FORECASTING THE INFLATION RATE FOR PADANG MUNICIPALITY BY USING BACKPROPAGATION METHOD

ATIKAH
NIM. 16030037/2016

UNDERGRADUATE PROGRAM OF MATHEMATICS
DEPARTMENT OF MATHEMATICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
PADANG STATE UNIVERSITY
2020

FORECASTING THE INFLATION RATE FOR PADANG MUNICIPALITY BY USING BACKPROPAGATION METHOD

UNDERGRADUATE THESIS

As a partial fulfillment for bachelor degree in science

By:
ATIKAH
NIM. 16030037/2016

UNDERGRADUATE PROGRAM OF MATHEMATICS
DEPARTMENT OF MATHEMATICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
PADANG STATE UNIVERSITY
2020

PERSETUJUAN SKRIPSI

Judul	: Peramalan Tingkat Inflasi di Kota Padang Menggunakan Metode *Backpropagation*
Nama	: Atikah
NIM	: 16030037
Program Studi	: Matematika
Jurusan	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam

Padang, 22 November 2020
Disetujui oleh,
Pembimbing

Defri Ahmad, S.Pd, M.Si
NIP.19880909 201404 1 002

HALAMAN PENGESAHAN LULUS UJIAN SKRIPSI

Nama	: Atikah
NIM / TM	: 16030037/2016
Program Studi	: Matematika
Jurusan	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam

Dengan Judul Skripsi

PERAMALAN TINGKAT INFLASI DI KOTA PADANG MENGGUNAKAN METODE *BACKPROPAGATION*

Dinyatakan lulus setelah dipertahankan di depan Tim Penguji Skripsi
Program Studi Matematika Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Padang

Padang, 22 November 2020

Tim Penguji

	Nama	Tanda Tangan
Ketua	: Defri Ahmad, S.Pd, M.Si	
Anggota	: Dra. Helma, M.Si	
Anggota	: Dra. Arnellis, M.Si	

SURAT PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama	: Atikah
NIM	: 16030037
Program Studi	: Matematika
Jurusan	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam

Dengan ini menyatakan, bahwa skripsi saya dengan judul **"Peramalan Tingkat Inflasi di Kota Padang Menggunakan Metode *Backpropagation*"** adalah benar merupakan hasil karya saya dan bukan merupakan plagiat dari karya orang lain atau pengutipan dengan cara-cara yang tidak sesuai dengan etika yang berlaku dalam tradisi keilmuan. Apabila suatu saat terbukti saya melakukan plagiat maka saya bersedia diproses dan menerima sanksi akademis maupun hukum sesuai dengan hukum dan ketentuan yang berlaku, baik di institusi UNP maupun di masyarakat dan negara.

Demikianlah pernyataan ini saya buat dengan penuh kesadaran dan rasa tanggung jawab sebagai anggota masyarakat ilmiah.

Padang, 22 November 2020

Diketahui oleh,
Ketua Jurusan Matematika,

Dra. Media Rosha, M.Si
NIP. 19620815 198703 2 004

Saya yang menyatakan,

METERAI TEMPEL
CDC3BAHF755632
6000 ENAM RIBU RUPIAH

Atikah
NIM. 16030037

Forecasting the Inflation Rate for Padang Municipality by Using Backpropagation Method

Atikah

ABSTRACT

Inflation is an indicator widely used to measure the economic power. The calculation of inflation rate for national level is aggregated from that of districts and municipalities in Indonesia. Padang municipality is one of those regions taking part in the calculation for national inflation rate which typically has one of the highest inflation rates in Indonesia. Therefore, in order to have a better planning at controlling the inflation rates in the future for Padang municipality, it is necessary to forecast the inflation first. This study is conducted to present a forecasting analysis of monthly inflation rates in Padang municipality for the next three years.

This research is classified as an applied research using secondary data, namely the inflation rate of Padang municipality from January 2013 to December 2019. In the analysis, the data is divided into two parts specifically for training and for testing the forecast model. The forecasting analysis technique uses a backpropagation method.

The analysis result shows that the forecasts for inflation rates from 2020 to 2022 are fluctuating by month, where the lowest rate is predicted to occur in February 2020 of -0.76 while the highest occurs in July 2020. The backpropagation method used is considered to yield accurate forecasts since the model parameters including the bipolar sigmoid activation function and the learning rate of 0.1 are able to give a relatively small error rate with a MSE value of 0.011956.

Keywords: Backpropagation*, MSE, Forecasting, Inflation Rate, Padang Municipality*

FOREWORD

Alhamdulillahi robbal"alamin I would like to praise Allah SWT for all His mercy, blessing and guidance so that I can complete the writing of the thesis entitled "**Forecasting the Inflation Rate for Padang Municipality by Using Backpropagation Method"**. This thesis is intended as a partial fulfillment to obtain a degree in science in the undergraduate study of mathematics in the Department of Mathematics, Padang State University.

Obviously, some obstacles and odds have been encountered during the completion of this research but with the support, counsels, and assistance from everyone, those problems can be overcome. In this occasion, I would like to deliver my gratitude to:

1. Mr. Defri Ahmad, S.Pd, M.Si, my academic and thesis advisor.
2. Mrs. Dra. Hj. Helma, M.Si and Mrs. Dra. Arnellis, M.Si, as the thesis examiners.
3. Mrs. Dra. Media Rosha, M.Si , as the Head of Undergraduate Study Program in Mathematics, Padang State University.
4. All lecturers and employees in the Department of Mathematics, Padang State University.
5. My parents and other families whose prayers, motivation and advices always accompany me.
6. The one who is always beside me during the research.
7. My classmates in Mathematics 2016.
8. All other people who have helped me in many ways during the completion of this thesis.

Hopefully the guidance, assistance and prayers that have been given to me can become deeds that only Allah SWT can reply with His blessing in life. I realize that this thesis is far from perfect for the limited capacity that I have. Therefore, I look forward to constructive criticism and suggestions from readers for the perfection of this thesis and make this thesis to be more useful for all.

Padang, 22 November 2020

Author

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I
INTRODUCTION

A. Research Background

Inflation is defined as a general and continuous increase in prices (Bank Indonesia, 2020). Inflation has become one of the main macro economic problems encountered by a region or country and an indicator to measure the success of a country's economy. Various problems may arise when the inflation rate increases continuously, such as an unstable economy, slow economic growth because the increase in prices for goods and services is not balanced by an increase in income, as well as a decline in the currency value which indirectly affects global trade activities.

According to Khalwaty (2000: 12) inflation that exceeds a double digit figure (above 10%), not only increases general prices and reduces the value of money, but also increases the unemployment rate, widens the gap between the rich and poor, between the large-scale entrepreneurs (conglomerates) with middle-to-lower entrepreneurs, between farmers owning large land areas and smallholders, between employers and workers, and can diminish the confidence of the international community (investors) in a country's government performance. These various impacts force every country to be able to maintain the inflation rate at the desired level. This is necessary in order to encourage maximum economic growth and create a controlled national economy.

However, for a developing country like Indonesia, maintaining the inflation rate is quite complicated. The national inflation rate is calculated as an aggregate of inflation in all urban and rural areas. For example, the inflation rate in Sumatera Barat Province, whose most of the population lives in rural areas, is calculated as an aggregate of urban and rural inflation. In fact, out of 19 districts/cities in Sumatera Barat Province, it is only Padang and Bukittinggi municipalities that are selected as the inflation measurement regions using the CPI (Consumer Price Index) method since both cities are considered the main traffic centers for goods and services from all other regions in Sumatera Barat. CPI is an index that represents the average price change of a number of goods and services consumed by households in a certain period of time. This study focuses on the inflation analysis for the city of Padang.

According to the Statistics (BPS) of Sumatera Barat Province, in November 2016, out of all inflation-calculating regions in Indonesia, the inflation rate in Padang municipality, of 1.13 percent, was ranked fourth nationally. In July 2017, with an inflation rate of 0.54 percent, Padang City ranks second out of all inflation-calculating regions in Sumatera and ranks sixteenth nationally. Meanwhile, at the end of 2018, the inflation rate of Padang City was recorded to be much lower, amounting to 0.16 percent, making it the nineteenth in Sumatera and 75th nationally. However, the success of controlling inflation in 2018 was not repeated in 2019. Even though inflation at the end of 2019 was considered to be under control, inflation during Ramadan and Lebaran remained the highest in 2019, amounting to 0.86 percent for May and 1.07 percent for the month. June.

Table 1. Comparison between inflation rates in Padang municipality and Indonesia

Month	Padang	Indonesia	Month	Padang	Indonesia
Jan 2016	0.02	0.51	Jan 2018	0.43	0.62
Feb 2016	0.86	-0.09	Feb 2018	-0.09	0.17
Mar 2016	0.55	0.19	Mar 2018	0.31	0.2
Apr 2016	-0.92	-0.45	Apr 2018	0.01	0.1
May 2016	-0.37	0.24	May 2018	0.46	0.21
Jun 2016	0.1	0.66	Jun 2018	0.39	0.59
Jul 2016	1.52	0.69	Jul 2018	0.62	0.28
Aug 2016	0.84	-0.02	Aug 2018	-0.4	-0.05
Sep 2016	0.58	0.22	Sep 2018	-0.35	-0.18
Oct 2016	0.56	0.14	Oct 2018	0.8	0.28
Nov 2016	1.13	0.47	Nov 2018	0.19	0.27
Dec 2016	0.07	0.42	Dec 2018	0.16	0.62
Jan 2017	0.57	0.97	Jan 2019	0.24	0.32
Feb 2017	-0.13	0.23	Feb 2019	-0.44	-0.08
Mar 2017	-0.01	-0.02	Mar 2019	0.33	0.11
Apr 2017	-0.31	0.09	Apr 2019	0.44	0.44
May 2017	-0.04	0.39	May 2019	0.86	0.68
Jun 2017	0.34	0.69	Jun 2019	1.07	0.55
Jul 2017	0.54	0.22	Jul 2019	0.89	0.31
Aug 2017	-0.36	-0.07	Aug 2019	-0.1	0.12
Sep 2017	0.13	0.13	Sep 2019	-0.95	-0.27
Oct 2017	0.19	0.01	Oct 2019	-0.34	0.02
Nov 2017	0.48	0.2	Nov 2019	-0.34	0.14
Dec 2017	0.72	0.71	Dec 2019	0.07	0.34

(BPS - Statistics Indonesia, 2020)

Table 1 shows that the inflation rate of Padang municipality has been recorded to be above that of Indonesia for some months. It suggests that the inflation rate of Padang municipality needs to be more maintained. In order to have better planning at maintaining or controlling the inflation rate in the future, it is necessary to have a good prediction on the inflation rate, which can be performed using a forecasting method.

Forecasting is an activity to predict what will happen in the future. Forecasting of inflation can be used to prepare government policies so that inflation remains more controlled in the future. In addition to that, the public may learn from the prediction o anticipate the negative impact of socio-economic conditions during inflation. To achieve this goal, an appropriate forecasting method is needed to forecast using inflation data.

According to Suparti (2013: 1) in 'Analysis of Inflation Data in Indonesia Using the Spline Regression Model', inflation data is one of the time series data in the past that can be modeled to predict future inflation data. However, inflation data has the irregular behaviors (suddenly rise or fall) which tends to be difficultly predicted. According to Sawitri, Sumarjaya, and Tastrawati (2018: 264), irregular data like inflation can be forecasted using a type of forecasting method, namely the backpropagation method of neural networks, which involves a training process on the data. The backpropagation neural network method then is able to recognize a pattern in the data using the training result and estimate future values.

Artificial neural networks do not require factors to be addressed, do not recognize formulas and rules, and apply in general. Therefore, the artificial neural network can be directly applied in predicting the inflation rate of Padang municipality without concerning the past data pattern since the method can learn the pattern by itself. The artificial neural network can also learn the relationship between variables to result the optimum prediction model. According to Siang (2009), one of the most suitable artificial neural methods applied to inflation forecasting is the backpropagation method, because this method has the advantage of minimizing errors in the output generated by the neural network..

Based on the background stated above, the implementation of backpropagation method to predict the inflation rate in Padang municipality is encouraging and important to perform within a research entitled **"Forecasting the Inflation Rate for Padang Municipality by Using Backpropagation Method"**

B. Problem Formulation

Based on the background stated above the problem discussed in this research is formulated as "*How to forecast the inflation rate for Padang municipality by using* backpropagation *method?"*

C. Research Questions

Based on the problem formulated above, the following research questions are proposed:

1. How to analyze the forecasting on inflation rate for Padang municipality for the next three years?
2. How is the error value of the forecast?

D. Research Objectives

This research is intended to to provide answers to the research questions proposed above, as follows:

1. Analyzing the forecasting on inflation rate for Padang municipality for the next three years.
2. Calculating the error value of the forecast.

E. Research Benefits

This research is expected to give the following benefits:

1. For author

As to widen the insight and knowledge on the forecasting on inflation rate for Padang municipality by using backpropagation method.

2. For stakeholders

   As an instrument for public in making investment decisions and for government in mitigating future inflations.

3. For readers

   As an advanced information for literatures on backpropagation method used to forecast inflation for Padang municipality and reference for further research.

CHAPTER II
THEORETICAL REVIEW

A. Inflation

Inflation is a measure that indicates a sharp (absolute) increase in prices continuously for quite a long time. As long as these prices increase, the value of money falls sharply in proportion to the increase in these prices (Khalwaty 2000:6). Conversely, a situation where there is a continuous decline in the price of goods or an increase in the value of money is called deflation.

Based on the severity, inflation level can be grouped into the following types:

a) Mild inflation (below 10% annually)

b) Medium inflation (10%-30% annually)

c) Severe inflation (30%-100% annually)

d) Hyperinflation (above 100% annually)

(Sinungan, 1995:51)

The percentage (%) change in the index or inflation rate each month is obtained by subtracting the consumer price index (CPI) of a month from the previous month's consumer price index (CPI) then divided by the previous month's consumer price index (CPI) multiplied by 100. Alternatively, a month’s CPI divided by the previous month's CPI, the result is subtracted by 1 and multiplied by 100, which is as follows

$$L(I)_n = \frac{I_n - I_{n-1}}{I_{n-1}} \times 100 \; atau \; \left(\frac{I_n}{I_{n-1}} - 1\right) \times 100$$

where:

$L(I)_n$: Inflation rate in month n

I_n : Inflation in month n

I_{n-1} : Inflation in month $n-1$

The inflation rate by calendar year (annual inflation) is calculated by dividing the inflation of month n by the inflation of December in previous year. The *year-on-year* (YoY) inflation is obtained by dividing the inflation a month in year n with that of the same month in year $n-1$ (BPS - Statistics of Sumatera Barat Province, 2020)

B. Artificial Neural Network

Artificial neural network is an information processing system that has characteristics similar to biological neural networks. In principle, the artificial neural network can compute all computable functions. Artificial neural networks are practically useful for classification and problems that can tolerate imprecision, have a large size of training data, but also have rules that cannot be applied easily.

