0.6520             | 0.5505           |
| Type 3       | 0.9080             | 0.7420           |

This training score does not indicate whether artificial neural networks can identify characters completely or not. The results of these calculations are shown in Table 6. These calculations are performed on each learning data used, both training data and testing data.

| Architecture | Data(training) | Data(testing) |
|--------------|----------------|---------------|
| Type 3       | 82.42%         | 8.152%        |
| Type 2       | 77.5%          | 46.25%        |
| Type 1       | 98.1%          | 82.5%         |

4. Conclusion
Learning that has been done shows the scores generated by the parameters used in the learning phase. Based on the learning results that have been shown in Table 5 and 6, it can be concluded that for each architecture the neural network architecture used to recognize Type 3 picture characters has the most optimal performance score on the second learning parameter (learning rate $\alpha = 0.003$ and momentum $\beta = 0.05$). The artificial neural network architecture used to recognize Type 1 characters have the most optimal performance score on the first learning parameter (learning rate $\alpha = 0.005$ and momentum $\beta = 0.1$). The artificial neural network architecture to recognize the Type 1 has the most optimal performance score on the second learning parameter (learning rate $\alpha = 0.003$ and momentum $\beta = 0.05$).

References

[1]. Wilamowski, B.M. Comparison of Training Algorithms and Network Architectures. 2013, p.4799.
[2]. Wilamowski, B.M. Neural Network Architectures and Learning. International Conference on Industrial Technology, 2003, pp.1–12
[3]. Rojas, R., 1996. Neural Networks A Systematic Introduction, Berlin: SpringerVerlag.
[4]. Cun L.Y., et al. Handwritten digit recognition with a back-propagation network. In Advances in neural information processing systems, 1990.
[5]. Y. LeCun, F.J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting In Computer Vision and Pattern Recognition. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on., 2004, volume 2, pages II–97. IEEE.
[6]. Nugroho, A. K., Permadi, I. *Composite Image with a Geographic Information System Approach*, IOP Conference Series: Earth and Environmental Science, 2020. Volume 406

[7]. Dash, T., Ranjan, S., Nayak, T. & Mishra, G., 2014. *Neural Network Approach to Control Wall Following Robot Navigation*. In IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT). pp. 1072–1076

[8]. Rajashekararadhya, S. V, and Ranjan, P. V. *Efficient Zone Based Feature Extraction Algorithm For Handwritten Numeral Recognition of Four Popular South Indian Scripts*. Journal of Theoretical and Applied Information Technology, 2008. Volume IV, pp. 1171-1181

[9]. Fisher, R., Perkins, S., Walker, A. & Wolfart, E., 2003. *Morphology – Thinning*

[10]. Siang, J.J, *Jaringan Syarat Tiruan dan Pemrogramannya menggunakan MATLAB*. 2009. Yogyakarta: Penerbit Andi