ISSN 1462 2815

UNVEILING THE POTENTIAL OF SECRET LAYERS: AN EXAMINATION OF THE EFFECTS OF HIDDEN LAYER DIFFERENCES ON NUMBER AND LETTER IDENTIFICATION THROUGH BACKPROPAGATION NEURAL NETWORKS

Abdul Tahir [1]*, Musakirawati [2], Dani Setiawan [3] and Jasman [4]

[1,2,3,4] Machine Maintenance and Repair, Soroako Technical Academy, Indonesia.
Email: [1]abdultahir0101@gmail.com (*Corresponding Author),
[2]musakirawati@ats-sorowako.ac.id, [3]dani@ats-sorowako.ac.id,
[4]jasman@ ats-sorowako.ac.id

DOI: 10.5281/zenodo.14058871

Abstract

This study aimed to create an optimal Artificial Neural Network (ANN) model for recognizing document identity patterns. Various experiments were conducted with different numbers of neurons in the hidden layer to determine the best ANN model that provided optimal accuracy in data validation and testing. During the training process, 70% of the research data (980 data points) were used as training data, whereas the remaining 30% (420 data points) were used as testing data. Each variation in the number of neurons in the hidden layer was evaluated to determine the performance and accuracy of the resulting model. The experimental results demonstrated that the number of neurons in the hidden layer significantly influenced the pattern recognition accuracy. Initially, an increase in the number of neurons resulted in a decrease in the Mean Squared Error (MSE) value and an increase in accuracy in the validation and testing stages. However, no significant improvement was observed when the number of neurons exceeded 30. A summary of the experimental results showed that the ANN model with 30 neurons in the hidden layer provided optimal performance with a validation accuracy of 99.49% and a testing accuracy of 97.62%. In the development of document control applications, this model was chosen due to its stable performance and relatively short computation time. Based on the results and analysis of this experiment, it was concluded that the number of neurons in the hidden layer of the ANN affected the performance of the model. To obtain optimal performance, the number of neurons in the hidden layer should not exceed 30. A model with 30 neurons in a hidden layer is the best choice for document identity pattern recognition, based on computational accuracy and time efficiency.

Keywords: Backpropagation Neural Networks, Hidden Layers, Letter Identification, Neural Network Architecture, Number Identification.

INTRODUCTION

Pattern recognition is one of the most vital areas of research in artificial intelligence and machine learning (LeCun et al., 2015; Schmidhuber et al., 2015). This is due to the numerous applications that require the automatic recognition and interpretation of data, such as handwriting recognition, speech recognition, and facial recognition. By utilizing advancements in computational technology and algorithms, researchers have been able to develop more intricate and efficient models for recognizing patterns in data. In recent years, deep learning techniques have emerged as a potent tool in pattern recognition, enabling the creation of sophisticated models that can accurately detect complex patterns in large datasets (LeCun et al., 2015; Schmidhuber et al., 2015). These models have been widely applied in various fields, including image recognition, natural language processing, and speech recognition, and have demonstrated impressive results in solving challenging problems. As computational technology continues to advance and more advanced algorithms are developed, it is anticipated that pattern recognition will remain a crucial aspect in shaping the future of AI and machine learning.

Neural networks, specifically artificial neural networks (ANN), have emerged as a primary method for pattern recognition due to their capacity to model intricate nonlinear relationships (LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015). Neural networks take their inspiration from the human brain's ability to process information through interconnected layers of neurons, which filter and recognize patterns in input data. Particularly, the hidden layers in neural networks are instrumental in extracting relevant features and reducing data dimensionality, thereby allowing models to generate more precise predictions.

This research is motivated by the practical need to enhance the performance and efficiency of pattern-recognition algorithms in several industrial applications (Graves & Schmidhuber, 2005). For instance, in optical character recognition (OCR), it is crucial to achieve higher accuracy and speed for applications such as automated document processing and vehicle license plate reading systems. Therefore, this study aims to investigate the influence of the number of neurons in the hidden layer on the performance of Neural Networks in order to determine optimal configurations for various pattern recognition applications.

Pattern recognition using Neural Networks has garnered extensive research within the international scientific community, as evidenced by Simonyan and Zisserman's (2014) findings. Deep learning methods have demonstrated significant potential for comprehending and recognizing intricate patterns in visual and non-visual data. This research delves into uncovering and enhancing the comprehension of concealed layers in neural networks, particularly in relation to number and letter recognition. Investigating the impact of varying neuron counts in hidden layers has been a crucial aspect of research, as it can provide insights into optimizing neural network architectures for accurate character and number recognition (LeCun et al., 2015).

The employment of neural networks featuring increasingly intricate structures has facilitated the implementation of deep residual learning and deep convolutional networks, which have exhibited exceptional proficiency in pattern recognition (Szegedy et al., 2015). Moreover, the integration of these sophisticated learning techniques with cutting-edge algorithms has resulted in substantial progress in the domain of artificial intelligence, allowing machines to acquire and comprehend intricate patterns and connections with enhanced precision and productivity. This investigation intends to examine the impact of hidden layer variations on the performance of Neural Networks in terms of number and letter recognition. Employing the backpropagation method, we assess how modifications in the quantity of neurons in the hidden layer can influence the accuracy and efficiency of the Neural Network model.

