The most optimal performance of the Levenberg-Marquardt algorithm based on neurons in the hidden layer

H Mustafidah1,*, C P Putri1, H Harjono1 and S Suwarsito2

1 Informatic Engineering - Universitas Muhammadiyah Purwokerto, Jl. Raya Dukuhwaluh Purwokerto, Indonesia 53182
2 Geography Education - Universitas Muhammadiyah Purwokerto, Jl. Raya Dukuhwaluh Purwokerto, Indonesia

*fida.mustafidah@ump.ac.id

Abstract. The training algorithm is the main driver in artificial neural networks. The performance of the training algorithm is influenced by several parameters including the number of neurons in the input and hidden layers, epoch maximum, activation function, and the size of the learning rate (lr). One of the benchmarks for optimizing the performance of training algorithms can be viewed from the error or MSE (mean square error) produced. The smaller the error, the more optimal the performance. The test conducted in the previous study obtained information that the most optimal training algorithm based on the smallest MSE produced was Levenberg-Marquardt (LM) with an average MSE=0.001 at the level of α=5% and using 10 neurons in the hidden layer. Therefore, this study aims to test the LM algorithm using several variations in the number of neurons in hidden layers. In this study, the LM algorithm was tested using 5 neurons in the input layer, and 2, 4, 5, 7, 9 neurons in hidden layers, and the same parameters as previous study. This study uses a mixed method that is developing computer programs and quantitative testing of program output data using statistical test. The results showed that the LM algorithm had the most optimal performance using 9 neurons in the hidden layer at the level of lr=0.5 with the smallest error of 0.000137501±0.000178355.

1. Introduction
Artificial neural network (ANN) is a biologically inspired computational model consisting of several processing elements (neurons) and there is a connection between these neurons. These neurons will transform information received by one neuron into another neuron. This relationship is called weight. Deboeck and Kohonen describe ANN as a collection of mathematical techniques that can be used for signal processing, forecasting and grouping and are referred to as parallel or multi-layered parallel regression techniques [1]. Areas of work that can be carried out with ANN implementation are numerous, including estimates or approaches, pattern classification, clustering, and forecasting [2]. The biggest problem domain is classification [3].

Some of the results of research on the implementation of artificial neural networks include predicting student achievement in the Informatics Engineering Study Program of Muhammadiyah University of Purwokerto based on the values of subjects tested in the National Examination when in high school [4] and for predicting levels Question validity [5]. Another development in the field of education is to improve the e-learning environment [6], a system of evaluating teachers [7], and [8] developing ANN to predict dropout rates on online education. Furthermore, the development of ANN is to diagnose breast
cancer [9] and [10] use ANN-based analysis models to reduce harmonic distortion on a Computer Centre Building. ANN-based systems were also developed by [11] to evaluate teacher performance and the academic level (achievement) of students. Next [12] uses ANN as a model that can be trained to produce qualitative data from a large number of categorical data about student attitudes.

ANN works based on the parameters owned. These parameters include epoch, learning rate (lr), neuron, error (MSE), and weight. Not all of these parameters correlate with each other, as the results of a study conducted by [13] that network errors correlate with the epoch, but network errors do not correlate with lr [14].

There are several training algorithms in ANN which are the driving forces of performance [15], but not all of these algorithms have the same level of accuracy [4]. Information regarding which training algorithms are most optimal based on empirical tests is also unknown. ANN-based system developments that have been carried out using one of the default training algorithms that exist in programming languages. Based on research conducted by [16] in testing 12 training algorithms in backpropagation networks [15], it was obtained that the Levenberg-Marquardt (LM) algorithm is the most optimal training algorithm at learning rate (lr) = 0.05 with an average error (MSE) of 0.0002196. Likewise the results of the study [17] produce information that the Levenberg–Marquardt algorithm is a training algorithm that produces the smallest error of 0.001001 with neurons in the input layer as many as 5, 10, and 15 neurons with 1 neuron in the output layer. In this study 10 neurons were used in hidden layers. Meanwhile, the number of neurons in hidden layers will greatly affect network performance, especially in the resulting errors that have an impact on the accuracy of network output. The more neurons in hidden layers, the more accurate the output will be, but the network performance will be slower [3].

