

Research Article

Predicting the Number of COVID-19 Sufferers in Malang City Using the Backpropagation Neural Network with the Fletcher–Reeves Method

Syaiful Anam , Mochamad Hakim Akbar Assidiq Maulana , Noor Hidayat ,
Indah Yanti , Zuraidah Fitriah , and Dwi Mifta Mahanani 

Department of Mathematics, Faculty of Mathematics and Natural Sciences, Brawijaya University, Malang, East Java, Indonesia

Correspondence should be addressed to Syaiful Anam; syaiful@ub.ac.id

Received 19 December 2020; Revised 20 March 2021; Accepted 12 April 2021; Published 29 April 2021

Academic Editor: Wan Hanna Melini

Copyright © 2021 Syaiful Anam et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

COVID-19 is a type of an infectious disease that is caused by the new coronavirus. The spread of COVID-19 needs to be suppressed because COVID-19 can cause death, especially for sufferers with congenital diseases and a weak immune system. COVID-19 spreads through direct contact, wherein the infected individual spreads the COVID-19 virus through cough, sneeze, or close contacts. Predicting the number of COVID-19 sufferers becomes an important task in the effort to curb the spread of COVID-19. Artificial neural network (ANN) is the prediction method that delivers effective results in doing this job. Backpropagation, a type of ANN algorithm, offers predictive problem solving with good performance. However, its performance depends on the optimization method applied during the training process. In general, the optimization method in ANN is the gradient descent method, which is known to have a slow convergence rate. Meanwhile, the Fletcher–Reeves method has a faster convergence rate than the gradient descent method. Based on this hypothesis, this paper proposes a prediction model for the number of COVID-19 sufferers in Malang using the Backpropagation neural network with the Fletcher–Reeves method. The experimental results show that the Backpropagation neural network with the Fletcher–Reeves method has a better performance than the Backpropagation neural network with the gradient descent method. This is shown by the Means Square Error (MSE) resulting from the proposed method which is smaller than the MSE resulting from the Backpropagation neural network with the gradient descent method.

1. Introduction

At the end of December 2019, Indonesia and the world were shocked by the emergence of an infectious disease that attacks the respiratory organs. This disease is called COVID-19 [1]. The COVID-19 disease is an infection caused by a new type of coronavirus. This virus was first discovered in Wuhan City, Hubei Province, China, and then it spread throughout the world, including Indonesia. It spread through direct contact with disease sufferers who traveled from the infected areas [2].

The effects of this disease are very serious because respiration is a vital human organ that helps metabolic processes and balances substances in the body. In addition,

COVID-19 can cause death for the sufferers [3], especially those with congenital diseases or a weak immune system. This disease quickly spreads because, like any other infectious respiratory disease, the transmission of the virus occurs through a droplet from the nose or mouth of the person with COVID-19 when they cough, sneeze, or are in close contacts. Therefore, during a pandemic, it is highly recommended to put on masks or protective equipment and carry out social restrictions to reduce the potential of spreading the virus [4].

The number of people with COVID-19 is increasing every day. The increment of the number of sufferers with this disease should be directly proportional to adequate health services. Predicting the number of COVID-19 sufferers based on the data of the number of preexisting sufferers is

necessary to slow down the spread of the disease and to sustain the provision of health service facilities in the future [5]. Predicting the number of COVID-19 sufferers is crucial for the effort in curbing, the rate of spread of the virus, and as a reference for health policy-making.

The tally of the number of COVID-19 sufferers is influenced by several factors related to the virus spread, including the number of deaths and the cases of patients' recovery. Also, the incubation period of the virus in the human body, which is 14 days, also affects the estimation of the tally on the following day [6].

Many methods have been proposed to predict the spread of viruses. Viruses can be modeled as a population influenced by the spread of the disease. One type of prediction methods is time series analysis, which is looking for variables with the variables that influence them and is associated with time or analysis of only cause and effect. This prediction of the disease spread leads to a time series analysis because the current number of COVID-19 sufferers is influenced by the number of sufferers from the previous time. Furthermore, the regression method is usually used in time series problems. There are two types of regression, namely, linear regression and nonlinear regression. The growth of the population in this study is considered unrealistic due to the conditions in the environment; therefore, nonlinear regression is used to overcome the prediction errors from the expected numbers. Even so, nonlinear regression is considered ineffective when working on more complex factors. Artificial neural network (ANN) is one of the suitable prediction methods. It is much more flexible and can handle more complicated and unassuming cases than the regression method.

Based on the prediction algorithm used, there are several types of ANN algorithm, one of which is the Backpropagation. The Backpropagation algorithm is a method that can be used to solve predictive problems with good results, but its performance is influenced by the optimization method used during training. In general, the optimization method used is the gradient descent method.

The downside of this method is that it has a slow convergence rate [7]. The Fletcher-Reeves method, however, has a better convergence rate than the gradient descent method [8].

