# Nonlinear Modeling of IHSG with Artificial Intelligence

1st Mutia Yollanda
*Department of Mathematics*
Andalas University
Padang 25163, Indonesia
mutiayollanda@gmail.com

2nd Dodi Devianto
*Department of Mathematics*
Andalas University
Padang 25163, Indonesia
ddevianto@sci.unand.ac.id

3rd Hazmira Yozza
*Department of Mathematics*
Andalas University
Padang 25163, Indonesia
hazmirayozza@sci.unand.ac.id

***Abstract* — Artificial Intelligence is the simulation of human intelligence processes by computer systems which can be used to model stock prices. Learning algorithms of artificial neural network used to train the network so far the weight of connection inter units can be suitable with error which have determined. The back propagation method is designed as operation of feed-forward network with multiple layers in order that the result of the weights is nonlinear. Nonlinear weights make a nonlinear model in artificial neural network. Time series data of Composite Stock Prices Index (IHSG) is trained using back propagation method in artificial neural network until error which is obtained in weights of the network become very small. The weights is used to model IHSG. Performance rate of time series data model of IHSG which started on January 2016 until December 2017 is measured using Mean Absolute Percentage Error (MAPE). Based on MAPE value of 1.74528596% indicates that the model obtained is very good used to forecast IHSG in the future.**

***Keywords — artificial neural network, back propagation, feed-forward***

## I. Introduction

Most of the researches have goals to measure the relationship inter variables which will affect the final result, in order that a model is obtained. In controlling behavior of the model as in [1], unknown constant values which are needed to predict a model, are called parameters. In the case of forecasting, usually parameter which is expected will make a linear model. In the fact, most of a model which approach actual result is not in linear form. In predicting nonlinear parameter, the least square method is commonly used. One of the procedures which used is gradient descent. In this procedure, there is a change in the initial estimation of the parameters so that the parameter which used is a nonlinear model.

The basic idea of stock is the investment which is needed in running a business. In starting a business which introduced in [2], there are risks that have to be borne by investors in running their business. Using investment, investors also share the risks so that the risks borne by each investor are reduced proportionately. In order to calculate the proportion legally recognized, the approval document sheet is made to strengthen the rights of investors, known as stock sheets. Every share issued by the company has a price. Stock forecasting is very useful for investors to be able to see how a company's stock investment in the future and anticipate the rise and fall in stock prices as a reference to take a decision to buy or sell shares of a company. In forecasting stock prices, there are many things which influence stock prices. They are inflation, interest rate, exchange rate, gold, earning per share, crude oil, and many observable variable. Using historical data, the data of stock prices and their fluctuation can be discovered as described in [3]. Improving neural network and analyzing historical data of stock prices might be predicting the price fluctuation on the stock market.

According [4], there are many techniques which have been expanded to forecast stock prices. One of the techniques is regression methods. According [5], the other technique that can be used for modeling stock price is Autoregressive Fractionally Integrated Moving Average (ARFIMA). Since stock prices data can be classified as time series data, nonlinear machine learning techniques have also been used. Artificial Neural Networks is machine learning algorithms which are most widely used for predicting stock and stock price index movement. Furthermore, risk management as described in [6], GARCH is a time series model that can be used for forecasting volatility of stock prices which is used to compute values at risk in measuring financial risk. Consistency and reliability of the parameter in statistical model can be tested using statistical test such as in [7], likewise for time series and regression model for financial data.

Regarding the forecast of stock price indexes, according [8], comparing intelligent systems to forecast stock exchange index is provided. Among some intelligent systems which had been used, it proved that Multilayer Perceptron presents more reliable results than the other intelligent systems.

An artificial neural network created in 1943 by Warren McCulloch and Walter Pits. According [9], there has been an explosion in the amount of available data which having millions of observations and hundreds or thousands of predictors. Examples include the point of sale data in marketing, credit card scoring data, online monitoring of production processes, optical character recognition, internet e-mail filtering data, microchip array data and computerized medical record data. This exponential growth in available data has motivated researchers in the' fields of statistics, artificial intelligence, and data mining to develop simple, flexible, powerful procedures for data model that can be applied to very large data sets.