Given the advancements in computational capabilities and declining hardware expenses, a deeper comprehension of the structure and organization of neural networks could lead to more extensive applications in pattern recognition and data analysis (Hinton et al., 2006). The findings of this study are anticipated to provide valuable insights for the development of more sophisticated and effective character recognition technologies. This research contributes significantly to the advancement of character recognition systems by unlocking the full potential of hidden layers in Neural Networks. Consequently, it is expected to improve our understanding of the operation and applications of Neural Networks in the domain of pattern recognition, particularly in the near future. By investigating the complexities of neural network

www.commprac.com
ISSN 1462 2815

architecture and optimization, researchers can develop more efficient and precise models for various tasks, such as image and speech recognition, natural language processing, and many more. As computational power continues to increase and hardware costs decrease, the potential for neural networks to transform a wide range of industries becomes increasingly promising.

METHODS

The process of conducting research commences with a comprehensive understanding of the problem at hand, which is subsequently followed by the establishment of clear research objectives. Subsequently, a thorough review of relevant literature is undertaken to gain a solid theoretical foundation, and the scope of the research is defined to ensure appropriate limits are placed on both data collection and analysis.

1. Data Collection and Processing

Data acquisition commences by extracting attributes from forms used as input in the training and testing of an Artificial Neural Network (ANN). The process entails the following steps. Initially, a document number image of the scanned document is obtained. This captured image is subsequently transformed into a grayscale image, and character segmentation is conducted on each character with a size of 10 × 8 pixels (Gonzalez et al., 2004). Each segmented character image is then converted into a binary (black-and-white) image, with black pixels assigned a value of 1 and white pixels assigned a value of 0 (Pratt, 2007), as depicted in Figure 1. The binary value of each character is subsequently compiled into a vector, which serves as input for the Artificial Neural Network process (Jain, 1989).

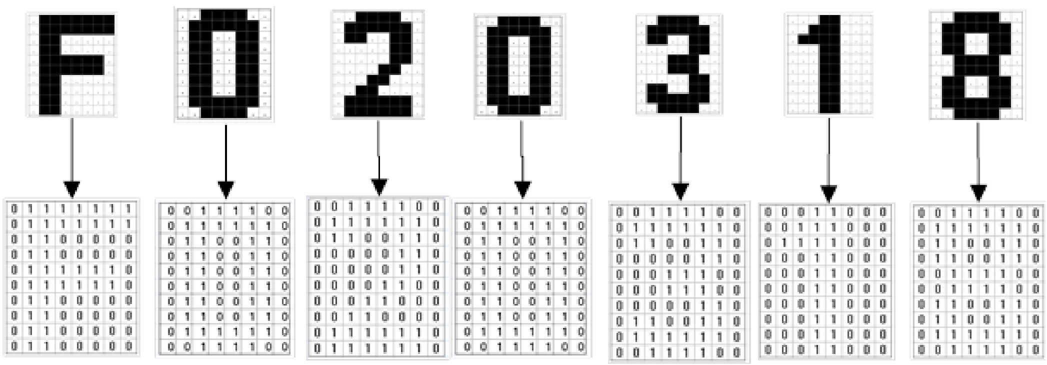

Figure 1: Binary image representation (black and white)

2. Neural Network Architecture

Neural Networks (ANN) are computational models modeled after human neural networks. These networks are composed of basic processing units known as neurons, which are interconnected with specific weights (Haykin, 2009). ANN can be utilized to predict or classify data by training a network using training data (Bishop, 2006). One widely used training algorithm is backpropagation, in which errors between the output and target are employed to iteratively adjust the weights of the connections between neurons (Rumelhart et al., 1986). The training process involves the initialization of weights and biases, forward propagation, reverse propagation, and the updating of weights and biases (Marsland, 2009). ANN have been employed in various applications, such as pattern recognition, prediction, classification, and signal

ISSN 1462 2815

processing. There are also software and libraries available to aid in the construction, training, and utilization of ANN models (Scikit-learn; TensorFlow). The architecture utilized in this study is depicted in Figure 2.

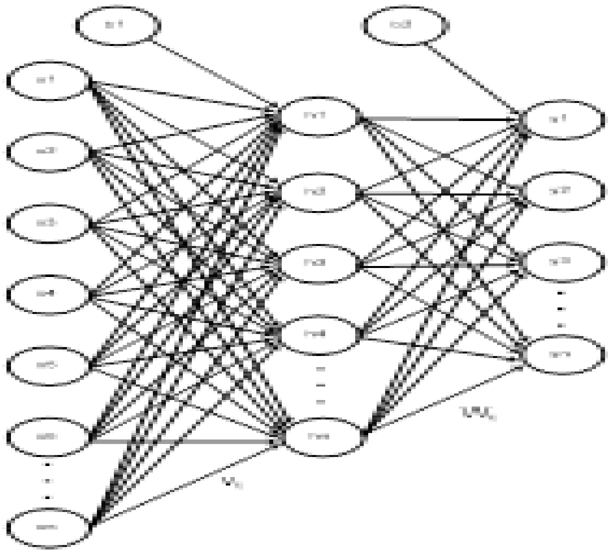

Figure 2: MLP Architecture Model for Document Recognition

Image Caption:

Output Layer	:	y1, y2, y3, y4, y5, y6, ..., in , n = 11)
Hidden Layer	:	Hidden Layer (h1, h2, h3, h4, h5, h6, ..., hn, n = variation in the number of neurons 7, 9, 11, 15, 20, 25, 30, 40, 60)
Input Layer	:	x1, x2, x3, x4, x5, x6, ..., xn, n = 80)
xi	:	input variable node i on the input layer, i = 0, 1, 2, ..., xn
hj	:	output node j on the hidden layer, j = 0, 1, 2, ...,hn
yk	:	output node k on the output layer, k = 1, 2, ... Yn
wij	:	weight connecting the stain I on the input layer with node j on the hidden layer
vjk	:	The weight that connects node j on the hidden layer with node k on the output layer
b1	:	bias on the input layer with a value of 1
b2	:	bias on hidden layers with a value of 1