This paper proposes a model prediction for the number of COVID-19 sufferers in Malang using the Backpropagation and the Fletcher-Reeves method. This experiment compared the Backpropagation neural network with the Fletcher-Reeves method with the Backpropagation neural network with the gradient descent method. The prediction model is obtained through experiments that combine network architecture and learning rate to get the most optimum prediction model.

2. Materials and Methods

This section explains the research dataset, optimization, the Fletcher-Reeves method, Backpropagation algorithm, and the proposed method. Because this section deals with the data set for the experiment, several theories related to

optimization, the Fletcher-Reeves method, and the Backpropagation algorithm are discussed here. The end of this section describes the proposed method, that is, the Backpropagation neural network optimized by the Fletcher-Reeves method, which is used to predict the number of sufferers of COVID-19.

2.1. Data Set. The data informing the number of COVID-19 sufferers used for evaluation were taken from the Gugus Tugas COVID-19 website, Malang city. The data were also published on the Instagram account of Malang City government, @pemkotmalang. In Table 1, it can be seen that as many as 206 data were taken from cases published from March to October 2020.

This research assumes that the factors affecting the number of cumulative confirmed positive cases today are the number of cumulative cases that have been confirmed positive within the previous 14 days (x_1, x_2, \dots, x_{14}), the cumulative number of deaths from the previous day (x_{15}), and the cumulative number of sufferers who recovered from the previous day (x_{16}). Furthermore, the data set for the ANN is formed based on the factors affecting the cumulative number of confirmed positive cases and the data sets in Table 1. From 192 data evaluated in this research, the results of the dataset are shown in Table 2.

Therefore, the data that have been obtained are classified into two parts: training and testing data. The training data uses 90% of the total data, while the testing data uses 10% of the total data.

2.2. Optimization and Fletcher-Reeves Method. Optimization is the process of finding the best solution or optimal value of a problem. An optimization method is used to find either the maximum value or minimum value. Optimization has been applied to solve everyday life problems, such as water resource management, medicine, agriculture, economics, and others [9–12].

The optimal value of an objective function can be found with optimization methods. Various optimization methods have been created, such as Golden Search and Quadratic Approximation for simple one-dimensional objective function problems, as well as Gradient Descent, Conjugate Gradient, Newton, and others.

Newton's method has a property called quadratic termination. Hence, it can precisely minimize quadratic functions in limited iterations, but it requires calculating and storing the second derivative of the function. Newton's method becomes impractical for calculating all the derivatives when there are too many parameters. ANN requires several hundred to thousands of weights, so the use of optimization methods that require the calculation of derivatives is less practical. Therefore, ANN requires an optimization method that considers only the first derivative and has squared stops [13].

Another optimization method is the Conjugate Gradient method, an iterative method to solve a linear equation system. This method is effective for systems with linear equations which have a positive definite symmetric

TABLE 1: Data of the cumulative number of COVID-19 cases in Malang City.

No.	Date	Cumulative number of cases		
		Confirmed positive	Dead	Recovered
1	03/27/20	3	0	0
2	03/28/20	3	0	3
3	03/29/20	4	0	3
4	03/30/20	4	0	3
5	03/31/20	4	0	3
6	04/01/20	4	0	3
7	04/02/20	5	0	3
8	04/03/20	5	0	3
9	04/04/20	5	0	3
10	04/05/20	5	0	3
11	04/06/20	8	0	3
12	04/07/20	8	0	4
13	04/08/20	8	0	4
14	04/09/20	8	0	4
15	04/10/20	8	0	4
⋮	⋮	⋮	⋮	⋮
206	10/18/20	1929	190	1691

TABLE 2: The dataset in this research.

No.	x_1	...	x_{14}	x_{15}	x_{16}	T
1	3	...	8	0	4	8
2	3	...	8	0	4	8
3	4	...	8	0	4	8
4	4	...	8	0	4	8
⋮	⋮	...	⋮	⋮	⋮	⋮
192	1815	...	1921	190	1691	1691

coefficient matrix. In general, this method generates conjugate vectors and is also a gradient of the quadratic function. It solves a linear equation system by finding the minimum point of the quadratic function. One of the variants of the Conjugate Gradient Method is the Fletcher–Reeves method.

The following is the algorithm of the Fletcher–Reeves method.

2.2.1. The Fletcher–Reeves Algorithm

- (1) Input the initial point \mathbf{x}_0 , the stopping criteria ε , and the number of maximum iteration k_{\max} .
- (2) Initialize $k = 0$.
- (3) Calculate \mathbf{p}_0 by using

$$\mathbf{p}_0 = -\mathbf{g}_0 = \nabla \mathbf{f}(\mathbf{x}_0), \quad (1)$$

which is a search direction. It is defined by a negative of the gradient of the function.

- (4) While $(\nabla f(\mathbf{x}_k) > \varepsilon)$ or $(k \leq k_{\max})$, do

- (1) Calculate \mathbf{p}_k by using

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}. \quad (2)$$

- (2) Choose α , which minimizes $f(\mathbf{x}_k + \alpha \mathbf{p}_k)$.

