Media Statistika

p-ISSN 1979 – 3693 e-ISSN 2477 – 0647
MEDIA STATISTIKA 13(2) 2020: 116-124
http://ejournal.undip.ac.id/index.php/media_statistika

SELECTION OF INPUT VARIABLES OF NONLINEAR AUTOREGRESSIVE NEURAL NETWORK MODEL FOR TIME SERIES DATA FORECASTING

Hermansah[1,2], Dedi Rosadi[1], Abdurakhman[1], Herni Utami[1]
[1] Mathematics Study Program, Gadjah Mada University
[2] Mathematics Education Study Program, Riau Kepulauan University

e-mail: bankhermansah@gmail.com

DOI: 10.14710/medstat.13.1.116-124

Article Info:
Received: 20 April 2020
Accepted: 19 October 2020
Available Online: 28 December 2020

Keywords:
Stepwise Method, Learning Method, Activation Function, Ensemble Operator, NARNN Model
.

Abstract: NARNN is a type of ANN model consisting of a limited number of parameters and widely used for various applications. This study aims to determine the appropriate NARNN model, for the selection of input variables of nonlinear autoregressive neural network model for time series data forecasting, using the stepwise method. Furthermore, the study determines the optimal number of neurons in the hidden layer, using a trial and error method for some architecture. The NARNN model is combined in three parts, namely the learning method, the activation function, and the ensemble operator, to get the best single model. Its application in this study was conducted on real data, such as the interest rate of Bank Indonesia. The comparison results of MASE, RMSE, and MAPE values with ARIMA and Exponential Smoothing models shows that the NARNN is the best model used to effectively improve forecasting accuracy.

1. INTRODUCTION

Time series data forecasting is carried out by studying existing patterns using numerical values and estimating future values based on these patterns (Abraham & Ledolter, 2005). Its forecasting process can be classified into linear and nonlinear methods. The popular linear forecasting methods include the Auto-Regressive Integrated Moving Average (ARIMA) and Exponential Smoothing models. Although this linear model successfully analyzes linear time series data, it is not good at modeling nonlinear data (Samarasinghe, 2007). Generally, forecasting nonlinear time series provides the requirements for the needed specification. According to Zhang (2003), the difficulty in determining the nonlinear function makes this model less useful in modeling nonlinear time series data. Artificial Neural Network (ANN) was introduced as an alternative model to overcome the nonlinear functions used in time series data forecasting. ANN can take a universal approach and does not need processed data knowledge (Walczak, 2001). Previous studies show that ANN with Nonlinear Auto-Regressive Neural Network (NARNN) has good performance for nonlinear data models. Similarly, several studies have shown exemplary performance in long-term forecasting, of monthly and quarterly time-series data replaced with general statistical methods, such as linear regression (Warner & Misra, 1996).

The use of NARNN model to determine time series data forecasting can be carried out using the univariate model, which utilizes past predicted data. This model contains a limited number of parameters in its application. Therefore, this study aims to determine the appropriate NARNN model, for the selection of input variables of the nonlinear autoregressive neural network model for time series data forecasting, using the stepwise method. Initial input variables are taken from the time series data frequency, which was selected using the stepwise method. The optimal number of neurons in the hidden layer is selected over several architectures, using the trial and error method. This study is an innovation from the research carried out by Suhartono, Subanar, & Guritno (2006) and Crone & Kourentzes (2010), which introduced the procedure for forming the NARNN model applied to forecasting time series data. In this study, the NARNN model is combined in three parts, namely the learning method, the activation function, and the ensemble operator, to obtain the best single model.

There are five learning methods used to calculate the NARNN model, namely backpropagation, resilient backpropagation with weight backtracking, resilient backpropagation without weight backtracking, globally convergent algorithm with smallest absolute gradient, and globally convergent algorithm with smallest learning rate. The detailed descriptions of each learning method can be seen from the research carried out by Riedmiller & Braun (1993), and Anastasiadis, Magoulas, & Vrahatis (2005). Two activation functions are used, namely logistic and tangent hyperbolic. Furthermore, three ensemble operators are used including mean, median, and mode with a detailed description of each ensemble operator indicated in Kourentzes, Barrow, & Crone (2014).

The application of the NARNN model in this study was carried out on real data, such as the interest rate of Bank Indonesia. The fully manual specification is carried out using the R program of the mlp function of the nnfor package with a detailed description in Kourentzes (2019). The measurement of forecasting accuracy is carried out using Mean Absolute Scaled Error (MASE), Root Mean Squared Error (RMSE), and Mean Absolute Percent Error (MAPE) values. The values of MASE, RMSE, and MAPE are taken due to their ability to measure the time series data forecasting studies accurately. Finally, the comparison of MASE, RMSE, and MAPE values was carried out with the ARIMA and Exponential Smoothing models.

