P-ISSN: 2528-0260 E-ISSN: 2579-5392
p.675-686

FORECASTING OMSET PRINTING OF PRINTING SALES IN CV SEMBILAN JAYA WITH NEURAL NETWORK METHOD

[1] RENI VIVIT AYU MAWARTI, [2]WIWIET HERULAMBANG, [3]R DIMAS ADITYO

[1]Informatics Engineering Study Program, Faculty of Engineering

Bhayangkara University – Surabaya

e-mail: renivivit6@gmail.com, wiwiet@ubhara.ac.id, dimas@ubhara.ac.id

ABSTRACT

Forecasting is a process for estimating several needs in the future which includes needs in order to meet the demand for goods and services. Neural Network Backpropagation Method is a time series forecasting method. The purpose of this study is to predict the turnover results in the next period obtained by CV. Nine Jaya every week. This study uses sales data obtained from the printing of food boxes, shoe boxes, watch boxes from January 2014 to December 2018. The results of this forecasting are done using the Neural Network method, the smallest MSE value obtained is 0.004211 with 1000 times iteration and learning rate 0.2. The MSE value obtained meets the condition or condition value as a good forecasting method because it is able to meet the MSE value requirement <0.1.

Keywords: *Forecasting, Printing Turnover, Artificial Neural Networks, Backpropagation Algorithms.*

1. INTRODUCTION

CV Sembilan Jaya is a company engaged in printing. Products produced at CV Sembilan Jaya are food boxes, shoe boxes and watch boxes. However, this company still has problems in recording the sales of its sales which are done manually in the form of sales records written in the books so that the data is often damaged or lost. The company needs to predict what will happen in the future, the aim is to see a picture of the company's sales whether it has increased or decreased. The research of forecasting sales turnover is carried out using the Neural Network (NN) method is an attempt to imitate the structure / architecture and the workings of the human brain so that it can replace some human work. Jobs such as recognizing patterns, predictions, classifications, function approaches, optimization [1]. Based on the description above, this study aims to forecast the turnover of printing sales at CV Sembilan Jaya by the Neural Network (NN) method. This forecasting can be used to facilitate the prediction of sales levels in the next marketing process at CV Sembilan Jaya and it is hoped that the problems encountered in the marketing process can be resolved in minimizing the level of error in the prediction system that already exists.

2. BASIC THEORY

2.1 Artificial Neural Network

The neural network is one of the artificial representations of the human brain that always tries to simulate the learning process in the human brain. The term artificial here is used because this neural network is implemented using a computer program that is able to complete a number of calculation processes during the learning process [2].

P-ISSN: 2528-0260 E-ISSN: 2579-5392
p.675-686

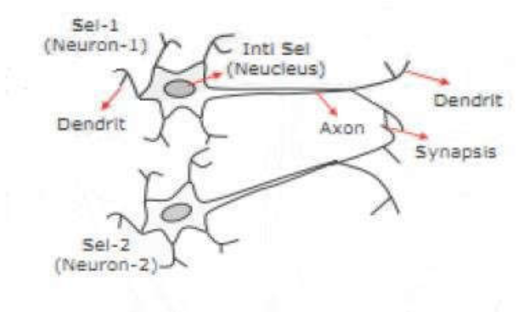

Figure 2.1 Nerve Cells as a whole

2.2 Backpropagation Neural Network

Backpropagation is a learning algorithm that is inherited and is usually used by perceptron with many layers to change the weights associated with neurons in the hidden layer. The backpropogation algorithm uses error output to change its weight values in the backward direction. Forward propagation stage (forward propagation) must be done first to get the error value. When forward propagation of neurons is activated using the binary sigmoid activation function:

$$F(x) = \frac{1}{1 + e^{-x}}$$

Backpropagation neural network architecture as shown in the image below:

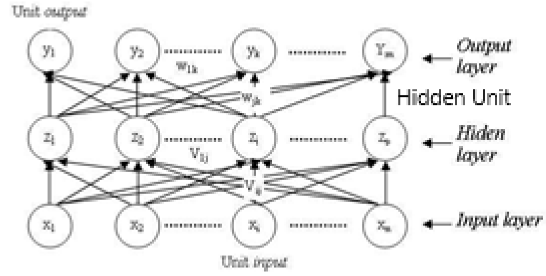

Figure 2.2 Backpropagation architecture

According to Kusumadewi and Hartati (2010), the backpropagation algorithm is as follows:

- Weight initialization.
- Set: Maximum epoch, target error, and learning rate (α), hidden neurons;
- Initialization: Epoch = 0.
- Perform the following steps during (Epoch <maximum Epoch) or (MSE (error) <target error):
  1. **Epoch = epoch +1**
  2. **The training algorithm for a network with a hidden screen (with a binary sigmoid activation function) is as follows:**
     Step 0: Initialize all weights with small random numbers
     Step 1: If the termination conditions have not been met, do it rare 2-9
     Step 2: For each pair of training data, do steps 3-8
     Phase I: Forward Propogation
     Step 3: Each input unit receives a signal and passes it to the hidden unit above it
     Step 4: Calculate all outputs in hidden units zj (j = 1, 2, ..., p)

$$z\_net_j = v_{jo} + \sum_{i=1}^{n} x_i v_{ji}$$

$$z_j = f(z_{net\ j}) = \frac{1}{1 + e^{-z_{net\ j}}}$$

Step 5: Calculate all network outputs in unit yk (k = 1, 2, ..., m)

$$y_{net_k} = w_{ko} + \sum_{j=1}^{p} z_j w_{kj}$$

$$y_k = f(y_{net\,k}) = \frac{1}{1 + e^{-y_{net\,k}}}$$

Phase II: Backward propagation

Step 6: Calculate the factor keluaran output unit based on the error in each output unit yk (k = 1, 2, ..., m)

$$\delta_k = (t_k - y_k) f'(y\_net_k) = (t_k - y_k) y_k (1 - y_k)$$

δ_k is the unit of error that will be used in changing the screen weights below (step 7)

Calculate the weight change w_{kj} (which will be used later to change the weight w_{kj}) with the accelerating rate α

$$\Delta w_{kj} = \alpha \delta_k z_j \, ; k = 1,2, \ldots, m \, ; j = 0,1, \ldots, p$$

Step 7: Calculate the factor tersembunyi hidden units based on the error in each hidden unit zj (j = 1, 2, ..., p)

$$\delta\_net_j = \sum_{k=1}^{m} \delta_k w_{kj}$$

Calculate the weight change term vji (which will be used later to change the weight vj)

$$\Delta w_{ji} = \alpha \delta_j x_i \, ; j = 1,2, \ldots, p \, ; i = 0,1, \ldots, n$$

Phase III: Change in weights

Step 8: Calculate all weight changes

Changes in line weights leading to the output unit:

$$w_{kj}(baru) = w_{kj}(lama) + \Delta w_{kj}(k = 1,2, \ldots, m \, ; j = 0, 1, \ldots$$

Changes in line weights leading to hidden units:

$$v_{ji}(baru) = v_{ji}(lama) + \Delta v_{ji}(j = 1,2, \ldots, p \, ; i = 0, 1, \ldots$$

3. **Calculate MSE (Mean Square Error)**

According to Hansun. (2013) MSE criteria state the magnitude of the mean square error of a forecasting method with the calculation formula:

$$MSE = \sum \frac{(t - y_k)^2}{n}$$

Information:

t = target output value
yk = network output value
n = amount of data

4. **Normalization**

The normalization method used is min-max normalization, this method changes the data to another new range that is between 0 to 1.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Information :

x' = normalization result
x = original data
$xmax$ = maximum value of all original data
$xmin$ = minimum value of all original data

5. **Selection Of Initial Weight And Bias**

The steps of the Ngunyen-Widrow initialization are as follows:

1. Initialize all initial weights v_{ji} long with random numbers in the interval [-0.5, 0.5]

$$\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + v_{jn}^2}$$

2. Weight used for initialization

$$v_{ji} = \frac{\beta v_{ji}(old)}{\|v_j\|}$$

6. Denormalization

According to Hidayat et al (2012) denormalisasai can provide or return data, so we get predicted sales from training data.

According to Indra (2014), the denormalization formula in the range [0,1] is:

$$Xi = y\ (Xmax - Xmin) + Xmin$$

Information:
Xi = Normal data value
y = network output results
$Xmin$ = data with a minimum value
$Xmax$ = data with a maximum value

3. SYSTEM DESIGN

3.1 System Planning

Artificial neural network architecture that is built consists of several layers, namely: the input layer (input layer), one hidden layer (hidden layer), and the output layer (output layer). The connecting of each layer is weight.