- (3) Calculate \mathbf{x}_{k+1} by using

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k, \quad (3)$$

where α is the learning rate.

- (4) Calculate β_k by using

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}, \quad (4)$$

where β_k is called the Fletcher–Reeves method.

- (5) End While

2.2.2. Backpropagation Algorithm. ANN is a method for a system of information processing. ANN is analogous to the generalization of the mathematical model of human understanding (human cognition). ANN contains several neurons, which are connected. Neurons transform the information which is received into other neurons. This relationship in ANN is then known as weight [14].

ANN has three components, which are an architecture, a learning algorithm, and activation functions. The architecture of ANN is the pattern of the relationships between each neuron. It also determines the weight of each relationship between neurons [15].

The Backpropagation algorithm is a systematic method for conducting training at the ANN layer. The Backpropagation algorithm is an algorithm that is often used in solving complex problems. It has been used in many applications, such as rainfall prediction [16] and compound function prediction [17]. The Backpropagation algorithm has several layers, which are input, hidden, and output layers. A hidden layer is consisting of m units and a bias. Biases V_{0j} and W_{0k} behave like weights where the output bias is always equal to 1 [14].

The training process for the Backpropagation has three stages, which are feedforward step from pattern input training, the Backpropagation of associated errors, and weight updating. During the advanced step, each input unit will be counted in the hidden layers to get the output of the pattern. During the training process, the output from the network will be compared with the target, and then the error is calculated. Subsequently, the optimization is carried out so that the factors that distribute the error are obtained. This factor is used for updating the weight between the input layer and the output layer [15].

2.2.3. Backpropagation Algorithm

- (a) Initialize weights
- (b) While termination condition is false

- (1) Feedforward

- (i) An input listsignal is sent to input units (X_i , $i = 1, 2, 3, \dots, n$) and is passed on to the hidden layer.
- (ii) Output of each hidden unit (z_j , $j = 1, 2, 3, \dots, m$) is obtained by using equation (5). The value is propagated back to the next layer:

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}, \quad (5)$$

$$z_j = f(z_in_j).$$

- (iii) Each unit output (y_k , $k = 1, 2, 3, \dots, p$) adds up the input that has been weighted using the following equation:

$$y_in_k = w_{0k} + \sum_{j=1}^m z_j w_{jk}. \quad (6)$$

- (iv) The output signal is obtained by calculating the activation function in the following equation:

$$y_k = f(y_in_k). \quad (7)$$

- (2) Backpropagation of error:

- (i) Each output unit (y_k , $k = 1, 2, 3, \dots, p$) obtains a pattern that is associated with the input pattern, and then the error information is calculated using the following equation:

$$\delta_k = (t_k - y_k) f'(y_in_k). \quad (8)$$

- (ii) Next, the error correction is calculated to update the weights later with

$$\Delta w_{jk} = \alpha \delta_k z_j. \quad (9)$$

- (iii) The bias correction is also calculated with equation (10) and then passes δ_k to the layer units afterward:

$$\Delta w_{0k} = \alpha \delta_k. \quad (10)$$

- (iv) The error information from the units of the previous layer is multiplied with the output weight and the result is added as the input delta is calculated using the following equation:

$$\delta_in_j = \sum_{k=1}^p \delta_k w_{jk}. \quad (11)$$

- (v) The first derivative of the activation function is calculated. Therefore, it is multiplied with the error information with

$$\delta_j = \delta_in_j f'(z_in_j). \quad (12)$$

- (vi) Next, the error correction is calculated with equation (13) to update the weights later:

$$\Delta v_{ij} = \alpha \delta_j x_i. \quad (13)$$

- (vii) Furthermore, the bias correction is calculated using the following equation:

$$\Delta v_{0j} = \alpha \delta_j. \quad (14)$$

- (3) Updating weights and biases:

- (i) The weights and biases of each output unit are updated by using the formulation in equation (15), to minimize the error:

$$w_{jk}^{baru} = w_{jk}^{lama} + \Delta w_{jk}. \quad (15)$$

- (ii) Therefore, the weights and biases of each hidden unit are updated by using the formulation in the following equation:

$$v_{ij}^{baru} = v_{ij}^{lama} + \Delta v_{ij}. \quad (16)$$

- (c) Check the stop condition.

2.3. The Proposed Method. This paper proposes a method for predicting the number of COVID-19 sufferers using the Backpropagation neural networks with the Fletcher-Reeves method. The flowchart of the proposed method is shown in Figure 1. The short-term prediction of the number of COVID-19 sufferers in Malang City can be formulated based on several factors related to the spread of the COVID-19 virus.

In the study, the first step is preprocessing the data using data normalization. The data were normalized by transforming them into a range of 0 and 1. This is done by dividing all existing data by the number of population in a place (in this case, the population in Malang City). However, the number of COVID-19 sufferers is still too small, so that the pattern of output obtained is not optimal, and the divider for normalization is only 25% of the population of Malang City. The current total population of Malang City is 874890 people. The dataset in Table 2 is divided by 218722.5, 25% of the total population in Malang City. The data after normalization can be seen in Table 3.