Information about the number of neurons in the hidden layer that has the most optimal performance is unknown. Therefore, this study conducted an analysis and testing of variations in the number of neurons in hidden layers using the Levenberg–Marquardt training algorithm. In determining the number of neurons used in hidden layers there are several rules or methods, namely: 1) between the size of the input layer and the output layer, 2) 2/3 of the size of the input layer plus the size of the output layer, or 3) smaller or less than two times the size of neurons in the input layer [3].

2. Method

This research is a development research by testing using statistical methods. In this study, the LM algorithm was tested using: 5 neurons in the input layer; 2, 4, 5, 7, and 9 neurons in hidden layers; epoch = 10000 and target error = 0.001 (same parameters as previous study). These values are used for reasons as a continuation of previous research as stated in the introduction to maintaining standards in data analysis. This study uses a mixed method for developing computer programs as in Figure 1 and quantitative testing of program output data using statistical tests as designed in Figure 2.

2.1. Data Collecting

The data of this study were taken from research [18], namely random data on 5 input layer neurons and 1 output layer neuron. Random data is used with the assumption that it can represent in various conditions.

2.2. Data Analysis

Data were analysed using ANOVA test with the following stages [19]:

- hypothesis formulation.
  \( H_0 \): no difference in the error generated by the Levenberg-Marquardt training algorithm based on the number of neurons in the hidden layer in each value of lr.
  \( H_1 \): there are differences in errors generated by the Levenberg-Marquardt training algorithm based on the number of neurons in the hidden layer in each value of lr.
- determine \( \alpha = 5\% \)
- make decision. \( H_0 \) is rejected if sig. < \( \alpha \)
3. Results and discussion

Data from previous study [18] were random data consisting of 5 input data \( (X_i; i = 1 \ldots 5) \) and 1 target data \( (Y) \) as presented in Table 1.

| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) | \( Y \) |
|---|---|---|---|---|---|
| 9.5013 | 7.6210 | 6.1543 | 4.0571 | 0.5789 | 2.0277 |
| 2.3114 | 4.5647 | 7.9194 | 9.3547 | 3.5287 | 1.9872 |
| 6.0684 | 0.1850 | 9.2181 | 9.1690 | 8.1317 | 6.0379 |
| 4.8598 | 8.2141 | 7.3821 | 4.1027 | 0.0986 | 2.7219 |
| 8.9130 | 4.4470 | 1.7627 | 8.9365 | 1.3889 | 1.9881 |

The data in Table 1 are run on computer programs that have been built based on the configuration in Figure 2 using the MATLAB programming language [21] with various advantages [22]. Furthermore, the output data from the program were analysed using the ANOVA test. Of the five tests based on the
number of neurons in hidden layers for various values of lr as mentioned previously (there are 12 kinds of value: 0.01, 0.05, ... 1.0), values of significance (sig.) are obtained as in Table 2.

Table 2. Sig. values obtained from the ANOVA test.

| Num. of neurons in HL | Sig.   |
|-----------------------|--------|
| 2                     | .870   |
| 4                     | .401   |
| 5                     | .583   |
| 7                     | .107   |
| 9                     | .286   |

Based on the ANOVA test results in Table 2, it was concluded that from the five test groups on the LM training algorithm based on the number of neurons in the hidden layer, the sig values were obtained, all of which were greater than α. This shows no difference in MSE generated by the LM training algorithm in each group of the number of neurons in the hidden layer. Although there is no difference in MSE produced, but from each group the number of neurons in the hidden layer can be obtained information about the amount of MSE in each value of lr (0.01, 0.05, 0.1, …, 1.0 as mentioned before) as presented in Table 3. The blue highlight shows the minimum MSE generated by the LM algorithm on the use of a number of neurons in HL (column) on a corresponding lr value (row). The smallest MSE in each group of the number of neurons in hidden layers based on the value of lr is shown in Table 4.