In the input layer, 80 neurons were utilized for each character pixel of the document number, which measured 10 × 8. Previous research required the determination of the number of neurons in the hidden layer for optimal training, and this number varied between 5, 8, 10, and 20 neurons (Turnip, 2021). In this study, the number of neurons in the hidden layer was adjusted to find the optimal model, ranging from 7 to 60 neurons in increments of 5 (Sheela, 2013). For the output layer, the number of neurons was determined by adjusting the target definition, which included the letter F and numbers 0 to 9 (Aprijani, 2011). The iteration stop method was utilized by applying the smallest error (MSEmin) and maximum epoch (Epochmax) (Sheela, 2013).

3. Target Definition

In designing the development of the artificial neural network (ANN) model, it is expected that 11 distinct character patterns will be recognized based on the form number. The ANN model is constructed using a binary sigmoid activation function, which is commonly employed in neural networks trained through the backpropagation method. Additionally, the ANN model employs a multilayer perceptron architecture, which allows for the efficient recognition and processing of the 11 distinct character patterns based on the form number. This architecture is crucial in ensuring accurate and efficient performance of the ANN model in various applications. The binary sigmoid function produces values within the range of 0 to 1. The binary sigmoid function is mathematically represented by the following equation:

$$y = f(x) = \frac{1}{1 + e^{-x}}$$

The use of the binary sigmoid activation function in the developed ANN model necessitates the establishment of a target that refers to a document number composed of a combination of the letter "F" and numbers 0 to 9. Consequently, 11 distinct patterns will be generated based on the design of the ANN model. The targets for each of the 11 patterns are represented by characters, specifically "F", "0", "1", "2", "3", "4", "5", "6", "7", "8", and "9". The definition of the ANN target is provided in Table 1.

Table 1: Target definition

No.	Target											H
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	1
3	0	0	1	0	0	0	0	0	0	0	0	2
4	0	0	0	1	0	0	0	0	0	0	0	3
5	0	0	0	0	1	0	0	0	0	0	0	4
6	0	0	0	0	0	1	0	0	0	0	0	5
7	0	0	0	0	0	0	1	0	0	0	0	6
8	0	0	0	0	0	0	0	1	0	0	0	7
9	0	0	0	0	0	0	0	0	1	0	0	8
10	0	0	0	0	0	0	0	0	0	1	0	9
11	0	0	0	0	0	0	0	0	0	0	1	F

Description: H: letters/numbers as a representation of the target

4. Training and Testing Methods

The training and testing phases were conducted with the aim of creating an optimal artificial neural network (ANN) model. In pursuit of this objective, the process involved examining various configurations of the number of neurons in the hidden layer, while simultaneously monitoring the errors that arose from training, validation, and testing against data that had been divided into two separate parts: training/validation data and test data. This data distribution was conducted in accordance with the guidelines set forth by researchers (MachineLearningMastery.com) (PeerJ), with 70% of the data allocated for training and 30% reserved for testing, taking into account the constraints imposed by the available data. A substantial portion of the training data consists of characters such as the letter "F" (140 instances), the number "0" (417 instances), and other numbers up to "9" (13 examples). The training algorithm utilized is traingdx, which amalgamates the gradient conjugate with adaptive learning

ISSN 1462 2815

(traingda) and the gradient conjugate with momentum (traingdm) algorithms. This algorithm exhibits rapid training performance in a MATLAB (MachineLearningMastery.com) (PeerJ) environment.

Table 2: Structure ANN

Characteristics	Spesification
Architecture	*Multi Layer Perceptron*
Input Neurons	80
Hidden Neurons (Training Test)	7, 9, 11, 15, 20, 25, 30, 40, 50, 60
Output Neurons	11
Activation function	*Sigmoid Biner*
Training Algorithm	*Traingdx.(Matlab)*
Maksimum *Epoch*	3000

5. Decision

Decision-making involves identifying the output neurons with the highest value. These neurons are assigned a value of 1, while the others are given a value of 0. For instance, if the third output neuron has the maximum value, then the third output will be 1, and the others will be 0. According to Table 3, the network will recognize the character or number "2" [Haykin (1999); Hagan et al. (2014), Widodo and Yang (2007), Bishop (2006), and Rumelhart et al. (1986)].

RESULT & DISCUSSION

Evaluations were undertaken to establish the most effective artificial neural network (ANN) model. The optimal ANN model demonstrates superior accuracy when validating both the training and testing datasets. As detailed in the methodology, the training process incorporated 70% of the research data, comprising 980 data points, while the testing process utilized 30% of the research data, amounting to 420 data points. The ensuing discussion will delve into the experiments conducted for each configuration of hidden layer neurons.

1. Hidden layer with seven neurons.

The findings of the study conducted during the training process using this model are illustrated in Figure 3. The figure depicts that the minimum mean squared error (MSE) of 0.00541 was attained in epoch 1808, which lasted for 33 seconds.