Furthermore, the factors that affect the spread of the number of COVID-19 sufferers determine the input variables for the Backpropagation algorithm. The factors related to the spread of the COVID-19 virus are the data of sufferer deaths, recovery cases, and the increase in the number of sufferers 14 days before the predicted day. The weights initialization of the Backpropagation algorithm is determined randomly. Therefore, the Backpropagation algorithm conducts learning based on the training data. In the learning step, the weights of the neural network are updated by minimizing the error between the output of the neural network and the actual value or target. This error is optimized by using the Fletcher-Reeves method. The final weight is used for the network test step. In the testing step, the final weight is used for predicting the number of COVID-19 sufferers. In the testing step, the training data and the testing data are used for validating the accuracy method.

The hypothesis of this research is as follows: the cases of patient deaths, the cases of sufferers recovery, and the increase in the number of sufferers within the previous 14 days influence the number of sufferers of COVID-19 today. The variable inputs of the prediction system are the number of

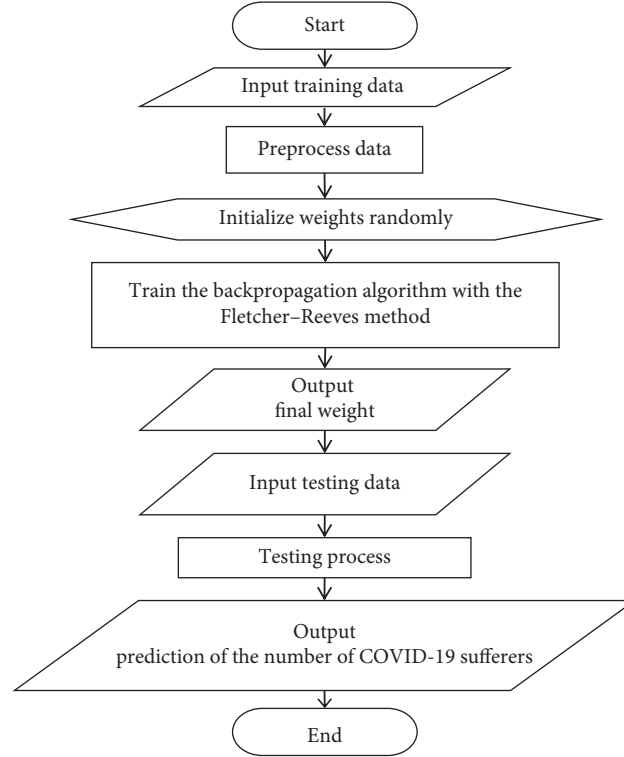


FIGURE 1: The flowchart of the system for predicting the number of COVID-19 sufferers with the Backpropagation neural network with the Fletcher-Reeves method.

TABLE 3: Dataset of this research after being normalized.

No.	x_1	...	x_{14}	x_{15}	x_{16}	T
1	1.37×10^{-5}	...	3.66×10^{-5}	3.66×10^{-5}	3.66×10^{-5}	3.66×10^{-5}
2	1.37×10^{-5}	...	3.66×10^{-5}	3.66×10^{-5}	3.66×10^{-5}	3.66×10^{-5}
3	1.83×10^{-5}	...	3.66×10^{-5}	3.66×10^{-5}	3.66×10^{-5}	3.66×10^{-5}
4	1.83×10^{-5}	...	3.66×10^{-5}	3.66×10^{-5}	3.66×10^{-5}	3.66×10^{-5}
\vdots	\vdots	...	\vdots	\vdots	\vdots	\vdots
192	8.30×10^{-3}	...	8.71×10^{-3}	8.74×10^{-3}	8.78×10^{-3}	8.69×10^{-4}

sufferer deaths, the cases of sufferers recovery, and the number of COVID-19 sufferers within the previous 14 days.

The network architecture is built based on several variables that influence the spread of COVID-19 and the number of COVID-19 sufferers. The variables used as network input are 16 variables, namely, data on the increase of COVID-19 sufferers within the previous 14 days ($x_1, x_2, x_3, \dots, x_{14}$), the number of sufferer deaths (x_{15}), and cases of sufferer recovery (x_{16}), whereas the variable used as the network output is the increase of COVID-19 sufferers (y). The network architecture is illustrated in Figure 2. In Figure 2, the Backpropagation neural network architecture has three layers, which are the input layer, the hidden layer, and the output layer.

The goal of the experiment is to find the best architecture and the appropriate learning rate for predicting the number of COVID-19 sufferers accurately. Until now, there has been no precise method to decide the number of neurons in the

hidden layer. Therefore, the number of neurons in the hidden layer is determined experimentally in this research. The number of neurons in the hidden layer is decided based on previous research. In this research, several architectural models are trialed in the experiment: 16-5-1, 16-20-1, 16-50-1, 16-100-1, and 16-150-1. The 16-5-1 architectural model means that the neural network has 16 neurons in the input layer, one neuron in the output layer, and five neurons in the hidden layers. The Backpropagation neural network has three steps, which are the feedforward step, the Backpropagation step, and the weight update step.