Artificial neural networks are data processing techniques that study the relationship between input data and output data. The artificial neural network works based on the input pattern that significantly affects the output.

1. Architecture of neural network

Biological neural network is one of the artificial representations of the human brain that simulate the learning process in the human brain. The biological neural network is illustrated by the following figure.

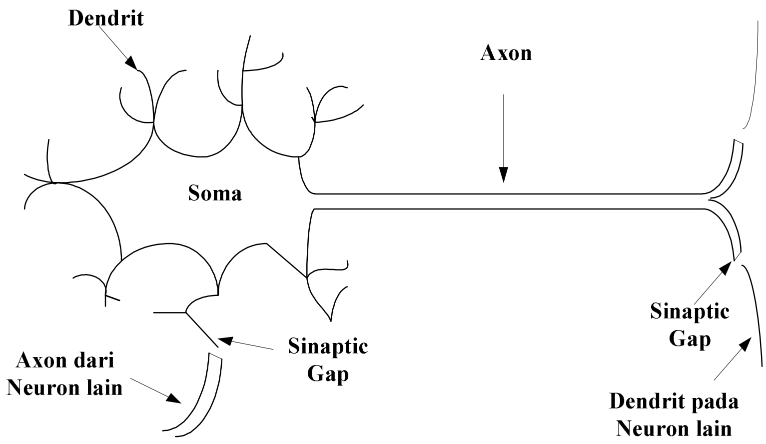

Figure 1 Biological neural network

Each nerve cell (neuron) has a cell nucleus which is responsible for processing information. The incoming information is received by the dendrites. Beside receiving information, dendrites also accompany axons as the output of information processing. This processed information becomes input for other neurons where the dendrites of the two cells are met with synapsis. The information sent between neurons is in the form of stimuli that are passed through the dendrites. Information that comes and is received by dendrites is summed and sent via axon to the final dendrite in contact with dendrites from other neurons. This information will be accepted by other neurons if it meets a certain threshold which is often known as the threshold value, which is said to be activated.

The relationship between the concept of biological neural networks and artificial neural networks is presented in the following table:

Table 2. The relationship between the concept of biological and artificial neural networks

Biological	Artificial
Soma	Node
Dendrite	Input
Axon	Output

Synapse	Weight (bobot)
Slow Speed	Fast Speed
Consists of numerous neuron (10^9)	Consists of several neuron (unit)

(Desiani,2006: 161)

In general, artificial neural networks consist of several layers, each of which has a different number of neurons (units). The following is the layer architectural drawing.

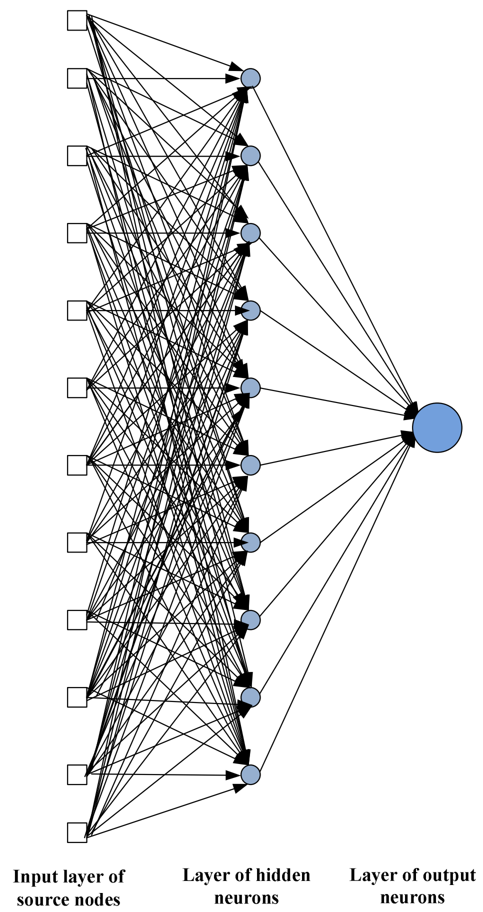

Figure 2 The acchitecture of artificial neural network

1) Input layer

This layer contains a number of units receiving signals (outside data), which describe a problem and then will forward the information to other units in the network. In this study, the input layer contains 12 units, referring to the number of months in a year, from January to December in which the inflation of Padang municipality occurs monthly.

2) Hidden layer

This layer contains units which cannot be directly observed. This layer serves to increase the network's ability to solve problems. In this study, the number of hidden layer units is twelve.

3) Output layer

This layer serves to distribute the output signals from network processing. In this study the output layer only contains one unit, which refers to the time of forecasting that is carried out in the next one month, namely the results of forecasting inflation in Padang municipality monthly.

By the number of layers, in general, there are two types of artificial neural network architecture, as follow:

1) Single layered artificial neural network

Single layered neural network has only one input layer and one output layer with connected weights, without any hidden layers. This artificial neural network only accepts input, then directly processes it into output without having to go through hidden layers.

2) Multilayered artificial neural network

The multilayered artificial neural network has one or more layers that lie between the input layer and the output layer (hidden layer). Usually there are weights located between two adjacent layers. This network can solve more difficult problems than a single layer neural network, but with more complex learning.

2. Learning process

Based on the input and output functions, it can be said that the function of the artificial neural network is determined by the connecting weights. A process of sequential adjustment of parameters is carried out in order to approximate the desired function. The process of adjusting parameters or weights in an artificial neural network is called the learning or training process. This process aims to regulate the weights that exist in the artificial neural network, so that the final weight is obtained according to the trained data pattern. According to Nugraha (2008) there are basically two learning methods, namely:

1) Supervised learning

   The learning method in artificial neural networks is called supervised if the expected output is known in advance. The expected output here is a final value that is used as a benchmark for the forecast results for the following year. Suppose the artificial neural network that is owned will be used to recognize the pattern pairs. In the learning process, an input pattern will be assigned to a unit in the input layer, which means one month of data for one input layer. This pattern will be propagated along the artificial neural network to the unit in the output layer. Then, this output layer will generate an output pattern which will be matched with the target output pattern. If there is a difference between the learning (training) output pattern and the target pattern, an error will appear. If the error value is still large enough, it means that more learning is needed.

2) Unsupervised learning

This method does not require a target output and it cannot be determined what results are expected during the learning process. When the learning process progresses, the weight values are arranged in a certain range depending on the input value given. This learning aims to group units that are almost the same in a certain area. This learning method is generally suitable for grouping (classification) patterns.

3. Activation function

A mathematical function that is used for restricting and determining the output range of a unit is called the activation function. This restriction is closely related to the specific range that the later layers in the network can handle. Selection of the right activation function in an artificial neural network application will significantly affects its performance both in terms of data processing speed and in terms of the accuracy of the results (Sri Redjeki, 2014). The properties that must be possessed by the activation function are to be continuous and not monotonically decreasing. The activation function is expected to approach the maximum and minimum values asymptotes (Puspitaningrum, 2006).

According to Desiani (2006: 172-174) there are several activation functions used in artificial neural networks, as follow:

1) Binary sigmoid function

Sigmoid biner function depends on a stepness parameter (σ). In order for this function yields a binary value (0 or 1) then for each $\sigma = 1$ a nonlinear continuous graph is resulted. The function is defined as follows:

$$f(x) = \frac{1}{1 + e^{-\sigma x}} = \frac{1}{1 + e^{-x}} \tag{1}$$

where the derivative of $f(x)$ is

$$f'(x) = \frac{d}{dx}f(x) \quad = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right)$$

$$= \frac{\frac{d}{dx}(1)(1+e^{-x}) - \frac{d}{dx}(1+e^{-x})(1)}{(1+e^{-x})^2}$$

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

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

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

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

$$= \left(1 - \frac{1}{1+e^{-x}}\right)f(x)$$

$$= \big(1 - f(x)\big)f(x)$$

2) Bipolar sigmoid function

This function is a binary sigmoid function yang diperluas extended to reach negative values on the x axis. Similar to the binary sigmoid function, it also depends on a stepness parameter (σ). Thus for $\sigma = 1$, this function returns the value between -1 to 1 and is defined below

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

$$= \frac{1-e^{-\sigma x}}{1+e^{-\sigma x}}$$

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

with its first derivative is:

$$g'(x) = \frac{d}{dx}g(x) \quad = \frac{d}{dx}\left(\frac{1-e^{-x}}{1+e^{-x}}\right)$$

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

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

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

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

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

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

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

$$= \big(1+f(x)\big)\big(1-f(x)\big)$$

$$= \left(1+\frac{g(x)-1}{2}\right)\left(1-\frac{g(x)-1}{2}\right)$$

$$= \left(\frac{2+g(x)-1}{2}\right)\left(\frac{2-g(x)-1}{2}\right)$$

$$= \frac{\big(1+g(x)\big)\big(1-g(x)\big)}{2}$$

3) Linear/identity function

A linear function returns a value that equals its argument, written as

$$f(x) = x \tag{3}$$

with its derivative is equal to 1, that is

$$f'(x) = \frac{d}{dx}f(x) = 1 \tag{3}$$

4) Hyperbolic tangent function

This function is quite similar with the bipolar sigmoid function as it returns a value ranging from -1 to 1. A hyperbolic tangent function is defined as below

$$\sinh(x) = \frac{e^x - e^{-x}}{2}$$

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

$$y = f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4}$$

or,

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

$$= \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

with the derivative of $f(x)$ is

$$f'(x) = \frac{d}{dx} f(x) \quad = \frac{d}{dx}\left(\frac{1 - e^{-2x}}{1 + e^{-2x}}\right)$$

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

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

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

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

$$= \frac{4e^{-2x}}{(1 + e^{-2x})^2}$$

$$= \left(\frac{2e^{-2x}}{1+e^{-2x}}\right)\left(\frac{2}{1+e^{-2x}}\right)$$

$$= \left(\frac{e^{-2x}+1-1+e^{-2x}}{1+e^{-2x}}\right)\left(\frac{1+e^{-2x}+1-e^{-2x}}{1+e^{-2x}}\right)$$

$$= \left(1-\left(\frac{1-e^{-2x}}{1+e^{-2x}}\right)\right)\left(1+\left(\frac{1-e^{-2x}}{1+e^{-2x}}\right)\right)$$

$$= (1-f(x))(1+f(x))$$

Some of the four activation functions described above will be applied to the backpropagation method if it meets several conditions: continuous, easily differentiated, and non-decreasing. Then the functions that fulfill the three requirements are the binary sigmoid, bipolar sigmoid, and identity functions.

4. Backpropagation method

The backpropagation method is an advanced learning method developed from the perceptron rules, namely the network algorithm stages. This method is a systematic method for training of multilayer artificial neural networks. This method has a strong and objective mathematical basis, this algorithm obtains the form of equations and coefficient values in the formula by minimizing the number of squared error errors through the developed model. The backpropagation model generally has multiple layers with one hidden layer which aims to minimize errors in the output generated by the network. The more hidden units and the number of layers used, the more complex the network is built, the better the forecasting results and the longer the training time required.

In the backpropagation method, the input layer $\mathbf{X}_i$ containing the units ($i = 1, \dots, n$), the output layer $\mathbf{Y}_k$ ($k = 1, \dots, m$), and the hidden layer $\mathbf{Z}_j$ ($j = 1, \dots, p$). The weight from the input unit to the hidden unit $\mathbf{Z}_j$ is denoted by $\mathbf{V}_{ij}$ while the bias contained in the hidden unit is denoted by V_{0j}. The weight from the hidden unit to the output $\mathbf{Y}_k$ is denoted by $\mathbf{W}_{jk}$ and the bias occurs in the output layer is denoted by $\mathbf{W}_{0k}$. This bias acts as a weight on the connection coming from a unit whose output is always one. The signal flow in the figure is represented by the arrow direction. As for the backpropagation phase, the signal is sent in the opposite direction.

The number of hidden layers in backpropagation method is determined experimentally. The higher the number of hidden layers it is expected that the network will give more accurate results, but with more complicated and time-consuming training process. The number of units in the hidden layer is determined by the number of units in the input layer.

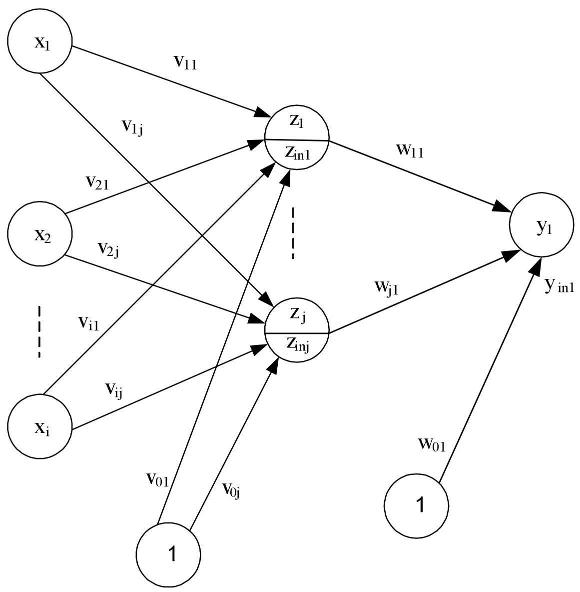

Figure 3. Backpropagation architecture

Where:

X : Input layer

V : Weight between X and Z

Z : Hidden layer

W : Weight between Z and Y

1 : Bias between Z and Y

Y : output layer

5. Backpropagation algorithm

Backpropagation is a supervised learning method and is usually used by multi-layer perceptrons to change the weights associated with the units in the hidden layer. The backpropagation algorithm uses an output error to change the weight values in the backward direction. The forward propagation stage must be done first to get this error. At the time of forward propagation, the units are activated using the activation function.

