Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 144 (2018) 269–276

Procedia Computer Science

www.elsevier.com/locate/procedia

INNS Conference on Big Data and Deep Learning 2018

Design of Experiment to Optimize the Architecture of Deep Learning for Nonlinear Time Series Forecasting

Suhartono, Novri Suhermi*, Dedy Dwi Prastyo

Department of Statistics, Institut Teknologi Sepuluh Nopember, Jl Raya ITS Sukolilo, Surabaya 60111, Indonesia

Abstract

The neural architecture is very substantial in order to construct a neural network model that produce a minimum error. Several factors among others include the input choice, the number of hidden layers, the series length, and the activation function. In this paper we present a design of experiment in order to optimize the neural network model. We conduct a simulation study by modeling the data generated from a nonlinear time series model, called subset 3 exponential smoothing transition auto-regressive (ESTAR ([3]). We explore a deep learning model, called deep feedforward network and we compare it to the single hidden layer feedforward neural network. Our experiment resulted in that the input choice is the most important factor in order to improve the forecast performance as well as the deep learning model is the promising approach for forecasting task.

Keywords: Deep learning, deep feedforward network, design of experiment, forecasting, time series

1. Introduction

Time series is an observational data that is collected over time with the same time periods, such as in hours, days, weeks, months, and years [11]. Based on the data pattern, time series models are divided into two, namely linear time series model and nonlinear time series model. One very flexible method of forecasting time series data that contains both linear and nonlinear patterns is the neural network. The advantage of using a neural network is that it is not necessary to determine the shape of a particular model because the model is adaptively formed based on the features presented from the data [23].

Neural network adopts the biological neuron workings consisting of neurons as input processing, then the existing input value will be summed by a function of the summing function, and gives output based on the weight. Neural network models are widely applied in various fields of forecasting such as stock prices [10, 8, 14, 12, 6, 7], inflow-outflow [18], electricity consumption [4], and interest rates [19].

* Corresponding author. Tel.: +62-31-5943352 ; fax: +62-31-5922940.
E-mail address: novri.suhermi@gmail.com

1877-0509
10.1016/j.procs.2018.10.528

In neural network it is required an appropriate architecture to get the forecast result that produces a minimum error. The architecture within the neural network includes the number of inputs and what variables are used, the number of hidden layers, the number of neurons in each hidden layer, and the activation function. Therefore, it is necessary to construct a design of experiment in order to determine the best architecture on the neural network. We conduct a simulation study with various combinations on neural network architecture.

Many studies also shown that deep neural network model, also called deep learning model, a neural network with multiple hidden layers, has been widely applied for forecasting task and resulted in great performance with high accuracy [1, 13, 15, 16, 22, 24, 17]. In this paper, we will compare the deep feedforward network to the single hidden layer feedforward network. We also aim to identify which factors that can significantly improve the forecast performance. The experiment results are then discussed.

2. Neural Network Model

2.1. Artificial Neural Network

Artificial neural network is a process information system that has certain performance characteristics in biological neural networks. In this context, neural network is viewed as a mathematical object or specifically as a time series model, with the following assumptions.

1. Information processing occurs in many simple elements called neurons.
2. The signal is passed between the neurons above the connection links.
3. Each connection link has a weight multiplied by the transmitted signal.
4. Each neuron uses an activation function determine the output signal.

A neural network is classified according to the connection pattern between neurons (also called architecture), methods of weighing the connection (called training, learning, or algorithm), and activation functions [5]. There are many types of neural network architecture, including feedforward network, recurrent network, radial basis, and function network. The feedforward neural network (FFNN) is one of the model that is widely used in forecasting.

In FFNN, the process begins with inputs received by neurons, where these neurons are grouped in layers. Information received from the input layer proceeds to the layer in FFNN sequentially to reach the output layer. Different layers between input and output are called hidden layers. The input used in the neural network is the lag of the previous observation and the output is the forecast result. An example of an FFNN model architecture consisting of p input, a hidden layer consisting of m nodes connecting to the output, is shown in Figure 1 [3].

