Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 3 (2022) 397 - 402 ISSN Media Electronic: 2580-0760

The Formula Study in Determining the Best Number of Neurons in Neural Network Backpropagation Architecture with Three Hidden Layers

Syaharuddin[1], Fatmawati[2], Herry Suprajitno[3]
[1,2,3]Department of Mathematics, Faculty of Science and Technology, Universitas Airlangga, Surabaya, Indonesia
[1]syaharuddin-2020@fst.unair.ac.id, [2]fatmawati@fst.unair.ac.id, [3]herry-s@fst.unair.ac.id

Abstract

The researchers conducted data simulation experiments, but they did so unstructured in determining the number of neurons in the hidden layer in the Artificial Neural Network Back-Propagation architecture. The researchers also used a general architecture consisting of one hidden layer. Researchers are still producing minimal research that discusses how to determine the number of neurons when using hidden layers. This article examines the results of experiments by conducting training and testing data using seven recommended formulas including the Hecht-Nelson, Marchandani-Cao, Lawrence & Fredrickson, Berry-Linoff, Boger-Guterman, JingTao-Chew, and Lawrence & Fredrickson modifications. We use rainfall data and temperature data with a 10-day type for the last 10 years (2012-2021) sourced from Lombok International Airport Station, Indonesia. The training and testing data used showed the results that in determining the number of neurons on the hidden-1 screen, it was more appropriate to use the Hecht-Nelson formula and the Lawrence & Fredricson formula which is more suitable for use in the 2nd & 3rd hidden layer. The resulting research was able to provide an accuracy rate of up to 97.79% (temperature data) and 99.94% (rainfall data) with an architecture of 36-73-37-19-1.

Keywords: Neural Network, Backpropagation, 3-Layer Hidden, Number of Neurons

1. Introduction

Generally, the network architecture of Backpropagation consists of three layers, namely the input layer, the hidden layer, and the output layer [1]. The process of determining the number of neurons in the input layer and output layer is not too difficult because it depends on the number of inputs and the desired amount of output. This is not the same as the number of neurons in the hidden layer. Deciding how many neurons to use in the hidden layer is one of the most important properties of a neural network (NN). If the number of neurons is too small, the NN cannot model complex data and the results may not be acceptable. Using too many neurons will not only increase the training time, but also degrade the performance of the NN [2]. Therefore, many researchers conduct experiments to determine the optimal number of neurons. The number of neurons used during the data training process becomes an important point in building the architecture network of Back Propagation in order to obtain high accuracy. The architecture found in the data training process will be used as a reference for the prediction process.

Park [3] has written three formulas that can be used in determining the number of neurons in the hidden layer, namely, the formulas of Hecht-Nelson (1987), Lawrence-Fredrickson (1988), and Marchandani-Cao (1989). In addition, there are also formulas that are often used by the other researchers, namely, the Berry-Linoff (1997), Boger-Guterman (1997), JingTao-Chew (2001), and Lawrence-Fredrickson modified formula. These formulas calculate the number of neurons in the first hidden layer by looking at the number of neurons in the input layer and the output layer. So, the calculation is not for the second, third, and so on hidden layers. These formulas are divided into two classifications, namely (1) the number of neurons in the input layer that is greater than the hidden layer and (2) the number of neurons in the input layer that is smaller than the hidden layer. However, there are also the researchers who use architectures with the number of neurons in the hidden layer equal to the number of neurons in the input layer [4], [5].

Some studies on the prediction of time series data using NNBP architecture with one hidden layer has been widely done namely classification of Australian credit card [6], diabetic detection [7], identification for a single-shaft gas turbine [8], particle swarm optimization [9], measuring the severity of osteoarthritis [10], and

Accepted: 26-04-2022 | Received in revised: 19-06-2022 | Published: 30-06-2022

classification of acute myelogenous leukemia [11]. Nawi et al. conducted experiment [7] with three types of data, namely Australian credit cards with architecture 14-5-1, diabetes mellitus with architecture 193-10-2, and glass dataset with architecture 13-10-2. Likewise, Asgari et al. conducted experiment [9] with architecture 40-20-1, and obtained a coefficient of correlation value of 0.990. The average architecture they used is that the number of neurons in the hidden layer is smaller than the number of neurons in the input layer. Furthermore, the results of research conducted by Bay et al. [4] and Jayalakshmi & Santhakumaran [5] achieved accuracy rates of 92.6% and 72.55%. This result is obtained with the number of neurons in the input layer and the hidden layer being the same.