The backpropagation training algorithm according to Fausett (1994) is carried out through the following steps:

Step 0 : Initiate weight and bias;

Weight and bias can be initiated by a small random number ranging from -1 to 1

Step 1 : If training process has not stopped, do steps 2-9;

Step 2 : For each training process, do steps 3-8;

Phase I : Feedforward

Step 3 : Each input unit $(X_i,\ i = 1, \dots, n)$; receives input signal X_i and forward the signal to every unit in the hidden layer.

Step 4 : For each unit in hidden layer $(Z_j,\ j = 1, \dots, p)$; add the weight and bias to the respective input signal

$$Z_{in_j} = V_{0j} + \sum_{i=1}^{n} X_i V_{ij}$$

Then calculate the output signal value of the hidden layer using the predefined activation function:

$$Z_j = f\left(Z_{in_j}\right) = \frac{1 - e^{-Z_{in_j}}}{1 + e^{-Z_{in_j}}}$$

This output signal is then sent to all units in the next layer (output layer).

Step 5 : For each unit in the output layer $(Y_k, k = 1, \ldots, m)$; add the weight and bias to the input signals

$$Y_{in_k} = W_{0k} + \sum_{i=1}^{p} Z_j W_{jk}$$

then calculate the output signal from the respective output unit using the predefined activation function

$$Y_k = f\left(Y_{in_k}\right) = \frac{1 - e^{-Y_{in_k}}}{1 + e^{-Y_{in_k}}}$$

This output signal is then sent to all units at the network output.

Phase II: Backpropagation of Error

Step 6 : Each unit in output layer $(Y_k, k = 1,2, \ldots, m)$; accepts a network-generated output pattern. To calculate the error between the input target and the output generated by the network.

$$\delta_k = (T_k - Y_k) f'\left(Y_{in_k}\right) = (T_k - Y_k) Y_k (1 - Y_k)$$

The error factor δ_k used to correct (W_{jk}) in the hidden layer with a training rate (learning rate) α.

$$\Delta W_{jk} = \alpha \delta_k Z_j; \quad (k = 1,2, \dots, m\,;\ j = 1, \dots, p)$$

Also calculate the bias correction factor that will be used to correct the bias weights W_{0k}

$$\Delta W_{0k} = \alpha \delta_k; \qquad (k = 1,2, \dots, m)$$

The factor δ_k is sent to layer in the step 7.

Step 7 : Each unit of hidden layer $(Z_j, j = 1, \dots, p)$; receive the δ_k input from step 6. Then calculate δ for each hidden unit based on the error of each hidden unit.

$$\delta in_j = \sum_{k=1}^{m} \delta_k W_{jk}$$

Then calculate factor δ in the hidden unit using the derived activation function to yield an error value, using

$$\delta_j = \delta in_j f'\left(Z_{in_j}\right) = \delta in_j Z_j\left(1 - Z_j\right)$$

Calculate changes in the weight V_{ij} to correct the weight V_{ij},

$$\Delta V_{ij} = \alpha \delta_j X_i; (j = 1,2, \dots, p\,;\ i = 1, \dots, n)$$

Then calculate the change in bias to correct the bias V_{0j} using

$$\Delta V_{0j} = \alpha \delta_j$$

Phase III: Update bias and weight

Step 8 : Calculate the change in weight and bias in the hidden unit towards the unit of output.

$$W_{jk}(baru) = W_{jk}(lama) + \Delta W_{jk}; \quad (k = 1, 2, \ldots, m\ ; j = 1, \ldots, p)$$

Then calculate the change in weight and bias in the input unit towards the hidden unit.

$$V_{ij}(baru) = V_{ij}(lama) + \Delta V_{ij}; \quad (j = 1, 2, \ldots, p\ ; i = 1, \ldots, n)$$

Step 9 : The training processes will stop when the conditions are met, but if not, then take steps 2-9.

The terms and symbols used in the steps described above are.

X_i : Input units

V_{ij} : weight of the input unit relative to the hidden unit

Z_j : Output from the hidden units

Z_{in_j} : Output factors in the hidden units

Y_k : Output units

W_{jk} : weight of the hidden unit relative to the output unit

Y_{in_k} : Output factors in the output units

V_{0j} : biases in the hidden units

W_{0k} : biases in the output units

δ_j : Error factors in the hidden layer

δ_k : Error factors in the output layer

α : Learning rate

ΔW_{jk} : terms of weight change

T_k : input target

e : exponential constant of 2,718

The backpropagation neural network algorithm is described as follows:

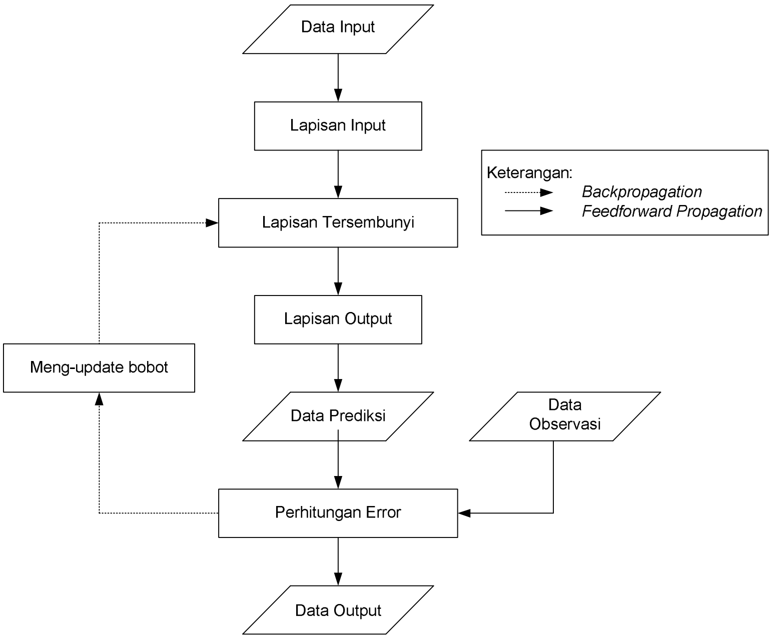

Figure 4. Backpropagation algorithm

Based on the figure above, at the feedforward stage, the network output can be determined. The next step is the calculation of errors, where the observational data will be used, namely the actual data for the last year which has been transformed for the activation function. Then, update the weight that will be used in the hidden unit. This stage is called backpropagation. The three phases of the algorithm are repeated until the stop conditions are met. The stop condition used is the smallest MSE value that is close to the specified tolerance limit.

The backpropagation calculation of the error is as follows.

1) Each output unit $(Y_k, k = 1, \dots, m)$ receives one target pattern (desired output) appropriate to the input training pattern to calculate the error between the target and the output generated by the network, $\delta_k = (T_k - Y_k)f'(Yin_k)$. Like the input training data, the output training data T_k has also been scaled for the activation function used. The factor δ_k is used

to calculate the error correction (ΔW_{jk}) which is later used to update the W_{jk}, where $\Delta W_{jk} = \alpha \delta_k Z_j$. Besides, the bias correction ΔW_{0k} is alsos calculated to update the bias W_{0k}, where $\Delta W_{0k} = \alpha \delta_k$. The factor δ_k is forwarded to the layer in the next step.

2) Each hidden unit ($Z_j, j = 1, \dots, p$)) adds a weighted delta input (sent from the previous step) $\delta in_j = \sum \delta_k w_{jk}$. Then the result is multiplied by the derivative of the activation function used by the network to produce an error correction factor δ_j, where $\delta_j = \delta in_j f'(Zin_j)$.

The δ_j factor is used to calculate the error correction (ΔV_{ji}) which will later be used to update V, where $\Delta V_{ij} = \alpha \delta_j X_i$. Besides, the bias correction is also calculated for ΔV_{0j}which will later be used to update V_{0j}, where $\Delta V_{0j} = \alpha \delta_j$.

Weight and bias adjustment

All biases and weights in the output units ($Y_k, k = 1, \dots, m$) will be updated of the hidden units: ($j = 0, \dots, p$), $W_{jk}(baru) = W_{jk}(lama) + \Delta W_{jk}$. As for those of the hidden units ($Z_j, j = 1, \dots, p$) will be updated of the input units: ($i = 0, \dots, n$), $V_{ij}(baru) = V_{ij}(lama) + \Delta V_{ij}$.

Checking the stopping condition

Neural network training can be stopped if the stopping conditions are met. There are two ways that can be used to check the stopping conditions, namely:

1) Restrict the iteration number
2) Determine certain error value

In the backpropagation method, to calculate the average error between the desired output on the training data and the output generated by the network, the Mean

Square Error (MSE) measure is used. For example, if the error has reached 0.01 (1%), then the training is stopped.

6. Optimizing the Backpropagation architecture

1) Selection of initial weights and bias

How fast the network reaches a local or global minimum and how fast it converges are affected by the initial weights. Weights that produce a small value of the activation derivative should be avoided as much as possible because they yield very small changes in weight. On another side, the initial weight value should not be too large because the value of the derivative of the activation function resulted would be very small. Therefore, in the backpropagation standard, weights and bias are assigned with some small random numbers.

2) Then number of the hidden units

A network with one hidden screen is sufficient for backpropagation to recognize any companionship between the input and the target with a specified level of accuracy. However, increasing the number of hidden layers will make training easier.

If the network has more than one hidden layer, the predefined training algorithms need to be revised. In feedforward propagation, the output must be calculated for each layer, specifically, starting from the lowest hidden layer (the closest one to the input). Conversely, in backpropagation, the factor δ needs to be calculated for each hidden layer starting from the output layer.

C. Applying the Backpropagation method in forecasting

The backpropagation method is applicable for forecasting purposes, common examples are the forecast for rainfall, the amount of sales, stock price index, inflation, and river water flow. According to Siang (2009: 119) the problem of forecasting can be stated as follows, given a time series data $x_1, x_2, \ldots, x_n$, then x_{n+1} I estimated by using $x_1, x_2, \ldots, x_n$ known earlier.

In predicting inflation for Padang municipality using the backpropagation method, the past data on inflation in this region must be provided. In this method the past data are trained to find optimal weights. For this reason, it is necessary to determine the length of period during which the data fluctuates. This period is determined intuitively. In the case of monthly fluctuating inflation data in Padang municipality, the data period can be taken for one year because the inflation reports take place at the beginning of the oncoming year. The one-year period data is then used as the input for the backpropagation method. The next month after the last period taken can be determined as the forecast target. In the monthly data with a period of one year, the number of backpropagation input is 12, that is the inflation data from January to December in that period. The output is inflation in the following month, called the inflation forecast for Padang municipality. The tricky part is determining the number of layers as well as the hidden units, as no theory can be used with certainty to underlie it. However, practically, the smallest network is tried first (for example, it consists of 1 hidden layer with only a few units). If it fails, in the sense that the error does not decreases after numerous iterations, then the network is enlarged by adding hidden units or even adding hidden layers.

1. Normalizing the data

Normalization of the data is performed in order for the network output matches the activation function used. The activation function used in this study is the sigmoid activation function. The sigmoid function is an asymptotic function which never reaches 0 and 1. Therefore the data is transformed within the smaller intervals, namely [0.1, 0.9], as demonstrated by the following equation:

$$x' = \frac{0.8(x-a)}{b-a} + 0.1 \tag{5}$$

(Wanto, 2017:374)

Where:

x' : normalized data

x : actual data

a : maximum value of actual data

b : minimum value of actual data

2. **Forecast accuracy measure**

The forecast accuracy of the network being built is assessed with MSE (Mean Square Error). Forecast is said to be accurate if it produces a small error value and a bias that is considered is MSE. MSE is a measure of model accuracy based on the pattern of the mean squared error. MSE is defined as:

$$MSE = \frac{\sum_{k=1}^{m}(T_k - Y_k)^2}{m}$$

where T_k is the target value and Y_k is the output yielded by the network. An optimum network associates with the smallest MSE value resulted by the training process.

CHAPTER III
RESEARCH METHODOLOGY

A. Research Classification

This research is classified as applied research that begins with the analysis of relevant theories and continues with data collection and analyzing the data using methods introduced in the theories. An applied research aims at finding solutions to certain problems in order to develop practical knowledge where the solution resulted is beneficial for either individuals or group of people.

B. Data Type and Source

The data used in this study is secondary data collected from the official website of a state statistical agency, BPS-Statistics of Sumatera Barat Province, namely https://sumbar.bps.go.id/. The data consists of monthly inflation data for Padang municipality from 2013 to 2019. Padang is the capital city of Sumatera Barat province which has become one of the inflation-calculating regions in Indonesia. The monthly inflation data collected will be used as the input for the neural network to be built and has the following actual structures:

Table 3. Monthly inflation data for Padang municipality, 2013-2019

Month	Year						
	2013	2014	2015	2016	2017	2018	2019
January	1,19	1,89	-1,98	0,02	0,57	0,43	0,24
February	0,70	-0,64	-2,07	0,86	-0,13	-0,09	-0,44
March	0,31	-0,39	0,01	0,55	-0,01	0,31	0,33
April	0,58	-0,09	0,56	-0,92	-0,31	0,01	0,44
May	0,69	0,05	0,65	-0,37	-0,04	0,46	0,86
June	1,45	0,31	0,83	0,10	0,34	0,39	1,07
July	2,43	0,81	1,21	1,52	0,54	0,62	0,89

August	0,89	1,83	0,38	0,84	-0,36	-0,4	-0,1
September	-0,05	0,33	-0,49	0,58	0,13	-0,35	-0,95
October	0,77	1,18	-0,44	0,56	0,19	0,8	-0,34
November	0,46	3,44	0,47	1,13	0,48	0,19	-0,34
December	0,50	2,66	1,79	0,07	0,72	0,16	0,07

C. Data Analysis Technique

To analyze the data in this study, an artificial neural network with a learning method, namely the backpropagation method is used. The calculation is performed by using Microsoft Excel 2010™ and Matlab™ 2016b software with the following steps:

1. Collecting the inflation data.
2. Determining the activation function, which for the case of this study, bipolar sigmoid activation function is used.
3. Normalizing data into the range of 0.1-0.9
4. Determining the target by using rotary.
5. Dividing the data into two parts namely for training and for testing.
6. Calculating one by one process in each iteration of backpropagation algorithm using Microsoft Excel™ (epoch).
7. Completing the Backpropagation Method Artificial Neural Network. In this case, MATLAB™ 2016b is used to test the suitability of targets by comparing the target error (MSE) obtained using the algorithm (manual) with that of obtained from the software.
8. Obtaining the forecast for inflation rate in Padang municipality.
9. Drawing conclusion.