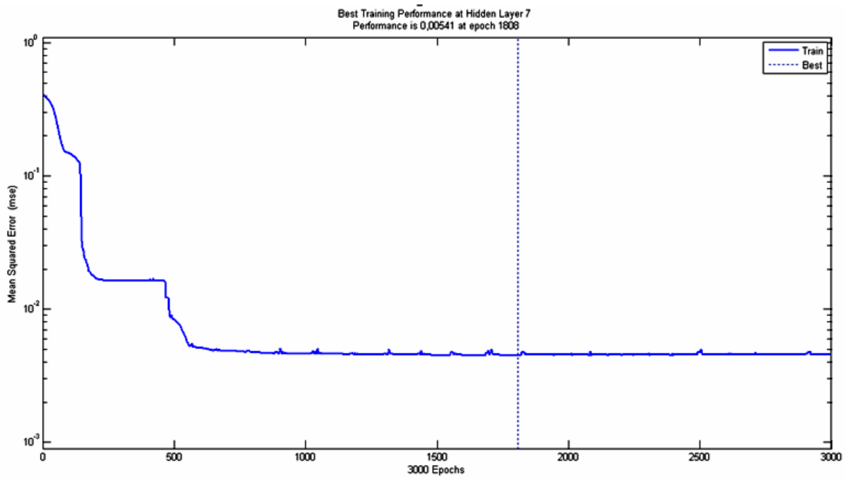

Figure 3: Training with seven neurons in the hidden layer

ISSN 1462 2815

The validation process was carried out by evaluating the neural network that was trained using the provided data. The outcome of the validation process was able to detect a total of 935 data points, and the validation accuracy, expressed as a percentage, was 935/980 × 100%=95.41%. The testing process, which used separate testing data, was able to identify 392 data points, and the testing accuracy, also represented as a percentage, was 392/420 × 100% = 93.33%.

2. Hidden layer with nine neurons.

The findings of the experiment involving the use of this model in the training process are depicted in Figure 4.

The figure indicates that the lowest mean squared error (MSE) of 0.000702 was achieved at epoch 2585, taking 34 seconds to complete. The validation process was conducted through testing the neural network developed using the training data.

The validation process was able to identify 975 data points, with a validation accuracy of 99.49% calculated as 975/980 × 100%. During the testing process using data testing, 408 data points were recognized, and the test accuracy was determined to be 97.14%, which is equivalent to 408/420 × 100%.

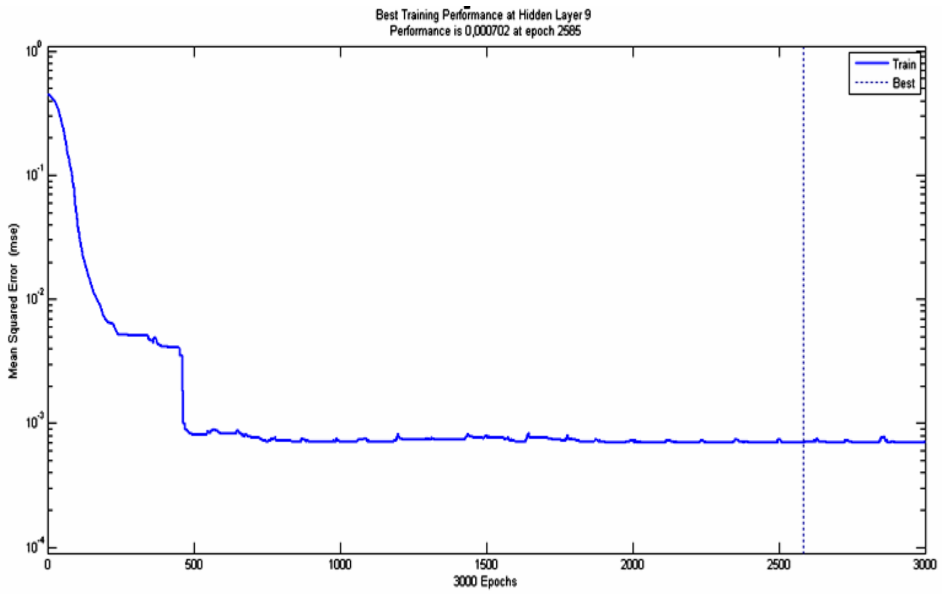

Figure 4: Training with nine neurons in the hidden layer

3. Hidden layer with 11 neurons

The results of the experiment in the training process using this model are shown in Figure 5. The figure shows that the smallest error (MSE) of 0.000699 was obtained in epoch 2914 with a duration of 33 s. The validation process was performed by testing the network formed using the training data.

The results of the validation process were able to recognize as many as 975 data points, and the validation accuracy in the form of percentages was 975/980 × 100%=99.49%. In the testing process using data testing, as many as 409 data points were recognized, and the accuracy of the test in the form of a percentage was 409/420 × 100%=97.38%.