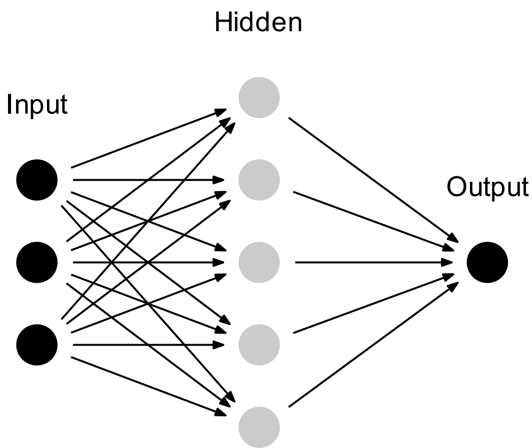

Fig. 1. The example of FFNN architecture

The mathematical expression of the FFNN is defined as follows [20]:

$$f(\mathbf{x}_t, \mathbf{v}, \mathbf{w}) = g_2\left\{ \sum_{j=1}^{m} v_j g_1\left[\sum_{i=1}^{n} w_{ji} x_{it} \right]\right\}, \tag{1}$$

where $\mathbf{w}$ is the weight connecting the input layer to the hidden layer, $\mathbf{v}$ is the weight connecting the hidden layer to the output layer, $g_1(.)$ and $g_2(.)$ is the activation function as well as w_{ji} and v_j the weights that we want to find by training the model. There are three activation functions that are commonly used, namely logistic function, tanh function, and reLU function. The activations functions are given in the following, respectively.

$$g(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

$$g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

$$g(x) = \max(0, x) \tag{4}$$

2.2. Deep Feedforward Network

Deep feedforward network or deep neural network (DNN) is a feedforward neural network model with more than one hidden layer. It is one of the basic deep learning model [9]. The feedforward network aims to approximate some function f^*. A feedforward network finds the best function approximation by learning the value of the parameters θ from a mapping $y = f(x; \theta)$, where the neural architecture is shown in Figure 2. In terms of time series model, the relationship between the output (Y_t) and the inputs ($Y_{t-1}, Y_{t-2}, ..., Y_{t-p}$) in a DNN model with 3 hidden layers is described in the following equation.

$$Y_t = \sum_{i=1}^{s} \alpha_i\, g\left(\sum_{j=1}^{r} \beta_{ij}\, g\left(\sum_{k=1}^{q} \gamma_{jk}\, g\left(\sum_{l=1}^{p} \theta_{kl} Y_{t-l} \right)\right)\right) + \varepsilon_t, \tag{5}$$

where ε_t is the error term, $\alpha_i(i = 1, 2, ..., s)$, $\beta_{ij}(i = 1, 2, ..., s; j = 1, 2, ..., r)$, $\gamma_{jk}(j = 1, 2, ..., r; k = 1, 2, ..., q)$, and $\theta_{kl}(k = 1, 2, ..., q; l = 1, 2, ..., p)$ are the model parameters called the connection weights, p is the number of input nodes, and q, r, s are the number of nodes in the first, second, and third hidden layers, respectively. Function $g(.)$ denotes the hidden layer activation function.