CHAPTER IV
RESULTS AND DISCUSSION

A. Results

1. Data description

Data on the inflation rate in Padang municipality obtained from the website of BPS-Statistics of Sumatera Barat province. The inflation rate data used in this study is monthly data, from January 2013 to December 2019 (84 periods in total). The graph for the entire data is shown in the Figure 5 below:

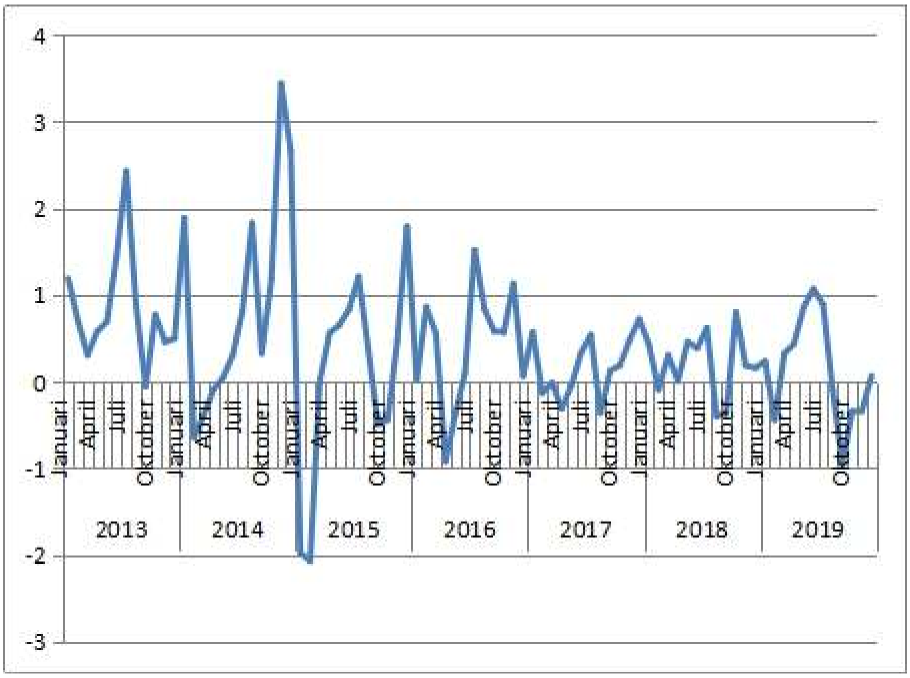

Figure 5. Inflation rate in Padang municipality, 2013-2019

Figure 5 shows that the inflation rates in Padang municipality over the recent seven years fluctuate significantly yet follow a declining trend. The lowest inflation rate is recorded in February 2015 with a value of -2.07 and the highest one of 3.44 was spotted in November 2014. In every February the inflation rate always decreases from January. In the mean time, while compared to previous months, in every May, June, July, and November the inflation rate always increases. This pattern will be learnt by the backpropagation method in training the data.

2. Data analysis

The Backpropagation Method Neural Network was used with calculation is performed using Microsoft Excel™ and Matlab™ programs. Since this method does not have special requirements for the type of data to be processed, the forecasting can be started right away. The initial step taken is to determine the activation function. According to Imelda (2014), the appropriate type of activation function to predict the inflation rate using the backpropagation method is sigmoid bipolar. The next step is to normalize the actual data in order to match the requirement of the activation function used. Since the sigmoid function is an asymptotic function (never reaches 0 or 1), the data is necessarily transformed into an interval of [0.1-0.9] (Wanto, 2017) by using transformation formula in Eq. (5). The architecture of the neural network built with the backpropagation method consists of twelve input units, ten hidden layer units, and one output unit. The next step is to determine the input pattern and target, maximum iteration (for this case 1000 iterations are used), target error, and learning rate (0.1). The final step is to complete the backpropagation algorithm so it can be used for forecasting the inflation figures.

a. Processing the data using the algorithm

Monthly inflation data for 7 consecutive years (January 2013 to December 2019) are used as the input of the backpropagation algorithm as shown in the Table 2 in the research methodology section (Chapter II). The normalized data resulted by using Equation (4) are shown in the Table 4 as follows:

Table 4. Transformed inflation rate in Padang municipality, 2013-2019

Transformed/normalized inflation data							
No	YEAR	JAN	FEB	MAR	APR	MAY	JUN
1	2013	0.5733	0.5022	0.4456	0.4848	0.5007	0.6111
2	2014	0.6750	0.3076	0.3439	0.3875	0.4078	0.4456
3	2015	0.1131	0.1000	0.4020	0.4819	0.4949	0.5211
4	2016	0.4034	0.5254	0.4804	0.2670	0.3468	0.4151
5	2017	0.4833	0.3817	0.3991	0.3555	0.3947	0.4499
6	2018	0.4630	0.3875	0.4456	0.4020	0.4673	0.4572
7	2019	0.4354	0.3367	0.4485	0.4644	0.5254	0.5559
Transformed/normalized inflation data							
No	YEAR	JUL	AUG	SEP	OCT	NOV	DEC
1	2013	0.7534	0.5298	0.3933	0.5123	0.4673	0.4731
2	2014	0.5181	0.6662	0.4485	0.5719	0.9000	0.7868
3	2015	0.5762	0.4557	0.3294	0.3367	0.4688	0.6604
4	2016	0.6212	0.5225	0.4848	0.4819	0.5646	0.4107
5	2017	0.4789	0.3483	0.4194	0.4281	0.4702	0.5051
6	2018	0.4906	0.3425	0.3497	0.5167	0.4281	0.4238
7	2019	0.5298	0.3860	0.2626	0.3512	0.3512	0.4107

After the data have been transformed, the data is divided into two parts, namely for training and for testing. Then the target of training and testing is determined in order to generate the following pattern:

Table 5. Input and target data pattern

Pattern	Data input	Target
1	Data on the 1st to 12th period	Data on the 13th period
2	Data on the 2nd to 13th period	Data on the 14th period
3	Data on the 3rd to 14th period	Data on the 15th period
⋮	⋮	⋮
72	Data on the 72nd to 83rd period	Data on the 84th period

Table 5 shows the rotation applied in the data forecasting, that is, each dataset has the equal chance to achieve the target. Pattern 1 is obtained by learning the relationship between the first input value, which consists of inflation rates from January to December 2013, and the target value, which consists of inflation rate in January 2014. Pattern 2 is obtained by learning the relationship between the second input value, which consists of inflation rates from February to December 2013, and the target value, which consists of inflation rate in February 2014. So forth, until the pattern 72 is obtained by learning the relationship between the input value consisting of inflation rates from the 72nd to 83rd period, and the target value consisting of the inflation rate in the 84th period. The generation of the pattern represents a rotation in which all data are used in the learning. Furthermore, the training process uses the 1st to 36th pattern, consisting of the inflation rates from the 1st month to the 48th month (2013 to 2016), while the testing uses the 37th to 72nd pattern, consisting of the inflation rates from the 37th to 84th month (2016-2019). The patterns of the input and target values in the training and testing processes are listed in Appendix 1.

b. Calculation in the training process using backpropagation method

According to the backpropagation algorithm, the initial step is to initialize the weights with small random numbers. Matlab™ program is used to generate some random weights as listed in Table 6 and Table 7.

Table 6. Weights (V_{ij}) and biases (V_{0j}) from input layer to hidden layer

	V_0	V_1	V_2	V_3	V_4	V_5	V_6
1	-0.4920	1.8217	0.4161	-2.2534	-0.3140	-0.3728	-0.6662
2	-0.7951	-1.5708	-0.8410	1.6584	-1.3214	1.7315	1.3939
3	4.0075	-1.1038	-0.1582	1.8199	-0.3708	-1.9787	-1.3675
4	1.1348	-1.5020	-0.5068	-0.4247	-1.8810	-1.9273	0.7861
5	2.9852	-1.2773	-1.4530	0.1220	-1.2726	-1.1412	-1.1543
6	-0.2784	1.8822	-1.2489	-0.5646	2.4028	0.8437	-0.5976
7	0.5379	0.2703	-1.3281	1.6397	2.1698	1.3220	0.7074
8	1.9639	0.4061	-1.7489	-0.0490	1.0305	0.6962	1.3886
9	2.4968	-1.6056	-1.1221	-1.8876	-2.1602	-0.3347	-2.0305
10	0.8533	1.6311	-0.2191	1.3948	-0.9252	0.3461	1.9394

	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}
1	1.1535	-0.3303	-0.9636	1.7434	-1.0765	-0.9003
2	-0.1802	1.5196	-0.2534	-0.3899	0.6185	1.5792
3	-0.3807	0.0993	-1.1744	-1.2949	0.9562	-2.0256
4	-0.4085	-0.7869	1.1035	1.3140	-1.5068	1.4489
5	-0.6883	1.5283	-0.9831	1.8130	-1.2853	0.8548
6	0.2111	1.9184	-1.2964	-0.2667	-0.9212	-0.0980
7	-0.1102	0.0500	-1.2193	-1.6744	-1.1854	0.7753
8	1.9599	0.9665	-1.5932	-1.4957	-0.0820	-1.8961
9	1.2380	0.2825	-0.4106	-0.6258	-0.2158	-0.2811

10	0.8073	-1.0394	-0.5811	0.6667	-1.5005	2.0569

Table 7. Weights (W_{jk}) and biases (W_{0k}) from hidden layer to output layer

	W0	**W1**	**W2**	**W3**	**W4**	**W5**
Y1	1.0535	0.5456	0.2237	0.0974	0.1787	-0.0708
	W6	**W7**	**W8**	**W9**	**W10**	
Y1	-0.0293	0.4247	-0.5795	-0.0685	-0.0644	

Feedforward phase

The random weights in Table 7 are then calculated for the 4th step of the algorithm which begins with adding the weights to the input signal using the data provided in Table 6:

$$Z_{in_j} = V_{0j} + \sum_{i=1}^{n} X_i V_{ij}$$

Then Z_{in_j} are evaluated in the activation function to return the output signals to be forwarded to the next layer, as expressed follows:

$$Z_j = f\left(Z_{in_j}\right) = \frac{1 - e^{-Z_{in_j}}}{1 + e^{-Z_{in_j}}}$$

The results are listed in the following table:

Table 8. Hidden layer output for 2020 forecasts

j	**January**		**February**		**March**		**April**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-0.7094	-0.3406	-1.6859	-0.6874	-1.7772	-0.7107	-0.9570	-0.4451
2	1.1052	0.5025	1.8405	0.7260	1.4712	0.6265	0.4739	0.2326
3	0.2289	0.1140	-0.4443	-0.2186	0.6705	0.3232	0.6591	0.3181
4	-1.2027	-0.5380	-0.6600	-0.3185	-2.1755	-0.7961	-1.3689	-0.5944
5	0.3210	0.1592	0.1002	0.0501	-0.1699	-0.0848	0.7145	0.3428
6	1.1961	0.5356	0.6807	0.3278	1.1414	0.5159	1.5800	0.6584
7	1.3214	0.5788	1.8204	0.7212	1.5907	0.6614	1.7809	0.7116
8	2.6535	0.8685	1.9370	0.7480	2.3375	0.8239	1.9616	0.7534
9	-1.9174	-0.7437	-2.5681	-0.8576	-2.4441	-0.8403	-2.6135	-0.8635
10	3.6523	0.9495	4.1713	0.9696	2.3439	0.8249	2.9844	0.9037
j	**May**		**June**		**July**		**August**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-2.1279	-0.7872	-1.0433	-0.4790	-0.2548	-0.1267	-1.6579	-0.6799
2	1.0174	0.4689	1.1209	0.5083	0.0834	0.0417	1.3380	0.5843

3	1.2760	0.5635	1.0040	0.4637	0.4016	0.1981	0.5709	0.2779
4	-0.9697	-0.4501	-1.6757	-0.6847	-1.5229	-0.6419	-0.6917	-0.3327
5	0.0201	0.0100	0.4276	0.2106	-0.2036	-0.1015	0.4284	0.2110
6	0.4687	0.2301	1.2371	0.5501	1.3627	0.5924	0.7095	0.3406
7	1.6556	0.6793	1.3199	0.5783	1.4819	0.6297	1.8173	0.7205
8	1.6677	0.6825	2.0496	0.7718	2.4489	0.8410	1.9034	0.7406
9	-2.6718	-0.8707	-2.0630	-0.7745	-1.9480	-0.7504	-2.4787	-0.8453
10	3.2059	0.9221	3.1320	0.9164	3.7081	0.9521	3.7210	0.9527
j	**September**		**October**		**November**		**December**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.7257	-0.6977	-1.3209	-0.5786	-1.4599	-0.6230	-1.5098	-0.6381
2	1.5260	0.6428	0.2671	0.1328	1.3144	0.5765	1.3276	0.5809
3	0.4088	0.2016	1.3273	0.5808	1.0089	0.4656	-0.2711	-0.1347
4	-1.0885	-0.4962	-1.3255	-0.5802	0.0523	0.0262	0.0552	0.0276
5	0.7339	0.3513	0.4875	0.2390	1.5631	0.6536	0.7647	0.3647
6	0.7065	0.3393	1.0016	0.4628	0.0493	0.0247	-0.0051	-0.0025
7	1.5070	0.6372	1.0707	0.4895	0.6805	0.3277	0.2541	0.1264
8	1.0160	0.4684	1.4826	0.6299	0.6745	0.3250	0.0394	0.0197
9	-1.6744	-0.6843	-2.1848	-0.7978	-1.8814	-0.7355	-1.5342	-0.6452
10	2.9547	0.9010	2.2937	0.8167	3.5518	0.9443	3.2391	0.9246

