IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Rainfall prediction using backpropagation algorithm optimized by Broyden-Fletcher-Goldfarb-Shanno algorithm

Rainfall prediction using backpropagation algorithm optimized by Broyden-Fletcher-Goldfarb-Shanno algorithm

S Anam[1]

[1] Department of Mathematics, Faculty of Sciences, Brawijaya University, Malang 65145, Indonesia

Corresponding author: syaiful@ub.ac.id

Abstract. An extreme climate change results in a long dry season and an extreme rainfall results the losses in various areas of life. Rainfall prediction becomes an important thing for planning in many life sectors. Many prediction methods have been proposed, such as Autoregressive Integrated Moving Average (ARIMA) and Artificial Neural Networks (ANN). ANN has some advantages compared with the ARIMA model. Backpropagation algorithm is one of the ANN which has been successfully used in various fields. However, the performance of the backpropagation algorithm depends on the architecture and the optimization method used. The standard backpropagation algorithm optimized by gradient descent method works slowly to get a small error. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm works faster than gradient descent method. For this reason, this paper proposes the rainfall prediction using the backpropagation algorithm optimized by the BFGS algorithm. From the experiment results, it can be shown that the backpropagation algorithm optimized by the BFGS algorithm gives better result compared with the standard backpropagation algorithm for rainfall prediction. The big number of neuron hidden causes overfitting and the small number of neuron hidden make the worst accuracy. Choosing the right learning rate will produce better accuracy.

1. Introduction

Climate is one of the important factors of agricultural productivity [1]. Climate change will give many impacts in agricultural and livestock production, hydrologic balances and others. Agriculture is heavily dependent on climate. Climate depends on rainfall, sunshine hours, temperature, relative humidity and length of the drought period, which are called climatic factors. The climatic factors result in cycle-to-cycle variability of crops production. The uncontrollable natures of climate factors are changing over time which affects the agricultural, social, economic and environmental sustainability of a country. Climate change also influences potentially the earth's biological systems and human health [2]. Temperature, humidity, and rainfall patterns are the meteorological factors that influence transmission intensity of infectious diseases Developing country are expected to face a host of health effects due to climate change, including water-borne and vector-borne diseases such as malaria and dengue.

Global warming causes climate change in almost all parts of the world. An extreme climate change results in a long dry season and an extreme rainfall results the losses in various areas of life. High rainfall can cause floods, disease outbreaks and health problems, transportation problems, landslide, and crop failure, while low rainfall has a negative impact on the agricultural sector, i.e. land drought and crop failure [1,2,3].

The weather or rainfall is influenced by several factors, namely temperature, relative humidity, air pressure, wind speed, total cloud cover, and sun exposure. The weather or rainfall prediction becomes an important thing for planning in many life sectors. Many prediction methods for rainfall prediction have been proposed, such as Autoregressive Integrated Moving Average (ARIMA) [4]. Artificial Neural Network (ANN) is an alternative method for prediction. The ARIMA model works on the assumption of stationary, while ANN can work without assumptions. It means that the ANN method is more flexible compared with the ARIMA model. Adebiyi et al. in 2014 have proven that the ANN model is superior over the ARIMA model [5].

ANN model is also able to acquire knowledge even though there is no certainty, generalize and extract from a particular data pattern, create a pattern of knowledge through self-organizing. One of the popular ANN models that have been successfully used is backpropagation neural networks. It has been successfully used in various fields. However, the performance of the backpropagation neural networks depends on the architecture and the optimization method used.

The standard backpropagation neural networks use the gradient descent method for optimizing the error. The gradient descent method is very slow to get a small error. Quasi-Newton method is one of the good methods to solve the minimization problem. The size of the gradient change is constructed well enough to result in superlinear convergence. The quasi-Newton method has better convergence compared with the gradient descent method. Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm is one of type quasi-Newton methods which is a good iterative method for obtaining a local optimum. This method has also been proven to be better than gradient descent when applied to the neural network [6]. For this reason, this paper proposes the rainfall prediction by using the backpropagation neural networks optimized by Broyden–Fletcher–Goldfarb–Shanno algorithm. This research uses the average temperature, average relative humidity, and average wind speed for predicting rainfall. An automatic rainfall prediction based on weather data will realize a better life.