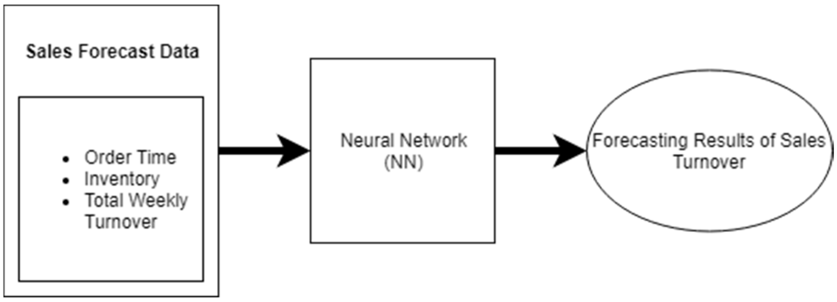

Figure 3.1 Block Diagram

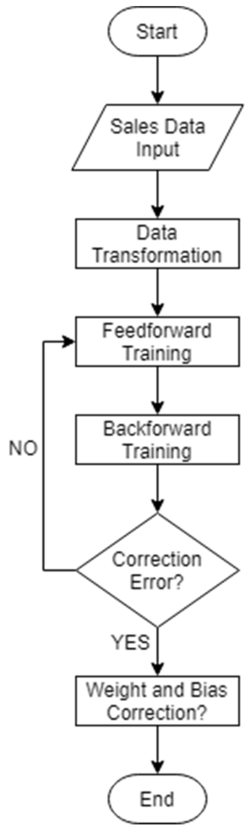

Figure 3.2 Artificial Neural Network Flowchart

The learning process itself begins with the feedforward process, and then continues with the backpropagation process. After the backpropagation process, a check will be made whether the target error value has been reached, if the target error has been reached, then the learning process is complete, which results in a correction of the network weight. If not, it will return to the feedforward process. This will continue until you find the maximum epoch value.

The input unit is denoted by the variable x, the hidden variable z and the output value are denoted by the variable y, while the weight value between x and z is denoted by the variable v and the weight value between z and y is denoted by the variable w.

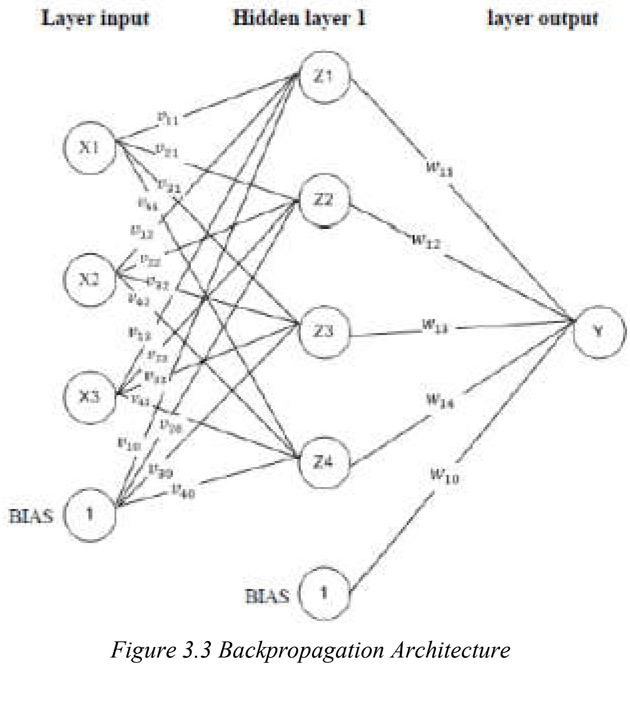

Figure 3.3 Backpropagation Architecture

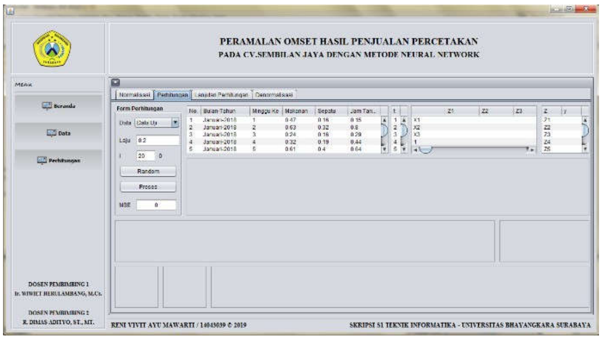

Figure 3.4 Display calculation form

4. DATA ANALYSIS

4.1 Neural Method Training

a. Data Processor

Data taken from CV Sembilan Jaya is the result of printing sales over the past 4 years as follows:

Table 4.1 Weekly sales data for January 2014

No	Month	Year	Week	Result		
				Food Dos	Shoe Dos	Watches Dos
1	January	2014	1	6350000	6175000	19500000
2	January	2014	2	12000000	17000000	10000000
3	January	2014	3	29750000	25250000	32850000
4	January	2014	4	76075000	27825000	37500000

Sales normalization data settlement is used min-max normalization, this method changes the data to another new range that is between 0 to 1 with the following formula:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Table 4.2 Data for normalization of weekly sales in January 2014

No	Month	Year	Week	Normalization Result		
				Food Dos	Shoe Dos	Watches Dos
1	January	2014	1	0.06	0.04	0.23
2	January	2014	2	0.14	0.17	0.11
3	January	2014	3	0.38	0.27	0.4
4	January	2014	4	1.00	0.3	0.46

b. Settlement Forecasting Using The Neural Network Method

$$\boldsymbol{z\_net_j = v_{j0} + \sum_{i=1}^{n} x_i v_{ji}}$$

$$z_{net_1} = 0.1 + 0.06 * 0.2 + 0.04 * 0.5 + 0.23 * 0.4 = 0.224$$
$$z_{net_2} = (-0.1) + 0.06 * 0.1 + 0.04 * 0.4 + 0.23 * 0.3 = -0.009$$
$$z_{net_3} = 0.1 + 0.06 * 0.4 + 0.04 * 0.3 + 0.23 * 0.5 = 0.251$$
$$z_{net_3} = (-0.1) + 0.06 * 0.3 + 0.04 * (-0.2) + 0.23 * 0.2 = -0.044$$

$$\boldsymbol{z_j = f\left(z_{net_j}\right) = \frac{1}{1 + e^{-z\_net\,j}}}$$

$$z_1 = \frac{1}{1 + e^{-0.224}} = 0.555767$$
$$z_2 = \frac{1}{1 + e^{-0.009}} = 0.49775$$
$$z_3 = \frac{1}{1 + e^{-0.251}} = 0.562423$$

$z_4 = \frac{1}{1 + e^{-0.044}} = 0.489002$

Step 5: Calculate all network outputs in unit yk (k = 1, 2, ..., m)

$$\boldsymbol{y\_net_k = w_{ko} + \sum_{j=1}^{p} z_j w_{kj}}$$

$y_{net_1} = -0.1 + 0.555767 * 0.5 + 0.49775 * (-0.3) + 0.562423 * (-0.4) + 0.489002 * 0.2 = -0.09861$

$$\boldsymbol{y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}}$$

$y_k = \frac{1}{1 + e^{-0.09861}} = 0.475367$

Calculate the δ unit of output factor based on the error in each unit of output yk (k = 1, 2, ..., m)

$$\boldsymbol{\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k)}$$

$\delta_k = (0 - 0.475367) * 0.475367 * (1 - 0.475367) = -0.11855342$

Calculate the weight change rate wkj (which will be used later to change the weight wkj) with the accelerating rate α

$$\boldsymbol{\Delta w_{kj} = \alpha \delta_k z_j \; ; k = 1, 2, \dots, m \; ; j = 0, 1, \dots, p}$$

$\Delta w_{10} = 0.2 * (-0.11855342) * 1 = -0.02371$
$\Delta w_{11} = 0.2 * (-0.11855342) * 0.555767 = -0.01318$
$\Delta w_{12} = 0.2 * (-0.11855342) * 0.49775 = -0.0118$
$\Delta w_{13} = 0.2 * (-0.11855342) * 0.562423 = -0.01334$
$\Delta w_{14} = 0.2 * (-0.11855342) * 0.489002 = -0.01159$

Calculate the factor tersembunyi hidden unit based on the error in each hidden unit zj (j = 1, 2, ..., p)

$$\boldsymbol{\delta\_net_j = \sum_{k=1}^{m} \delta_k w_{kj}}$$

$\delta\_net_1 = -0.11855342 * 0.5 = 0.05927671$
$\delta\_net_2 = -0.11855342 * (-0.3) = 0.035566$
$\delta\_net_3 = -0.11855342 * (-0.4) = 0.047421$
$\delta\_net_4 = -0.11855342 * 0.2 = 0.02371$