Table 3. MSE data in each group of numbers of neurons in hidden layers based on each value of lr.

| lr     | The number of neurons in hidden layers | MSE                        |
|--------|---------------------------------------|----------------------------|
| 0.01   | 2                                    | 0.008440/758±               |
|        | 4                                    | 0.008799/824±               |
|        | 5                                    | 0.000395/029±               |
|        | 7                                    | 0.0003821/03±               |
|        | 9                                    | 0.0002847/85                |
| 0.05   | 2                                    | 0.087628/783±               |
|        | 4                                    | 0.002937/105±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 0.1    | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 0.2    | 2                                    | 0.000272/81±                |
|        | 4                                    | 0.000198/337±               |
|        | 5                                    | 0.000120/115                |
|        | 7                                    | 0.0002344/72                |
|        | 9                                    | 0.0002344/72                |
| 0.3    | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 0.4    | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 0.5    | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 0.6    | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 0.7    | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 0.8    | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 0.9    | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |
| 1      | 2                                    | 0.084550/864±               |
|        | 4                                    | 0.000282/272±               |
|        | 5                                    | 0.000155/244±               |
|        | 7                                    | 0.0002417/28±               |
|        | 9                                    | 0.0002133/26                |

Table 4. The smallest MSE based on lr of every n neuron in hidden layer.

| Neurons in HL | Learning Rate (lr) | MSE                        |
|---------------|--------------------|----------------------------|
| 2             | 1                  | 0.144278644 ± 0.545408489  |
| 4             | 0.7                | 0.004157556 ± 0.01742286   |
| 5             | 0.5                | 0.000160917 ± 0.00223816   |
| 7             | 0.1                | 0.000155244 ± 0.00164561   |
| 9             | 0.5                | 0.000137501 ± 0.00178355   |
Table 4 shows that of the five groups of neurons in the hidden layer, the smallest MSE was achieved by the LM training algorithm on 9 neurons with \( lr = 0.5 \). From these data, we can see a pattern of decreasing MSE values along with the increase in the number of neurons in hidden layers. After the bivariate correlation test between the number of neurons in the hidden layer and the MSE value using the Pearson Correlation method, the \( r \) value of -0.716 is obtained. This means that there is a relatively high correlation but the correlation is reversed. The more neurons in the hidden layer increase, the smaller the MSE value is generated as shown in Figure 3. This is in accordance with what was stated by [3].

![Graph of the relationship between the number of neurons in hidden layers and MSE.](image)

4. Conclusion
This study produces information that by using 5 neurons in the input layer and 1 neuron in the output layer, the Levenberg-Marquardt (LM) training algorithm produces the smallest MSE of 0.000137501 ± 0.000178355. This condition is achieved in the use of 9 neurons in the hidden layer with the value of learning rate = 0.5. This configuration can be referred to as the 5-9-1 model. The resulting model information can be used as a consideration in developing artificial neural network-based applications, especially the use of the backpropagation method.

Acknowledgments
Authors thank to DP2M - The Ministry of Research, Technology, and Directorate General of Higher Education (DIKTI) through KOPERTIS Region VI and Universitas Muhammadiyah Purwokerto who has provided funds and facilities in the implementation of this research.