Table 9. Hidden layer output for 2021 forecasts

j	**January**		**February**		**March**		**April**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.2146	-0.5422	-0.6114	-0.2965	-0.7792	-0.3710	-1.9985	-0.7613
2	1.6531	0.6786	0.7607	0.3630	0.7335	0.3511	1.8041	0.7173
3	0.1486	0.0741	1.0545	0.4833	-0.0769	-0.0384	0.2851	0.1416
4	-1.0514	-0.4821	-0.7177	-0.3442	0.3039	0.1508	-1.2426	-0.5520
5	1.0185	0.4694	0.8960	0.4203	1.1524	0.5199	0.6432	0.3110
6	1.1699	0.5263	-0.0178	-0.0089	0.4656	0.2287	1.8949	0.7386
7	0.5856	0.2847	-0.4602	-0.2261	0.4159	0.2050	2.1803	0.7969
8	1.5091	0.6379	2.1214	0.7859	1.7732	0.7097	2.8044	0.8858
9	-1.5872	-0.6604	-1.2447	-0.5528	-1.9463	-0.7501	-1.2430	-0.5522
10	3.1276	0.9160	1.9935	0.7602	2.7626	0.8812	2.4596	0.8425
j	**May**		**June**		**July**		**August**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.5116	-0.6386	-1.4093	-0.6073	-1.7725	-0.7095	-1.7302	-0.6989
2	1.7175	0.6956	1.7538	0.7049	1.4445	0.6183	0.5703	0.2776
3	1.3951	0.6028	0.6496	0.3138	-0.6338	-0.3067	0.3230	0.1601
4	-2.2298	-0.8058	-1.0991	-0.5002	-1.4024	-0.6051	-2.5818	-0.8594
5	0.4087	0.2015	-0.4521	-0.2223	-0.8902	-0.4179	0.1707	0.0851
6	2.3862	0.8315	0.2130	0.1061	0.3340	0.1655	2.3506	0.8260
7	2.8809	0.8938	2.3648	0.8282	2.0285	0.7675	2.5549	0.8558
8	4.3658	0.9749	3.2288	0.9238	1.2242	0.5456	0.9957	0.4604
9	-1.3813	-0.5984	-2.5750	-0.8585	-3.6149	-0.9476	-3.1373	-0.9168
10	3.0891	0.9129	4.9468	0.9859	3.8903	0.9599	2.4146	0.8359
j	**September**		**October**		**November**		**December**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j

1	-2.1156	-0.7848	-1.0816	-0.4936	0.6981	0.3355	-0.1113	-0.0556
2	0.2150	0.1071	0.5492	0.2679	-0.3331	-0.1650	0.5192	0.2539
3	2.6745	0.8710	2.1524	0.7918	-0.1665	-0.0830	-0.2066	-0.1029
4	-1.3507	-0.5885	-0.0675	-0.0337	-0.6340	-0.3068	-1.0011	-0.4626
5	1.0124	0.4670	0.9741	0.4519	0.1439	0.0718	0.8347	0.3947
6	0.9226	0.4312	-0.8844	-0.4155	0.5598	0.2728	2.2622	0.8114
7	1.4767	0.6282	-0.4643	-0.2281	-0.6127	-0.2971	1.4522	0.6207
8	0.9046	0.4238	0.3968	0.1958	1.2085	0.5401	2.9607	0.9015
9	-2.6522	-0.8683	-2.0394	-0.7698	-1.3011	-0.5720	-0.9065	-0.4246
10	2.1762	0.7962	3.3456	0.9319	3.3129	0.9297	3.2416	0.9247

Table 10. Hidden layer output for 2022 forecasts

j	**January**		**February**		**March**		**April**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-2.1488	-0.7911	-2.1264	-0.7869	-1.2954	-0.5701	-1.8504	-0.7283
2	2.2633	0.8116	1.7562	0.7055	1.0887	0.4963	0.7934	0.3771
3	0.8416	0.3976	1.3791	0.5977	0.2555	0.1271	0.9294	0.4339
4	-0.2447	-0.1217	-0.7745	-0.3690	-0.5169	-0.2529	-1.1075	-0.5033
5	1.5135	0.6391	0.6473	0.3128	0.9402	0.4383	0.3829	0.1891
6	0.8291	0.3923	-0.0611	-0.0305	0.5713	0.2781	0.7640	0.3645
7	2.1134	0.7844	1.2921	0.5690	1.2650	0.5598	1.2061	0.5392
8	2.6307	0.8656	2.1057	0.7829	1.0382	0.4770	1.1403	0.5155
9	-0.6109	-0.2963	-1.4682	-0.6256	-2.2049	-0.8014	-2.1047	-0.7827
10	2.9921	0.9044	2.4217	0.8369	3.2542	0.9256	2.3905	0.8322
j	**May**		**June**		**July**		**August**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.0700	-0.4892	-0.5111	-0.2501	-1.1681	-0.5256	-1.1569	-0.5216
2	0.7131	0.3422	0.3452	0.1709	1.1201	0.5080	1.5271	0.6432
3	1.7979	0.7158	0.8435	0.3984	0.4229	0.2084	0.2563	0.1275
4	-1.2575	-0.5572	-0.3094	-0.1535	-0.7377	-0.3530	-1.2648	-0.5597
5	1.1752	0.5282	0.6633	0.3200	0.3476	0.1720	0.8860	0.4161
6	1.0829	0.4941	0.3450	0.1708	0.8998	0.4218	1.5711	0.6559
7	0.8341	0.3944	0.6069	0.2945	1.3056	0.5736	2.0007	0.7617
8	1.9747	0.7562	1.8195	0.7210	2.2936	0.8167	2.3975	0.8333
9	-1.6547	-0.6791	-1.3626	-0.5924	-1.6946	-0.6897	-1.0976	-0.4996
10	2.3315	0.8229	3.3751	0.9338	3.2962	0.9286	3.2273	0.9237
j	**September**		**October**		**November**		**December**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.7998	-0.7163	-1.9369	-0.7480	-1.1747	-0.5280	-1.1982	-0.5364
2	1.0188	0.4695	1.6436	0.6761	0.3597	0.1780	0.7479	0.3575
3	1.1166	0.5067	1.0761	0.4915	0.5999	0.2913	0.9697	0.4501
4	-0.9938	-0.4597	-0.4755	-0.2334	-0.9082	-0.4253	-0.7832	-0.3728
5	0.3622	0.1791	1.0944	0.4984	0.4720	0.2317	1.1954	0.5354
6	0.6996	0.3362	0.0117	0.0058	0.4906	0.2405	1.3100	0.5750
7	1.7977	0.7157	1.1971	0.5360	0.5620	0.2739	1.0858	0.4952
8	2.1615	0.7935	1.0031	0.4633	0.5824	0.2832	1.4097	0.6074
9	-1.9362	-0.7479	-2.2189	-0.8039	-1.8719	-0.7333	-1.9307	-0.7466
10	2.8463	0.8903	3.2022	0.9218	2.3336	0.8232	3.0428	0.9089

After the hidden layer outputs (Z_j) have been obtained, the calculation is continued to the 5th step of the algorithm to obtain the output layer outputs Y_k. The values of output signals Y_{in_k} are calculated using the following formula

$$Y_{in_k} = W_{0k} + \sum_{i=1}^{p} Z_j W_{jk}$$

The output signal values Y_k are then evaluated by the activation function to yield

$$Y_k = f\left(Y_{in_k}\right) = \frac{1 - e^{-Y_{in_k}}}{1 + e^{-Y_{in_k}}}$$

as listed in the following Table 11:

Table 11. Output layer outputs

Month	Y_{in_k} Tahun 2020	Y_k Tahun 2020	Y_{in_k} Tahun 2021	Y_k Tahun 2021	Y_{in_k} Tahun 2022	Y_k Tahun 2022
January	0.60	0.29	0.52	0.25	0.56	0.27
February	0.62	0.30	0.37	0.18	0.53	0.26
March	0.49	0.24	0.58	0.28	0.74	0.35
April	0.61	0.30	0.48	0.23	0.60	0.29
May	0.59	0.29	0.53	0.26	0.50	0.25
June	0.59	0.29	0.65	0.31	0.63	0.30
July	0.66	0.32	0.70	0.34	0.57	0.28
August	0.63	0.30	0.67	0.32	0.59	0.29
September	0.70	0.34	0.61	0.30	0.55	0.27
October	0.53	0.26	0.68	0.33	0.72	0.35
November	0.79	0.37	0.66	0.32	0.68	0.33
December	0.83	0.39	0.65	0.31	0.61	0.30

Backpropagation of error phase

In the 6th step of the algorithm, the factor δ in the output units will be calculated based on the error resulted in each output unit. This factor is later be used to calculate the term of change in weight W_{jk} using $\propto = 0.1$. By calculating $\delta_k = (T_k - Y_k)f'\left(Y_{in_k}\right) = (T_k - Y_k)Y_k(1 - Y_k)$, the following results are obtained:

Table 12. Error factor for each output unit

Month	$\boldsymbol{\delta_k}$**for 2020**	$\boldsymbol{\delta_k}$**for 2020**	$\boldsymbol{\delta_k}$**for 2020**
January	0.1226	-0.0515	0.0455
February	0.0024	-0.0351	0.0958
March	0.0385	0.0398	0.0290
April	0.0287	0.0952	-0.0077
May	0.0397	0.0838	0.0373
June	0.0518	0.0610	0.0339
July	0.0568	0.0600	0.1158
August	0.1092	0.0362	0.0763
September	0.0284	0.0101	0.0754
October	0.1109	0.0028	0.0325
November	0.1006	0.0416	0.0628
December	0.0620	0.1014	0.0352

Then the changes in weights from hidden units to output units are calculated using $\Delta W_{jk} = \alpha\delta_k Z_j$ while the biases are calculated using $\Delta W_{0k} = \alpha\delta_k$. The calculation results are listed in Appendix 2.

In the 7th step of the algorithm, the error factors of the hidden units are calculated based on the error in each hidden unit by using $\delta in_j = \sum_{k=1}^{m} \delta_k W_{jk}$ multiplied with the derivative of the activation function evaluated at Z_{in_j} to obtain the error value δ_j, expressed as $\delta_j = \delta in_j f'\left(Z_{in_j}\right) = \delta in_j Z_j\left(1 - Z_j\right)$. From the values of δ_j obtained, the changes in weights ΔV_{ij} and the changes in biases ΔV_{ij} are calculate using $\Delta V_{ij} = \alpha\delta_j X_i$ and $\Delta V_{0j} = \alpha\delta_j$, The values of δin_j and δ_j are listed in Appendix 3 and 4.

Updating the weights and biases

Update the weights and forwarded to output layer using $W_{jk}(new) = W_{jk}(old) + \Delta W_{jk}$. Also update the changes in weights and biases from input to hidden layer by using $V_{ij}(new) = V_{ij}(old) + \Delta V_{ij}$. Thus, the following new weights and biases are obtained:

Table 13. New weights (W_{jk}) and biases (W_{0k}) from input to hidden layer

	W_0	W_1	W_2	W_3	W_4	W_5
Y_1	1.0657	0.5414	0.2299	0.0001	-0.0012	-0.0001
	W_6	W_7	W_8	W_9	W_{10}	
Y_1	-0.0002	0.0030	-0.0062	0.0006	-0.0008	

Table 14. New weights (V_{ij}) and biases (V_{0j}) from hidden to output layer

	V_0	V_1	V_2	V_3	V_4	V_5	V_6
1	-0.4904	1.8227	0.4169	-2.2527	-0.3132	-0.3720	-0.6651
2	-0.7954	-1.5710	-0.8411	1.6583	-1.3216	1.7313	1.3937
3	4.0081	-1.1034	-0.1579	1.8201	-0.3705	-1.9784	-1.3672
4	1.1343	-1.5023	-0.5071	-0.4249	-1.8813	-1.9275	0.7858
5	2.9848	-1.2775	-1.4532	0.1219	-1.2728	-1.1414	-1.1545
6	-0.2783	1.8822	-1.2488	-0.5646	2.4029	0.8437	-0.5975
7	0.5359	0.2692	-1.3291	1.6388	2.1689	1.3211	0.7062
8	1.9854	0.4184	-1.7381	-0.0394	1.0409	0.7070	1.4017
9	2.4979	-1.6050	-1.1215	-1.8871	-2.1597	-0.3341	-2.0298
10	0.8582	1.6339	-0.2167	1.3970	-0.9229	0.3486	1.9423
	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	
1	1.1547	-0.3294	-0.9630	1.7442	-1.0757	-0.8995	
2	-0.1804	1.5195	-0.2535	-0.3900	0.6183	1.5791	
3	-0.3802	0.0996	-1.1742	-1.2946	0.9565	-2.0254	
4	-0.4089	-0.7871	1.1033	1.3137	-1.5071	1.4487	
5	-0.6886	1.5281	-0.9832	1.8128	-1.2855	0.8546	
6	0.2112	1.9184	-1.2963	-0.2667	-0.9212	-0.0979	
7	-0.1116	0.0489	-1.2201	-1.6754	-1.1864	0.7743	
8	1.9760	0.9778	-1.5848	-1.4847	-0.0720	-1.8860	
9	1.2388	0.2831	-0.4101	-0.6252	-0.2153	-0.2805	
10	0.8110	-1.0368	-0.5792	0.6692	-1.4982	2.0592	

The next step is testing the data using backpropagation method.

c. Manual testing on training results using backpropagation method

The new weights and biases resulted from manual training as listed in Table 11 and Table 12 will be used to test the data from 2016 to 2019. The processes performed in the testing stage are similar to those of training stage. The only aspect that distinguishes both stages is that the testing is performed until outputs Y_k are obtained.