In addition, the architecture with a greater number of neurons in the input layer than in the hidden layer has been used by Suryani et al. [12] for the prediction of acute myelogenous leukemia, by Gowda & Mayya [13] for stream-flow prediction, and by Abdulkadir et al. [14] for moisture prediction in maize. They used architecture for relatively small amounts of data, such as 6-8-2, 5-10-1, 2-5-1, and 1-4-1 architectures. The accuracy rate resulted in an average of 0.931. This result is higher compared to architectures with the number of neurons in the input layer greater than that in the hidden layer, such as 13-10-2 [7], 40-20-1 [9], and 24-13-1 [10]. However, against a larger amount of input data, it certainly cannot be drawn from this case.

The development of neural networks continues to be carried out by the researchers with conducting experiments with two hidden layers. However, research on methods using three hidden layers and determining the number of neurons in the hidden layer has not been intensively studied, since increasing the number of hidden layers means duration of the data training process. Determination of the number of neurons in the hidden layer is certainly not done. Calculations should be based on existing rules that have been tested for the results of experiments both for the case of two or three hidden layers. Unlike Karsoliya [15] has recommended determining the number of neurons that should fit the criteria of Berry & Linoff [16], did not conduct experiments regarding the number of neurons used. Therefore, there has been no in-depth study of how to calculate the number of neurons in the NNBP architecture with two or three hidden layers based on existing formulas. In this article, we calculated the number of neurons in the first hidden layer, the second hidden layer, and the third hidden layer based on the formulas mentioned. We used two types of data with different patterns and trends, where rainfall data has extreme trends and temperature data has monotonous trends. The results of this study are expected to provide an easier concept in determining the number of neurons in the NNBP architecture with more hidden layers.

2. Research Methods

2.1 Formula for Determining the Number of Neurons in the Hidden Layer

Generally, the number of neurons in the hidden layer is divided into two types, namely: the number of neurons in the input layer that is greater than the hidden layer $(N_x > N_z)$; and the number of neurons in the input layer that is smaller than the hidden layer $(N_x < N_z)$. According to Table 1, some researchers have proposed a formula for calculating the number of neurons in the hidden layer, with N_z is the number of hidden layer neurons, N_x is the number of input layer neurons, N_y is the number of neurons of the output layer, and N_t is the number of training data.

Table 1. Formula for calculating the number of hidden layer neurons

No	Formula	Source
P1	$N_z = 2 \cdot N_x + 1$	[17], [18], [3]
P2	$N_z = \log_2 N_x$	[19], [3]
P3	$N_z = \dfrac{N_x + N_y}{2}$	[20] [21], [3], [15], [22]
P4	$N_z = \dfrac{2}{3} N_x$	[16], [15]
P5	$N_z = 2 \cdot N_x$	[23] [15]
P6	$N_z = \dfrac{N_x}{2} + 1$	[24]
P7	$N_z = \dfrac{1}{2}(N_x + N_y) + \sqrt{N_t}$	[25]

If we look at the architectural construction results of some research results, the number of these neurons is only for the hidden-1 layer, especially the equations P1, P2 and P5. That is, the process of determining the number of neurons in the second and third hidden layers does not apply. The use of a two-layer hidden architecture where the number of neurons in the hidden layer is higher than the number of neurons in the input layer has been carried out by Mislan et al. [26] in monthly rainfall predictions. However, the method of calculating the number of neurons in the second hidden layer uses the third equation model (P3).