In this study, the learning rate is determined through experience. Thus far, the method to precisely determine the learning rate in artificial neural networks is yet to be found. In general, if the learning rate is large, then the learning rate is fast. Nevertheless, a fast learning rate often results in the divergence of MSE, which in turn results in errors that cannot be minimized even though iteration is heavily used.

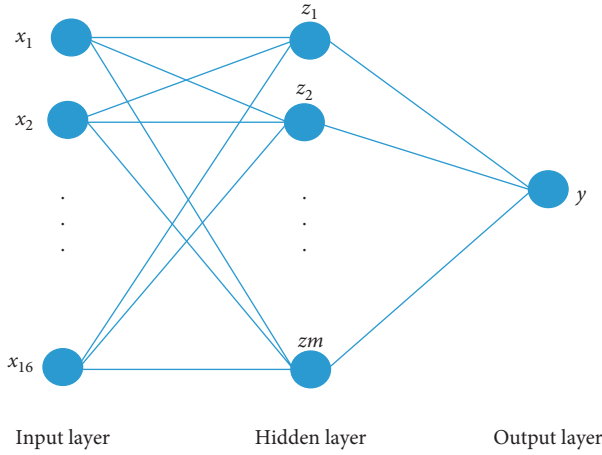


FIGURE 2: Backpropagation neural network architecture.

In this study, several learning methods were used, namely, 0.001, 0.005, 0.01, 0.1, and 0.2. The learning is selected based on previous research.

Backpropagation algorithm with the Fletcher–Reeves method:

- (a) Initialized $it = 1$ and weight, v_{ij} and w_{jk}
- (b) Input it_{\max}
- (c) While $it < it_{\max}$ do

(1) Feedforward

- (i) An input signal is sent to input units (X_i , $i = 1, 2, 3, \dots, n$) and is passed on to the hidden layer.
- (ii) Output of each hidden unit (z_j , $j = 1, 2, 3, \dots, m$) is obtained by using equations (17) and (18). The output value will be propagated back to the next layer:

$$z_in_j = v_{0j} + \sum_{i=1}^{16} x_i v_{ij}. \quad (17)$$

- (iii) The activation function is calculated to obtain the output signal. Then, therefore, the value is propagated back to the next layer. The activation function is used for the hidden layer and the output layer is the log-sigmoid function which is determined by calculating the following equation:

$$z_j = f(z_in_j) = \frac{1}{1 + e^{-z_in_j}}. \quad (18)$$

- (iv) The output unit (y) totals the weighted inputs, represented by

$$y_in = w_0 + \sum_{j=1}^m z_j w_j. \quad (19)$$

- (v) The activation function for calculating the output signal used is defined by

$$y = f(y_in) = \frac{1}{1 + e^{-y_in}}. \quad (20)$$

(2) Backpropagation of error:

- (i) Each output unit (y_k , $k = 1, 2, 3, \dots, p$) obtains a pattern that is associated with the input pattern. Then, the error information is calculated by using equation (11):

$$\delta = (t - y) f'(y_in). \quad (21)$$

- (ii) Each hidden unit (z_j , $j = 1, 2, 3, \dots, m$) adds up to the delta input from the hidden units in the previous layer:

$$\delta_in_j = \delta w_j. \quad (22)$$

(3) Updating weights and biases:

- (i) The value of p_{it} is calculated by the following equation:

$$p_{it} = -g_{it} + \beta_{it} p_{it-1}. \quad (23)$$

- (ii) The value of β_{it} is obtained using the following equation:

$$\beta_{it} = \frac{g_{it}^T g_{it}}{g_{it-1}^T g_{it-1}}. \quad (24)$$

- (iii) The biases and weights in the output unit ($j = 0, \dots, m$) are updated with equation (5).

$$w_j^{it+1} = w_j^{it} + \alpha p_{it} (\delta z_j). \quad (25)$$

- (iv) The biases and weights ($i = 0, \dots, 16$) are updated for each hidden unit (Z_j , $j = 1, 2, 3, \dots, m$):

$$v_{ij}^{it+1} = v_{ij}^{it} + \alpha p_{it} (\delta_j x_i). \quad (26)$$

- (v) Check the stop condition.

3. Results and Discussion

To implement the prediction system, the hardware used is a laptop with a 7th Gen Core i3 processor, 2.30 GHz, 8192 MB RAM, 250 SSD. The programming software is MATLAB R2014b.

The input variables from the ANN are the number of COVID-19 sufferers within the previous 14 days, the number of deaths, and the number of recoveries up to the previous day, while the output from the network is the number of confirmed cases to date. Tables 1 and 2 show the performance comparison between the Fletcher–Reeves and gradient descent methods for optimizing Backpropagation ANN with several different architectures. From these tables, it can be concluded that the Fletcher–Reeves method has much better accuracy than the gradient descent method. The Fletcher–Reeves method and gradient descent methods require a similar computation time for Backpropagation neural

TABLE 4: Evaluation results using training data for several architectures with a learning rate of 0.005.