The testing begins at accumulating all forwarded signals using $Z_{in_j} = V_{0j} + \sum_{i=1}^{n} X_i V_{ij}$, then the signal values to be forwarded to the next layer is obtained by evaluating Z_{in_j} using the activation function: $Z_j = f\left(Z_{in_j}\right) = \frac{1-e^{-Z_{in_j}}}{1+e^{-Z_{in_j}}}$. The calculation results the following hidden layer outputs:

Table 15. Hidden layer outputs for 2020 forecasts

j	January		February		March		April	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.1648	-0.5244	-0.5292	-0.2586	-1.2980	-0.5710	-1.4633	-0.6241
2	1.1203	0.5081	0.8619	0.4061	1.2979	0.5710	1.2876	0.5675
3	1.3328	0.5826	0.0040	0.0020	0.3264	0.1618	0.8075	0.3832
4	-0.5403	-0.2638	-0.2151	-0.1071	-1.0314	-0.4744	-0.9634	-0.4476
5	0.9423	0.4391	0.8329	0.3939	0.4826	0.2367	1.0986	0.5000
6	0.0562	0.0281	0.6115	0.2965	1.0085	0.4654	1.3336	0.5829
7	0.2695	0.1340	0.5414	0.2643	1.2860	0.5670	1.7287	0.6985
8	1.7666	0.7081	1.7642	0.7075	2.4188	0.8365	2.4563	0.8420
9	-0.9997	-0.4620	-1.5310	-0.6443	-1.3559	-0.5902	-1.5131	-0.6390
10	2.7589	0.8808	3.3977	0.9353	2.8047	0.8859	2.4746	0.8447
j	May		June		July		August	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.8021	-0.7168	-1.4593	-0.6229	-1.0286	-0.4733	-1.3564	-0.5903
2	1.2212	0.5446	1.1286	0.5112	0.7171	0.3439	0.7393	0.3537
3	1.3028	0.5726	0.9637	0.4477	0.6557	0.3166	0.9750	0.4522
4	-1.0038	-0.4636	-0.9916	-0.4588	-1.4763	-0.6280	-1.1152	-0.5062
5	0.4869	0.2388	0.4818	0.2364	0.1394	0.0696	0.4706	0.2310
6	0.6579	0.3176	0.7125	0.3419	1.1680	0.5256	1.0603	0.4855
7	1.7414	0.7017	1.4968	0.6342	1.6486	0.6774	1.6380	0.6745
8	2.3571	0.8270	1.9470	0.7502	2.0323	0.7683	1.9046	0.7408
9	-1.7730	-0.7097	-2.1548	-0.7922	-1.9556	-0.7521	-2.1669	-0.7945
10	2.9529	0.9008	3.0938	0.9133	3.3148	0.9299	3.1043	0.9141
j	September		October		November		December	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.3317	-0.5822	-0.9439	-0.4398	-1.2824	-0.5657	-1.0593	-0.4851
2	0.8384	0.3963	0.3496	0.1730	0.7445	0.3560	0.8252	0.3907
3	1.3833	0.5991	0.8909	0.4182	1.1963	0.5357	1.1631	0.5238
4	-1.1146	-0.5060	-0.6303	-0.3051	-0.8720	-0.4103	-0.6018	-0.2922
5	0.5983	0.2905	0.7826	0.3725	0.7798	0.3713	1.1274	0.5107
6	0.5392	0.2633	0.6958	0.3345	0.8437	0.3985	0.7082	0.3400
7	1.1936	0.5348	0.9684	0.4496	1.1953	0.5354	0.9570	0.4450
8	1.7046	0.6923	1.3047	0.5732	1.6890	0.6882	1.7030	0.6918
9	-1.7617	-0.7068	-1.7887	-0.7135	-1.4695	-0.6260	-1.1447	-0.5171
10	2.8768	0.8934	2.8273	0.8883	2.9021	0.8959	2.7433	0.8791

Table 16. Hidden layer outputs for 2021 forecasts

j	January		February		March		April	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.0893	-0.4965	-1.3348	-0.5833	-1.1344	-0.5133	-1.4401	-0.6169

2	0.8894	0.4175	1.1101	0.5043	0.8118	0.3850	0.8792	0.4133
3	0.8428	0.3981	0.8212	0.3890	0.9734	0.4516	1.0917	0.4974
4	-0.4509	-0.2217	-0.4977	-0.2438	-0.7688	-0.3665	-0.3520	-0.1742
5	0.8336	0.3942	0.9722	0.4511	1.0412	0.4782	1.1703	0.5264
6	0.4763	0.2338	0.4699	0.2307	0.7838	0.3730	0.6527	0.3152
7	0.9953	0.4603	0.9154	0.4282	0.9867	0.4569	1.0886	0.4962
8	1.7023	0.6917	1.5593	0.6525	1.7327	0.6995	1.7005	0.6912
9	-1.2678	-0.5607	-1.3705	-0.5949	-1.2039	-0.5384	-1.4115	-0.6080
10	3.1890	0.9208	2.7068	0.8749	2.4642	0.8432	2.6776	0.8714
j	**May**		**June**		**July**		**August**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.4630	-0.6240	-0.8889	-0.4173	-1.3159	-0.5770	-1.5061	-0.6370
2	1.1819	0.5306	0.8377	0.3959	1.0518	0.4823	1.2953	0.5701
3	1.2638	0.5594	0.6834	0.3290	0.7471	0.3571	0.7925	0.3768
4	-0.6855	-0.3299	-0.6627	-0.3197	-0.7717	-0.3678	-0.7129	-0.3421
5	0.9354	0.4363	0.8609	0.4057	0.6084	0.2951	0.9715	0.4508
6	0.4841	0.2374	0.8189	0.3880	0.7556	0.3608	0.7134	0.3423
7	0.9428	0.4393	1.0163	0.4685	1.3383	0.5844	1.3221	0.5791
8	2.0093	0.7635	1.9703	0.7553	2.0675	0.7754	1.7129	0.6944
9	-1.2239	-0.5455	-1.3202	-0.5784	-1.5347	-0.6454	-1.5505	-0.6500
10	2.7303	0.8776	3.0133	0.9063	3.1187	0.9153	2.8171	0.8872
j	**September**		**October**		**November**		**December**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.3108	-0.5753	-1.2008	-0.5373	-1.3513	-0.5887	-1.1413	-0.5158
2	0.7188	0.3447	0.7496	0.3582	1.0155	0.4682	0.8618	0.4061
3	1.2031	0.5382	1.1722	0.5271	0.8125	0.3853	1.1302	0.5118
4	-1.0749	-0.4910	-0.6207	-0.3007	-0.6326	-0.3062	-1.1622	-0.5235
5	0.6057	0.2939	1.0073	0.4650	0.7489	0.3579	0.5834	0.2837
6	0.7356	0.3520	0.7363	0.3524	0.6567	0.3170	0.9902	0.4583
7	1.1545	0.5207	1.1548	0.5208	1.2552	0.5564	1.2063	0.5393
8	1.8870	0.7368	1.7423	0.7019	1.5741	0.6567	2.2500	0.8093
9	-1.5917	-0.6617	-1.7433	-0.7022	-1.4297	-0.6137	-1.4980	-0.6346
10	2.4878	0.8466	2.8289	0.8884	3.1188	0.9153	2.7582	0.8807

Table 17. Hidden layer outputs for 2022 forecasts

j	**January**		**February**		**March**		**April**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-0.9364	-0.4367	-1.5008	-0.6354	-1.3354	-0.5835	-1.0693	-0.4889
2	0.8753	0.4117	0.8603	0.4055	0.9138	0.4276	0.5675	0.2764
3	0.8658	0.4077	0.7573	0.3615	1.2956	0.5702	1.1819	0.5306
4	-0.6787	-0.3269	-0.5073	-0.2483	-1.1607	-0.5229	-0.5736	-0.2792
5	0.9039	0.4235	0.6031	0.2927	0.9766	0.4529	1.0518	0.4822
6	0.6780	0.3266	0.4087	0.2016	1.0649	0.4873	0.8060	0.3825
7	1.1744	0.5279	1.1390	0.5150	1.2728	0.5624	1.2076	0.5398
8	1.9752	0.7563	1.5200	0.6411	1.9022	0.7403	1.7390	0.7011
9	-1.4502	-0.6200	-1.7126	-0.6944	-1.4483	-0.6195	-1.3053	-0.5735
10	3.2357	0.9243	2.7884	0.8841	2.3760	0.8300	2.8202	0.8875
	May		**June**		**July**		**August**	

j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.5182	-0.6405	-0.9827	-0.4553	-1.1494	-0.5188	-1.8385	-0.7255
2	1.1099	0.5042	1.1147	0.5060	0.7157	0.3433	1.4796	0.6290
3	1.2051	0.5389	0.5968	0.2899	0.5476	0.2672	1.0721	0.4900
4	-0.5083	-0.2488	-0.7557	-0.3608	-0.7104	-0.3410	-0.4151	-0.2046
5	0.7086	0.3402	0.9091	0.4256	0.6830	0.3288	1.2594	0.5579
6	0.3530	0.1747	0.6857	0.3300	0.8632	0.4067	0.3895	0.1923
7	1.1410	0.5157	0.9882	0.4575	1.2228	0.5451	1.0850	0.4949
8	1.7887	0.7135	1.6070	0.6660	1.6859	0.6874	1.3596	0.5914
9	-1.5648	-0.6541	-1.2507	-0.5548	-1.5631	-0.6536	-1.6163	-0.6686
10	3.1667	0.9191	3.0353	0.9083	2.9761	0.9030	2.8003	0.8854
j	**September**		**October**		**November**		**December**	
	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j	Z_{in_j}	Z_j
1	-1.2311	-0.5480	-0.7988	-0.3794	-1.1153	-0.5062	-1.1050	-0.5024
2	0.8064	0.3827	0.3874	0.1913	1.0615	0.4860	0.9524	0.4432
3	1.1027	0.5015	1.0735	0.4905	0.8866	0.4164	0.9519	0.4430
4	-0.6832	-0.3289	-0.6476	-0.3129	-0.4873	-0.2389	-1.0839	-0.4944
5	0.9559	0.4446	0.9246	0.4320	1.0285	0.4733	0.5178	0.2533
6	0.3075	0.1525	0.8449	0.3990	0.7732	0.3684	1.1462	0.5176
7	0.4429	0.2179	0.6660	0.3212	1.0982	0.4999	1.4952	0.6337
8	1.2999	0.5716	1.9205	0.7444	2.1335	0.7882	2.7756	0.8827
9	-1.2950	-0.5700	-1.5690	-0.6553	-1.1135	-0.5056	-1.1571	-0.5216
10	2.2688	0.8125	2.6000	0.8617	2.9378	0.8994	2.7310	0.8777

The values of Y_{in_k} are calculated to obtain the outputs Y_k by using $Y_{in_k} = W_{0k} + \sum_{i=1}^{p} Z_j W_{jk}$ and $Y_k = f\left(Y_{in_k}\right) = \frac{1-e^{-Y_{in_k}}}{1+e^{-Y_{in_k}}}$. The following results are obtained:

Table 18. Output unit outputs for all forecasts

Month	Y_{in_k} for 2020	Y_k for 2020	Y_{in_k} for 2021	Y_k for 2021	Y_{in_k} for 2022	Y_k for 2022
January	0.89	0.42	0.87	0.41	0.91	0.43
February	1.00	0.46	0.84	0.40	0.81	0.39
March	0.88	0.41	0.86	0.41	0.83	0.39
April	0.84	0.40	0.82	0.39	0.85	0.40
May	0.79	0.38	0.84	0.40	0.82	0.39
June	0.83	0.39	0.92	0.43	0.92	0.43
July	0.88	0.41	0.86	0.40	0.86	0.41
August	0.83	0.39	0.84	0.40	0.81	0.38
September	0.83	0.39	0.82	0.39	0.85	0.40
October	0.86	0.41	0.84	0.40	0.89	0.42
November	0.84	0.40	0.84	0.40	0.89	0.42
December	0.89	0.42	0.87	0.41	0.88	0.41

The output values Y_k resulted in the algorithm are substituted into Eq. 4 in which Y_k substitute x' then transformed into x which is the actual forecast values. The following transformed forecast values are obtained:

Table 19. Forecasts with single iteration (epoch)

Month	2020	2021	2022
January	0.13	0.06	0.20
February	0.44	-0.01	-0.09
March	0.09	0.04	-0.04
April	-0.01	-0.05	0.007
May	-0.16	0.002	-0.08
June	-0.04	0.22	0.21
July	0.08	0.03	0.04
August	-0.05	-0.02	-0.11
September	-0.05	-0.08	0.02
October	0.05	-0.02	0.14
November	-0.02	-0.03	0.13
December	0.11	0.06	0.10

All the steps of the backpropagation algorithm above have only been performed using single iteration. The number of iterations is increased until the smallest MSE or the smallest error value is obtained. For performing numerous iterations, the algorithm is implemented by using Matlab™ commands in order to yield the desired forecast result. The result also shows a MSE value graph in addition to forecast values. The process using Matlab™ is shown in the Figure 6.

From the architecture model of 12-10-1 as shown by Figure 6, the iterations of 1000 times can be performed within 8 seconds. The MSE graph for the forecast results (the inflation rates for the next three years) is shown by Figure 7. Take note that in Figure 6, the MSE value is represented by ‘Performance’ parameter.

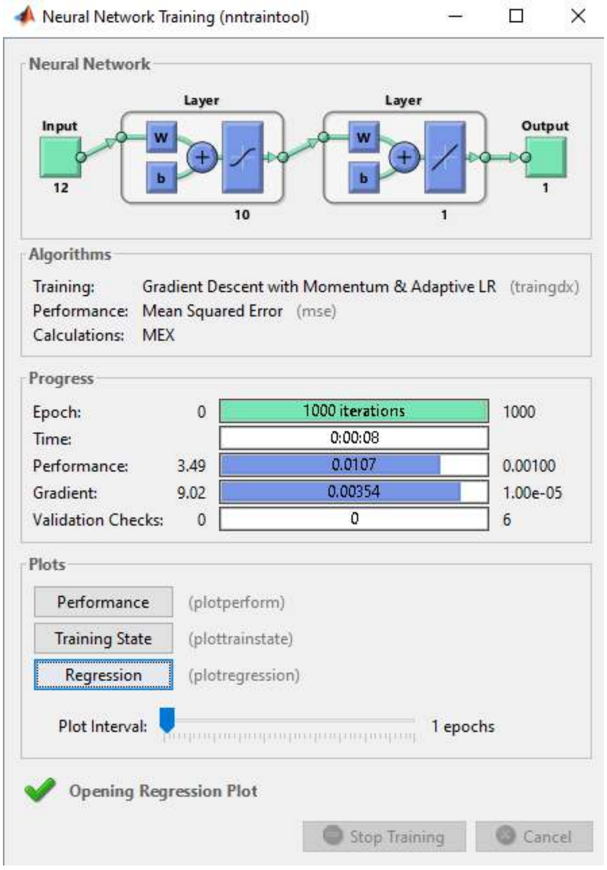

Figure 6. Training result using backpropagation method implemented in Matlab™ 2016b

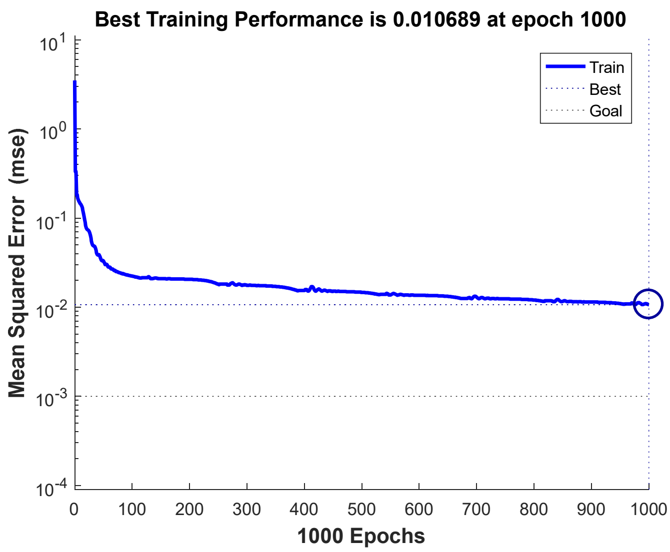

Figure 7. MSE values for monthly inflation forecasts from 2020 to 2022

In the Figure 7, it is shown that the MSE value has reached 0.010689 when the 1000th epoch (iteration) stops with a tolerance limit of 10^{-2} which is greater than the specified tolerance limit (10^{-4}). With the MSE value obtained (as listed at the top of the chart in Fig. 7) the iteration process is stopped automatically because the desired target is considered to be reached. The value of correlation coefficient R is 0.75541 as shown in the following figure

