International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Backpropagation Neural Networks Implementation for JKSE Forecasting

Seng Hansun

Abstract: ***Neural networks is a type of soft computing methods that widely has been used and implemented in many fields, including time series analysis. One of the goals of time series analysis is to predict future data value.In this study, we try to implement another approach using the backpropagation neural networks method to forecast the Jakarta Stock Exchange (JKSE) composite index data, which is one of the stock market change indicators in Indonesia.The study then is continued by calculating the accuracy and robustness levels of Backpropagation NN in forecasting JKSE data. The experimental result on the case taken shows an encouraging and promising result.***

Index Terms: time series analysis, backpropagation, neural networks, JKSE forecasting.

I. INTRODUCTION

A set of regular time-ordered observations of a quantitative characteristic of an individual or collective phenomenon taken at successive periods is known as time-series data [1]. One of the main goals of time series data analysis is to find a pattern that can be used to predict future data or events [2].

Recently, we can find so many works regarding time series analysis, using different kinds of approaches. Some were using the fuzzy time series approach [3-7], while some others were using hybrid approach [8-12], combining two or more methods to forecast the time series data. In this paper, we usea different approach, using backpropagation neural networks to forecast Jakarta Stock Exchange (JKSE) data. By conducting this research, we hope it can give an illustration for the market traders about the condition of stock market, so they can make the best decision on buying or selling their shares. JKSE forecasting results accuracy level will also be calculated by using Mean Square Error (MSE) and Mean Absolute Percentage Error (MAPE).

II. BACKPROPAGATION NEURAL NETWORKS

Artificial neural networks is a network that consists of groups of small processing units, which has been modeled imitating the biological human nervous systems.

Manuscript published on November 30, 2019.
* Correspondence Author
Seng Hansun*, Informatics Department, Universitas Multimedia Nusantara, Tangerang, Indonesia.

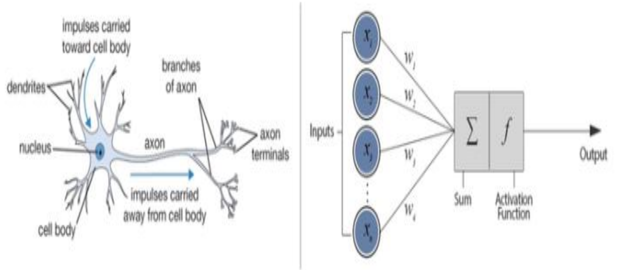

Figure 1. A neuron in human nervous systems (left) and a simple neuron model in artificial neural networks (right) [13]

There are some neural networks techniques that have been developed by researchers, such as delta learning rule, generalized delta learning rule, radial basis function, self-organizing map, polynomial neural networks, backpropagation, counter-propagation,etc. [14]. In this research, we will use backpropagation method to predict future data of JKSE composite index data.

Backpropagation is one type of supervised learning artificial neural networks (ANN) techniques, and consist of three main processes,

1. Feedforward process – data are inputted into the networks,
2. Backpropagation process – calculation and back-propagation of errors found,
3. Updating networks' weights and bias.

Usually, the backpropagation method been used in multi-layer networks, which could be consist of some hidden units, to minimalize errors found in the output of the networks. The steps of backpropagation method can be described as following [14]:

1. Weights and bias initialization.
   Weights and bias can be initialized with any random numbers
2. If the stop condition is not fulfilled yet, do step 3 to 10.
3. For each training data, do steps 4 to 9.

<u>Feedforward</u>

4. Every input unit (X_i, $i = 1, \ldots, n$) receives input signal x_i and distributes the signal to all units in the hidden layer. The input signals x_i been used here are normalized input data training. To normalize them, first find the minimum and maximum value from the input training data. Then normalize them using an activation function. If activation function been used is sigmoid binary, which has minimum value 0 and maximum value 1, then the lowest input value is also 0, and the highest input value is 1. Here we will use sigmoid binary function.