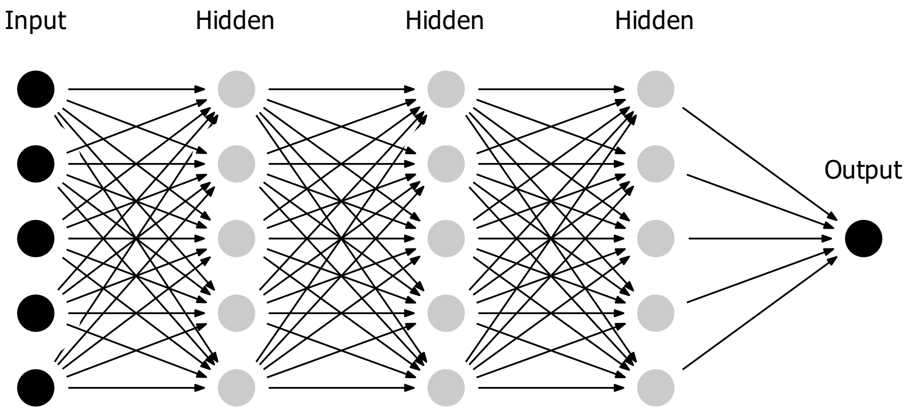

Fig. 2. The Example of DNN architecture

3. Experimental Plan

Our experiment starts with generating the series from a nonlinear time series model, called Exponential Smoothing Transition Auto-regressive (ESTAR) ([3]) which is defined in the following [21].

$$Y_t = 6.5Y_{t-3}\exp(-0.25Y_{t-3}) + a_t \tag{6}$$

We use several factors, namely the input, the number of hidden layer, the number of nodes in hidden layer, the activation function, and the length of series. As the inputs, we use the lag variables of the series, which are lag 1, lag 2, lag 3, lag 4, and lag 5. We use up to three hidden layers in our experiment. The number of nodes in each hidden layer is limited to 5. We use logistic, tanh, and reLU as the activation functions. We also use two different length of series, which are 1000 and 10000. Each model with the combination of several factors is replicated 3 times. In total, we will have 86490 model combinations.

We split the data into two parts, which are training and testing data. The model is trained using the training data. Meanwhile the data used for the prediction is the testing data. For the series with size of 1000, we split the data where 900 data is for training and 100 data is for testing. The series with size of 10000 is splitted in to two parts, where 9900 data is for training and 100 data is for testing. In order to forecast the testing data, we conduct one-step ahead forecast. Then, the forecast accuracy measure that we use is root mean squared error prediction (RMSEP). RMSEP is calculated using the following equation [2].

$$\text{RMSEP} = \sqrt{\frac{1}{L}\sum_{l=1}^{L}\left(Y_{n+l} - \hat{Y}_n(l)\right)^2} \tag{7}$$

where L denotes the size of testing data, Y_{n+l} denotes the l-th actual value of the testing data, and $\hat{Y}_n(l)$ denotes the l-th forecast.

4. Results and Discussion

The first analysis starts with visualization of the series. We present the lag plot to see the pattern of the series with their lags, as shown in Figure 3. We could see that there is a nonlinear pattern on lag 3, as our data is generated from nonlinear model with lag 3, i.e. ESTAR([3]).

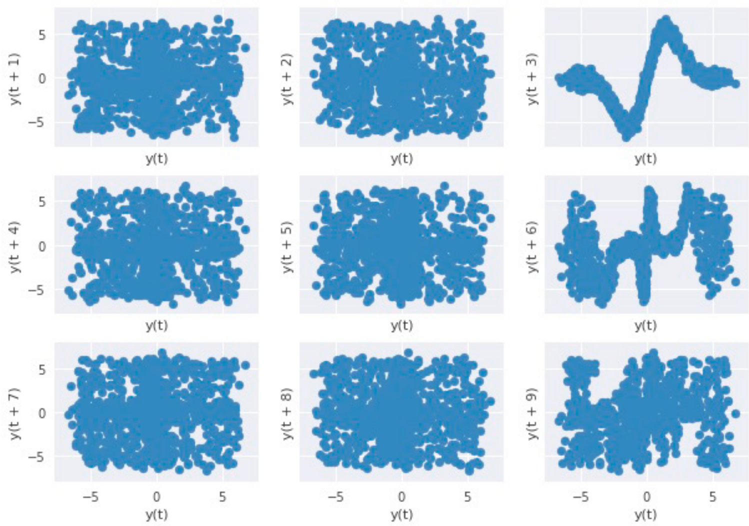

Fig. 3. The lag plot of the simulation data

Then, Figure 4 shows the effect of number of inputs in the neural architecture. We compare RMSEP between the model with lag 3 input and the model without lag 3 input. It can obviously be seen that the increase of input does not significantly improve the forecast performance. In fact, there is a drastic difference of the average of RMSEP between these two types of models. Excluding the lag 3 input affects the increase of RMSEP. It indicates that the input choice plays a very important role in the forecast performance, instead of the number of the input. When we choose the the wrong input, the forecast performance will be bad whatsoever the number of the input.