References
[1] Shanmuganathan S and Samarasinghe S 2016 Artificial Neural Network Modelling 628 (Springer International Publishing)
[2] Samarasinghe S 2007 Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition
[3] Heaton J 2015 Artificial Intelligence for Humans, Volume 3: Neural Networks and Deep Learning, 1.0 (Chesterfield, USA: Heaton Research Inc.)
[4] Mustafidah H, Hakim D K and Sugiyanto S 2013 Tingkat Keoptimalan Algoritma Pelatihan pada Jaringan Syaraf Tiruan (Studi Kasus Prediksi Prestasi Belajar Mahasiswa) JUITA (Jurnal Inform. Ilm. 3 159–166
[5] Mustafidah H, Hartati S, Wardoyo R and Harjoko A 2013 Prediction of Test Items Validity Using Artificial Neural Network in Proceeding International Conference on Education, Technology, and Science (NETS) 2013, "Improving The Quality Of Education To Face The Impact Of Technology"’
[6] Kaur P, Kaur K and Singh G 2012 Improving E-Learning with Neural Networks Int. J. Comput. Bus. Res. Proceeding, no. of ‘1-Society 2012’ at GKI, Talwandi Sabo Bathinda (Punjab)
[7] Kooshki S A, Zeinabadi H, Jafarnezhad A and Kooshki H A 2016 Applications of Fuzzy Logic
and Artificial Neural Networks in Evaluation and Ranking of Teachers Based on ‘Framework for Teaching’ Model, Int. Acad. J. Innov. Res. 3 2 1–10
[8] Tan M and Shao P 2014 Predicting Dropout from Online Education based on Neural Networks Open Cybern. Syst. J. 8 3 2 623–627
[9] Ulum M B and Laday R K 2016 Implementasi Metode Backpropagation Neural Network untuk Mendiaagnosis Penyakit Kanker Payudara Seminar Nasional APTIKOM (SEMNASTIKOM) 581–583
[10] Yasri I, Ervianto E and Noviando E S 2016 Model Analisa Berbasis Jaringan Saraf Tiruan untuk Mengurangi Distorsi Harmonisa pada Gedung Pusat Komputer Seminar Nasional APTIKOM (SEMNASTIKOM) 675–680
[11] Qianyin X and Bo L 2015 Multiple Evaluation Models for Education Based on Artificial Neural Networks Int. J. Hybrid Inf. Technol. 8 9 1–10
[12] Nurettin Y and Ilker U 2015 A CFBPN artificial neural network model for educational qualitative data analyses: Example of students attitudes based on Kellert’s typologies Educ. Res. Rev. 10 18 2606–2616
[13] Mustafidah H and Harjono 2017 Korelasi Tingkat Kesalahan dan Epoh dalam Jaringan Backpropagation,” Prosiding SEMNASTIKOM 55–61
[14] Mustafidah H and Suwarsito S 2017 Correlation Analysis between Error Rate of Output and Learning Rate in Backpropagation Network International Summit on Knowledge Advancements (ISKA 2017)
[15] Kusumadewi S 2004 Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK (Yogyakarta: Graha Ilmu)
[16] Mustafidah H and Suwarsito S 2015 Model Parameter Jaringan Syaraf Tiruan untuk Pemilihan Algoritma Pelatihan Jaringan Backpropagation yang Paling Optimal (Purwokerto, Central Java)
[17] Mustafidah H and Suwarsito S 2016 Inferensi Tingkat Kesalahan dalam Jaringan Backpropagation Berdasarkan Laju Pemahaman Prosiding Seminar Nasional Aptikom 2016 (SEMNASTIKOM) 576–580
[18] Mustafidah H and Suwarsito S 2015 Error Rate Testing of Training Algorithm in Backpropagation Network Int. J. Soft Comput. Eng. 5 4 46–50
[19] Taniredja T and Mustafidah H 2011 Penelitian Kuantitatif (Sebuah Pengantar) (Bandung: ALFABETA)
[20] Mustafidah H and Suwarsito S 2016 Testing Design of Neural Network Parameters in Optimization Training Algorithm International Conference of Result and Community Services 139-146
[21] Lent S L 2013 Learning to Program with MATLAB: Building GUI Tools. Department of Electrical Engineering (University of Notre Dame: John Wiley & Sons, Inc.)
[22] Paulus E and Nataliani Y 2007 GUI Matlab Yogyakarta: ANDI, 2007.