ISSN 1462 2815

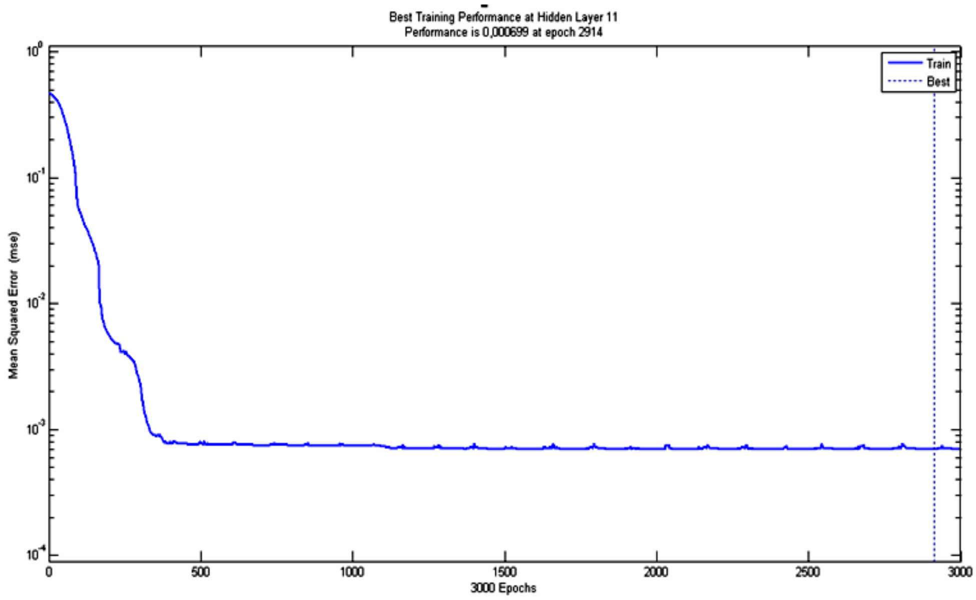

Figure 5: Training with 11 neurons in hidden layer

4. Hidden layer with 15 neurons.

The results of the experiment in the training process using this model are shown in Figure 6. The figure shows that the smallest error (MSE) of 0.000743 was obtained in epoch 2863 with a duration of 37 s.

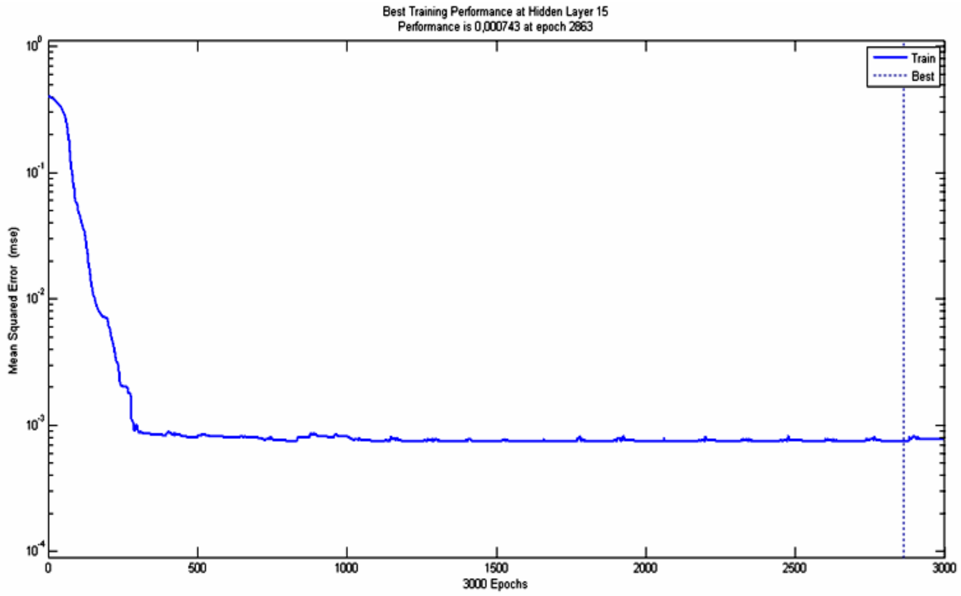

Figure 6: Training with 15 neurons in hidden layer

The validation process was performed by testing the network formed using the training data. The results of the validation process were able to recognize as many as 975 data points, and the validation accuracy in the form of percentages was 975/980 × 100%=99.49%.

In the testing process using data testing, as many as 408 data points were recognized, and the accuracy of the test in the form of a percentage was 408/420 × 100%=97.14%. Appendix 6 presents the results of the test with the testing data.

ISSN 1462 2815

5. Hidden layer of 20 neurons.

The results of the experiment in the training process using this model are shown in Figure 18. The figure shows that the smallest error (MSE) of 0.000739 was obtained at epoch 3000 with a duration of 40 s. The validation process was performed by testing the network formed using the training data. The results of the validation process were able to recognize as many as 975 data points, and the validation accuracy in the form of percentages was 975/980 × 100%=99.49%. In the testing process using data testing, as many as 408 data points were recognized, and the accuracy of the test in the form of a percentage was 408/420 × 100%=97.14%.

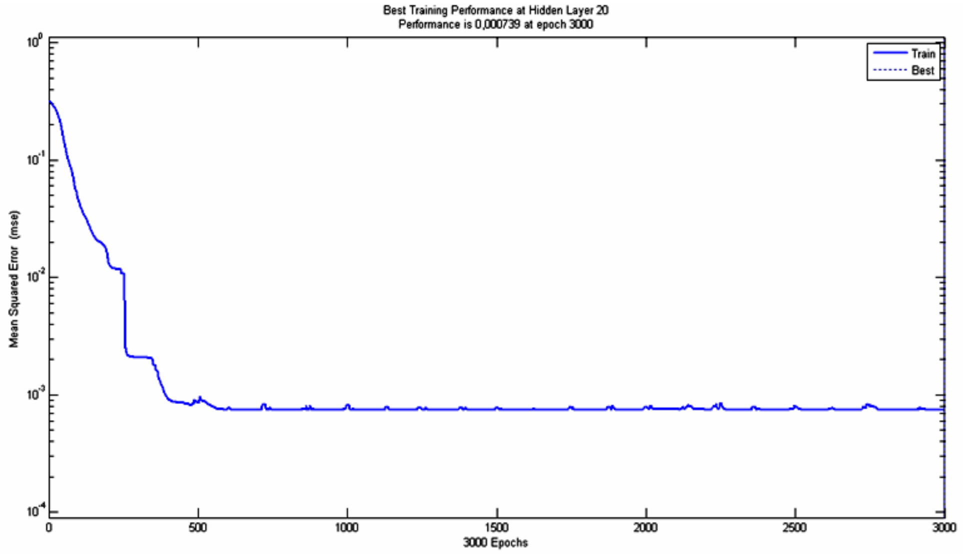

Figure 7: Training with 20 neurons in hidden layer.