The artificial neural network imitates the workings of neural networks. Based on [10], artificial neurons are designed to imitate the characteristics of biological neurons. The properties of artificial neural networks are determined by the pattern of relationships between neurons that are usually called network architecture, weighting methods and activation functions. Artificial neural networks require data of the problem to be solved and then filter the information obtained from the data through training.

In training the network, back propagation algorithm is applied to train the network in a large number of experiments. According [10], many topics that have used the artificial neural network on gradient techniques for training the network, especially some variation of the back propagation algorithm. After training the network, validating the network is needed. According [11], the performance of each neural network was determined by calculating of MAPE and RMSE.

In artificial neural networks, the parameters which used to make a model is the weights that have connections inter units. In learning algorithm of the artificial neural network, the procedure which is used is the gradient descent procedure. Therefore, the model to be formed from artificial neural networks is a nonlinear model.

## II. NONLINEAR MODEL WITH ARTIFICIAL INTELLIGENCE

Artificial intelligence is one of the newest fields in science and engineering. The name was created in 1956 and started after World War II. Artificial intelligence covers a huge variety of subfields such as playing chess, proving mathematical theorems, writing poetry, driving a car on a crowded street, and diagnosing diseases which is explained in [12].

Neural networks are networks of nerve cells in the brains of humans and animals. The human brain has about 100 billion nerve cells. The humans have the intelligence and ability to learn various motors and intellectual capabilities. For many centuries biologists, psychologists and doctors have tried to understand how the brain functions. Around 1900 there was an awareness that the nerve cells and their connections are responsible for awareness, associations, thoughts, consciousness, and the ability to learn which is described in [12]. The signal is accepted by a neuron cell from dendrites. When a signal is accepted by neuron, that neuron may fire. When a neuron fires, a signal is transmitted over the axon. Ultimately, the signal will leave the neuron as it travels to the axon terminals. Then, the signal will be transmitted to other neurons or nerves.

The first big step toward neural networks in artificial intelligence was made 1943 by McCulloch and Pitts in an article entitled “A logical calculus of the ideas immanent in nervous activity.” They were the first to present a mathematical model of the neuron as the basic switching element of the brain. This article laid the foundation for the construction of artificial neural networks as in [10]. A simple mathematical model of the neuron will be shown in the following figure.

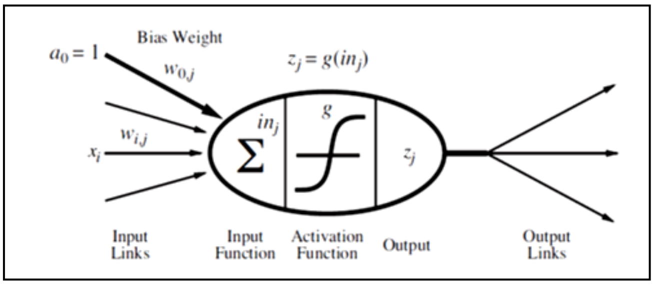


Fig. 1. A simple mathematical model for a neuron

Based on Fig.1, the neural networks are composed of nodes or units connected by directed links. A link from unit $i$ to unit $j$ provides to propagate the activation $x_i$ from i to j. Each link also has weight $w_{ij}$ associated with it, which determines the strength and sign of the connection. Each unit $j$ first computes a weighted sum of its inputs:

$$in_j = \sum_{i=0}^{n} x_i w_{ij} \qquad (1)$$

Then the activation function $g$ is applied to this sum to obtain the output $z_j = g(in_j) = g\left(\sum_{i=0}^{n} x_i w_{ij}\right)$ and continued to other neurons as a new set of input data.

According [13], there are two fundamentally distinct ways to make the network after deciding on the mathematical model of a neuron. They are feed-forward and recurrent network. A feed-forward network has connections only in one direction, that is, it forms a directed acyclic graph. Feed-forward networks are usually arranged in layers, such that each unit receives input only from units in the immediately preceding layer. In single layer networks, every unit connects directly from the network’s inputs to its outputs, while on a multilayer network, which have one or more layers of hidden units that are not connected to the outputs of the network. A recurrent network, on the other hand, feeds its outputs back into its own inputs. This means that the activation levels of the network form a dynamical system that may reach a stable state or exhibit oscillations or even chaotic behavior. Moreover, the response of the network to a given input depends on its initial state, which may depend on previous inputs. Hence, recurrent networks can support short-term memory which makes the networks more interesting as models of the brain, but also more difficult to understand.