2. LITERATURE REVIEW

2.1. NARNN Model

The Artificial Neural Network (ANN) model is generally the most widely used in engineering or engineering applications. It is a Nonlinear Auto-Regressive Neural Network (NARNN) model, which is also known as the Multi-Layer Perceptron (MLP) or Feed-Forward Neural Network (FFNN). Typically, applications for time series data modeling are based on the NARNN architecture, with regression modeling, time series, and signal processing among the ANN applications usually based on the NARNN architecture.

In statistical modeling, NARNN can be viewed as a flexible class of nonlinear functions. Generally, this model works by accepting a vector of input x and then calculating a response or output $\hat{y}(x)$ by processing (propagating) x through interrelated process elements arranged in layers. The input data, x, sequentially flows from one layer to the next with the inputs nonlinearly transformed into layers by processing elements in each layer. Finally, the output values of $\hat{y}$, which can be scalar or vector, are computed at the output

layer (Suhartono et al., 2006). The output values of $\hat{y}$ are calculated as follows in Equation 1.

$$\hat{y}_{(k)} = f^o[\sum_{j=1}^{q}[w_j^o f_j^h\left(\sum_{i=1}^{p} w_{ji}^h x_{i(k)} + b_j^h\right) + b^o]] \tag{1}$$

with

$x_{i(k)}$: input variable as much as p, $(i = 1, 2, \dots, p)$

$\hat{y}_{(k)}$: estimated value of the output variable

k : input-output data pair index $(x_{i(k)}, \hat{y}_{(k)})$, $k = 1, 2, \dots, n$

w_{ji}^h : weight of the i-th input to the j-th neuron in the hidden layer, $(j = 1, 2, \dots, q)$

b_j^h : bias in j-th neuron in hidden layer, $(j = 1, 2, \dots, q)$

f_j^h : activation function in j-th neuron in hidden layer

w_j^o : weight of the j-th neuron in the hidden layer leading to the neuron in the output layer

b^o : bias in the neurons in the output layer

f^o : activation function on neurons in the output layer.

The nonlinear form of $\hat{y}$ function occurs through the activation function f_j^h and f^o in the hidden and output layer, using a logistic or tangent hyperbolic function.

Several notations are used to clarify the NARNN input-output process. The superscripts h and o are used as an index representing the hidden and output layers. It also used v_j^h to express a vector of values after the sum of the inputs and weights (bias is included) in the hidden layer in the j-th neuron, namely

$$v_j^h = \sum_{i=1}^{p} w_{ji}^h x_i + b_j^h \tag{2}$$

or for the k-th data obtained

$$v_{j(k)}^h = \sum_{i=1}^{p} w_{ji}^h x_{i(k)} + b_j^h \tag{3}$$

The output in the hidden layer, which is processed in the j-th neuron, is

$$a_j^h = f_j^h(v_j^h) \tag{4}$$

or for the k-th data obtained

$$a_{j(k)}^h = f_j^h(v_{j(k)}^h) = f_j^h(\sum_{i=1}^{p} w_{ji}^h x_{i(k)} + b_j^h) \tag{5}$$

Similarly, several notations that state the sum of the inputs and weights in the output layer are

$$v^o = \sum_{j=1}^{q} w_j^o a_j^h + b^o \tag{6}$$

or for the k-th data obtained

$$v_{(k)}^o = \sum_{j=1}^{q} w_j^o a_{j(k)}^h + b^o \tag{7}$$

The output at the output layer is

$$\hat{y}_{(k)} = a_{(k)}^o = f^o(v_{(k)}^o) \tag{8}$$

Therefore, the relationship between the input $x_{i(k)}$, $i = 1, 2, \dots, p$ and $k = 1, 2, \dots, n$, with the output $\hat{y}_{(k)}$ is

$$\begin{aligned} \hat{y}_{(k)} &= f^o[\sum_{j=1}^{q} w_j^o f_j^h(v_{j(k)}^h) + b^o] \\ &= f^o[\sum_{j=1}^{q} w_j^o f_j^h(\sum_{i=1}^{p} w_{ji}^h x_{i(k)} + b_j^h) + b^o] \\ &= F(x_{1(k)}, x_{2(k)}, \dots, x_{p(k)}) \end{aligned} \tag{9}$$