5. On every hidden unit (Z_j, $j = 1, \ldots, p$), sum all the weighted input signals including the bias,
$$z_{in_j} = v_{0j} + \sum^{n} x_i v_{ij} \quad (1)$$
Then, calculate the output signal from every hidden unit using activation function chosen,
$$z_j = f(z\_in_j) \quad (2)$$
The output signal has then been delivered to all output units in output layer.
6. On every output unit (Y_k, $k = 1, \ldots, m$), sum all the weighted input signals including the bias,
$$y\_in_k = w_{0k} + \sum^{p} z_j w_{jk} \quad (3)$$
Then, calculate the output signal from every output unit using the activation function chosen,
$$y_k = f(y\_in_k) \quad (4)$$
The output signal has then been delivered to all units in the networks' output.

<u>Back-propagation of error</u>

7. Every output unit (Y_k, $k = 1, \ldots, m$) receives a target pattern (desired output) that suitable with input training pattern to calculate the error between target and output given by the networks,
$$\delta_k = (t_k - y_k) f'(y_k) \quad (5)$$
δ_k factor will be used to calculate error correction (Δw_{jk}), which will be used to update w_{jk}, where
$$\Delta w_{jk} = \alpha \delta_k z_j \quad (6)$$
The bias correction (Δw_{0k}) also be calculated to update w_{0k}, where
$$\Delta w_{0k} = \alpha \delta_k \quad (7)$$
δ_k factor then is distributed to the layer in step 8.
8. Every hidden unit (Z_j, $j = 1, \ldots, p$) receives a weighted delta input (from step 7),
$$\delta\_in_j = \sum^{m} \delta_k w_{jk} \quad (8)$$
Then the result will be multiplied with derivative of activation function been used by the network to get error correction factor δ_j, where
$$\delta_j = \delta\_in_j f'(z_j) \quad (9)$$
δ_j factor will be used to calculate error correction (Δv_{ij}), which will be used to update v_{ij}, where
$$\Delta v_{ij} = \alpha \delta_j x_i \quad (10)$$
$\Delta v_{0j} = \alpha \delta_j$The bias correction ($\Delta v_{0j}$) also be calculated to update v_{0j}, where
$$\Delta v_{0j} = \alpha \delta_j \quad (11)$$

<u>Weights and bias adjustment</u>

9. Every output unit (Y_k, $k = 1, \ldots, m$) will update the weights and bias from every hidden unit ($j = 0, \ldots, p$)
$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (12)$$
Every hidden unit (Z_j, $j = 1, \ldots, p$) will update the weights and bias from every input unit ($i = 0, \ldots, n$)
$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (13)$$
10. Check the stop condition.

If the stop condition had been fulfilled, then the training process can be stopped. To check the stop condition, we use error limitation as below

$$error = 0.5 \times \{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \cdots + tkm{-}ykm2 \quad (14)$$

After the networks been trained successfully with the training data, we will get weights and bias values that best represent the networks' architecture. The new weights and bias values can be used in testing data to predict time series data desired.

Testing

After the training process, if we give some inputs to the networks, we will get prediction outputs. The processes are similar to feedforward process explained above (steps 4 to 6). However, we need to remember that y_k variable still needs to be converted again into corresponding value since we had normalized it before based on the range of activation function used.

III. RESULTS AND ANALYSIS

In this section, the backpropagation neural networks method explained above will be incorporated to forecast JKSE data. There is a total of 50 data been used, which was taken weekly from August, 13rd 2012 to July, 29th 2013 [15]. From those 50 data, the first 25 data was used as the training data to get the best weights and bias values in the networks, while the last 25 data was used as the testing data. MSE and MAPEare used to calculate and get the forecasted data accuracy level.

The networks' architecture been used in this research is shown in Fig. 2 below. The network is multi-layer networks that consist of an input layer, one hidden layer, and an output layer. The input layer has three input units plus one bias, the hidden layer has one hidden unit plus one bias, and the output layer has one output unit. For the first input unit, we will use training data X_i; for the second input unit, we use training data X_{i+1}; and for the third input unit, we use training data X_{i+2}, where $i = 1, \ldots, n-2$.