Most neural networks pass the output of their layers through activation functions. This function can be delivered from the mathematical transformation. There exists Fourier-Stieltjes transform to characterize property of probability distributions such as explained in [14], [15], and [16] for certain distributions. However, activation function may characterize by using this transform to be a unique form. These activation functions scale the output of the neural network into proper ranges. There are three activation functions provided which is explained in [12]: sigmoid, hyperbolic tangent, and linear. A sigmoid activation function uses the sigmoid function to determine its activation. The sigmoid function is defined as follows:

$$f(x) = \frac{1}{1+e^{-x}} \qquad (2)$$

The result of sigmoid function is a value between 0 and 1. Sigmoid function only returns positive values. If the neural network needed to return negative numbers, the sigmoid function will be unsuitable. The activation function that can be used to solve that problem is a hyperbolic tangent function. The hyperbolic tangent function is defined as follows:

$$f(x) = \frac{e^{2x}-1}{e^{2x}+1} \qquad (3)$$

The hyperbolic tangent activation function returns both positive and negative values. The linear activation function is essentially no activation function at all. It is probably the least commonly used of the activation functions. The linear activation function does not modify a pattern before outputting it. The linear function is defined as follows:

$$f(x) = x \qquad (4)$$

The linear activation function might be useful in situations when the entire range of numbers to be output needed. Because the hyperbolic tangent and sigmoid activation functions both have established upper and lower bounds, they tend to be used more for Boolean type operations. The linear activation function is useful for presenting a range.

According [12], there are two steps to construct a model of the artificial neural network. They are training and validation neural network. In training neural network, a neuron that makes a network is interconnected through their synapses. These connections allow the neurons to signal each other as information is processed. Not all connections are equal. Each connection is assigned a connection weight. If there is no connection between two neurons, then their connection weight is zero. These weights determine the output of the neural network. Therefore, it can be said that the connection weights from the memory of the neural network. There are many ways to train neural networks. Neural network training methods generally fall into the categories of supervised, unsupervised, and various hybrid approaches.

Supervised training is accomplished by giving the neural network a set of sample data along with the anticipated outputs from each of these samples. Supervised training is the most common form of neural network training. As supervised training proceeds, the neural network is taken through a number of iterations, or epochs, until the output of the neural network matches the anticipated output, with a reasonably small rate of error. Each epoch is one pass through the training samples. Unsupervised training is similar to supervised training, except that no anticipated outputs are provided. Unsupervised training usually occurs when the neural network is being used to classify inputs into several groups. The training involves many epochs, just as in supervised training. There are several hybrid methods that combine aspects of both supervised and unsupervised training. One such method is called reinforcement training. In this method, a neural network is provided with sample data that does not contain anticipated outputs, as is done with unsupervised training. However, for each output, the neural network is told whether the output was right or wrong given the input.

The final step, validating a neural network is very important because it is possible to determine if additional training is required. The data which are used for validating the networks have to be set aside that is completely separate from the training data. Training a neural network with a given sample set and using this same set might be used to predict the anticipated error the network for a new arbitrary set will surely lead to bad results. The error achieved using the training set will almost always be substantially lower than the error on a new set of sample data. The integrity of the validation data must always be maintained. If the validation is performing poorly, it is likely that there was data present in the validation set that was not available in the training data. The way this situation should be rectified is to try a different random approach to separating the data into training and validation sets. If this fails, the training and validation sets have to be combined into one large training set. New data must then be acquired to serve as the validation data.