The overall mapping that occurred in NARNN can then be written in the following form

$$\begin{bmatrix} \hat{y}_{(1)} \\ \hat{y}_{(2)} \\ \vdots \\ \hat{y}_{(n)} \end{bmatrix} = \begin{bmatrix} F(x_{1(1)}, x_{2(1)}, \dots, x_{p(1)}) \\ F(x_{1(2)}, x_{2(2)}, \dots, x_{p(2)}) \\ \vdots \\ F(x_{1(n)}, x_{2(n)}, \dots, x_{p(n)}) \end{bmatrix} \tag{10}$$

2.2. Stepwise Method

Before using the stepwise method, the first step is to determine the lag of the data based on the number of frequencies. When the data frequency is m, then successive lag m from lag one is used, for example, 1:4 and 1:12 for quarterly and monthly data lag, respectively. The optimal lag (variable) is then selected using the stepwise method, which selects the variable based on the most considerable partial correlation included in the model. Variables that are already in the model can be removed again, however, when one of them is entered into the model, then the other does not need to be included again because the effect is represented by the variables that have been included. Therefore, there is no multicollinearity in the resulting model (Sembiring, 1995).

2.3. Learning Method

The learning method consists of two methods, namely, supervised and unsupervised. Its main objective is to regulate the weights that exist in NARNN to ensure the final weight is obtained according to the trained data pattern (Yeung, Cloete, Shi, & Ng, 1998). In the supervised learning process, one input given to a neuron in the input layer runs along the NARNN to the neuron in the output with the results matched with the target. An error tends to appear when there is a difference, with learning conducted when the value is significant. Meanwhile, in the unsupervised learning process, and weight values are arranged in a

specific interval depending on the input. Unsupervised learning aims to group similar units in a specific area. This study makes use of five supervised learning methods, namely backpropagation, resilient backpropagation with weight backtracking, resilient backpropagation without weight backtracking, globally convergent algorithm with smallest absolute gradient, and globally convergent algorithm with smallest learning rate. Descriptions and implementation details for each learning method are shown in Riedmiller & Braun (1993), and Anastasiadis et al. (2005) research.

2.4. Activation Function

The activation function is used to determine the output of a neuron as well as to activate or deactivate the neurons used in the network. Some of the activation functions often used in NARNN are linear, logistic, and tangent hyperbolic functions. Linear functions have an output value equal to their input. Meanwhile, logistic functions have values between 0 to 1, and tangent hyperbolic is often used as activation functions when the desired output value ranges from -1 to 1 (Fausett, 1994). This study uses a hidden layer with the logistic or tangent hyperbolic activation function and an output layer with the linear activation function.

2.5. Ensemble Operator

Many experiments have shown that the generalization results of NARNN are not unique (singular), therefore, the solution is unstable. This is because a small change in the parameter leads to a massive change in the forecasting output. Furthermore, several structures with different connection weights give different generalization results. Therefore, the limitation in choosing the best model from a single NARNN becomes a problem because the approach is introduced by assuming that the discarded model has the potential as a candidate model. Furthermore, combining several NARNN can help overcome the weaknesses in choosing a single model. This merger of several NARNN is known as the NARNN ensemble, which uses mean, median and mode operators for forecasting. This study combined 20 networks with different connection weights using mean, median, and mode operators. Kourentzes et al. (2014) research contain descriptions and detailed implementation of each ensemble operator.

3. METHODOLOGY

This study aims to select the optimal input variable from the NARNN model using the stepwise method for forecasting the interest rate data of Bank Indonesia. Primary data were obtained from the monthly interest rate data of Bank Indonesia from July 2005 to August 2016, which consists of 134 observations, while secondary data were obtained from www.bi.go.id. In this NARNN model application, data is divided into two parts, namely, training and testing data. The first and last data consisting of 129 and 5 observations, was used as training, and testing data, with a frequency of 12, therefore 12 consecutive lags starting from lag one are used as input variables. The optimal input variable is then selected using the stepwise method, while the optimal number of neurons in the hidden layer is selected using trial and error. Furthermore, the NARNN model is combined into three parts, namely the learning method, the activation function, and the ensemble operator, to get the best single model. Finally, the values of Mean Absolute Scaled Error (MASE), Root Mean Squared Error (RMSE) and Mean Absolute Percent Error (MAPE) are compared in testing data forecasting with ARIMA and Exponential Smoothing models.