2.2 Dataset

The data used in the data training process is were rainfall data and temperature data taken from Lombok International Airport station (latitude: -8.560555556 and longitude: 116.0938889). The data taken were the data over the last 10 years (2012-2021) with 10 days of data types so that the total input data (N_x) were 36 data. However, we divided the data into 80% for training (2012-2019) and 20% for testing (2020-2021). So that the total input data (N_t) in the training process were 288 data (36 x 8 years) and the testing process were 72 data (36 x 2 years). Meanwhile, the number of neurons in the output layer (N_y) was 1. In this study, only a data

DOI: https://doi.org/10.29207/resti.v6i3.4049

training process was carried out to see the accuracy of each equation (P1-P7) with a different number of neurons in each hidden layer. The data training process was carried out by each data, so there were 14 data training processes.

2.3 The Architecture of Backpropagation

In this study, we used the NNBP architecture with three hidden layers, the learning rate of 0.1, the momentum of 0.9, the training function is trainlm, activation function on each layer namely logsig-logsig-logsig-purelin, the maximum epoch of 1,000 (show step epoch of 50), and target error of 0.001. Meanwhile the number of neurons in each hidden layer was determined according to the equation in Table 1, because the number of neurons in the first hidden layer is affected by the number of neurons in the input layer, the number of neurons in the second hidden layer is affected by the number of first hidden layers, and so on. Therefore, the combination is obtained according to Table 2.

Table 2 shows that the data training process will be carried out seven times according to their respective architectures. The results of the calculation of the number of neurons show that the architectures of P1, P2, P3, P4, P5, and P6 have the same pattern, namely the number of neurons in the next hidden layer is getting smaller, except for P7 where each hidden layer the number of neurons is the same. Furthermore, the accuracy level indicators used in this study are acquired iteration (epoch), mean square error (MSE), the coefficient of correlation, and accuracy ($R^2 \times 100$).

3. Result and Discussions

The simulation used applications that have been developed using Graphical User Interface (GUI) of MATLAB with a three-layer hidden architecture. This was done to facilitate the data training and testing process. In the first stage, we input the data and the number of neurons on each layer. Then, we select a combination of activation functions, select the training function, and input training parameters such as maximum epoch, learning rate, momentum, and maximum error. Finally, we conducted data training for each architecture and tabulate each training result.

The data training process was carried out seven times each by inputting the number of neurons of each hidden layer. Since, the number of equations is seven (Table 2), the number of training and testing data is seven times. The training and testing process is performed step-by-step based on the number of neurons in each architecture. Then each result was tabulated and the graph modelled the actual data and prediction data. The simulation results are presented in Table 3 and Table 4.

The results of training and testing data in Table 3 and Table 4 show that each formula has a different of accuracy rate. Each formula or architecture produces a different number of epochs, MSE values, and accuracy rate. The number of epochs in each data obtained an average of fewer than 15 iterations, this is according to the maximum target of 1000 epochs. Table 3 shows that the architecture with the highest accuracy rate in the training process was architecture 36-35-35-35-1 which is formed from the seventh equation (P7), in training process with epoch of 7, MSE of 0.0243, correlation coefficient of 0.9906, and accuracy of 98.13%; and in testing process with epoch of 2, MSE of 5.82×10^{-4}, correlation coefficient of 0.9992, and accuracy of 99.84%. But the number of iterations was still greater compared to the number of iterations of the first equation (P1), and the difference in accuracy rate was not too high, which was 0.34%. Hence, in the testing process, the accuracy rate of the first equation (P1) is higher than seventh equation (P7), in testing process with epoch of 4, MSE of 1.63×10^{-6}, correlation coefficient of 0.9999, and accuracy of 99.99%.

The training results in Table 4, it can be seen that the architecture formed by the first equation (P1) in the training process had the smallest MSE value of 0.5307 with 6 epoch, correlation coefficient 0.9997, and accuracy rate of 99.94%; in the testing process obtained an MSE value of 0.00088 with 4 epoch, correlation coefficient of 0.9999, and accuracy rate of 99.99%. Therefore, we recommend the architecture used which was 36-73-37-19-1.