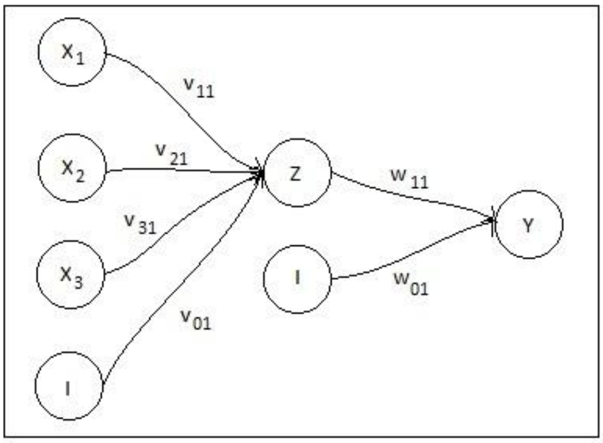

Figure 2. Networks architecture has been used in the research

For the initial values, we use this assumption:

1. Learning rate : 0.1
2. Error tolerance : 0.05
3. v_{01} : -1
4. v_{11} : 1.5
5. v_{21} : 0
6. v_{31} : 2.5
7. w_{01} : -5
8. w_{11} : 7

Then the networks will be trained with the above initial values to get the best weights and bias values within the error tolerance range, as follow:

1. Error tolerance : 0.0482
2. v_{01} : -0.8628
3. v_{11} : 1.5323
4. v_{21} : 0.0384
5. v_{31} : 2.5380
6. w_{01} : -4.9175
7. w_{11} : 7.0209

The weights and bias values above then will be used on testing data. The forecasting results on testing data using backpropagation neural networks method are shown in Table 1. The values of MSE and MAPE are quite small, which means that the backpropagation method can be used to forecast JKSE data quite well.

Table 1. Actual versus Forecasted Data

Date	***Data***	***Forecast***	$(X_t - \hat{X}_t)^2$	$\frac{\lvert X_t - \hat{X}_t \rvert}{X_t}$
3/5/2013	4874.5	4858.48	256.75	0.0033
3/11/2013	4819.32	4896.23	5915.74	0.0160
3/18/2013	4723.16	4911.15	35339.22	0.0398
3/25/2013	4940.99	4890.04	2595.51	0.0103
4/1/2013	4926.07	4945.84	390.75	0.0040
4/8/2013	4937.21	4927.81	88.44	0.0019
4/15/2013	4998.46	4961.33	1378.56	0.0074
4/22/2013	4978.51	4970.74	60.44	0.0016
4/29/2013	4925.48	4968.65	1863.44	0.0088
5/6/2013	5105.94	4965.76	19650.23	0.0275
5/13/2013	5145.68	4990.10	24205.28	0.0302
5/20/2013	5155.09	4991.29	26831.55	0.0318
5/27/2013	5068.63	5002.73	4342.74	0.0130
6/3/2013	4865.32	4997.18	17386.72	0.0271
6/10/2013	4760.74	4971.91	44593.81	0.0444
6/17/2013	4515.37	4939.11	179554.12	0.0938
6/24/2013	4818.9	4779.70	1537.00	0.0081
7/1/2013	4602.81	4898.47	87415.97	0.0642
7/8/2013	4633.11	4687.77	2987.19	0.0118
7/15/2013	4724.41	4831.83	11539.87	0.0227
7/22/2013	4658.87	4810.04	22852.45	0.0324
7/29/2013	4640.78	4781.51	19804.06	0.0303
			MSE = **23208.63**	MAPE = **2.4113**

IV. CONCLUSION

We have tried to applythe backpropagation neural networks method to forecast JKSE composite index data in this research. From the experimental result, encouraging and promising results are gotten. By usingmulti-layer networks architecture, with three input units (plus one bias) and one hidden unit (plus one bias), we could get quite similar forecasted and actual data. The forecasted data accuracy level had also been compared to the actual data using MSE and MAPE whose values are 23208.63 and 2.4113 consecutively.

From the experiment conducted, we could conclude that backpropagation neural networks using multi-layer networks can be incorporated to predict the JKSE data so that the market traders can use it to make the best decision on buying or selling their shares based on the condition of the stock market. Next, we could try to build a new method to decide the best initial weights and bias values so that we can get a better forecasting result.