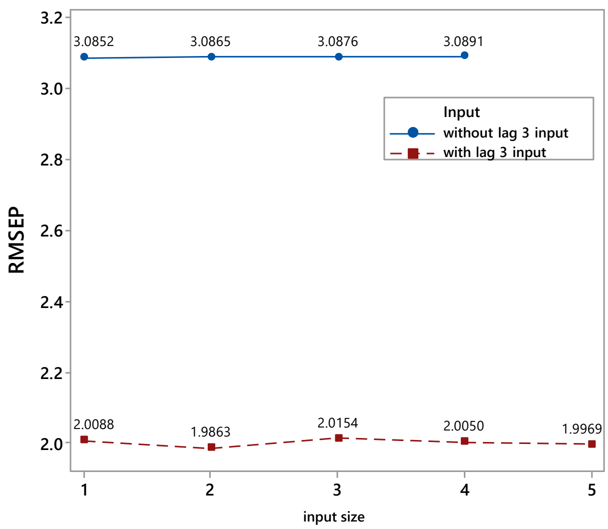

Fig. 4. The effect of the number of input in neural network model

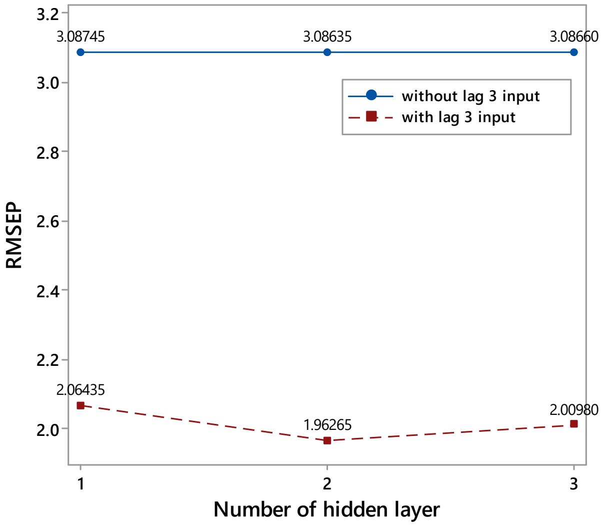

Fig. 5. The effect of the number of hidden layer in neural network model

We continue by showing the effect of the number of hidden layers. Figure 5 shows that the deep feedforward network resulted in a slightly better forecast performance than the single hidden layer model. In our experiment, in average, the optimal number of hidden layer is 2. Compared to the 3 hidden layers model, the RMSEP is getting higher instead. Adding more layer in our architecture does not necessarily improve the forecast accuracy.

In terms of the choosing the activation function, we compare the logistic, tanh, and reLU functions as it is presented in Figure 6. Surprisingly, the model with tanh activation function outperforms among the other models with lag 3 input. But, the activation functions in model without lag 3 input give no significant effect, as the RMSEPs are bad. Our results show that tanh is the optimal activation function in the context of forecasting task.