4. RESULTS

In this case study, the optimal input variable (lag) in the interest rate data of Bank Indonesia is lag-1, lag-6, and lag-12. Furthermore, the optimal neuron in the hidden layer is determined using neurons 1:10 as a training model. The best training model is selected based on the smallest MASE, RMSE, and MAPE values from the several models built. Based on trial and error, the optimal number of neurons in the hidden layer is 5 with MASE of 0.331768, RMSE of 0.053238, and MAPE of 0.005223. The complete results of the training model used to determine the number of neurons in the hidden layer are shown in Table 1, therefore, the NARNN model that is formed is an architecture with three neurons in the input layer and five in the hidden layer.

Furthermore, the NARNN model is combined in three parts, namely the learning method, the activation function, and the ensemble operator, to get the best single model. The five learning methods used to calculate the NARNN model are backpropagation (backprop), resilient backpropagation with weight backtracking (rprop+), resilient backpropagation without weight backtracking (rprop-), globally convergent algorithm with the smallest absolute gradient (sag), and globally convergent algorithm with smallest learning rate (slr). There are two activation functions, namely the logistic and tangent hyperbolic (tanh), with three ensemble operators, consisting of mean, median, and mode. Each ensemble operator is compared using empirically different learning methods and activation functions. There are thirty models of forecasting results from the formation of the NARNN model based on learning methods, activation functions, and ensemble operators. The model function is determined with the smallest MASE, RMSE, and MAPE values in training data forecasting. The complete results of the modeling are shown in Table 2.

Table 1. Value of MASE, RMSE, and MAPE Based on Hidden Layer Neurons

Number of neurons	MASE	RMSE	MAPE
1 **	0.819916	0.113456	0.012099
2 **	0.689173	0.100596	0.010413
3 **	0.548739	0.085728	0.008187
4 **	0.598546	0.096921	0.009121
5 **	0.331768	0.053238	0.005223
6 **	0.608426	0.094700	0.009252
7 **	0.457622	0.079324	0.007061
8 **	0.350492	0.055376	0.005367
9 **	0.541127	0.086116	0.008223
10 **	0.587925	0.091228	0.008950

Note: ** is training using the learning method of resilient backpropagation with weight backtracking, the activation function of logistic, and without the ensemble operator (individual NARNN)

Based on the results of model formation in Table 2, empirically, it can be seen that the NARNN model uses the learning method of resilient backpropagation with weight backtracking. The activation function of hyperbolic tangent and the ensemble operator of median gives the smallest errors in the MASE, RMSE and MAPE values of 0.404058, 0.070705 and 0.006296. Furthermore, the learning method of resilient backpropagation with weight backtracking shows that the learning steps is significantly reduced and effective. The convergence time and resilience are the most promising compared to other learning methods. Similarly, the activation function of a hyperbolic tangent makes the overall model results better and consistent. Meanwhile, the ensemble operators of mean, median, and mode show

that the difference is not significant. In this case study, the median ensemble operator provides the smallest error in forecasting accuracy.

Finally, three forecasting models are used for comparison, namely, the ARIMA, Exponential Smoothing, and NARNN. The best model is determined by using the cross-validation method, which gives the smallest error in forecasting data testing. Comparisons were made using the MASE, RMSE, and MAPE values in each model. The complete results of comparing the accuracy of the three forecasting models are shown in Table 3. The best model obtained in the testing data is the NARNN model with a MASE, RMSE and MAPE of 0.513163, 0.084999, and 0.007875, respectively. From Table 3, it can be concluded that the NARNN model using the learning method of resilient backpropagation with weight backtracking, the activation function of hyperbolic tangent and the ensemble operator of the median is the best model.

Table 2. Summary of the Results of the Formation of the NARNN Model

Combination			Accuracy Measure		
Learning Method	Activation Function	Ensemble Operator	MASE	RMSE	MAPE
backprop	logistic	mean	0.700823	0.102030	0.010452
rprop+	logistic	mean	0.472535	0.078817	0.007274
rprop-	logistic	mean	0.484726	0.079372	0.007405
sag	logistic	mean	0.525671	0.083282	0.008028
slr	logistic	mean	0.499525	0.081900	0.007684
backprop	tanh	mean	0.636371	0.096726	0.009606
rprop+	tanh	mean	0.444651	0.073125	0.006854
rprop-	tanh	mean	0.433606	0.072987	0.006696
sag	tanh	mean	0.496871	0.084458	0.007592
slr	tanh	mean	0.441291	0.078794	0.006899
backprop	logistic	median	0.705963	0.103704	0.010460
rprop+	logistic	median	0.487490	0.081316	0.007490
rprop-	logistic	median	0.524638	0.087260	0.008044
sag	logistic	median	0.486105	0.080839	0.007452
slr	logistic	median	0.505134	0.084623	0.007774
backprop	tanh	median	0.635339	0.095196	0.009556
rprop+	tanh	median	0.404058	0.070705	0.006296
rprop-	tanh	median	0.440080	0.077191	0.006833
sag	tanh	median	0.452027	0.078396	0.007034
slr	tanh	median	0.460600	0.072756	0.007009
backprop	logistic	mode	0.677271	0.102008	0.010077
rprop+	logistic	mode	0.511154	0.088301	0.007853
rprop-	logistic	mode	0.475049	0.081216	0.007315
sag	logistic	mode	0.568988	0.090844	0.008551
slr	logistic	mode	0.528960	0.087294	0.008117
backprop	tanh	mode	0.635551	0.098845	0.009585
rprop+	tanh	mode	0.431016	0.078721	0.006721
rprop-	tanh	mode	0.450797	0.080183	0.007006
sag	tanh	mode	0.517586	0.086009	0.008029
slr	tanh	mode	0.550592	0.089785	0.008420