Furthermore, if you look at the results of the MSE and the accuracy rate, the architecture can be sorted from the most accurate, namely P1-P7-P5-P4-P6-P3-P2. This result shows that the architecture 36-8-5-3-1 calculation of the second equation (P2) provided the lowest accuracy rate and the highest number of epochs. Therefore, the P2 architecture is not recommended for use in the data prediction process. The graph model of the actual data approach with forecasting data from the results of the training process using the first equation can be seen in Figure 1 and Figure 2.

From Figure 1 to Figure 2 it can be seen that the actual data (blue) approach model and predictions (red) were already good. Figure 1 shows static data patterns, so the actual data is easily approached by predictive data. This result has implications for the MSE value of the temperature training data for a relatively low, average of 0.039. While, Figure 2 shows a data pattern with a larger range, so that the MSE value obtained is relatively large, average of 0.827.

Therefore, the predicted temperature in 2022 (Figure 1) averages 26.22°C, the maximum temperature appears in the second 10-days in December at 28.34°C, and the minimum temperature appears in the third 10-days in August at 24.34°C.

Table 2. Results of calculation of the number of neurons in the hidden layer

No	N_x	N_{z1}	N_{z2}	N_{z3}	Architecture
P1	36	73	37	19	36-73-37-19-1
P2	36	$8{,}16 \approx 8$	$4{,}58 \approx 5$	$2{,}79 \approx 3$	36-8-5-3-1
P3	36	$18{,}5 \approx 19$	$9{,}75 \approx 10$	$5{,}375 \approx 5$	36-19-10-5-1
P4	36	24	16	$10{,}67 \approx 11$	36-24-16-11-1
P5	36	72	$36{,}5 \approx 37$	$18{,}75 \approx 19$	36-72-37-19-1
P6	36	19	$10{,}5 \approx 11$	$6{,}25 \approx 7$	36-19-11-7-1
P7	36	$35{,}47 \approx 35$	$35{,}20 \approx 35$	$35{,}07 \approx 35$	36-35-35-35-1

Table 3. Temperature data training and testing results

No	Architecture	Training				Testing			
		Epoch	MSE	R	Accuracy (%)	Epoch	MSE	R	Accuracy (%)
P1	36-73-37-19-1	6	0.0291	0.9889	97.79	4	1.63×10^{-6}	0.9999	99.99%
P2	36-8-5-3-1	12	0.0455	0.9815	96.33	10	4.63×10^{-6}	0.9999	99.99%
P3	36-19-10-5-1	12	0.0486	0.9794	95.92	5	5.64×10^{-4}	0.9993	99.86%
P4	36-24-16-11-1	13	0.0419	0.9849	97.00	5	1.27×10^{-5}	0.9999	99.99%
P5	36-72-37-19-1	6	0.0472	0.9837	96.77	3	2.33×10^{-4}	0.9998	99.96%
P6	36-19-11-7-1	12	0.0355	0.9878	97.57	5	3.65×10^{-4}	0.9995	99.90%
P7	36-35-35-35-1	7	0.0243	0.9906	98.13	2	5.82×10^{-4}	0.9992	99.84%

Table 4. Rainfall data training and testing results

No	Architecture	Training				Testing			
		Epoch	MSE	R	Accuracy (%)	Epoch	MSE	R	Accuracy (%)
P1	36-73-37-19-1	6	0.5307	0.9997	99.94	4	0.00088	0.9999	99.99%
P2	36-8-5-3-1	14	6.9610	0.9973	99.46	16	0.762	0.9998	99.96%
P3	36-19-10-5-1	10	1.3062	0.9995	99.90	7	2.694	0.9997	99.94%
P4	36-24-16-11-1	8	0.5482	0.9996	99.92	5	1.686	0.9997	99.94%
P5	36-72-37-19-1	6	0.8470	0.9998	99.96	3	1.982	0.9998	99.96%
P6	36-19-11-7-1	11	0.9226	0.9994	99.88	9	0.386	0.9995	99.90%
P7	36-35-35-35-1	5	0.8097	0.9996	99.92	4	0.0065	0.9999	99.99%