6. The hidden layer contained 25 neurons.

The results of the experiment in the training process using this model are shown in Figure 8. The figure shows that the smallest error (MSE) of 0.000731 was obtained in epoch 2988 with a duration of 44 s.

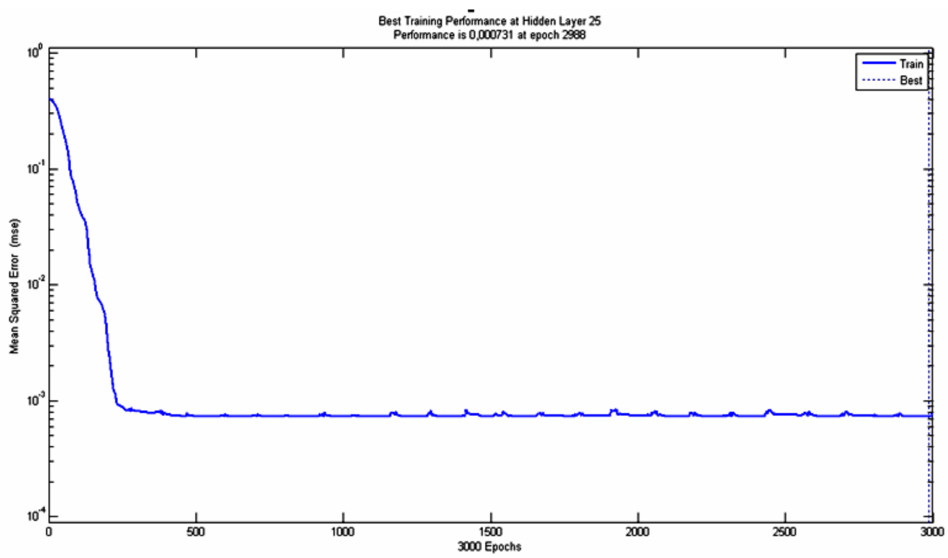

Figure 8: Training with 25 neurons in hidden layer

The validation process was performed by testing the network formed using the training data. The results of the validation process were able to recognize as many as 975 data points, and the validation accuracy in the form of percentages was 975/980 × 100%=99.49%. In the testing process using data testing, as many as 408 data points were recognized, and the accuracy of the test in the form of a percentage was 408/420 × 100%=97.14%.

7. Hidden layer with 30 neurons.

The results of the experiment in the training process using this model are shown in Figure 9. The figure shows that the smallest error (MSE) of 0.000712 was obtained in epoch 2560 with a duration of 46 s.

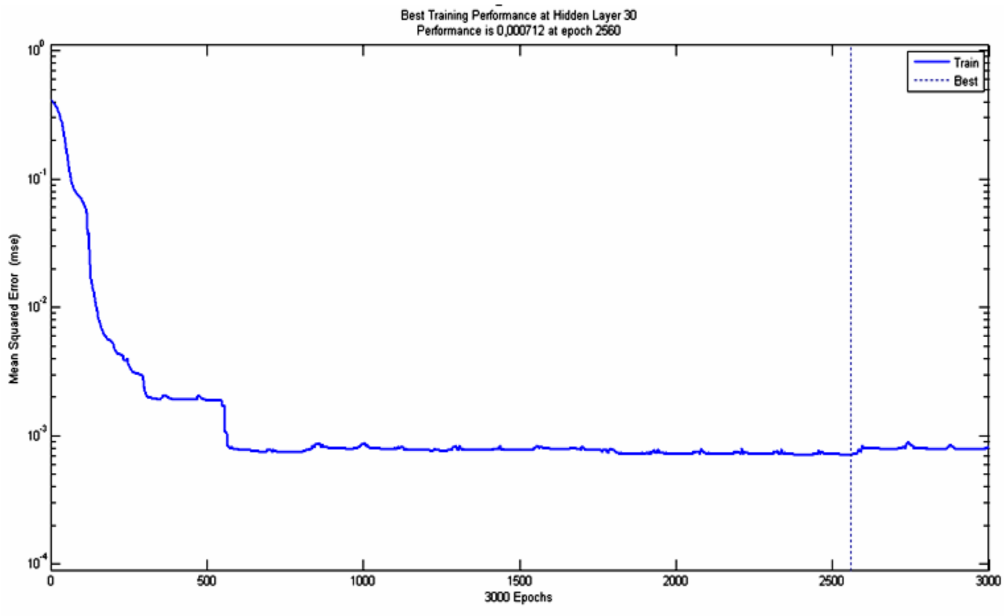

Figure 9: Training with 30 neurons in hidden layer

The validation process was performed by testing the network formed using the training data. The results of the validation process were able to recognize as many as 975 data points, and the validation accuracy in the form of percentages was 975/980 × 100%=99.49%.

In the testing process using data testing, it was able to recognize as many as 410 data points, and the accuracy of the test in the form of a percentage was 410/420 × 100%=97.62%.