2. Related Works

This section will describe some theories and the previous research related to this research, such as rainfall, BFGS algorithm, and neural networks.

2.1. Rainfall

Rainfall is defined as the amount of water that falls on a flat ground surface during a certain period measured by unit height (mm) above the horizontal surface if it does not occur evaporation, runoff, and infiltration. One rainfall millimeters (1 mm), meaning in one-meter area square on a flat place where water is high one millimeter.

Rainfall is very important in determining the yield of crop cultivation. Increasing rainfall in an area raises the potential for flooding. Conversely, if there is a decrease in rainfall in one area, will cause drought [7]. According to Wilson (1993), influencing factors a lot of rainfall is humidity, air pressure, temperature and wind speed [8].

2.2. BFGS Algorithm

Quasi-Newton method is an alternative method that can be used for solving optimization problems. This method was successfully used for minimizing errors on artificial neural networks [9]. Quasi-Newton method is a method that is used when the calculation of the Hessian matrix is difficult or time-consuming. This method has a rapid convergence when compared with the method of gradient descent. One of quasi-Newton method types is the BFGS method. To solve the problem of minimization of the Eq. (1), the BFGS method can be used for solving it.

$$\min f(\mathbf{x}_k) \tag{1}$$

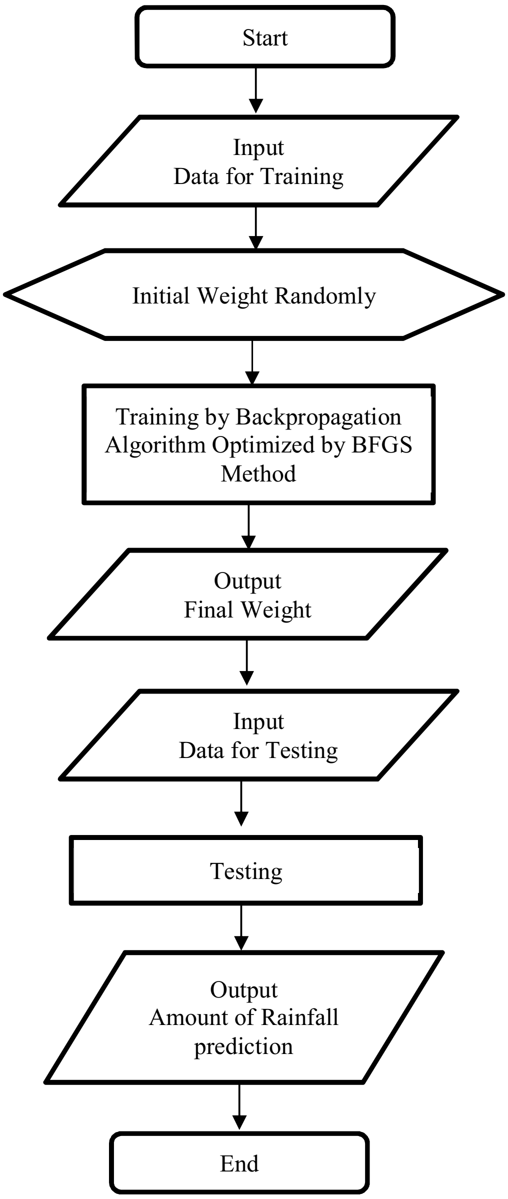


Figure 1. Rainfall prediction system by backpropagation neural network optimized by the BFGS method.

The BFGS algorithm can be seen in Algorithm 1 [6].