δ hidden unit factor:

$$\boldsymbol{\delta_j = \delta_{net_j} f'\left(z_{net_j}\right) = \delta_{net_j} z_j (1 - z)}$$

$\delta_1 = (-0.05927671) * 0.555767 * (1 - 0.555767) = -0.01463$
$\delta_2 = 0.035566 * 0.49775 * (1 - 0.49775) = 0.008891$
$\delta_3 = 0.047421 * 0.562423 * (1 - 0.562423) = 0.011671$
$\delta_4 = (-0.02371) * 0.489002 * (1 - 0.489002) = -0.005924803$

Calculate the weight change term v_{ji} (which will be used later to change the weight v_j)

$$\boldsymbol{\Delta v_{ji} = \alpha \delta_j x_i \; ; j = 1, 2, \dots, p ; i = 0, 1, \dots, n}$$

$\Delta v_{10} = 0.2 * (-0.01463) * 1 = -0.00292697$
$\Delta v_{20} = 0.2 * 0.008891 * 1 = 0.001778$
$\Delta v_{30} = 0.2 * 0.011671 * 1 = 0.002334$
$\Delta v_{40} = 0.2 * (-0.005924803) * 1 = -0.00118$

$\Delta v_{11} = 0.2 * (-0.01463) * 0.06 = -0.00018$
$\Delta v_{21} = 0.2 * 0.008891 * 0.06 = 0.000107$
$\Delta v_{31} = 0.2 * 0.011671 * 0.06 = 0.00014$
$\Delta v_{41} = 0.2 * (-0.005924803) * 0.06 = -7.10976E - 05$
$\Delta v_{12} = 0.2 * (-0.01463) * 0.04 = -0.00012$
$\Delta v_{22} = 0.2 * 0.008891 * 0.04 = 7.11306E - 05$
$\Delta v_{32} = 0.2 * 0.011671 * 0.04 = 9.33645E - 05$
$\Delta v_{42} = 0.2 * (-0.005924803) * 0.04 = -4.73984E - 05$
$\Delta v_{13} = 0.2 * (-0.01463) * 0.23 = -0.00067$
$\Delta v_{23} = 0.2 * 0.008891 * 0.23 = 0.000409$
$\Delta v_{33} = 0.2 * 0.011671 * 0.23 = 0.000537$
$\Delta v_{43} = 0.2 * (-0.005924803) * 0.23 = -0.00027$

Phase III: Change in weights
Step 8: Calculate all weight changes
Changes in line weights leading to the output unit:

$$\boldsymbol{w_{kj}(new) = w_{kj}(old) + \Delta w_{kj}(k = 1, 2, \dots, m\, ; j = 0, 1, \dots, p)}$$

$w_{10} = (-0.1) + (-0.02371) = -0.12371068$
$w_{11} = 0.5 + (-0.01318) = 0.486822$
$w_{12} = (-0.3) + (-0.0118) = -0.3118$
$w_{13} = (-0.4) + (-0.01334) = -0.41334$
$w_{14} = 0.2 + (-0.01159) = 0.188405$

Changes in line weights leading to hidden units:

$$\boldsymbol{v_{ji}(new) = v_{ji}(old) + \Delta v_{ji}(j = 1, 2, \dots, p\, ; i = 0, 1, \dots, n)}$$

$v_{10} = 0.1 + (-0.00292697) = 0.09707303$
$v_{20} = (-0.1) + 0.001778 = -0.09822$
$v_{30} = 0.1 + 0.002334 = 0.102334$
$v_{40} = (-0.1) + (-0.00118) = -0.10118$
$v_{11} = 0.2 + (-0.00018) = 0.199824$
$v_{21} = 0.1 + 0.000107 = 0.100107$
$v_{31} = 0.4 + 0.00014 = 0.40014$
$v_{41} = 0.3 + (-7.10976E - 05) = 0.299928902$
$v_{12} = 0.5 + (-0.00012) = 0.499883$
$v_{22} = 0.4 + 7.11306E - 05 = 0.400071131$
$v_{32} = 0.3 + 9.33645E - 05 = 0.300093364$
$v_{42} = (-0.2) + (-4.73984E - 05) = -0.200047398$
$v_{13} = 0.4 + (-0.00067) = 0.399327$
$v_{23} = 0.3 + 0.000409 = 0.300409$
$v_{33} = 0.5 + 0.000537 = 0.500537$
$v_{43} = 0.2 + (-0.00027) = 0.199727$

5. SYSTEM IMPLEMENTATION AND TRIAL

5.1 System Implementation

Implementation stage is a continuation of the forecasting system design stages that have been described in chapter IV, in this chapter designated as the steps of making the system described in the screen shoot display program and source code system. The work process of the forecasting system uses the Neural Network (NN) method.