According [17], error is differences between the actual and the desired output. Gradient descent procedures calculate the gradient of an arbitrary but finite-dimensional function $Err(W)$ and move down against the direction of the gradient until a minimum is reached. The change in all weights is denoted as $\Delta W$ and change the gradient of error function $Err(W)$ is denoted to as $\nabla Err(W)$. The relation between change of every single weight and error function can be expressed in the following equation.

$$\Delta w_{jk} = -\eta \frac{\partial Err(W)}{\partial w_{jk}} \tag{5}$$

where $w_{jk}$ is single weight between the $j$th hidden layer and the $\underline{k}$th output layer.

During training neural network as in [17], the networks try to minimize the error. Error for $k$th training data can be expressed in the following equation.

$$Err_k(W) = \frac{1}{2}\sum_{j=1}^{p}(t_{kj} - y_{kj})^2 \tag{6}$$

where $t_{kj}$ is $j$th output value which want to be achieved or target value for $k$th data and $y_{kj}$ is $j$th actual output for $k$th data.

According [4], a set of input data $x_i, i = 0,1,\ldots,n$ are applied to the input layer of the network. The input units distribute the values to the hidden layer. The net input to the $p$th hidden unit is

$$in_{kp}^{h} = \sum_{i=1}^{N} w_{pi}^{h} x_{ki} + w_{p0}^{h} \tag{7}$$

where $w_{pi}^{h}$ is the weight on the connection from the $i$th input unit, and $w_{p0}^{h}$ is the bias. The "h" refers to quantities on the hidden layer. Assume that the activation of this node $f_p^h$ is equal to the net input. The output of this node is

$$i_{kp} = f_p^h(in_{kp}^h) \tag{8}$$

The equations for the output nodes are

$$in_{kj}^{o} = \sum_{p=1}^{L} w_{jp}^{o} i_{kp} + w_{j0}^{o} \tag{9}$$

$$y_{kj} = f_j^o(in_{kj}^o) \tag{10}$$

where "o" refers to quantities on output layer.

Updating output layer weights can apply the chain rule to factorize the derivative $\frac{\partial\, Err_k(W)}{\partial\, w_{jp}^{o}}$ which is explained in [18] .The result can be expressed in the following equations

$$\frac{\partial Err_k(W)}{\partial w_{jp}^{o}} = -(t_{kj} - y_{kj}) f_j^{o'}(in_{kj}^{o}) i_{kp}, \tag{11}$$

while updating output layer weights can apply the chain rule to factorize the derivative $\frac{\partial\, Err_k(W)}{\partial\, w_{pi}^{h}}$ which is explained in [17]. The result might be expressed in the following equations.

$$\frac{\partial Err_k(W)}{\partial w_{pi}^{h}} = -\sum_{j=1}^{Q}(t_{kj} - y_{kj}) f_j^{o'}(in_{kj}^{o}) f_p^{h'}(in_{kp}^{h}) x_{ki} \tag{12}$$

where $Err_{k(W)}$ is error function which maps the values of weight W onto the normalized error as output error. The weights which have updated can be used in the next steps if stopping criterion is not available.

Learning data a neural network uses a specific learning algorithm. There are many learning algorithms, but in general they all train the network by iteratively modifying the connection weights until the error between the output produced by the network and the desired output fall below a pre-specified threshold. The back propagation algorithm was the first learning algorithm and is still widely used. It uses gradient descent as the core learning mechanism. Starting from random weights the back propagation algorithm calculates the network weights making small changes and gradually making adjustments determined by the error between the result produced by the network and the desired outcome.

Error propagation is applied by the algorithm from outputs to inputs and gradually fine tunes the network weights to minimize the sum of error using the gradient descent technique as in [19]. The learning network consists of the following steps. First, initialize the network. The initial values of the weights need to be determined. A neural network is generally initialized with random weights. Second, feed-forward. Information is passed forward through the network from input to hidden and output layer via node activation functions and weights. Third, assess the error. The output of the network is assessed relative to known output. If the error is below a pre-specified threshold the network is trained and the algorithm terminated. Fourth, propagate. The error at the output layers is used to re-modify the weights. The algorithm propagates the error backwards through the network and computes the gradient of the change in error with respect to changes in the weight values. Fifth, adjust. Make adjustments to the weights using the gradients of change with the goal of reducing the error. The weights and biases of each neuron are adjusted by a factor based on the derivative of the activation function, the differences between the network output and the actual target outcome and the neuron outputs. Through this process the network will learn. If the partial derivative is negative, the weight is increased, if the partial derivative is positive, the weight is decreased. Each cycle through this process is called an epoch.