Table 3. Summary of Comparison Results

Forecasting Method	MASE	RMSE	MAPE
ARIMA	0.778748	0.167984	0.013166
Exponential Smoothing	0.758449	0.172114	0.012466
NARNN	0.513163	0.084999	0.007875

5. CONCLUSION

In conclusion, the optimal selection of input variables using the stepwise method in the NARNN model can provide good accuracy. Furthermore, combining the NARNN model using the learning method of resilient backpropagation with weight backtracking, the activation function of hyperbolic tangent and the median ensemble operator also provides the best results. The learning method of resilient backpropagation with weight backtracking, different activation functions and ensemble operators are the most effective. Meanwhile, the hyperbolic tangent is the most consistent activation function, while the median is the best ensemble operator in combining these two parameters. This study also shows that the forecasting accuracy of the NARNN model is significantly different from the ARIMA and Exponential Smoothing results introduced by Hyndman & Khandakar (2008). Further research on forecasting time series data using the NARNN model can be focused on finding the best number of neurons in the hidden layer that tends to produce unstable forecasts.

ACKNOWLEDGMENTS

The authors are grateful to the Ministry of Education and Culture Republic of Indonesia for their financial support through the Doctoral Research Grant (Penelitian Disertasi Doktor, PDD) 2020.

REFERENCES

Abraham, B., & Ledolter, J. (2005). *Statistical Methods for Forecasting*. New York: John Wiley and Sons.

Anastasiadis, A. D., Magoulas, G. D., & Vrahatis, M. N. (2005). New Globally Convergent Training Scheme Based on the Resilient Propagation Algorithm. *Neurocomputing*, *64*, 253–270.

Crone, S. F., & Kourentzes, N. (2010). Feature Selection for Time Series Prediction - A Combined Filter and Wrapper Approach for Neural Networks. *Neurocomputing*, *73*(10), 1923–1936.

Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. United States: Prentice-Hall.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The Forecast Package for R. *Journal of Statistical Software*, *27*(3), 1–22.

Kourentzes, N. (2019). *Package nnfor: Time Series Forecasting with Neural Networks*. Retrieved from https://cran.r-project.org/web/packages/nnfor/nnfor.pdf.

Kourentzes, N., Barrow, D. K., & Crone, S. F. (2014). Neural Network Ensemble Operators for Time Series Forecasting. *Expert Systems with Applications*, *41*(9), 4235–4244.

Riedmiller, M., & Braun, H. (1993). A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. *IEEE International Conference on Neural Networks*, *1*, 586–591.

Samarasinghe, S. (2007). *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*. New York: Auerbach Publications.

Sembiring, R. K. (1995). *Analisis Regresi*. Bandung: ITB.

Suhartono, Subanar, & Guritno, S. (2006). Model Selection in Neural Networks by Using Inference of $R^2_{Incremental}$, PCA, and SIC Criteria for Time Series Forecasting. *Journal of Quantitative Methods: Journal Devoted to The Mathematical and Statistical Application in Various Fields*, *2*(1), 41–57.

Walczak, S. (2001). An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks. *Journal of Management Information Systems*, *17*(4), 203–222.

Warner, B., & Misra, M. (1996). Understanding Neural Networks as Statistical Tools. *The American Statistician*, *50*(4), 284–293.

Yeung, D. S., Cloete, I., Shi, D., & Ng, W. W. Y. (1998). *Sensitivity Analysis of Neural Networks*. New York: Springer.

Zhang, G. P. (2003). Time Series Forecasting using A Hybrid ARIMA and Neural Network Model. *Neurocomputing*, *50*, 159–175.