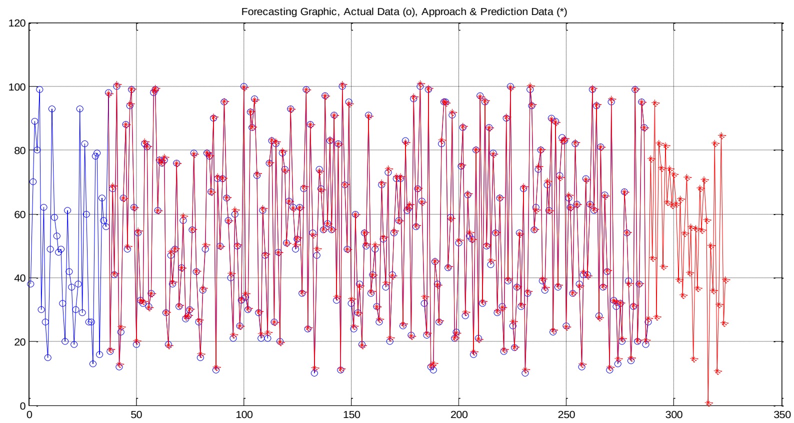

Figure 1. Actual data approach and predictions of temperature data

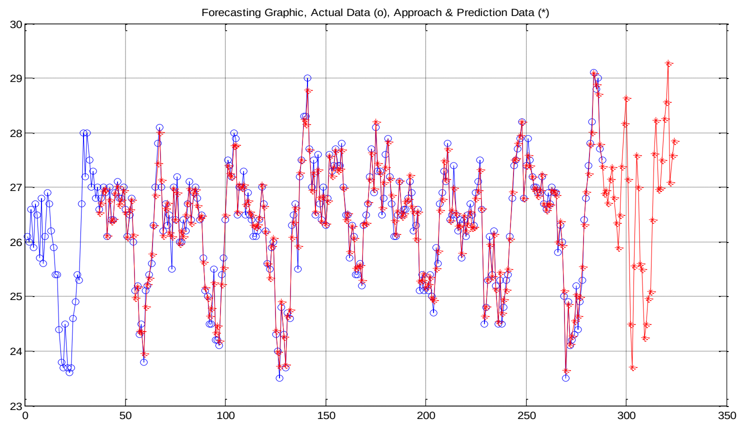

Figure 2. Actual data approach and predictions of rainfall data

DOI: https://doi.org/10.29207/resti.v6i3.4049

Hence, the rainfalls prediction results in 2022 (Figure 2) show that the rainfalls in December is the highest at 456.19 mm (wet category), and the rainfalls in September is the lowest at 40.88 mm (dry category). This recommended architectural model by Hecht-Nelson has been widely used in data prediction, where the number of neurons in the hidden layer is greater than the number of neurons in the input layer [27], [28], [29], [2]. After looking at the simulation results in Table 3 and Table 4, it can be seen that the first formula, the fifth formula, and the seventh formula give almost the same accuracy results. However, the output of the first equation has higher accuracy rate in both given cases. Therefore, we recommend that the number of neurons in the first hidden layer is determined using the Hecht-Nelson formula (P1) while the second and third hidden layers is determined using the Lawrence-Fredrickson (P3) formula.

4. Conclusions

During data training and testing, the number of hidden layers has a great impact on improving the performance of the NNBP network. The same applies to the number of neurons that must be used when building an architecture with a number of hidden layers greater than one. The training and testing data results for temperature and rainfalls show that the seven formulas offered had different accuracy rate. However, the first formula with the architecture 36-73-37-19-1 showed the highest accuracy rate when training temperature data of 97.79% and 99.99% for testing data, while in the process of rainfall data with first formula obtained accuracy rate of 99.94% for training data and 99.99% for testing data. Hence, from this architecture obtained predictions of the highest rainfall in 2022 occurred in December and the lowest in September, and the temperature prediction obtained an average of 26.22^{0}C. Therefore, we recommend the Hecht-Nelson formula for counting the number of neurons on the first hidden layer and for the counting of neuron on the second and third hidden layers, we recommend the Lawrence-Fredrickson formula. These results open opportunities for new research to make modifications or combinations of the entire formulas to find the best architecture through many experiments and data training.