There are a number of measures of accuracy in the forecasting literature. The accuracy of the model can be measured using the following two measures which are the most frequently used. First, Mean Square Error (MSE). Mean Square Error (MSE) is a measure used to measure the average error of a model. MSE can be formulated as follows:

$$MSE = \frac{SSE}{N} = \frac{\sum_{i=1}^{N}(t_i - y_i)^2}{N} \tag{13}$$

Second, Mean Absolute Percentage Error (MAPE) is a measure used to measure the accuracy of the model using certain forecasting methods in a given data set. Mean Absolute Percentage Error (MAPE) can be formulated as follows:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|t_i - y_i|^2}{t_i} \times 100\% \tag{14}$$

where SSE is sum square error), N is the number of observational data, $y_i$ is the observed or actual output data, and $t_i$ is the output data using a particular method as in [20].

## III. IHSG MODEL WITH ARTIFICIAL INTELLIGENCE

In determining a nonlinear model of the artificial neural network with back propagation method, it consists of the following steps.

The first step is to define the input and output data which want to be obtained. Input data which will be used are BI rate, inflation, crude oil, gold, and exchange rate. Target data which want to be obtained is Composite Stock Price Index (IHSG). Data which used is historical data from January 2016 until December 2017 that is 462 data. The data were accessed formally via https://yahoofinance.com and http://www.bi.go.id.

The second step is normalization of data. Activation function which used in artificial neural network is sigmoid function, thus all of sample data have to transform into a value between 0 and 1.

The third step is to construct the architecture of artificial neural network. Based on the data, the architecture of artificial neural network, which used is 5-3-1, it means that the network consists of five input units, three hidden units, and one output unit. The architecture of artificial neural network will be shown in Fig.2.

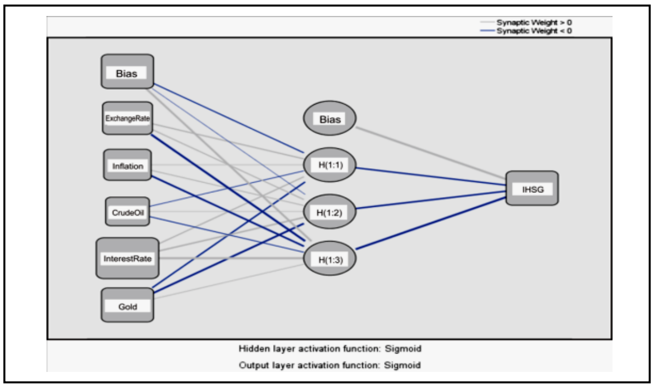


Fig.2. The architecture of artificial neural network

The fourth step is training neural networks. Learning algorithms of the artificial neural network consist of initialization, feed-forward, error assessment, propagation, and adjustment.

1. Initialization of the network.

A neural network is generally initialized with random weights. Based on Fig.1, initial values of the weights are

$w_{10}^h = -0.5, w_{11}^h = 0.5, w_{12}^h = 0.5, w_{13}^h = -0.5,$
$w_{14}^h = 0.5, w_{15}^h = -0.5, w_{20}^h = -0.5, w_{21}^h = 0.5,$
$w_{22}^h = 0.5, w_{23}^h = 0.5, w_{24}^h = 0.5, w_{25}^h = -0.5$ ,
$w_{30}^h = 0.5, w_{31}^h = -0.5, w_{32}^h = -0.5, w_{33}^h = -0.5,$
$w_{34}^h = 0.5, w_{35}^h = 0.5, w_{10}^o = 0.5, w_{11}^o = -0.5,$
$w_{12}^o = -0.5,$ and $w_{13}^o = -0.5.$

In the training process, the network has 378 data, learning rate ($\eta$) is 0.01, activation function which used is sigmoid function, and error assessment of 0.001 as a threshold value so far algorithm can be stopped.