Algorithm 1: Algorithm of BFGS method

```
a. Input $\mathbf{x}_0$, ε (stopping criteria) and $k_{max.}$
b. Set an initial iteration k = 0 and $\mathbf{B} = \mathbf{I}$ where I is an identity matrix.
c. Calculate $f(\mathbf{x}_k)$
```

```
d. While (||∇ ($\mathbf{x}_k$)||>ε) or (k<=$k_{max}$) do
    1. Calculate $\mathbf{d}_k = -\mathbf{B}_k\, f(\mathbf{x}_k)$, where $\mathbf{d}_k$ is a generating direction.
    2. Select α that is able to minimize $f(\mathbf{x}_k + \alpha_k\, \mathbf{d}_k)$
    3. $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\, \mathbf{d}_k$
    4. Calculate $\mathbf{B}_{k+1}$ by using Eq. (2).
       $\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k\mathbf{s}_k(\mathbf{B}_k\mathbf{s}_k)^T}{\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k} + \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k} + \phi_k\left[\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k\right]\mathbf{v}_k\mathbf{v}_k^T$    (2)
       where $\phi_k \in [0,1]$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$,
       $\mathbf{y}_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ and $\mathbf{v}_k = \left[\frac{\mathbf{y}_k}{\mathbf{y}_k^T\mathbf{s}_k} - \frac{\mathbf{B}_k\mathbf{s}_k}{\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k}\right]$.
    5. k = k + 1
e. End while
```

2.3. Backpropagation Neural Networks

Artificial neural network (ANN) is an information processing system that has characters such as neural networks in the human brain. There are two types of learning in ANN to modify their weight, namely supervised learning and unsupervised learning. Backpropagation learning model is a learning model used for multilayer networks. The optimization method commonly used in backpropagation learning is the gradient descent method. In this method, the weight of the network moves along the negative direction of the gradient [6].

3. Proposed Method

The system of the proposed method is used to predict the amount of rainfall as shown in figure 1. By using this system, the amount of rainfall in an area can be predicted based on the factors of the rainfall. The first step, the training data of rainfall which are the previous month's rainfall, average temperature, average relative humidity, and average wind speed are inputted in the system. The initial weight of the backpropagation neural networks is generated randomly. The next step is that the backpropagation

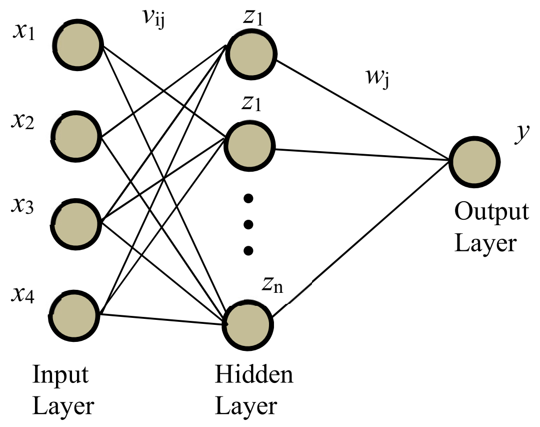


Figure 2. Architecture of backpropagation neural network.

neural networks will do training based on the training data. In this step, the weight of the neural networks will be improved by minimizing an error between the output of networks and target. The error is minimized by the BFGS method. The final weight will be used for testing step. In the testing step, the data test is inputted and the output is a prediction of the amount of rainfall.

The hypothesis in this study is that the previous month's rainfall, average temperature, average relative humidity and average wind speed in the previous month will affect rainfall this month. The data input of this program/system is the average rainfall in the previous month, average temperature, average relative humidity, and average wind speed.

The network architecture is developed by using the number of variables that affect many factors that influence rainfall. These variables include the average rainfall in the previous month average temperature, average relative humidity, average wind speed, and rainfall. Consequently, the variables used as network inputs are four variables which are the previous month average temperature (x_1), average temperature (x_2), average relative humidity (x_3), average wind speed (x_4), while the variables used as network output are the amount of rainfall (y). The network architecture is shown in figure 2. It has a three-layer, they are the input layer, the hidden layer and the output layer as shown in figure 2. Conversely for the number of hidden in the network is determined by trial (trial and error). The aim is to find out which architecture is most suitable for accurately recognizing patterns. Therefore, in this thesis an architectural model is taken, including: 3-5-1, 3-20-1, 3-50-1, 3-100-1, and 3-150-1. The meaning of architectural model 3-5-1 is that the network is built by 3 neurons in the input layer and 1 neuron in the output layer while the number of neurons in the hidden layer is 5 neurons.