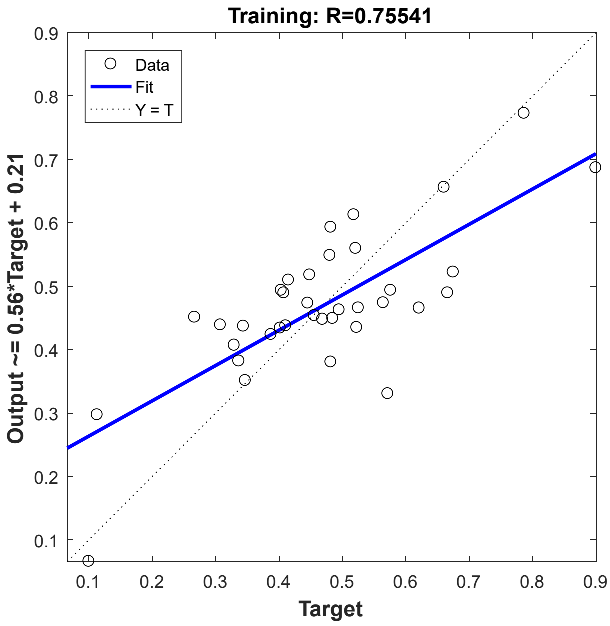

Figure 8. Correlation coefficient

The main output, the forecasts for the inflation rate in Padang municipality, is obtained as shown below:

Table 20. Matlab™ forecast results for inflation rates in Padang municipality

Month	2020	2021	2022
January	0.72	0.67	0.89
February	-0.76	-0.49	-0.47
March	0.27	0.58	1.00
April	0.72	0.25	1.12
May	1.49	0.89	0.69
June	0.46	0.43	0.34
July	1.90	0.51	0.48
August	0.63	0.01	0.11
September	1.62	1.06	0.39
October	0.47	0.57	0.50
November	1.53	1.15	0.38
December	0.62	0.90	1.03

The comparison between the forecasts and target on the inflation rates for Padang municipality in 2017-2019 is shown in Figure 9 below:

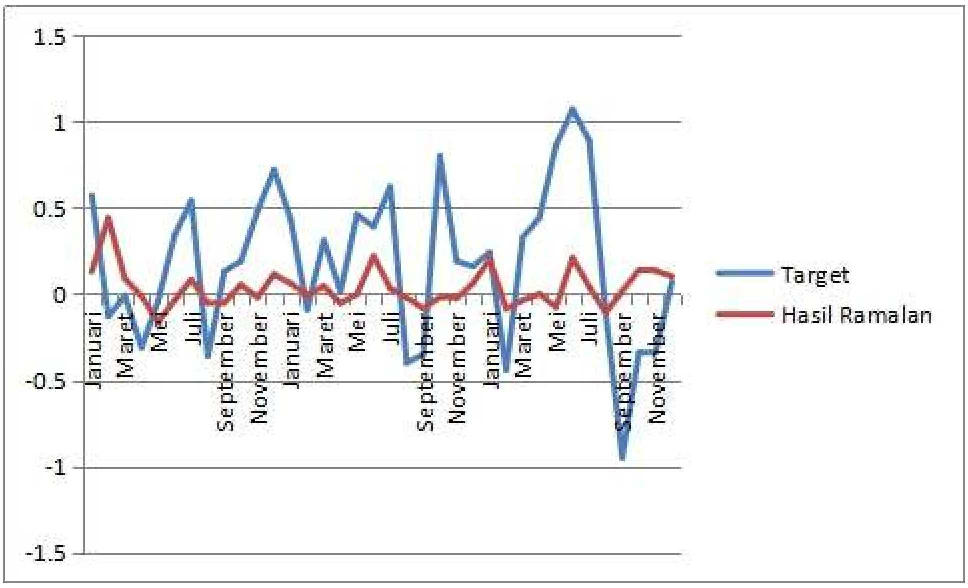

Figure 9. Forecasts versus target for inflation rates in 2017-2019

B. Discussion

Based on the analysis results, the monthly inflation forecasts for Padang municipality for the next three years have been obtained. The analysis is performed using the steps that begins at describing the characteristics of the research data, determining the activation function to be applied, transforming the data using equation 4 and dividing the transformed data for training and testing. The next steps are performing manual calculations on the algorithm steps using Microsoft Excel and automatic calculations using the Matlab™ 2016b program.

In the programming, the first step taken is making network initialization. First the desired parameter values are determined in order to obtain optimal results. The preferred network architecture used in this study consists of 12 input layers, 10 hidden layers and 1 output layer. The parameters used in executing Matlab™ command for the backpropagation algorithm are as follows.

```
net.performFcn = 'mse';
net.trainParam.goal = 0.001;
net.trainParam.show = 20;
net.trainParam.epochs = 1000;
net.trainParam.mc = 0.95;
net.trainParam.lr = 0.1;
```

In both processes, the parameter $\alpha = 0.1$ and the equal initial weights and bias are used. The iteration stopped automatically when the expected tolerance limit was reached. The results are obtained with a MSE value of 0.010689 is reached at the 1000th iteration (epoch). The resulting MSE value differs from that assigned in the syntax since at the 1000th iteration the MSE resulted is considered to be very small that even if the iteration is continued, a relatively equally small MSE value will be obtained, thus there will be no significant improvement in MSE no matter large the iteration number is increased. Therefore, the program decides to automatically stop at the 1000th iteration with a tolerance limit of 10^{-2}

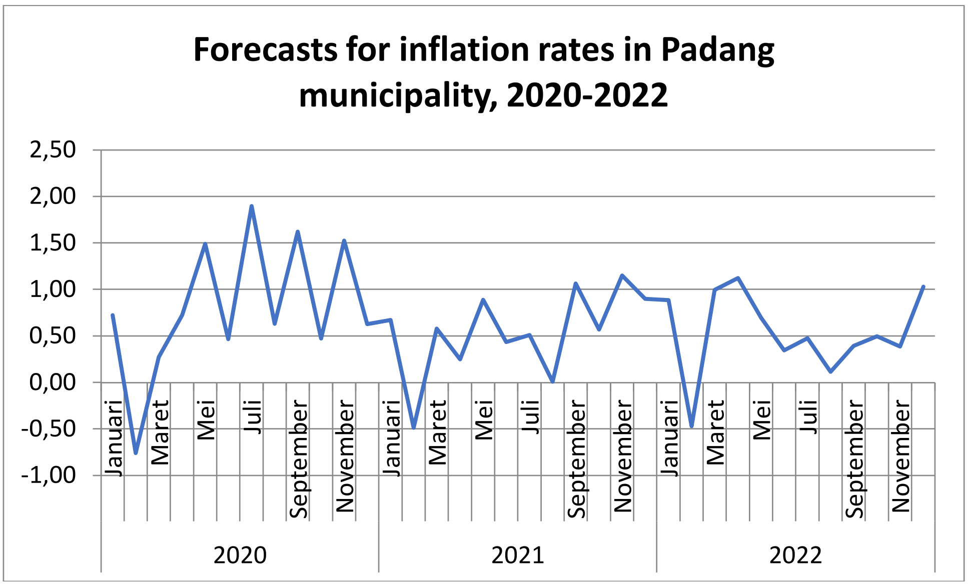

Figure 10. Forecasts for inflation rates in Padang municipality, 2020-2022

Figure 10 shows that the forecasts for the inflation rates in Padang municipality follow an increasing tendency, even though some decreases also occur. However, compared to the inflation rate in previous years, the forecast results seem to be more stable and standardized. More specifically, the inflation rate starts at a relatively high value in the year beginning then decreases in February but increases again until July. It declines again until September and increases during the last three months of the year. This fluctuation is caused by several factors.

However, technically from the methodology point of view, it can be seen that the Backpropagation Method can maintain the intrinsic properties (true value) of forecasting. The smaller the MSE value, the better the forecasting results. In this study the MSE value obtained is 0.010689 which is smaller than the MSE target inputted in the Matlab™ command. This result confirms that resulted in Amrin (2014) in which the usage of the backpropagation method of neural networks to predict the monthly inflation rate in Indonesia also exposed the maintaining of the intrinsic properties. Amrin's work resulted in a fairly good level of prediction accuracy based on the MSE value of 0.0171 with a correlation coefficient of 0.75541. According to Sarwono (2006), a correlation value within ranging from 0.75 to 0.99 is considered to be very strong.

From the results discussed it can be concluded that the forecasts for the inflation rates in Padang municipality using the Backpropagation Method Artificial Neural Network for 2020-2022 are desirable and accurate.

CHAPTER V
SUMMARY

A. Conclusion

The analysis of forecasting the inflation rates for Padang municipality using the backpropagation method has been implemented. Based on the results discussed in the previous chapter, it is concluded that the forecasts of the inflation rates in Padang municipality in 2020 to 2022 fluctuate monthly. The lowest inflation rate is predicted to occur in February 2020 as of -0.76 while the highest one occurs in July 2020. However the data fluctuates, it also can be observed that the forecasts for February, June, and August tend to decrease while the forecasts for March, July, September tend to increase. This pattern is resulted in by the backpropagation algorithm that is capable of analyzing input data patterns sourced from the inflation rate data in previous years.

The use of the Backpropagation method in this study with parameters and a bipolar sigmoid activation function as well as a learning rate of 0.1 in fact can produce a fairly good forecast with an error rate represented by a MSE value of 0.010689. This suggests that the Backpropagation method can be an appropriate alternative method in predicting the inflation rate in Padang municipality.

B. Recommendation

Based on the conclusion drawn above, the following recommendations are suggested

1. The network architecture used in this study is not yet optimum where there are still many other possible combinations can be used, and other factors to adjust including data input patterns, the number of neurons in the hidden

layer, the maximum number of epochs (iterations) and the target MSE value. The architecture used in this study is considered the best to implement based on ‘trial and error’ process have been done while there are many other possible ways to find the best architecture.

2. Further researches on the inflation rate in Padang municipality necessarily need to involve some other variables that may affect the inflation rate so that further research can produce more accurate forecasts.
3. The forecasts may be useful for both public and private stakeholders to anticipate the negative impact of socio-economic conditions during inflation.

BIBLIOGRAPHY

Amrin. (2014). Peramalan Tingkat Inflasi Indonesia Menggunakan Neural Network Backpropagation Berbasis Metode Time Series. *Jurnal Techno Nusa Mandiri.*

Badan Pusat Statistik Sumatera Barat. (2020). *Inflasi Kota Padang, Kota Bukittinggi dan Gabungan 2 (dua) Padang dan Bukittinggi, 2014-2019.* Diunduh 20 January 2020, http://sumbar.bps.go.id

Bank Indonesia. (2020). *Pengenalan Inflasi.* Diakses 20 January 2020, http://www.bi.go.id

Desiani, A. (2006). *Konsep Kecerdasan Buatan.* Yogyakarta: Andi Offset.

Fausett, L. (1994). *Fundamental of Neural Networks: Architectures, Algorithms, and Applications.* New Jersey: Prentice Hall.

Jonathan, Sarwono. (2006). *Metode Penelitian Kuantitatif dan Kualitatif* . Yogyakarta: Graha Ilmu.

Khalwaty, T. (2000). *Inflasi dan Solusinya.* Jakarta: Gramedia Pustaka Utama.

Nugraha, D. A. (2008). Software Pembelajaran Neural Networks dengan Algoritma Backpropagation. *Proyek Akhir ITS.*

Sawitri, M. N., Sumarjaya, I. W., & Tastrawati, N. K. (2018). Peramalan Menggunakan Metode Backpropagation Neural Network. *E-Jurnal Matematika Vol. 7(3)*, 264-270.

Siang, J. J. (2009). *Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan MATLAB.* Yogyakarta: Andi Offset.

Sinungan, M. (1995). *Uang dan Bank.* Jakarta: Rineka Cipta.

Suparti. (2013). Analisis Data Inflasi di Indonesia Menggunakan Model Regresi Spline. *Media Statistika, Vol.6, No. 1*, 1-9.

Wanto, Anjar. (2018). Optimasi Prediksi Dengan Algoritma Backpropagation Dan Conjugate Gradient Beale-Powell Restars. *Jurnal Nasional Teknologi dan Sistem Informasi-Vol.03 (2017)* 370-380.

