2018 e3sconf_icenis18_05017

by Hasbi Yasin

Feed Forward Neural Network Modeling for Rainfall Prediction

Hasbi Yasin[1*], *Budi* Warsito[2], *Rukun* Santoso[3]

[1,2,3]Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang – Indonesia

Abstract. Prediction of rainfall data by using Feed Forward Neural Network (FFNN) model is proposed. FFNN is a class of neural network which has three layers for processing. In time series prediction, including in case of rainfall data, the input layer is the past values of the same series up to certain lag and the output layer is the current value. Beside a few lagged times, the seasonal pattern also considered as an important aspect of choosing the potential input. The autocorrelation function and partial autocorrelation function patterns are used as aid of selecting the input. In the second layer called hidden layer, the logistic sigmoid is used as activation function because of the monotonic and differentiable. Processing is done by the weighted summing of the input variables and transfer process in the hidden layer. Backpropagation algorithm is applied in the training process. Some gradient based optimization methods are used to obtain the connection weights of FFNN model. The prediction is the output resulting of the process in the last layer. In each optimization method, the looping process is performed several times in order to get the most suitable result in various composition of separating data. The best one is chosen by the least mean square error (MSE) criteria. The least of in-sample and out-sample predictions from the repeating results been the base of choosing the best optimization method. In this study, the model is applied in the ten-daily rainfall data of ZOM 136 Cokrotulung Klaten. Simulation results give a consecution that the more complex architecture is not guarantee the better prediction.

Keywords: rainfall; FFNN; prediction, RMSE

1. INTRODUCTION

Neural network modeling for rainfall prediction has been rapidly developed in recent many years. It desiderated can respond the need of reliable modelling in rainfall prediction. Understanding the pattern of rainfall data and get an accurate prediction are very important tasks in hydrology and climate fields. Reliable rainfall prediction also give a great impact in forecasting the rainfall data on daily, ten-daily or monthly and seasonal time scales. It provide useful information for water resource management, agricultural planning, and associated crop insurance application significant implications for food production, securing water supplies for major population centres, and minimizing flood risks. Some researches about rainfall prediction by using neural network have been conducted conducted (Cigizoglu et al, 2009; Benmahdjouba et al., 2013; Asadi et al., 2013) [1-3]. Some other publications related to comparison and developing new models also have been conducted (Lin and Wu, 2009; Mekanik et al, 2013 ; Kim et al., 2016) [4-6]. In this research we applied Feed Forward Neural Network (FFNN), a main class of neural network modelling, for rainfall prediction in ZOM 136 Cokrotulung Klaten.

Some of the interesting parts of neural network modeling are procedures of determining the optimal input, the number of hidden unit, the activation function used in the hidden layer and the choosing of optimization method for obtaining the weights of the network. Various procedures have been evolved for obtaining the most efficient and reasonable network in both theoretically and practically. The optimization technique for obtaining network weights is one of the main focus in neural network modelling. Various standard gradient based methods have been used to estimate the weights. Therefore, the activation function permitted in the hidden layer should be continue and differentiable functions. In this research, the logistic sigmoid and the linear function are used as activation functions in hidden layer and output layer, respectively. Various architecture have been conducted for choosing the best one. In each architecture, the process is repeated several times for getting the least Mean Square Error (MSE) as the based of choosing the best architecture. The three gradient based optimization are used to optimize the neural network weights.

2. MATERIAL AND METHODS

In this section, the neural network model for time series prediction is explained. Neural network architecture consists of three layers namely input layer, hidden layer and output layer. Each layer has a connection with the previous layer and called weight. Unlike in classification or regression problems, in neural network modelling for time series prediction the input consists of the past values of the same variable. The lagged variables as input are determined by various techniques. Some researches employ the best ARIMA

---

* Corresponding author: hasbiyasin17@gmail.com

model for determining the input. While, the overfitting and underfitting technique also often be used as inquiring the alternative ones. Some other researches utilize certain algorithm so that neural network model is free-standing from other model. In this research the best ARIMA model is chosen for determining input. Architecture of neural network for time series prediction is described in Fig. 1. Output layer just have one neuron because of the univariate case. If the network is applied in multivariate time series, the more than one neuron is needed for describing the model.
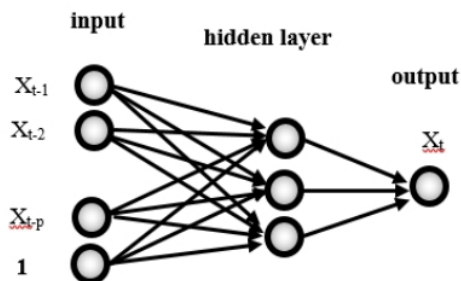