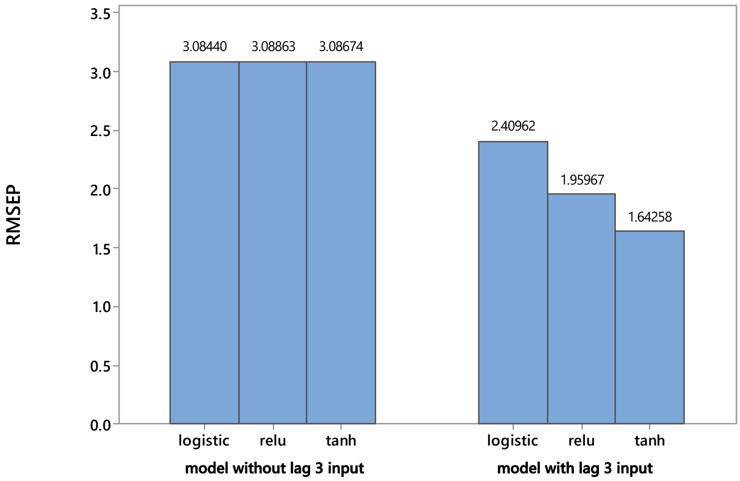

Fig. 6. The effect of the activation function in neural network model

Our next analysis shows the comparison of the series length. We compare two time series of size 1000 and 10000, respectively. We see in the Figure 7 that there is a huge difference of the forecast performance. The longer series shows a significantly forecast accuracy. But, it does not apply to the model without lag 3 input. As we see in our analysis, the models without lag 3 input have no significant factors that affect their forecast capability. We consider that the wrong input model cannot be well optimized by any means.

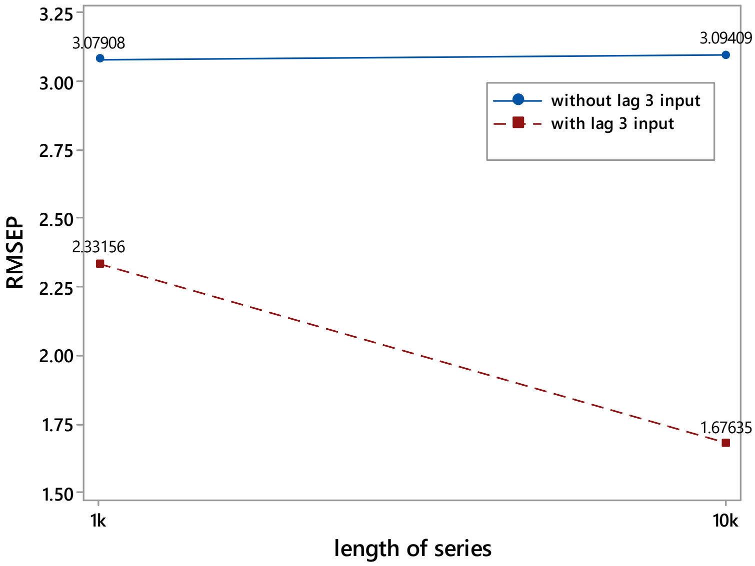

Fig. 7. The effect of series length in neural network model

5. Conclusions and Future Works

Our experiment has shown several discoveries in order to optimize the deep learning model for forecasting. The number of input in our case does not significantly affect the forecast performance. But, there are some factors that can be tuned in order to optimize the model, among other, include the number hidden layer, the choosing of activation function, and the length of series. The deep feedforward network outperforms compared to the single hidden layer network. In the context of forecasting, tanh activation function resulted in a very good accuracy as well as the longer series also increase the performance. Lastly, the most important factor of the forecast performance is the choosing the correct input. We have already seen that the correct input is the main and the first condition that we need to satisfy in order to obtain the best forecast result. Otherwise, the model will fail to give good forecast.

In the future works, we are confident that deep learning is the promising model for forecasting and we suggest to explore more deep learning architecture in order to improve the forecast performance. Some models such as recurrent neural network (RNN) and long short term memory (LSTM) network are highly recommended to explore and optimize for forecasting task.

6. Acknowledgments

This research was supported by DRPM under the scheme of “Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT)” no. 930/PKS/ITS/2018. The authors thank to the General Director of DIKTI for funding and to the referees for the useful suggestions.