References

[1] Haviluddin and R. Alfred, "A genetic-based backpropagation neural network for forecasting in time-series data," in *Proceedings - 2015 International Conference on Science in Information Technology: Big Data Spectrum for Future Information Economy, ICSITech 2015*, 2016, pp. 158–163, doi: 10.1109/ICSITech.2015.7407796.

[2] H. Azami, M. R. Mosavi, and S. Sanei, "Classification of GPS satellites using improved back propagation training algorithms," *Wirel. Pers. Commun.*, vol. 71, no. 2, pp. 789–803, 2013, doi: 10.1007/s11277-012-0844-7.

[3] H. Park, "Study for Application of Artificial Neural Networks in Geotechnical Problems," in *Artificial Neural Networks - Application*, Croatia: InTech, 2011, pp. 303–336.

[4] Y. Bai, Y. Li, X. Wang, J. Xie, and C. Li, "Air pollutants concentrations forecasting using back propagation neural network based on wavelet decomposition with meteorological conditions," *Atmos. Pollut. Res.*, vol. 7, no. 3, pp. 557–566, 2016, doi: 10.1016/j.apr.2016.01.004.

[5] T. Jayalakshmi and A. Santhakumaran, "Statistical Normalization and Back Propagationfor Classification," *Int. J. Comput. Theory Eng.*, vol. 3, no. 1, pp. 89–93, 2011, doi: 10.7763/ijcte.2011.v3.288.

[6] R. Bagaria, S. Wadhwani, and A. K. Wadhwani, "Bone fractures detection using support vector machine and error backpropagation neural network," *Optik (Stuttg).*, vol. 247, 2021, doi: 10.1016/j.ijleo.2021.168021.

[7] N. M. Nawi, F. Hamzah, N. A. Hamid, M. Z. Rehman, M. Aamir, and A. A. Ramli, "An optimized back propagation learning algorithm with adaptive learning rate," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 5, pp. 1693–1700, 2017, doi: 10.18517/ijaseit.7.5.2972.

[8] S. N. Endah, A. P. Widodo, M. L. Fariq, S. I. Nadianada, and F. Maulana, "Beyond back-propagation learning for diabetic detection: Convergence comparison of gradient descent, momentum and Adaptive Learning Rate," in *Proceedings - 2017 1st International Conference on Informatics and Computational Sciences, ICICoS 2017*, 2017, vol. 2018-January, pp. 189–193, doi: 10.1109/ICICOS.2017.8276360.

[9] H. Asgari, X. Chen, M. B. Menhaj, and R. Sainudiin, "Artificial neural network-based system identification for a single-shaft gas turbine," *J. Eng. Gas Turbines Power*, vol. 135, no. 9, pp. 1-7, 2013, doi: 10.1115/1.4024735.

[10] S. Ch and S. Mathur, "Particle swarm optimization trained neural network for aquifer parameter estimation," *KSCE J. Civ. Eng.*, vol. 16, no. 3, pp. 298–307, 2012, doi: 10.1007/s12205-012-1452-5.

[11] D. Pratiwi, D. D. Santika, and B. Pardamean, "An Application Of Backpropagation Artificial Neural Network Method for Measuring The Severity of Osteoarthritis," *Int. J. Eng. Technol.*, vol. 11, no. 3, pp. 102–105, 2011.

[12] E. Suryani, Wiharto, S. Palgunadi, and T. P. Nurcahya Pradana, "Classification of Acute Myelogenous Leukemia (AML M2 and AML M3) using Momentum Back Propagation from Watershed Distance Transform Segmented Images," in *Journal of Physics: Conference Series*, 2017, vol. 801, no. 1, pp. 1-8, doi: 10.1088/1742-6596/801/1/012044.