REFERENCES

1. OECD: Glossary of Statistical Terms, http://stats.oecd.org/glossary/about.asp, accessed on November 2019.
2. B. Render, R.M. Stair Jr., and M.E. Hanna, Quantitative Analysis for Management, 8th ed., New Jersey: Pearson Education, Inc., 2003.
3. M.S. Halim and S. Hansun, "Truck Spare Part Sales and Forecasting System using Fuzzy Time Series Method," *MALTESAS Multi-Disciplinary Research Journal (MIRJO)*, Vol. 3, No. 2, pp. 25-32, 2018.
4. S. Hansun, "Jakarta Stock Exchange (JKSE) Forecasting using Fuzzy Time Series," *Proc. of 2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems (ROBIONETICS)*, 25-27 November 2013, Yogyakarta, Indonesia.
5. D. Kuswoyo, N. Agani, "Model Perhitungan Kebutuhan Bandwidth Jaringan Komputer Menggunakan Sistem Pakar Fuzzy Dengan Metode Adaptive Neuro Fuzzy Inference System(ANFIS) : Studi Kasus PT.GMF Aero Asia Cengkareng," *Jurnal TICOM*, Vol. 3, No. 3, pp. 1-15, 2015.
6. W. Lu, X. Chen, W. Pedrycz, X. Liu, and J. Yang, "Using interval information granules to improve forecasting in fuzzy time series," *International Journal of Approximate Reasoning*, Vol. 57, pp. 1-18, 2015.

7. E. Ketaren and D.H. Sembiring, "Fuzzy Logic dalam Penentuan Prestasi Siswa Sesuai Standar Kompetensi Sekolah," *Publikasi Ilmiah Teknologi Informasi Neumann*, Vol. 2, No. 1, pp. 29-34, 2017.
8. A. Mauko, M. B., and P. Sugiartawan, "Sistem Pendukung Keputusan Kelompok pemilihan Saham LQ45 dengan menggunakan metode AHP, Promethee dan BORDA," *Jurnal Sistem Informasi dan Komputer Terapan Indonesia*, Vol. 1, No. 1, pp. 25-34, 2018.
9. A. Gunaryati, "Analisis dan Model Peramalan Data Ekspor-Impor dengan Metode Gabungan ARIMA-Neural Networks," *Jurnal STRING*, Vol. 2, No. 1, pp. 28-37, 2017.
10. N. Setyawan, N.A. Mardiyah, and K. Hidayat, "Deteksi dan Prediksi Trajektori Objek Bergerak dengan Omni-Vision Menggunakan Pso-Nn dan Interpolasi Polynomial," *Mutlitek Indonesia*, Vol. 13, No. 1, pp. 66-80, 2019.
11. L. Rosalina and B. Susilo, "Integrasi Pemodelan Cellular Automata dan Multilayer Perceptron untuk Prediksi Lahan Pertanian Sawah di Sebagian Kabupaten Sleman," *Jurnal Bumi Indonesia*, Vol. 7, No. 4, 2018.
12. I. Khandelwal, R. Adhikari, and G. Verma, "Time Series Forecasting Using Hybrid ARIMA and ANN Models Based on DWT Decomposition," *Procedia Computer Science*, Vol. 48, pp. 173-179, 2015.
13. K. Willems, Keras Tutorial: Deep Learning in Python, https://www.datacamp.com/community/tutorials/deep-learning-python, accessed on November 2019.
14. K. Setiawan, Paradigma Sistem Cerdas, 3rd ed., Surabaya: Sekolah Tinggi Teknik Surabaya.
15. Yahoo! Finance, http://finance.yahoo.com/q/hp?s=^JKSE&a=07&b=1&c=2012&d=07&e=7&f=2013&g=w, accessed on August, 2013.

AUTHORS PROFILE

Seng Hansunhad finished his Bachelor and Master's degree from Universitas Gadjah Mada, majoring Mathematics and Computer Science program. Since 2011, he has been a lecturer and researcher at Universitas Multimedia Nusantara and published more than 100 papers both nationally and internationally. His research interests mainly in time series analysis and machine learning domain where he has successfully granted some research grants from the government and UMN institution.