2. Feed-forward.

Each input unit ($x_{ki}, i = 1,2, \ldots ,5$) accepts input data and distribute the data into each hidden unit. Based on the first data which is taken on January 4th, 2016 which had been normalized into a value between 0 and 1, input values that are obtained is:

$x_{11} = 0.95369458$

$x_{12} = 0.81325301$

$x_{13} = 0.23954860$

$x_{14} = 1.00000000$

$x_{15} = 0.00413081$

and output value which want to be achieved or target value is $t_{11} = 0.05757991$. Feed-forward consists of the following steps.

First, each hidden units sum each input data which connected by their weights. Evaluate $in_{kp}^h = w_{po}^h + \sum_{i=1}^{5} x_{ki} w_{pi}^h$ for $k, j = 1, p = 1,2,3$ and the results which obtained are: $in_{11}^h = 0.76163409, in_{12}^h = 1.00118269$, and $in_{13}^h = -0.00118269$. Then, get hidden unit values by evaluating $i_{kp} = f_p^h(in_{kp}^h)$ for $k = 1, j = 1, p = 1,2,3$, and $f_p^h$ is sigmoid function between hidden and input layer and obtained the results: $i_{11} = 0.68170841, i_{12} = 0.73129105$, and $i_{13} = 0.49970433$.

Second, output unit sums each hidden unit data which connected by their weights by evaluating $in_{11}^o = w_{10}^o + \sum_{p=1}^{3} i_{1p} w_{1p}^o$ for $k, j = 1, p = 1,2,3$ and the result which obtained is $i_{11}^o = -0.45635190$. The output unit value of evaluating $y_{kj} = f_j^o(in_{kj}^o)$ for $k, j = 1$ and $f_j^o$ is sigmoid function between output and hidden layer and result which obtained is $y_{11} = 0.38785161$.

3. Error assessment.

If $|t_{k1} - o_{k1}| > 0.001$ for $k$=1th data, stopping criterion is not available, thus the training of the network has not finished. Because error between actual and predicted outputs is 0.33027170 which is greater than 0.001, thus the training process has to be continued.

4. Propagation.

The back propagation method consists of the following steps. First, to find out the error information on the output unit, calculate $\delta_{kj}^o = (t_{kj} - y_{kj}) f_j^{o'}(in_{kj}^o)$ for $k, j = 1, p = 1,2,3$. Second, to find out error on each hidden unit, calculate $\delta_{kj}^h = f_p^{h'}(in_{kj}^h) \delta_{kj}^o w_{jp}^o$ for $k, j = 1, p = 1,2,3$.

In hidden layer, calculate $\Delta w_{jp}^o = \eta \delta_{kj}^o i_{kp}$ for $j = 1, k = 1$ and $p = 1,2,3$ to find out the correction value of weights and bias. While in the input layer, calculate $\Delta w_{pi}^h = \eta \delta_{kp}^h x_{ki}$ for k=1, i=1,2,...,5 and $x_{k0} = 1$ to find out the correction value of weights and bias.

5. Adjustment.

The last step in learning algorithms of artificial neural network, these are two steps to adjust the weights. I n output layer into hidden layer, calculate $w_{jp}^o(t+1) = w_{jp}^o(t) + \Delta w_{pi}^o$ for $j = 1, p = 1,2,3$. Meanwhile in the input layer, calculate $w_{pi}^h(t+1) = w_{pi}^h(t) + \Delta w_{pi}^h$ for $j = 1, p = 1,2,3$, where $w_{jp}^o(t+1)$ and $w_{pi}^h(t+1)$ are new weights which will be used in the next process.

Before doing the next step, repeat (b),(c), and (d) until the error between actual output or target and predicted output is less than a value that has determined.

The fifth step is validation neural network. In the validation process, data which used to test a network are 84 data. The following table is a model summary of training and testing process.

TABLE I. MODEL SUMMARY

| Model Summary | | |
|---|---|---|
| ***Training*** | Sum of Squares Error | 0.754 |
| | Relative Error | 0.075 |
| | Stopping Rule Used | 1 consecutive step(s) with no decrease in error |
| | Training Time | 0:00:00:03 |
| | MSE | 0.00179762 |
| ***Testing*** | Sum of Squares Error | 0.151 |
| | Relative Error | 0.063 |
| | MSE | 0.00199471 |

Dependent Variable : IHSG
a. Error computations are based on the testing sample.