Figure 1. Neural network architecture for time series prediction

In the form of mathematical model, the network architecture can be described as follows.

$$x_t = f^o\left(w^b + \sum_{j=1}^{l} w_j^o f^h\left(w_j^b + \sum_{i=1}^{p} w_{ji}^h x_{t-i}\right)\right) \tag{1}$$

where, f^h is activation function at hidden layer and the weights vector of the network consists of weights from input layer to hidden layer (w_{ij}), weights from bias to hidden layer (w_{bj}), weights from hidden layer to output (w_j), and weights from bias to output (w_b). Therefore, the three optimization methods often used in neural network modeling for time series prediction is briefly discussed.

Backpropagation algorithm in the FFNN model consists of three stages, namely feedforward of the input pattern, calculating the error and adjusting the weights. In this algorithm, the initial weights are specified first. By the feedforward stage, the output of the initial points is calculated. Furthermore, the difference between output and target, namely error, is found. Weights updating is done by using a gradient based method for obtaining the smaller error. The termination criteria are specified for stopping the iteration process. Maximum epochs permitted and minimum error reached are the two basic rules commonly used as the stopping criteria. In this research we used the three gradient based methods for obtaining the weights of FFNN, that are Gradient Descent, Levenberg-Marquardt and Quasi Newton BFGS. Levenberg-Marquardt algorithm is a virtual standard in nonlinear optimization. It is the most widely used optimization algorithm. It outperforms simple gradient descent and other conjugate gradient methods in

a wide variety of problems. In many cases, it finds a solution even if it starts very far off the final minimum. It is a very popular curve-fitting algorithm used in many software applications for solving generic curve-fitting problems. However, as for many fitting algorithms, it finds only a local minimum, which is not necessarily the global minimum. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) method searches a stationary point of a (preferably twice continuously differentiable) function. A necessary condition for optimality of such problems is that the gradient be zero. If the function has not a quadratic Taylor expansion near an optimum, the BFGS method is not guaranteed to converge. This method uses both the first and second derivatives of the function. Gradient descent is a first-order optimization algorithm. To find a local minimum of a function, it need to takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. Momentum method is an extension of Gradient descent method, that reduces the risk of a getting stuck in a local minimum, as well as speeds up the convergence considerably in cases where the process would otherwise zigzag heavily. Momentum simply adds a fraction of the previous weight update to the current one. When the gradient keeps pointing in the same direction, this will increase the size of the steps taken towards the minimum. It is therefore often necessary to reduce the global learning rate when we use a lot of momentum.

3. RESULTS AND DISCUSSION

Procedure of neural network modelling by using standard gradient based optimization methods is applied in the ten-daily rainfall data of ZOM 136 Cokrotulung, Klaten, Central Java Indonesia from January 2010 until July 2018 with the length of 309. We tried to divide the data into two parts with various ways. In the first experiment, the first 216 data (70%) is used as training and the remaining 93 data (30%) as testing. In the second experiment, the first 247 data (80%) is used as training and the remaining 62 data (20%) as testing, and the last experiment the first 278 data (90%) is used as training and the remaining 31 data (10%) as testing. In each composition, the same architecture is repeated for 30 times with 1-10 hidden units, respectively. The activation function used in hidden layer is logistic sigmoid. The maximum number of epochs in each experiment is 1000. The minimum root mean square error of each architecture is chosen for obtaining the best one. We need to refer to the values of the series in previous periods as the input. Determining lagged variables as input is performed by ARIMA model. In early step, the investigations resulting the lagged variables of 1, 2 and 18 as input. The summarize of the processes is shown in Table 1 whereas plot of the in-sample prediction of training data and plot of the out-sample prediction of testing data by using the FFNN model in one of the iterations in the composition of 80:20 are shown in Fig. 1 and Fig. 2.
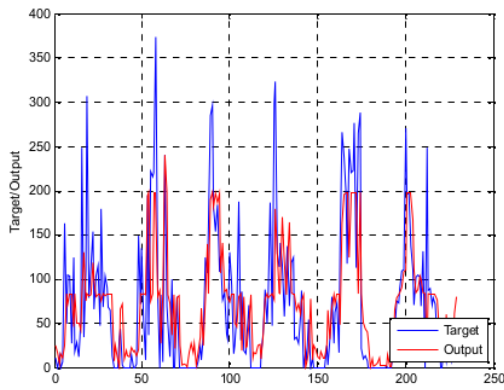
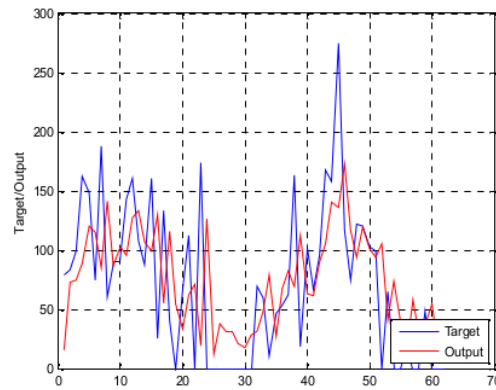Figure 1. Plot of the in-sample prediction of training data by using FFNN model of rainfall data in ZOM 136