The training of network by the backpropagation algorithm has three phases which are the feedforward of the input data, the backpropagation error, and the weights adjustment. The backpropagation algorithm is as in Algorithm 2.

Algorithm 2: Backpropagation neural networks

```
Step 0. Initialize the weights of the neural networks by random values and t=1 (iteration t^th).
Step 1. While error> min error or iteration < max iteration, do step 2-8
        Step 2. For each data, do step 3-8
        Feedforward :
        Step 3. Every input unit $x_i$, i=1,2,...,n gets an input $x_i$ and distributes it to all units in the hidden
                layer.
        Step 4. Every hidden unit $(z_j, j = 1,2,\dots,p\ )$ calculates its weighted $(v_{ij}(\boldsymbol{t}))$ by an input
                signal as in Eq.(3)
                    $\boldsymbol{z\_in_j = v_{0j}(t) + \sum_i x_i v_{ij}(t)},$          (3)
                where $v_{0j}$ is a bias on the hidden unit $j$.
               Calculate its output signal by an activation function as in Eq.(4) or Eq.(5)
                    $\boldsymbol{z_j = f(z\_in_j)}$          (4)
                or
                    $z_j = \frac{1}{1+\text{ex}^{\ (-z\_in_j)}}$          (5)
                then sends $z_j$ to all units in the output layer.
        Step 5. Every output unit $(y_{\_k}, k = 1,2,\dots,m\ )$ is calculated by considering its weighted
                input signals, it can be shown as in Eq.(6),
                    $y\_in_k = w_{0k}(\boldsymbol{t}) + \sum_{j=1}^{P} z_j\, w_{jk}(\boldsymbol{t})$          (6)
               where $w_{0k}\ (\boldsymbol{t})$ is a bias on the output unit $k$.
               Therefore, the output signal is calculated by an activation function as in Eq. (7) or Eq.(8)
```

$$y_k = f(y\_in_k) \quad (7)$$

$$\text{or } \boldsymbol{y_k} = \frac{\mathbf{1}}{\mathbf{1+exp}^{(-y\_in_k)}} \quad (8)$$

Backpropagation of error :

Step 6. Every output unit $(y_k, k = 1,2,\ldots,m\ )$ accepts a target pattern matching to the input data, and calculates its error as in Eq.(9),

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (9)$$

where , δ_k is error correction weight for w_{jk}

Step 7. Every hidden unit $(z_j, j = 1,2,\ldots,p\ )$ counts its $\delta_k$as in Eq.(10),

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \quad (10)$$

$$\delta_j = \delta\_in_j f'(\text{z\_}in_j) \quad (11)$$

Update weights and biases:

Step 8. Every output unit $(y_k, k = 1,2,\ldots,m\ )$ improves its bias and weights $j = 1,2,\ldots,p$ as in Eq.(12)

$$w_{jk}(t+1) = w_{jk}(t) + \alpha B_k(t)(\delta_k z_j(t)) \quad (12)$$

where $B_k(t)$ is an aproximation of Hessian matrix. In this case, the approach to the Hessian matrix is updated by formulation in Eq (13).

$$B_k(t+1) = B_k(t) + \frac{\Delta(-\delta_k z_j(t))\Delta(-\delta_k z_j(t))^T}{\Delta(-\delta_k z_j(t))^T \Delta w_{jk}(t)} + \frac{(-\delta_k z_j(t))(-\delta_k z_j(t))^T}{(-\delta_k z_j(t))^T \left({}^{\Delta w_{jk}(t)}\!\big/_{\alpha} \right)} \quad (13)$$

Each hidden unit $(Z_j, j = 1,2,\ldots,p\ )$ improves its bias and weights $i = 1,2,\ldots,n$ as in Eq.(14).