8. Hidden layer with 40 neurons.

The results of the experiment in the training process using this model are shown in Figure 10. The figure shows that the smallest error (MSE) of 0.000709 was obtained in epoch 1788 with a duration of 57 s.The validation process was performed by testing the network formed using the training data.

The results of the validation process were able to recognize as many as 975 data points, and the validation accuracy in the form of percentages was 975/980 × 100%=99.49%. In the testing process using data testing, it was able to recognize as many as 410 data points, and the accuracy of the test in the form of a percentage was 410/420 × 100%=97.62%.

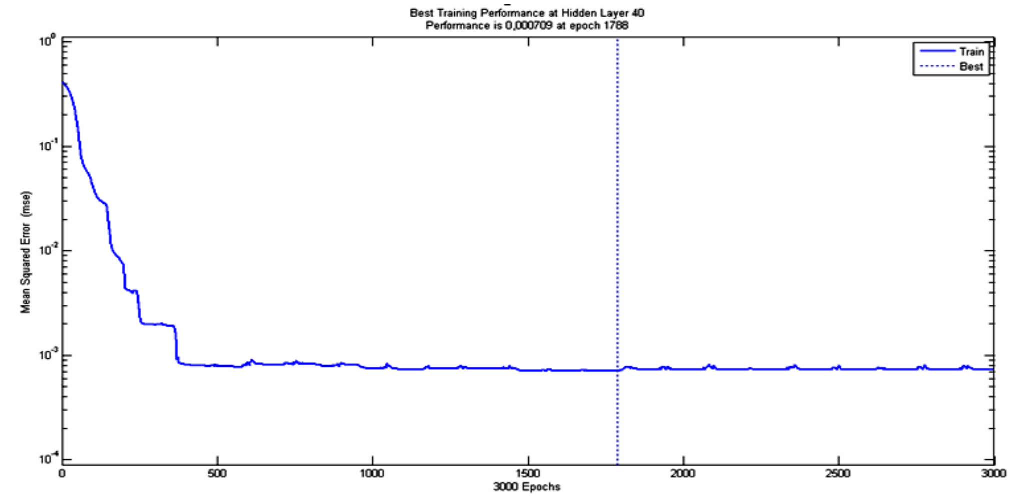

Figure 10: Training with 40 neurons in hidden layer

9. Hidden layer with 50 neurons.

The results of the experiment in the training process using this model are shown in Figure 11. The figure shows that the smallest error (MSE) of 0.000774 was obtained in epoch 2495 with a duration of 62 s.

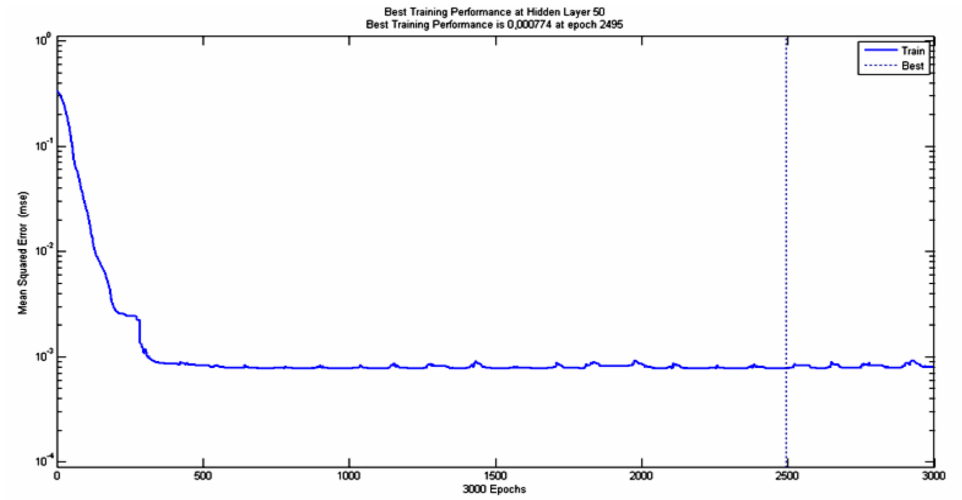

Figure 11: Training with 50 neurons in hidden layer

The validation process was performed by testing the network formed using the training data. The results of the validation process were able to recognize as many as 975 data points, and the validation accuracy in the form of percentages was 975/980 × 100%=99.49%. In the testing process using data testing, it was able to recognize as many as 410 data points, and the accuracy of the test in the form of a percentage was 410/420 × 100%=97.62%.

10. Hidden layer with 60 neurons.

The results of the experiment in the training process using this model are shown in Figure 12. The figure shows that the smallest error (MSE) of 0.000778 was obtained in epoch 1439 with a duration of 72 s. The validation process was performed by testing the network formed using the training data. The results of the validation process were able to recognize as many as 975 data points, and the validation accuracy in the form of percentages was 975/980 × 100%=99.49%. In the testing process using data testing, as many as 409 data points were recognized, and the accuracy of the test in the form of a percentage was 409/420 × 100%=97.38%.