Figure 2. Plot of the out-sample prediction of testing data by using FFNN model of rainfall data in ZOM 136

Table 1. MSE of the simulation results of the FFNN modelling for rainfall data in ZOM 136

Method	Number of hidden units	MSE (70:30)		MSE (80:20)		MSE (90:10)	
		training	testing	training	testing	training	testing
Levenberg Marquardt	1	4.1188	2.8422	3.9301	2.8022	3.8221	2.5314
	2	3.5636	2.6539	3.4541	2.7730	3.4382	2.5582
	3	3.3426	3.7274	3.0883	3.4473	3.2104	3.2408
	4	2.9685	3.2851	2.7897	3.2876	2.8928	3.3459
	5	2.6997	3.7548	2.6816	3.8313	2.6995	3.3035
	6	2.4385	4.5895	2.5553	4.0972	2.4444	4.0975
	7	2.1505	4.6786	2.2640	4.4995	2.2861	5.2426
	8	2.6979	4.2517	2.1055	3.7509	2.1644	3.6958
	9	1.8876	6.6311	1.9329	4.7806	1.9265	3.1208
	10	1.6476	5.2667	1.8566	4.7764	1.9886	6.3686
BFGS	1	4.1188	2.8422	3.9301	2.8022	3.8221	2.5314
	2	3.5524	3.1232	3.4511	2.8181	3.4029	2.4601
	3	3.3454	3.0057	3.2613	3.6890	3.1349	3.3095
	4	3.0568	3.9355	2.8354	3.3102	2.8291	4.4433
	5	2.6979	4.2517	2.7289	4.4600	2.6751	3.1794
	6	2.4150	5.1342	2.4647	4.1706	2.4924	3.2462
	7	2.1589	6.9416	2.1668	6.4382	2.2689	3.5214
	8	1.9230	8.3038	1.9874	5.9726	2.2900	4.7923
	9	2.0186	11.876	1.8498	3.1678	2.1067	5.1394
	10	1.7660	3.0204	1.8990	14.397	2.0371	3.5021
Gradient Descent	1	4.2300	2.6353	4.0235	2.7166	3.9598	2.4910
	2	4.3406	2.9412	4.0891	2.9414	3.9255	2.5881
	3	4.1520	2.9953	4.0321	2.6888	3.9116	2.4890
	4	4.0773	2.8096	3.9594	3.0428	3.8149	2.6279
	5	4.1800	2.8265	3.8858	2.7964	3.8464	2.7556
	6	4.1005	2.7761	3.9106	2.6953	3.7502	2.5104
	7	3.9610	3.2023	3.8622	2.7936	3.7699	2.4018
	8	3.9977	2.8670	3.8305	2.9564	3.7324	2.4815
	9	3.8898	2.8154	3.8532	2.8782	3.7338	2.5604
	10	3.9750	2.5958	3.8829	2.7293	3.7003	2.6720

Fig. 1 shows the actual versus in-sample prediction of FFNN model in the rainfall data of ZOM 136. Generally, the seasonal pattern of the in-sample data can be followed by the prediction. The period of repetitive phenomena between both data are the same. None the worse, the model has a blind side. There are still several points far from as proper. Some points that cannot be reached by the model are the extreme values. It needs more extensive discussion for using the more suitable optimization methods for optimizing neural network related to the data type. The further open question also can be discussed about the more appropriate model for the data type. Similarly, the graph of testing data versus out-sample prediction in Fig. 2 also shows the resemblance between the two. Actual data which has an extreme value cannot be reached by the prediction but in general the out-sample prediction can keep abreast the pattern of actual data.