$$v_{ij}(t+1) = v_{ij}(t) + \alpha A_j(t)(\gamma_j x_i(t)) \quad (14)$$

where A_j(t) is defined as in Eq (15).

$$A_j(t+1) = A_j(t) + \frac{\Delta(-\gamma_j x_i(t))\Delta(-\gamma_j x_i(t))^T}{\Delta(-\gamma_j x_i(t))^T \Delta v_{ij}(t)} + \frac{(-\gamma_j x_i(t))((-\gamma_j x_i(t))^T}{((-\gamma_j x_i(t))^T \left({}^{\Delta v_{ij}(t)}\!\big/_{\alpha} \right)} \quad (15)$$

Step 9. Check the stopping creteria.

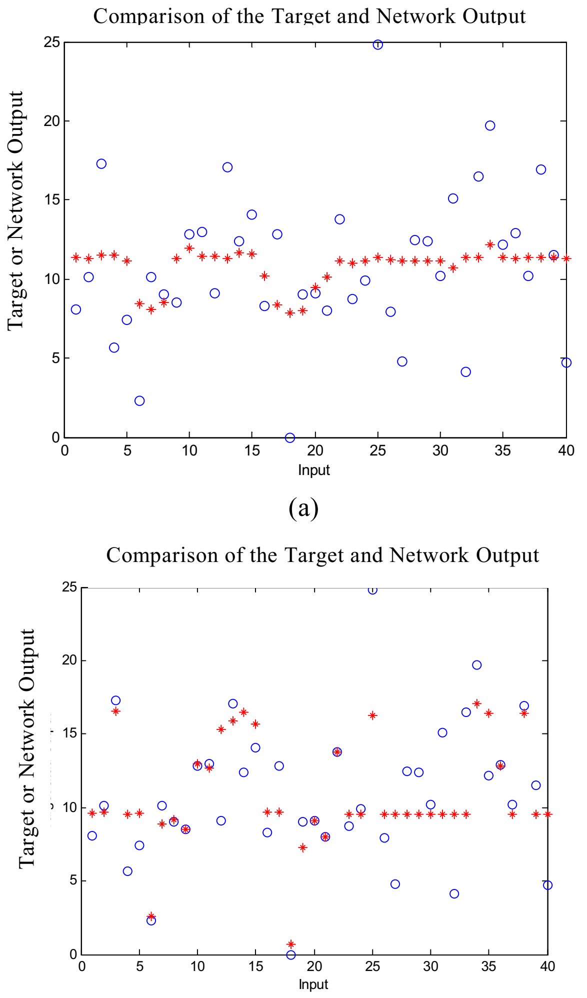
The activation functions used in this paper are the sigmoid function for the hidden layer and the identity function for the output layer. The learning rate used is static. The learning rate is intended to see how the influence given to ANN's ability to recognize patterns. In this case, the several learning rate (α) valaus are used which are 0.001, 0.005, 0.01, 0.1 and 0.2 to analyze the effect of the learning rate.

The stopping criteria is the maximum iteration amount of 1000. It means that when the neural networks learning has reached the number of iterations, the learning process will stop, if the number of iteration is less then 1000, the learning will continue.

4. Result and Discussion

For the evaluation of the system of rainfall prediction, the data used were taken from Karangploso Meteorology Climatology and Geophysics Council Station, Malang Regency. The number of data is 47 data from 2014 until 2017. The data are divided into data training and testing data. The number of data training is 40 data (85% from data total) and the number of data training is 7 data (15% from data total). For implementing the system, the hardware used is Core i5 processor, 1.70 GHz, 2. 4.0 GHz RAM, 500 GB hard drive and the software used is MATLAB R2013a. The input data of the neural networks are the previous month's rainfall, the average temperature, the average relative humidity, and the average wind speed and the output of the neural networks is rainfall this month.