Architecture	MSE		Computational time	
	Gradient descent method	Fletcher–Reeves method	Gradient descent method	Fletcher–Reeves method
16-5-1	14471.31	1.31	2.03	0.80
16-20-1	1458.48	1.44	2.13	1.09
16-50-1	394.77	1.27	2.46	2.01
16-100-1	168.11	1.30	4.37	5.09
16-150-1	105.45	1.97	5.37	3.40

TABLE 5: Evaluation of the performance of the prediction method for the number of COVID-19 sufferers by using testing data for several architectures for a learning rate equal to 0.005.

Architecture	MSE	
	Gradient descent method	Fletcher–Reeves method
16-5-1	32391.95	6.67
16-20-1	11092.24	11.49
16-50-1	2901.01	27.18
16-100-1	1259.15	881.52
16-150-1	915.89	595.43

TABLE 6: Evaluation of the performance of the prediction method for the number of COVID-19 sufferers by using training data for several learning rates with 50 hidden neurons.

Learning rate	MSE		Computational time	
	Gradient descent method	Fletcher–Reeves method	Gradient descent method	Fletcher–Reeves method
0.001	418.98	1.29	3.18	2.10
0.005	375.47	1.39	3.88	3.45
0.01	354.06	1.70	4.35	2.89
0.1	356.43	1.44	3.91	4.13
0.2	352.57	1.81	3.45	2.19

network architectures with a few hidden neurons. It means that the optimization method used in the Backpropagation neural network dramatically affects its performance.

Tables 1 and 2 show the architectural effects of the two Backpropagation neural network models. From Tables 1 and 2, it can be seen that the architecture of neural networks or the number of neurons in the hidden layer also affects the performance of the Backpropagation neural network. Referring to Tables 1 and 2, when the number of hidden neurons in the Backpropagation neural network algorithm using the Fletcher–Reeves method increases, the Means Square Error (MSE) generated from the Backpropagation neural network using the Fletcher–Reeves method also increases significantly in the testing data. However, MSE generated from the Backpropagation neural network using the Fletcher–Reeves method shows no significant changes in the training data. Furthermore, in the Backpropagation neural network algorithm with the gradient descent method, when the number of hidden neurons increases, the Means Square Error (MSE) resulted from the Backpropagation neural network using the Fletcher–Reeves method decreases for both training data and testing data. It means that an overfitting condition in the Backpropagation neural network algorithm with the Fletcher–Reeves method occurs. Contrastingly, overfitting condition does not occur in the

Backpropagation neural network algorithm with the gradient descent method. The number of hidden neurons is too many in the Backpropagation neural network algorithm with the Fletcher–Reeves method, which often leads to overfitting. The overfitting condition occurs when the accuracy is too extreme for training data, but the accuracy is too poor for the testing data. Furthermore, this condition causes a decrease in the generalization capability of the Backpropagation neural networks. However, generally, the performance of the Backpropagation neural network using the Fletcher–Reeves method is much better than the Backpropagation neural network using the gradient descent method. Drawing on the results of these experiments, the learning rate with as many as 50 neurons gives the best results. Therefore, the number of neurons to be used in determining the learning rate is 50 neurons.

Table 4 also shows the computational times of the two methods. The Backpropagation neural network algorithm with the Fletcher–Reeves method has a faster computational time than the Backpropagation neural network algorithm with the gradient descent method. In both methods, the increase in the number of neurons consequently increases the computational time required for training (Table 5). Table 5 shows the evaluation of the performance of the prediction method for the number of COVID-19 sufferers by using

TABLE 7: Evaluation of the performance of the prediction method for the number of COVID-19 sufferers by using testing data for several learning rates with 50 hidden neurons.

Learning rate	MSE	
	Gradient descent method	Fletcher–Reeves method
0.001	4627.75	27.49
0.005	2168.61	47.62
0.01	1607.21	9.95
0.1	2496.37	10.35
0.2	2022.84	31.37