Based on Table 1, the MSE value is obtained by using equation (13) for the testing process, 0.00179762 is less than training process, 0.00199471. It means that the artificial neural network has the ability to recognize new data. Finally, a model which obtained can be used to forecast composite stock price index value.

The last step is modeling of the artificial neural network. After learning algorithms have finished, obtained new weights which will be used to make a nonlinear model of IHSG. Parameter estimates are shown in Table 2.

TABLE II. IBM SPSS OUPUT TO ESTIMATE PARAMETER

| Parameter Estimates | | | | | |
|---|---|---|---|---|---|
| Predictor | | Predicted | | | |
| | | Hidden Layer 1 | | | Output Layer |
| | | H(1:1) | H(1:2) | H(1:3) | IHSG |
| Input Layer | (Bias) | -1.028 | -0.059 | 3.022 | |
| | Exchange Rate | 1.351 | 1.244 | -3.095 | |
| | Inflation | 0.209 | 0.875 | -2.004 | |
| | Crude Oil | -0.267 | 0.095 | -0.409 | |
| | Interest Rate | 1.340 | 1.755 | 4.372 | |
| | Gold | -1.687 | -2.601 | 0.320 | |
| Hidden Layer 1 | (Bias) | | | | 4.789 |
| | H(1:1) | | | | -1.752 |
| | H(1:2) | | | | -1.917 |
| | H(1:3) | | | | -4.291 |

Based on Table 2, IHSG model of artificial intelligence with back propagation method can be written to be:

$$y_{k1} = \frac{1}{1+e^{-(4.789-1.752H(1:1)-1.917H(1:2)-4.291H(1:3))}} \quad (15)$$

where

$$H(1:1) = \frac{1}{1+e^{-h_{11}}} \quad (16)$$

$$H(1:2) = \frac{1}{1+e^{-h_{12}}} \quad (17)$$

$$H(1:3) = \frac{1}{1+e^{-h_{13}}} \quad (18)$$

and we define $h_{11}, h_{12}, h_{13}$ respectively as follows

$$\begin{aligned} h_{11} = &-1.028 + 1.351\,x_{k1} + 0.209\,x_{k2} \\ &- 0.267\,x_{k3} + 1.340\,x_{k4} - 1.687\,x_{k5} \end{aligned} \quad (19)$$

$$\begin{aligned} h_{12} = &-0.059 + 1.244\,x_{k1} + 0.875\,x_{k2} \\ &+ 0.095\,x_{k3} + 1.755\,x_{k4} - 2.601\,x_{k5} \end{aligned} \quad (20)$$

$$\begin{aligned} h_{13} = &\ 3.022 - 3.095\,x_{k1} - 2.004\,x_{k2} \\ &- 0.409\,x_{k3} + 4.372\,x_{k4} + 0.320\,x_{k5} \end{aligned} \quad (21)$$

$y_{k1}$ is*k*th actual output which has been normalized, $x_{k1}, x_{k2}, x_{k3}, x_{k4}, x_{k5}$ are *k*th input data which have been normalized.

Based on the data, after forecasting IHSG value, the MAPE value of the network is 1.74528596%. This value is less than 10%. It means that the model that have been obtained having a good performance to forecast composite stock price index in the future. Furthermore, MAPE value of 1.74528596% states that the forecasting value of the composite stock price index in the future is almost approaching the real value. Moreover, many disadvantages could be avoided. From forecasting composite stock price index, we know that how the fluctuation of stock prices is. In other side, the investors can also determine when they can make investment so far the risks might be reduced from now.

The time series neural network is modeling for time series data using artificial neural network method which can be used for forecasting the data in the future. Because the composite stock price index data can be classified as time series data, it needs to be analyzed statistically using time series modeling. However, to get more precise results, it can be combined with neural networks.