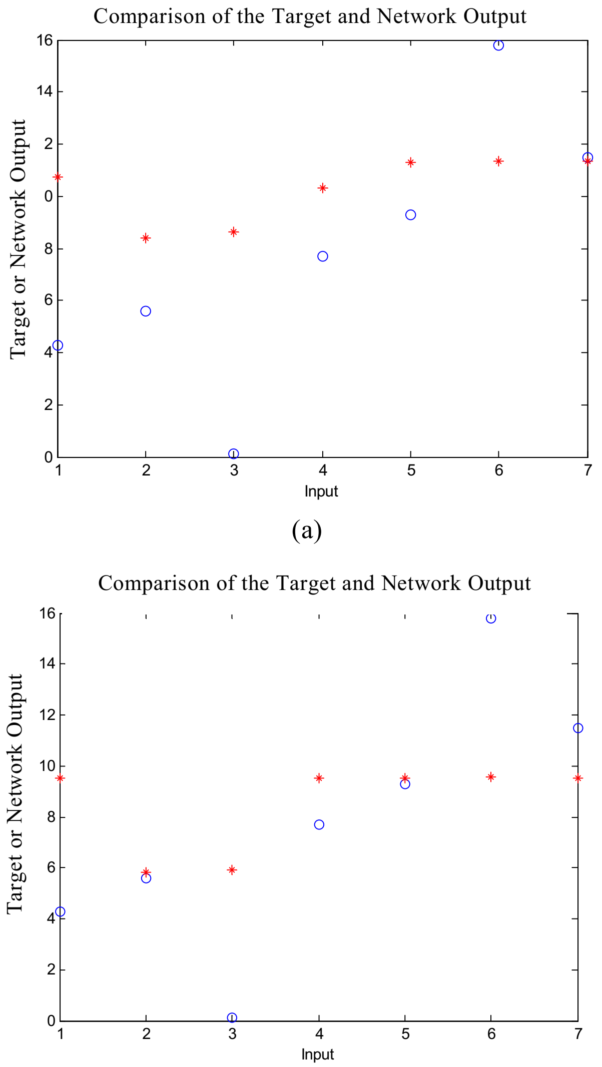
Table 1 and 2 show the comparison of the performance of the BFGS and gradient descent methods to optimize the backpropagation neural networks for the different architecture. They show that the BFGS method has better accuracy than the gradient descent. For a small number of neurons hidden, the BFGS method and the gradient descent almost have the same computational time. The number of the hidden layer should not be too small and too many. The big number of neuron hidden causes overfitting, it means that the accuracy for data training is too good but the accuracy for data



(a)

(b)

Figure 3. Comparison of the target and network output for training data. (a). the standard backpropagation (b) the backpropagation algorithm optimized by Broyden–Fletcher–Goldfarb–Shanno algorithm



(a)

(b)

Figure 4. Comparison of the target and network output for testing data. (a). the standard backpropagation (b) the backpropagation algorithm optimized by Broyden–Fletcher–Goldfarb–Shanno algorithm.

test is worst. Table 3 and 4 show the accuracy comparison of the BFGS method and the gradient descent method for predicting the amount of rainfall. It can be shown that the best learning is 0.01.

The backpropagation algorithm optimized by BFGS algorithm results in the best MSE (Mean Square Error) of 10.09 for training data and 20.13 for testing data, while the standard backpropagation algorithm results in MSE of 16.38 for training data and 18.98 for testing data. Figure 3 (a) and (b) show the comparison of the rainfall truth data, and the prediction results by the standard backpropagation algorithm and the backpropagation algorithm optimized by BFGS algorithm for training data, respectively. Figure 4 (a) and (b) show the comparison of the rainfall truth data, and the prediction results by the standard backpropagation algorithm and the backpropa-

Table. 1. Evaluation results by using data training for different architectures with learning rate 0.005.

Architecture	MSE (Means Square Error)		Computational Time	
	Gradient Descent	BFGS	Gradient Descent	BFGS
3-5-1	22.29	**22.29**	10.02	**8.03**
3-20-1	20.38	**13.96**	**11.46**	14.55
3-50-1	18.86	**7.85**	**10.14**	30.40
3-100-1	17.47	**0.32**	**10.02**	24.65
3-150-1	16.38	**0.32**	**11.73**	69.88

Table. 2. Evaluation results by using data test for different architecture with learning rate 0.005.