[13] C. C. Gowda and S. G. Mayya, "Comparison of Back Propagation Neural Network and Genetic Algorithm Neural Network for Stream Flow Prediction," *J. Comput. Environ. Sci.*, vol. 2014, pp. 1–6, 2014, doi: 10.1155/2014/290127.

[14] S. J. Abdulkadir, S. M. Shamsuddin, and R. Sallehuddin, "Moisture Prediction in Maize Using Three Term Back Propagation Neural Network," *Int. J. Environ. Sci. Dev.*, vol. 3, no. 2, pp. 199–204, 2012, doi: 10.7763/ijesd.2012.v3.215.

[15] S. Karsoliya, "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture," *Int. J. Eng. Trends Technol.*, vol. 3, no. 6, pp. 714–717, 2012.

[16] M. J. Berry and G. Linoff, "Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management," *John Wiley Sons, Inc.*, pp. 1–888, 2011.

[17] R. Hecht-Nielsen, "Kolmogorov's Mapping Neural Network Existence Theorem," *Proc. Int. Conf. Neural Networks*, pp. 11–14, 1987.

[18] Z. Yudong and W. Lenan, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 8849–8854, 2009, doi: 10.1016/j.eswa.2008.11.028.

[19] G. Mirchandani and W. Cao, "On Hidden Nodes for Neural Nets," *IEEE Trans. Circuits Syst.*, vol. 36, no. 5, pp. 661–664, 1989, doi: 10.1109/31.31313.

[20] J. Laurence, *Introduction to Neural Networks: Design, Theory,*

DOI: https://doi.org/10.29207/resti.v6i3.4049

and Applications. Nevada City: California Scientific Software, 1994.

[21] A. Blum, *Neural networks in C++: an object-oriented framework for building connectionist systems*, Cambridge . New York: John Wiley & Sons, Inc., 1992.

[22] R. Faisal, N. S. Surjandari, and S. Setiono, "Prediksi Stabilitas Lereng Menggunakan Adaptive Neuro-Fuzzy Metode Hybrid," *Matriks Tek. Sipil*, vol. 6, no. 3, pp. 439-450, 2018, doi: 10.20961/mateksi.v6i3.36549.

[23] Z. Boger and H. Guterman, "Knowledge extraction from artificial neural networks models," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1997, vol. 4, pp. 3030–3035, doi: 10.1109/icsmc.1997.633051.

[24] Y. JingTao and L. T. Chew, "Guidelines for financial forecasting with neural networks," . *Neural Inf. Process. Shanghai*, 2001.

[25] B. Setyonugroho, A. E. Permanasari, and S. S. Kusumawardani, "Perbandingan Akurasi Algoritme Pelatihan dalam Jaringan Syaraf Tiruan untuk Peramalan Jumlah Pengguna Kereta Api di Pulau Jawa," *J. Metik*, vol. 1, no. 1, pp. 50–62, 2017.

[26] Mislan, Haviluddin, S. Hardwinarto, Sumaryono, and M. Aipassa, "Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan - Indonesia," in *Procedia Computer Science*, 2015, vol. 59, pp. 142–151, doi: 10.1016/j.procs.2015.07.528.

[27] Z. Zhang, P. Yang, X. Ren, Q. Su, and X. Sun, "Memorized sparse backpropagation," *Neurocomputing*, vol. 415, pp. 397–407, 2020, doi: 10.1016/j.neucom.2020.08.055.

[28] S. Almaliki, R. Alimardani, and M. Omid, "Artificial neural network based modeling of tractor performance at different field conditions," *Agric. Eng. Int. CIGR J.*, vol. 18, no. 4, pp. 262–274, 2016.

[29] G. Lesinski, S. Corns, and C. Dagli, "Application of an Artificial Neural Network to Predict Graduation Success at the United States Military Academy," in *Procedia Computer Science*, 2016, vol. 95, pp. 375–382, doi: 10.1016/j.procs.2016.09.348.

DOI: https://doi.org/10.29207/resti.v6i3.4049