The results of Table 1 show that a better in-sample prediction is not give a guarantee that the out-sample also better. In consequence, we should make decision of fitting the best one by using the two criteria, both in-sample and out-sample predictions. For the first composition, the 70% as training and 30% as testing, it is reasonable if we choose architecture with ten neurons in hidden layer and BFGS as optimization technique for constructing the selected one. The in-sample prediction has minimum of mean square error, while the out-sample prediction give a near the least one. Similar reason is considerable for choosing architecture which has nine neurons and BFGS optimization as the best architecture of the second composition. In the third separation, 90% as training and 10% as testing, selecting the optimal number of hidden units is debatable. The one choice is two hidden neurons -it is more balance between in-sample and out-sample- and the other one is nine hidden neurons. Both of them use Levenberg Marquardt as optimization method. The decision is depended on various factors. If the simplicity, parsimony and balanced results take into consideration, the first choice is better.

The second choice is good for considering the strong capability in-sample prediction.

4. CONCLUSION

Procedure of rainfall prediction by utilizing Feed Forward Neural Network has been developed. The more complex network architecture not give a guarantee of a better prediction result. Levenberg Marquardt and BFGS algorithm can be regarded as the main optimization methods whereas Gradient Descent is not recommended for the data type. Comparison of the results with the other technique like heuristic optimization can be regarded as an interesting problem for the future work.

5. ACKNOWLEDGEMENT

REFERENCES

1. Asadi, S., Shahrabi, J., Abbaszadeh, P., Tabanmehr, S., 2013, *A new hybrid artificial neural networks for rainfall–runoff process modeling*, Neurocomputing, 121, pp 470–480
2. Benmahdjouba, K., Ameura, Z., Boulifa, M., 2013, *Forecasting of Rainfall using Time Delay Neural Network in Tizi-Ouzou (Algeria)*, Energy Procedia 36 (2013) 1138 – 1146
3. Cigizoglu, H.,K., Askin., P., Ozturk, A., Gurbuz, A., Ayhan, O., Yildiz, M and Ucar, I., 2008, *Artificial Neural Network Models in Rainfall-Runoff Modelling of Turkish Rivers*, International Congress on River Basin Management, pp. 560-571
4. Kim, S., Seo, Y., and Lee, C.J., 2016, *Modeling of Rainfall by Combining Neural Computation and Wavelet Technique*, Procedia Engineering, 154: pp 1231-1236
5. Lin, G.F dan Wu, M.C., 2009, *A hybrid neural network model for typhoon-rainfall forecasting*,

Journal of Hydrology 375 (2009) 450–458 (SOM &MLP)

6. Mekanik, F., Imteaz, M.A., Trinidad, S.G., Elmahdi, A., 2013, *Multiple Regression and Artificial Neural Network for Long-Term Rainfall Forecasting Using Large Scale Climate Modes*, Journal of Hydrology 503 (2013) 11–21

7. Wang, Y.M. and Traore, S., 2009, *Time-lagged recurrent network for forecasting episodic event suspended sediment load in typhoon prone area*, International Journal of Physical Sciences Vol. 4 (9), pp. 519-528, Available online at http://www.academicjournals.org/ijps

8. Yasin, H., 2014, *Prediction of Weekly Rainfall in Semarang City use Support Vector Regressions*, Prosiding Seminar Internasional the 4th ISNPINSA, Hotel Patra Jasa Semarang

9. Yasin, H. dan Prahutama, A., 2015, *Prediction of Weekly Rainfall in Semarang City Use Support Vector Regression (SVR) with Quadratic Loss Function*, International Journal of Science and Engineering (IJSE), ISSN: 1086-5023, Vol. 9 No.1

2018 e3sconf_icenis18_05017

Frequency Curves of Peninsular Malaysia under Changing Climate Scenarios", Water, 2018
Publication

7   Submitted to Malaviya National Institute of Technology
Student Paper   1%

8   Smart Innovation Systems and Technologies, 2017.
Publication   1%

9   Junaida Sulaiman, Siti Hajar Wahab. "Chapter 9 Heavy Rainfall Forecasting Model Using Artificial Neural Network for Flood Prone Area", Springer Nature, 2018
Publication   1%

10   www.academicjournals.org
Internet Source   1%

11   Submitted to Universiti Putra Malaysia
Student Paper   1%

12   Fahimi, Farzad, Zaher Mundher Yaseen, and Ahmed El-shafie. "Application of soft computing based hybrid models in hydrological variables modeling: a comprehensive review", Theoretical and Applied Climatology, 2016.
Publication   1%

13   zenodo.org

Internet Source

1 %

14 www.sensorsportal.com
Internet Source

<1 %

15 www.crojfe.com
Internet Source

<1 %

16 article.sapub.org
Internet Source

<1 %

17 shareok.org
Internet Source

<1 %

| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | Off | | |

2018 e3sconf_icenis18_05017

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5