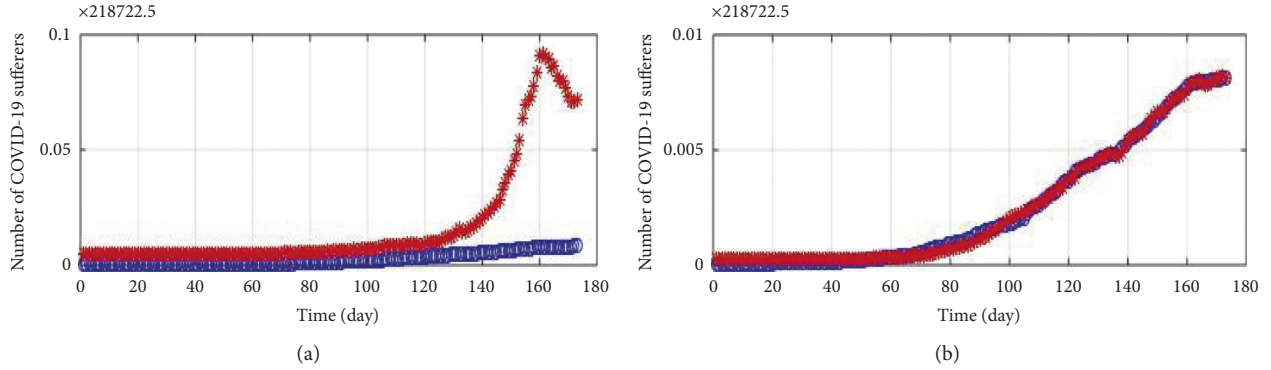


FIGURE 3: Comparison between the number of COVID-19 sufferers from the actual data and the number of COVID-19 sufferers predicted by the Backpropagation algorithm for training data. (a) Backpropagation neural network with the gradient descent method. (b) Backpropagation neural network with the Fletcher–Reeves method.

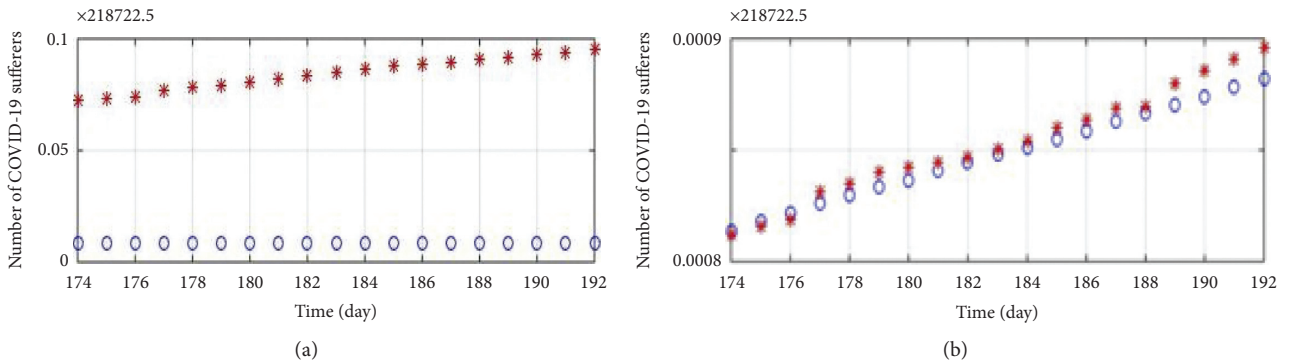


FIGURE 4: Comparison between the number of COVID-19 sufferers from the actual data and the number of COVID-19 sufferers predicted by the Backpropagation algorithm for testing data. (a) Backpropagation neural network with the gradient descent method. (b) Backpropagation neural network with the Fletcher–Reeves method.

testing data for several architectures for a learning rate equal to 0.005. Table 5 also shows that the backpropagation neural network with Fletcher–Reeves method gives better results than the backpropagation neural network with gradient descent method.

Tables 6 and 7 illustrate the performance evaluation of the Backpropagation neural network with Fletcher–Reeves and gradient descent methods to predict the number of confirmed cases of COVID-19 sufferers with different learning rates. The tables show that the best learning rate is 0.01. The best MSE generated by the Backpropagation neural

network with the Fletcher–Reeves method for the training data is 1.29 and 9.95 for the testing data. Contrastingly, the best MSE produced by the Backpropagation neural network with the gradient descent method for the training data is 105.45 and 915.89 for the testing data. From Tables 6 and 7, it can be stipulated that learning failure and poor accuracy result from an overly small or an overly large learning rate.

Figures 3(a) and 3(b) show the comparisons between the actual number of COVID-19 sufferers and the predicted results using the Backpropagation neural network with the gradient descent method and the Backpropagation neural

network with the Fletcher–Reeves method for training data. Figures 4(a) and 4(b) give details on the comparisons of the actual number of COVID-19 sufferers and the predicted results using the Backpropagation neural network with the gradient descent method and the Backpropagation neural network with the Fletcher–Reeves method for data testing. The x -axis shows the day after the first COVID-19 cases occurred, while the y -axis shows the cumulative number of COVID-19-confirmed positive cases and is scaled to 218722.5. Tables 6 and 7 show that the Backpropagation neural network with the Fletcher–Reeves method produces better results than the Backpropagation neural network with the gradient descent method to predict the number of COVID-19 sufferers in Malang City.