Architecture	MSE (Means Square Error)	
	Gradient Descent	BFGS
3-5-1	**31.74**	**31.74**
3-20-1	27.10	**23.76**
3-50-1	23.71	**20.13**
3-100-1	**18.98**	109.41
3-150-1	**25.17**	126.38

Table. 3. Evaluation results by using data training for different learning rate with 50 neurons hidden.

Learning rate	MSE (Means Square Error)		Computational Time	
	Gradient Descent	BFGS	Gradient Descent	BFGS
0,001	19.37	**12.094**	**9.94**	18.56
0,005	20.38	**13.96**	**11.46**	14.55
0,01	18.84	**10.09**	**9.98**	14.95
0,1	19.23	**10.25**	**7.71**	31.92
0,2	18.69	**10.87**	**10.99**	31.09

Table. 4. Evaluation results by using data test for different learning rate with 50 neurons hidden.

Learning rate	MSE (Means Square Error)	
	Gradient Descent	BFGS
0,001	27.73	**19.27**
0,005	27.10	**23.76**
0,01	27.64	**19.41**
0,1	24.90	**23.48**
0,2	26.08	**23.24**

gation algorithm optimized by the BFGS algorithm for testing data, respectively. From the figures 3 and 4, it can be seen that the backpropagation algorithm optimized by BFGS algorithm gives better result compared with the standard backpropagation algorithm for predicting the amount of rainfall.

5. Conclusion

The learning rate, the neurons number of the hidden layer and the maximum number of iterations should be determined. It can be concluded that the performance of the backpropagation neural network depends on the number of neurons in the hidden layer and the learning used. The number of the hidden layer should not be too small and too many. The big number of neuron hidden causes over fitting and the small number of neuron hidden make the worst accuracy. When the number of neurons in the hidden layer increases, the computational time also increases. The increasing of computational time does not increase significantly for the gradient descent, but the computational time for the BFGS method increase significantly. Learning rates that are too small or too large will result in poor accuracy. Therefore, choosing the right learning rate will produce better accuracy. The backpropagation algorithm optimized by the BFGS algorithm gives better result compared with the standard backpropagation algorithm for predicting the amount of rainfall.

Acknowledgments

We would like to say our greatest thanks to Brawijaya University who has supported to join this conference.

References

[1] Aydinalp C and Cresser M S 2008 The effects of global climate change on agriculture American-Eurasian *J. Agric. & Environ. Sci.* **3** (5) pp 672-676.

[2] Dhara V R, Schramm P J and Luber G 2013 Climate change & infectious diseases in India: Implications for health care providers *Indian J Med Res* **138** pp 847-852

[3] Xu F, He Z, Sha Z, Zhuang L and Sun W 2013 Assessing the impact of rainfall on traffic operation of urban road network *Procedia - Social and Behavioral Sciences* **96** pp 82-89

[4] Eni D and Adeyeye F J 2015 Seasonal ARIMA modeling and forecasting of rainfall in Warri Town, Nigeria *Journal of Geoscience and Environment Protection* **3** pp 91-98

[5] Adebiyi A A, Adewumi A O and Ayo C K 2014 Comparison of ARIMA and artificial neural networks models for stock price prediction *Journal of Applied Mathematics* **2014** pp1-7

[6] Anam S, Yanti I and Widodo A 2010 Penggunaan metode quasi Newton pada jaringan syaraf tiruan backpropagation *Jurnal Teknika* **2** (1) pp 43-56

[7] Novelina P and Brodjol S S U 2016 Pemodelan data curah hujan di Kabupaten Banyuwangi dengan metode ARIMA dan radial basis function neural network *Jurnal Sains dan Seni ITS* **5** (2) pp 2337-3520

[8] Wilson E M 1993 *Hidrologi Teknik*, Erlangga Jakarta