References

[1] Chen, Y., He, K., Tso, G.K., 2017. Forecasting crude oil prices: a deep learning based model. Procedia Computer Science 122, 300–307.
[2] De Gooijer, J.G., Hyndman, R.J., 2006. 25 years of time series forecasting. International journal of forecasting 22, 443–473.
[3] El-Telbany, M.E., 2014. What quantile regression neural networks tell us about prediction of drug activities, in: Computer Engineering Conference (ICENCO), 2014 10th International, IEEE. pp. 76–80.
[4] Fahmi, F., Sofyan, H., 2017. Forecasting household electricity consumption in the province of aceh using combination time series model, in: Electrical Engineering and Informatics (ICELTICs), 2017 International Conference on, IEEE. pp. 97–102.
[5] Fausett, L., 1994. Fundamentals of neural networks: architectures, algorithms, and applications. Prentice-Hall, Inc.

[6] Franses, P.H., Van Griensven, K., 1998. Forecasting exchange rates using neural networks for technical trading rules. Studies in Nonlinear Dynamics & Econometrics 2.

[7] Gencay, R., 1999. Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules. Journal of International Economics 47, 91–107.

[8] Gencay, R., Stengos, T., 1998. Moving average rules, volume and the predictability of security returns with feedforward networks. J. Forecast 17, 401–414.

[9] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

[10] Haefke, C., Helmenstein, C., 1995. A neural network model to exploit the econometric properties of austrian ipos, in: Computational Intelligence for Financial Engineering, 1995., Proceedings of the IEEE/IAFE 1995, IEEE. pp. 128–135.

[11] Hyndman, R.J., Athanasopoulos, G., 2014. Forecasting: principles and practice. OTexts.

[12] Kuan, C.M., Liu, T., 1995. Forecasting exchange rates using feedforward and recurrent neural networks. Journal of applied econometrics 10, 347–364.

[13] Liu, L., Chen, R.C., 2017. A novel passenger flow prediction model using deep learning methods. Transportation Research Part C: Emerging Technologies 84, 74–91.

[14] Qi, M., Maddala, G., 1999. Economic factors and the stock market: a new perspective. Journal of Forecasting 18, 151–166.

[15] Qin, M., Li, Z., Du, Z., 2017. Red tide time series forecasting by combining arima and deep belief network. Knowledge-Based Systems 125, 39–52.

[16] Qiu, X., Ren, Y., Suganthan, P.N., Amaratunga, G.A., 2017. Empirical mode decomposition based ensemble deep learning for load demand time series forecasting. Applied Soft Computing 54, 246–255.

[17] Suhartono, Endharta, A.J., 2009. Short term electricity load demand forecasting in indonesia by using double seasonal recurrent neural networks. International Journal of Mathematical Models and Methods in Applied Sciences 3, 171–178.

[18] Suhartono, Saputri, P.D., Amalia, F.F., Prastyo, D.D., Ulama, B.S.S., 2017. Model selection in feedforward neural networks for forecasting inflow and outflow in indonesia, in: Communications in Computer and Information Science, Springer. pp. 95–105.

[19] Swanson, N.R., White, H., 1995. A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks. Journal of Business & Economic Statistics 13, 265–275.

[20] Taylor, J.W., 2000. A quantile regression neural network approach to estimating the conditional density of multiperiod returns. Journal of Forecasting 19, 299–311.

[21] Teräsvirta, T., 1994. Specification, estimation, and evaluation of smooth transition autoregressive models. Journal of the american Statistical association 89, 208–218.

[22] Voyant, C., Notton, G., Kalogirou, S., Nivet, M.L., Paoli, C., Motte, F., Fouilloy, A., 2017. Machine learning methods for solar radiation forecasting: A review. Renewable Energy 105, 569–582.

[23] Zhang, G.P., Berardi, V.L., 1998. An investigation of neural networks in thyroid function diagnosis. Health Care Management Science 1, 29–37.

[24] Zhao, Y., Li, J., Yu, L., 2017. A deep learning ensemble approach for crude oil price forecasting. Energy Economics 66, 9–16.