4. Conclusions

From the experiment results and discussion, it can be concluded that the Backpropagation neural network performance depends on several factors, which are the total neurons on the hidden layer and the optimization algorithm for learning. When hidden neurons are excessive, the generalization capability of the method decreases. This condition is called the overfitting condition. The Backpropagation neural network algorithm with the Fletcher–Reeves method has a faster computational time than the Backpropagation neural network algorithm with the gradient descent method. In both methods, the increase in the number of neurons consequently increases the computation time required for training. The learning rate of 0.01 gives the best result. If the learning rate is excessively small or large, it will lead to learning failure and, as a result, poor accuracy. Therefore, it is very important to select the appropriate learning rate to get better accuracy. The Backpropagation neural network with the Fletcher–Reeves optimization method gives better results compared to the Backpropagation neural network with the gradient descent method to predict the number of COVID-19 sufferers in Malang City in the future.

Data Availability

The raw data of the number of COVID-19 sufferers used for evaluation were taken from the Gugus Tugas COVID-19 website, Malang City, <https://covid19.malangkota.go.id/beranda>. They were also published on the Instagram ID of Malang City government, <https://www.instagram.com/pemkotmalang/?hl=id>.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] I. M. Ibrahim, D. H. Abdelmalek, M. E. Elshahat, and A. A. Elfiky, "COVID-19 spike-host cell receptor GRP78 binding site prediction," *Journal of Infection*, vol. 80, no. 5, pp. 554–562, 2020.
- [2] D. S. Pradanti, "Evaluation of formal risk assessment implementation of Middle East respiratory syndrome coronavirus in 2018," *Jurnal Berkala Epidemiologi*, vol. 7, no. 3, pp. 197–206, 2019.
- [3] N. Barda, D. Riesel, A. Akviv et al., "Developing a COVID-19 mortality risk prediction model when individual-level data are not available," *Nature Communications*, vol. 11, no. 1, pp. 1–9, 2020.
- [4] D. P. Kavadi, R. Patan, M. Ramachandran, and A. H. Gandomi, "Partial derivative nonlinear global pandemic machine learning prediction of COVID 19," *Chaos, Solitons & Fractals*, vol. 139, pp. 1–7, 2020.
- [5] L. Wynants, B. V. Calster, G. S. Collins et al., "Prediction models for diagnosis and prognosis of COVID-19: systematic review and critical appraisal," *The BMJ*, vol. 369, p. m1328, 2020.
- [6] Y. Yi, P. N. P. Lagniton, S. Ye, E. Li, and R.-H. Xu, "COVID-19: what has been learned and to be learned about the novel coronavirus disease," *International Journal of Biological Sciences*, vol. 16, no. 10, pp. 1753–1766, 2020.
- [7] F. D. Marleny, "Komparasi Algoritma Conjugate gradient Dan gradient descent Pada MLPNN Untuk Tingkat Pengetahuan Ibu (Studi kasus Pemberian ASI Eksklusif)," *Prosiding Konferensi Nasional Sistem & Informatika*, pp. 507–512, 2017.
- [8] T. M. Bafitlhile, Z. Li, and Q. Li, "Comparison of levenberg marquardt and conjugate gradient descent optimization methods for simulation of streamflow using artificial neural network," *Advances in Ecology and Environmental Research*, vol. 3, pp. 217–237, 2018.
- [9] D. V. Morankar, K. S. Raju, D. Nagesh Kumar, and D. N. Kumar, "Integrated sustainable irrigation planning with multiobjective fuzzy optimization approach," *Water Resources Management*, vol. 27, no. 11, pp. 3981–4004, 2013.
- [10] P. Schulthess, V. Rottschäfer, J. W. T. Yates, and P. H. Van der Graaf, "Optimization of cancer treatment in the frequency domain," *The AAPS Journal*, vol. 21, no. 106, pp. 1–9, 2019.
- [11] S. D. Kumara, S. Esakkirajan, C. Vimalraj, and B. K. Veena, "Design of disease prediction method based on whale optimization employed artificial neural network in tomato fruits," *Materials Today: Proceedings*, vol. 33, no. 7, pp. 4907–4918, 2020.
- [12] S. Zhu, X. Hu, K. Huang, and Y. Yuan, "Optimization of product category Allocation in multiple warehouses to minimize splitting of online supermarket customer orders," *European Journal of Operational Research*, vol. 290, no. 2, pp. 556–571, 2021.
- [13] K. L. Du and N. S. Swamy, *Neural Networks in a Soft computing Framework*, pp. 1–566, Springer, London, England, 2006.
- [14] A. Sudarsono, "Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode Backpropagation," *Media Infotama*, vol. 12, no. 1, pp. 61–69, 2016.
- [15] L. V. Fausett, *Fundamentals of Neural Network, Architectures, Algorithm and Applications*, Prentice-Hall, Inc., Hoboken, NJ, USA, 1994.
- [16] S. Anam, "Rainfall prediction using backpropagation algorithm optimized by broyden-fletcher-goldfarb-shanno algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 567, no. 1, 2019.
- [17] D. E. Ratnawati, Marjono, Widodo, and S. Anam, "Features selection for classification of SMILES codes based on their function," in *Proceeding of the 2019 International Seminar on Research of Information Technology and Intelligent Systems*, pp. 103–108, Yogyakarta, Indonesia, December 2019.