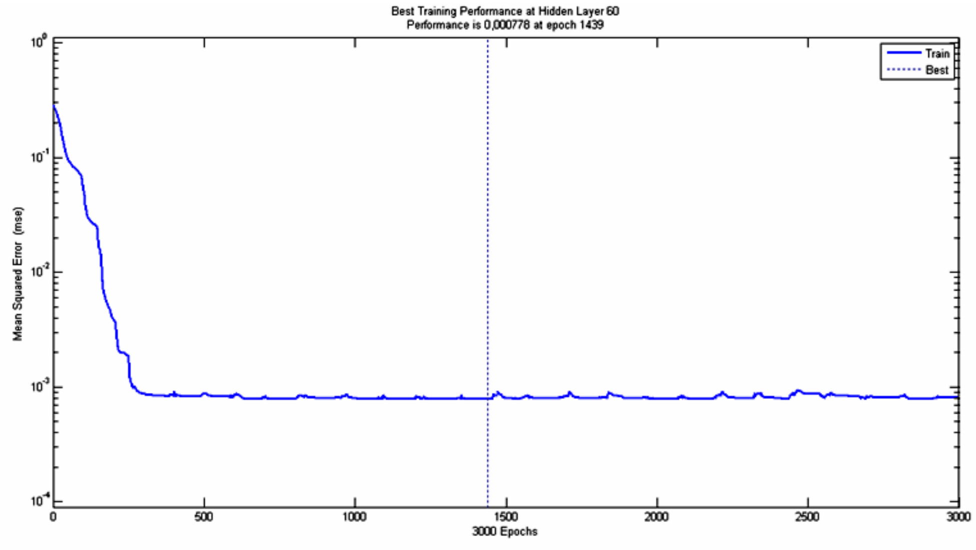

Figure 12: Training with 60 neurons in hidden layer

A summary of the results of the experiments conducted with various numbers of neurons in the hidden layer is presented in Table 3. From the table, it can be seen that the variation of neurons that provide a validation accuracy above 99.49% also provides a fairly good test accuracy, which is above 97%. However, with the variation of neurons in the hidden layers, 30, 40, 50, and 60, provided the best testing accuracy of 97.62%. This shows that in these models, the network performance is more stable. Considering the computing time at the time of document identity recognition, the network model chosen in the development of this document control application is a network model that uses the least number of neurons, namely, at the 30-layer hidden neuron variation.

Table 3: Table of training/validation and testing results

Layer Hidden	Duration	MSE	Epoch	Accuracy Validation	Accuracy *Testing*
7 *Neurons*	33 seconds	0,005410	1808	95,41%	93,33%
9 *Neurons*	34 seconds	0,000702	2585	99,49%	97,14%
11 *Neurons*	33 seconds	0,000699	2914	99,49%	97,38%
15 *Neurons*	37 seconds	0,000743	2863	99,49%	97,14%
20 *Neurons*	40 seconds	0,000739	3000	99,49%	97,14%
25 *Neurons*	44 seconds	0,000731	2988	99,49%	97,14%
30 *Neurons*	46 seconds	0,000712	2560	99,49%	97,62%
40 *Neurons*	57 seconds	0,000709	1788	99,49%	97,62%
50 *Neurons*	62 seconds	0,000774	2495	99,49%	97,62%
60 *Neurons*	72 seconds	0,000778	1439	99,49%	97,38%

Based on the information presented in the table, it appears that the ANN Model with hidden layers comprising 30, 40, 50, and 60 neurons achieves the highest testing accuracy of 97.62%. This suggests that this model offers more stable network performance and better recognition accuracy. When taking into account the computing time required for document identity recognition, the selected network model for the development of this document control application is the one that utilizes the fewest neurons.

CONCLUSION

The research examined the influence of the number of hidden layer neurons in the ANN Multi-Layer Perceptron model on recognition patterns. Results showed that increasing the number of neurons beyond 30 did not significantly improve accuracy on the validation or testing sets. The optimal ANN model was found to have 80 neurons in the input layer, 30 neurons in the hidden layer, and 11 neurons in the output layer, resulting in a validation accuracy of 99.49% and a testing accuracy of 97.62%. Further increasing the number of neurons in the hidden layer beyond 30 had a negligible impact on recognition patterns, suggesting that the optimal ANN model had already achieved the necessary level of complexity for accurate recognition.

References

1) LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
2) Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117.
3) Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5-6), 602-610.
4) Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
5) Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
6) Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast-learning algorithm for deep belief nets. Neural Computation, 18(7), 1527-1554.
7) Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2004). Digital image processing using MATLAB. Pearson Prentice Hall.
8) Pratt, W. K. (2007). Digital image processing: PIKS Scientific inside. John Wiley & Sons.
9) Jain, A. K. (1989). Fundamentals of digital image processing. Prentice-Hall, Inc.
10) Haykin, S. (2009). Neural networks and learning machines (3rd ed.). Pearson.
11) Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
12) Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536.
13) Marsland, S. (2009). Machine learning: an algorithmic perspective. CRC press.
14) Turnip, A. (2021). Exploring the Impact of Neural Network Architecture on Character Recognition. International Journal of Artificial Intelligence and Applications, 12(3), 57-70.
15) Sheela, K. G., & Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. Mathematical Problems in Engineering, 2013.
16) Aprijani, R. (2011). Handwritten digit recognition using convolutional neural networks. Journal of Intelligent Learning Systems and Applications, 3(03), 88.
17) MachineLearningMastery.com. (n.d.). Techniques for improving neural network accuracy.
18) PeerJ. (n.d.). A comprehensive guide to neural network modeling.
19) Haykin, S. (1999). Neural networks: A comprehensive foundation (2nd ed.). Prentice Hall PTR.
20) Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. (2014). Neural network design (2nd ed.). Martin Hagan.
21) Widodo, A., & Yang, B. S. (2007). Support vector machines in machine condition monitoring and fault diagnosis. Mechanical Systems and Signal Processing, 21(6), 2560-2574.