P-ISSN: 2528-0260 E-ISSN: 2579-5392
p.675-686

Figure 5.1 Login menu

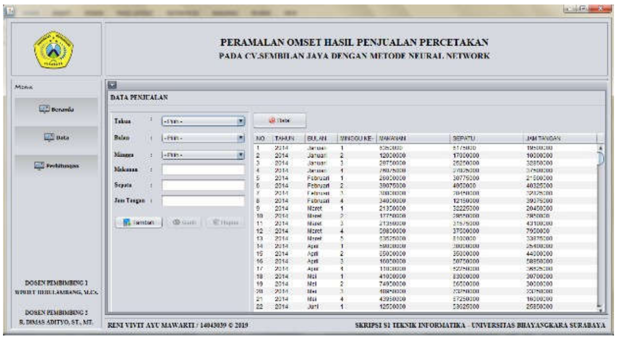

Figure 5.2 Sales Data Menu

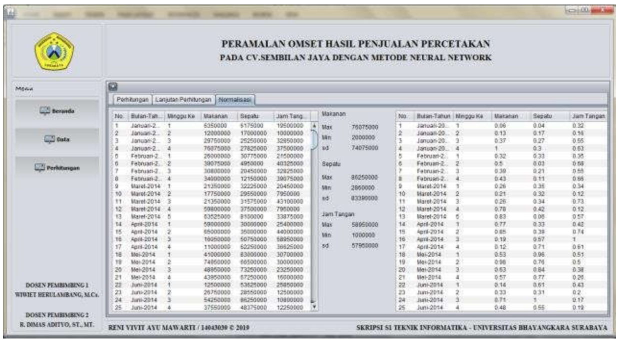

Figure 5.3 Normalization data menu

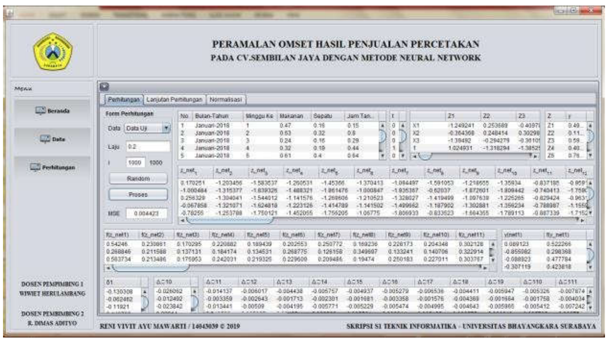

Figure 5.4 Calculation data menu

5.2 Trial Results

In the first trial process using data prices for food boxes, shoe boxes, watch boxes with weekly periods. Forecasting results using the backpropagation neural network method with α = 0.2 iteration = 1000 MSE = 0.004211. Like the picture below:

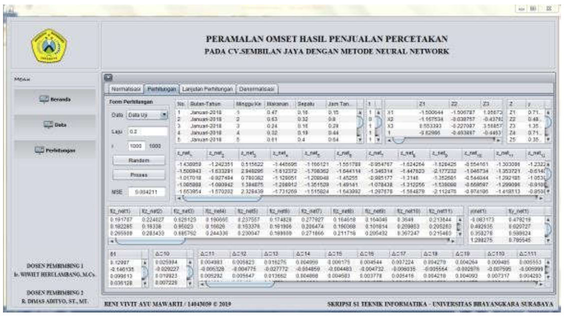

Figure 6.1 Display the results of the calculation of the first test data

In the trial process using data prices for food boxes, shoe boxes, watch boxes with weekly periods. Forecasting results using the backpropagation neural network method with α = 0.5 iteration = 1000 MSE = 0.015265. Like the picture below:

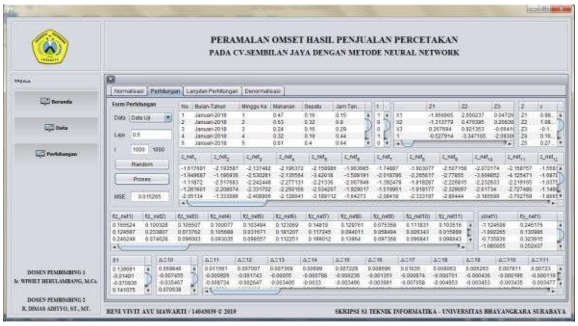

Figure 6.2 Display the results of the calculation of the second test data

In the trial process using data prices for food boxes, shoe boxes, watch boxes with weekly periods. Forecasting results using the backpropagation neural network method with α = 0.9 iteration = 1000 MSE = 0.008062. Like the picture below:

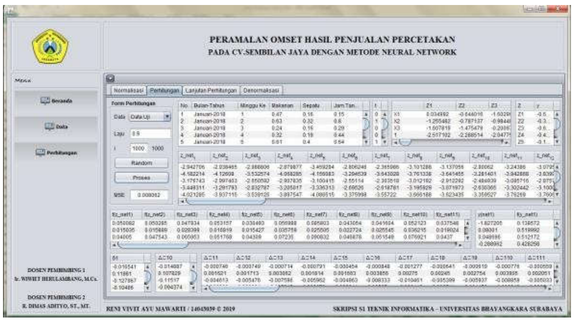

Figure 6.3 Display the results of the calculation of the third test data

From the test results in table 6.1, table 6.2 and table 6.3 we can calculate the average system error. Like the table below:

Table 6.4 Data for forecasting results 2018

No	Learning Rate	Epoch	MSE
1	0.2	1000	0.004211
2	0.5	1000	0.015265
3	0.9	1000	0.008062

In testing data using weights and biases that have been optimized with Artificial Neural Network iteration, the MSE value obtained with learning rate: 0.2 iteration: 1000 produces MSE 0.004211, meaning that the weights and biases used are optimal because it is able to produce MSE values ≤ 0.1. As in Figure 6.4 below:

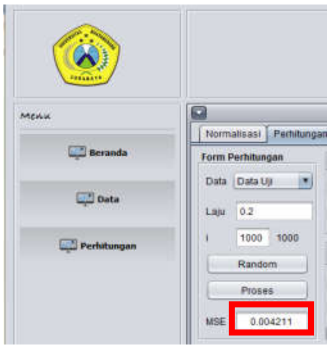

Figure 6.4 The smallest MSE value

6. CONCLUSION

The conclusion that can be drawn after several stages in completing the design and manufacturing of forecasting systems for printing turnover in CV. Sembilan Jaya Surabaya uses backpropagation neural network method by conducting an iteration search up to 1000 iterations, the smallest number of MSE is obtained at the 1000th iteration using a maximum error of 0.1, learning rate 0.2 obtains the smallest MSE value of 0.004211.

7. SUGGESTION

In the many deficiencies in the research of forecasting sales of printing with the neural network method is expected to create a forecasting system using the neural network method can then be further developed with the following development suggestions:

1. In forecasting it can be done periodically by using newer data.
2. By testing and comparing the existing systems, it is hoped that they can be refined further in the next research in order to create more complex applications so that there are many facilities in this program.

REFERENCES

[1] Santosa. B., *Data Mining, Teknik Pemanfaatan Data Untuk Keperluan Bisnis*. Graha Ilmu , 2007.
[2] Laurene V Fausett, *Fundamentals of neural networks | Guide books*. Florida Institute of Technology, Melbourne, 1992.
[3] Maria Agustin, "Penggunaan jaringan Syaraf Tiruan BACKPROPAGATION UNTUK SELEKSI PENERIMAAN MAHASISWA BARU PADA JURUSAN TEKNIK KOMPUTER DI POLITEKNIK NEGERI SRIWIJAYA," *Universitas Diponegoro Semarang*, Jul-2012. [Daring]. Tersedia pada: http://eprints.undip.ac.id/39239/1/Maria_Agustin.pdf.
[4] E. Krisnanik, K. Kraugusteeliana, dan V. Indriasari, "Desain Model Sistem Pakar Menu Sehat Wanita Hamil Berdasarkan Gizi Menggunakan Metode Cooper," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 6, hal. 643, Nov 2018.
[5] R. D. Adityo dan H. Miawarni, "Implementation of Naive Bayes Classifier Algorithm to Evaluation in Utilizing Online Hotel Tax Reporting Application," *VOLT J. Ilm. Pendidik. Tek. Elektro*, vol. 2, no. 2, hal. 125, Okt 2017.