## IV. CONCLUSION

The artificial neural network with back propagation method is a method which used to solve a problem related to a set of data based on the workings of the human brain by making randomly assigned weight changes so that the error is less than a value which have determined.

If the training process uses very small value on error assessment to be stopping rule, iteration that will be obtained from training process become larger.

The artificial neural network has accuration of 98.25471404%. It means that the method which used in forecasting composite stock price index (IHSG) has high performance in order that the model which is obtained very well using to forecast composite stock price index (IHSG) in the future.

## REFERENCES


[1] J. O. Rawlings, S. G Pantula and D. A Dickey, Applied Regression Analysis. USA: Springer, 2001.

[2] E. Tandelilin, Portofolio and Investation. Yogyakarta: Kanisius, 2010.

[3] F. Liu and J. Wang, “Fluctuation prediction of stock market index by Legendre neural network with random time strength function” in Neurocomputing, 2012, pp.12–21.

[4] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques” in Expert Systems with Applications, 2015, pp. 259–268.

[5] D. Devianto, Maiyastri and S. Damayanti, “Forecasting Long Memory Time Series for Stock Price with Autoregressive Fractionally Integrated Moving Average” in International Journal of Applied Mathematics and Statistics, 2015, Vol. 53, pp. 86-95.

[6] D. Devianto, Maiyastri, and D. R. Fadhilla, “Time Series Modeling for Risk of Stock Price with Value at Risk Computation” in Applied Mathematical Sciences, 2015, pp. 2779 – 2787.

[7] F. Yanuar, D. Devianto, S. Marisa and A. Zetra, “Consistency Test of Reliability Index in SEM Model,” Applied Mathematical Sciences, Vol. 9 (106), pp. 5283-5292, 2015.

[8] E. Guresen, G. Kayakutlu, and T.U. Daim, “Using artificial neural network models in stock market index prediction” in Expert Systems with Applications, 2011, pp. 10389–10397.

[9] J. Neter, W. Wasserman, and M.H. Kutner, Applied Linear Statistical Models. Singapore :The McGraw-Hill Companies, Inc, 2004.

[10] B. K. Wong, T. A. Bodnovich, and Y. Selvi, “A bibliography of neural network business applications research: 1988–September 1994” in Expert Systems with Applications, 1995, pp. 253–262.

[11] L. A. Laboissiere, R. A. S. Fernandes, and G. G. Lage, “Maximum and minimum stock price forecasting of Brazilian power distribution companies based on artificial neural networks” in Applied Soft Computing, 2015, pp. 66–74.

[12] J. Heaton, Introduction to Neural Networks for C. America: Heaton Research, Inc, 2008.

[13] S. Russel and P. Norvig, Artificial Intelligence: A Modern Approach. New Jersey: Pearson Education, 2010.

[14] D. Devianto, “The Uniform Continuity of Characteristic Function from Convoluted Exponential Distribution with Stabilizer Constant,” AIP Conference Proceedings 1707, American Institute of Physics, New York, pp. 1-5, 2016.

[15] D. Devianto, Maiyastri, L. Oktasari and M. Anas, “Convolution of Generated Random Variable from Exponential Distribution with Stabilizer Constant,” Applied Mathematical Sciences, Vol. 9 (96), pp. 4781-4789, 2015.

[16] D. Devianto, L. Oktasari and Maiyastri, “Some Properties of Hypoexponential Distribution with Stabilizer Constant,” Applied Mathematical Science, Vol.9 (142), pp. 7063-7070, 2015.

[17] J. L. McClelland and D.E. Rumelhart, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge: MIT Press, 1986.

[18] J. A. Freeman,J.A. and D.M. Skapura, Neural Network: Algoritms, Applications, and Programming Techniques. America: Addison-Wisley Publishing Company, 2004.

[19] N. D. Lewis, Build Your Own Neural Network Today !: With Step by Step Instructions Showing You How to Build Them Faster than You Imagined Possible Using R. Create Space Independent Publishing Platform. ISBN : 978-1519101235, 2015.

[20] S. Makridakis, S. C. Wheelwright and V. E. McGEE, Forecasting: Methods and Applications. New York : Wiley, 1983.