APPENDIX

Appendix 1. Pattern of input and target data

Pattern	X1	X2	X3	...	X10	X11	X12	Training target
1	0.5733	0.5022	0.4456	...	0.5123	0.4673	0.4731	0.6750
2	0.5022	0.4456	0.4848	...	0.4673	0.4731	0.6750	0.3076
3	0.4456	0.4848	0.5007	...	0.4731	0.6750	0.3076	0.3439
4	0.4848	0.5007	0.6111	...	0.6750	0.3076	0.3439	0.3875
5	0.5007	0.6111	0.7534	...	0.3076	0.3439	0.3875	0.4078
6	0.6111	0.7534	0.5298	...	0.3439	0.3875	0.4078	0.4456
7	0.7534	0.5298	0.3933	...	0.3875	0.4078	0.4456	0.5181
8	0.5298	0.3933	0.5123	...	0.4078	0.4456	0.5181	0.6662
9	0.3933	0.5123	0.4673	...	0.4456	0.5181	0.6662	0.4485
10	0.5123	0.4673	0.4731	...	0.5181	0.6662	0.4485	0.5719
11	0.4673	0.4731	0.6750	...	0.6662	0.4485	0.5719	0.9000
12	0.4731	0.6750	0.3076	...	0.4485	0.5719	0.9000	0.7868
13	0.6750	0.3076	0.3439	...	0.5719	0.9000	0.7868	0.1131
14	0.3076	0.3439	0.3875	...	0.9000	0.7868	0.1131	0.1000
15	0.3439	0.3875	0.4078	...	0.7868	0.1131	0.1000	0.4020
16	0.3875	0.4078	0.4456	...	0.1131	0.1000	0.4020	0.4819
17	0.4078	0.4456	0.5181	...	0.1000	0.4020	0.4819	0.4949
18	0.4456	0.5181	0.6662	...	0.4020	0.4819	0.4949	0.5211
19	0.5181	0.6662	0.4485	...	0.4819	0.4949	0.5211	0.5762
20	0.6662	0.4485	0.5719	...	0.4949	0.5211	0.5762	0.4557
21	0.4485	0.5719	0.9000	...	0.5211	0.5762	0.4557	0.3294
22	0.5719	0.9000	0.7868	...	0.5762	0.4557	0.3294	0.3367
23	0.9000	0.7868	0.1131	...	0.4557	0.3294	0.3367	0.4688
24	0.7868	0.1131	0.1000	...	0.3294	0.3367	0.4688	0.6604
25	0.1131	0.1000	0.4020	...	0.3367	0.4688	0.6604	0.4034
26	0.1000	0.4020	0.4819	...	0.4688	0.6604	0.4034	0.5254
27	0.4020	0.4819	0.4949	...	0.6604	0.4034	0.5254	0.4804
28	0.4819	0.4949	0.5211	...	0.4034	0.5254	0.4804	0.2670
29	0.4949	0.5211	0.5762	...	0.5254	0.4804	0.2670	0.3468
30	0.5211	0.5762	0.4557	...	0.4804	0.2670	0.3468	0.4151
31	0.5762	0.4557	0.3294	...	0.2670	0.3468	0.4151	0.6212
32	0.4557	0.3294	0.3367	...	0.3468	0.4151	0.6212	0.5225
33	0.3294	0.3367	0.4688	...	0.4151	0.6212	0.5225	0.4848
34	0.3367	0.4688	0.6604	...	0.6212	0.5225	0.4848	0.4819
35	0.4688	0.6604	0.4034	...	0.5225	0.4848	0.4819	0.5646

36	0.6604	0.4034	0.5254	...	0.4848	0.4819	0.5646	0.4107
37	0.4034	0.5254	0.4804	...	0.4819	0.5646	0.4107	0.4833
38	0.5254	0.4804	0.2670	...	0.5646	0.4107	0.4833	0.3817
39	0.4804	0.2670	0.3468	...	0.4107	0.4833	0.3817	0.3991
40	0.2670	0.3468	0.4151	...	0.4833	0.3817	0.3991	0.3555
41	0.3468	0.4151	0.6212	...	0.3817	0.3991	0.3555	0.3947
42	0.4151	0.6212	0.5225	...	0.3991	0.3555	0.3947	0.4499
43	0.6212	0.5225	0.4848	...	0.3555	0.3947	0.4499	0.4789
44	0.5225	0.4848	0.4819	...	0.3947	0.4499	0.4789	0.3483
45	0.4848	0.4819	0.5646	...	0.4499	0.4789	0.3483	0.4194
46	0.4819	0.5646	0.4107	...	0.4789	0.3483	0.4194	0.4281
47	0.5646	0.4107	0.4833	...	0.3483	0.4194	0.4281	0.4702
48	0.4107	0.4833	0.3817	...	0.4194	0.4281	0.4702	0.5051
49	0.4833	0.3817	0.3991	...	0.4281	0.4702	0.5051	0.4630
50	0.3817	0.3991	0.3555	...	0.4702	0.5051	0.4630	0.3875
51	0.3991	0.3555	0.3947	...	0.5051	0.4630	0.3875	0.4456
52	0.3555	0.3947	0.4499	...	0.4630	0.3875	0.4456	0.4020
53	0.3947	0.4499	0.4789	...	0.3875	0.4456	0.4020	0.4673
54	0.4499	0.4789	0.3483	...	0.4456	0.4020	0.4673	0.4572
55	0.4789	0.3483	0.4194	...	0.4020	0.4673	0.4572	0.4906
56	0.3483	0.4194	0.4281	...	0.4673	0.4572	0.4906	0.3425
57	0.4194	0.4281	0.4702	...	0.4572	0.4906	0.3425	0.3497
58	0.4281	0.4702	0.5051	...	0.4906	0.3425	0.3497	0.5167
59	0.4702	0.5051	0.4630	...	0.3425	0.3497	0.5167	0.4281
60	0.5051	0.4630	0.3875	...	0.3497	0.5167	0.4281	0.4238
61	0.4630	0.3875	0.4456	...	0.5167	0.4281	0.4238	0.4354
62	0.3875	0.4456	0.4020	...	0.4281	0.4238	0.4354	0.3367
63	0.4456	0.4020	0.4673	...	0.4238	0.4354	0.3367	0.4485
64	0.4020	0.4673	0.4572	...	0.4354	0.3367	0.4485	0.4644
65	0.4673	0.4572	0.4906	...	0.3367	0.4485	0.4644	0.5254
66	0.4572	0.4906	0.3425	...	0.4485	0.4644	0.5254	0.5559
67	0.4906	0.3425	0.3497	...	0.4644	0.5254	0.5559	0.5298
68	0.3425	0.3497	0.5167	...	0.5254	0.5559	0.5298	0.3860
69	0.3497	0.5167	0.4281	...	0.5559	0.5298	0.3860	0.2626
70	0.5167	0.4281	0.4238	...	0.5298	0.3860	0.2626	0.3512
71	0.4281	0.4238	0.4354	...	0.3860	0.2626	0.3512	0.3512
72	0.4238	0.4354	0.3367	...	0.2626	0.3512	0.3512	0.4107

Appendix 2. Changes in weights from hidden units to output units

Pattern	$W_{0.1}$	$W_{1.1}$	$W_{2.1}$	…	$W_{9.1}$	$W_{10.1}$
1	1.06573142	0.541411563	0.229859146	…	0.000624584	-0.000750031
2	1.053710969	0.545421299	0.223873197	…	1.42147E-05	-1.51178E-05
3	1.057321291	0.542849759	0.226110925	…	0.00022169	-0.000204716
4	1.056336182	0.544311565	0.22436449	…	0.000169557	-0.000166929
5	1.057435972	0.54246493	0.225557694	…	0.000236558	-0.000235655
6	1.058645954	0.543108086	0.22632905	…	0.000274611	-0.000305624
7	1.05914871	0.544867911	0.223934129	…	0.00029192	-0.00034838
8	1.064389393	0.538162717	0.230078683	…	0.000632188	-0.000670251
9	1.056312394	0.543603715	0.22552534	…	0.000133268	-0.000165039
10	1.064561774	0.539168828	0.225170426	…	0.000606068	-0.000583633
11	1.063526052	0.539321772	0.229495347	…	0.00050663	-0.000611772
12	1.059668744	0.541631715	0.227298766	…	0.000273975	-0.000369268
13	1.048323366	0.548377779	0.220205662	…	-0.000232747	0.00030365
14	1.049955923	0.546629299	0.222422238	…	-0.000132995	0.000172047
15	1.057453132	0.544109547	0.225096482	…	0.000204674	-0.000226185
16	1.062987337	0.538341441	0.230524951	…	0.000359955	-0.000516622
17	1.061848719	0.540236436	0.229526569	…	0.000343419	-0.000492807
18	1.059572553	0.541880834	0.227999764	…	0.00035886	-0.000387653
19	1.05946701	0.541331824	0.227406163	…	0.000389257	-0.000370924
20	1.057090692	0.543056476	0.224703057	…	0.000227411	-0.000195021
21	1.054478957	0.544794984	0.223805624	…	6.00626E-05	-5.18045E-05
22	1.053744565	0.545451616	0.223771321	…	1.45305E-05	-1.65471E-05
23	1.057625243	0.546982252	0.223011629	…	0.000162831	-0.000248941
24	1.063607002	0.545024234	0.226271676	…	0.000294794	-0.000603952
25	1.058015512	0.541990817	0.227387374	…	9.2266E-05	-0.000264905
26	1.063052879	0.538046243	0.230458714	…	0.000410615	-0.000516727
27	1.056366924	0.543935451	0.22513562	…	0.000159055	-0.00017281
28	1.052703433	0.546145211	0.223408798	…	-4.10367E-05	4.10399E-05
29	1.057195988	0.54376447	0.224972833	…	0.000173334	-0.000197582
30	1.056859247	0.544739649	0.224276853	…	0.000137543	-0.000203956
31	1.06504509	0.539503162	0.229578273	…	0.000546784	-0.000692498
32	1.061101279	0.541606991	0.228606252	…	0.000261166	-0.000454167
33	1.06100778	0.540187922	0.227236805	…	0.00038614	-0.000432364
34	1.056716391	0.543158472	0.225892926	…	0.00017879	-0.000192853
35	1.059745152	0.542273888	0.224814354	…	0.000315224	-0.000332851
36	1.056986538	0.543700735	0.224954889	…	0.000179875	-0.000205973

Appendix 3. Error factors in hidden units (δin_j)

Pattern	δin_1	δin_2	δin_3	δin_4	δin_5	δin_6	δin_7	δin_8	δin_9	δin_{10}
1	0.0669	0.0274	0.0119	0.0219	-0.0087	-0.0036	0.0521	-0.0711	-0.0084	-0.0079
2	0.0013	0.0005	0.0002	0.0004	-0.0002	-0.0001	0.0010	-0.0014	-0.0002	-0.0002
3	0.0210	0.0086	0.0038	0.0069	-0.0027	-0.0011	0.0164	-0.0223	-0.0026	-0.0025
4	0.0156	0.0064	0.0028	0.0051	-0.0020	-0.0008	0.0122	-0.0166	-0.0020	-0.0018
5	0.0216	0.0089	0.0039	0.0071	-0.0028	-0.0012	0.0168	-0.0230	-0.0027	-0.0026
6	0.0282	0.0116	0.0050	0.0093	-0.0037	-0.0015	0.0220	-0.0300	-0.0035	-0.0033
7	0.0310	0.0127	0.0055	0.0102	-0.0040	-0.0017	0.0241	-0.0329	-0.0039	-0.0037
8	0.0596	0.0244	0.0106	0.0195	-0.0077	-0.0032	0.0464	-0.0633	-0.0075	-0.0070
9	0.0155	0.0064	0.0028	0.0051	-0.0020	-0.0008	0.0121	-0.0165	-0.0019	-0.0018
10	0.0605	0.0248	0.0108	0.0198	-0.0079	-0.0032	0.0471	-0.0643	-0.0076	-0.0071
11	0.0549	0.0225	0.0098	0.0180	-0.0071	-0.0029	0.0427	-0.0583	-0.0069	-0.0065
12	0.0338	0.0139	0.0060	0.0111	-0.0044	-0.0018	0.0263	-0.0359	-0.0042	-0.0040
13	-0.0281	-0.0115	-0.0050	-0.0092	0.0036	0.0015	-0.0219	0.0298	0.0035	0.0033
14	-0.0192	-0.0079	-0.0034	-0.0063	0.0025	0.0010	-0.0149	0.0204	0.0024	0.0023
15	0.0217	0.0089	0.0039	0.0071	-0.0028	-0.0012	0.0169	-0.0231	-0.0027	-0.0026
16	0.0519	0.0213	0.0093	0.0170	-0.0067	-0.0028	0.0404	-0.0552	-0.0065	-0.0061
17	0.0457	0.0187	0.0082	0.0150	-0.0059	-0.0025	0.0356	-0.0486	-0.0057	-0.0054
18	0.0333	0.0137	0.0059	0.0109	-0.0043	-0.0018	0.0259	-0.0354	-0.0042	-0.0039
19	0.0327	0.0134	0.0058	0.0107	-0.0042	-0.0018	0.0255	-0.0348	-0.0041	-0.0039
20	0.0198	0.0081	0.0035	0.0065	-0.0026	-0.0011	0.0154	-0.0210	-0.0025	-0.0023
21	0.0055	0.0023	0.0010	0.0018	-0.0007	-0.0003	0.0043	-0.0059	-0.0007	-0.0007
22	0.0015	0.0006	0.0003	0.0005	-0.0002	-0.0001	0.0012	-0.0016	-0.0002	-0.0002
23	0.0227	0.0093	0.0040	0.0074	-0.0029	-0.0012	0.0177	-0.0241	-0.0028	-0.0027
24	0.0553	0.0227	0.0099	0.0181	-0.0072	-0.0030	0.0431	-0.0588	-0.0069	-0.0065
25	0.0248	0.0102	0.0044	0.0081	-0.0032	-0.0013	0.0193	-0.0263	-0.0031	-0.0029
26	0.0523	0.0214	0.0093	0.0171	-0.0068	-0.0028	0.0407	-0.0555	-0.0066	-0.0062
27	0.0158	0.0065	0.0028	0.0052	-0.0021	-0.0008	0.0123	-0.0168	-0.0020	-0.0019
28	-0.0042	-0.0017	-0.0007	-0.0014	0.0005	0.0002	-0.0033	0.0044	0.0005	0.0005
29	0.0203	0.0083	0.0036	0.0067	-0.0026	-0.0011	0.0158	-0.0216	-0.0026	-0.0024
30	0.0185	0.0076	0.0033	0.0061	-0.0024	-0.0010	0.0144	-0.0196	-0.0023	-0.0022
31	0.0632	0.0259	0.0113	0.0207	-0.0082	-0.0034	0.0492	-0.0671	-0.0079	-0.0075
32	0.0416	0.0171	0.0074	0.0136	-0.0054	-0.0022	0.0324	-0.0442	-0.0052	-0.0049
33	0.0411	0.0169	0.0073	0.0135	-0.0053	-0.0022	0.0320	-0.0437	-0.0052	-0.0049
34	0.0177	0.0073	0.0032	0.0058	-0.0023	-0.0010	0.0138	-0.0188	-0.0022	-0.0021
35	0.0342	0.0140	0.0061	0.0112	-0.0044	-0.0018	0.0267	-0.0364	-0.0043	-0.0040
36	0.0192	0.0079	0.0034	0.0063	-0.0025	-0.0010	0.0149	-0.0204	-0.0024	-0.0023

Appendix 4. Error values in hidden units (δin_j)

Pattern	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8	δ_9	δ_{10}
1	0.0166	-0.0030	0.0057	-0.0049	-0.0039	0.0008	-0.0194	0.2147	0.0112	0.0487
2	-0.0012	-0.0006	0.0001	0.0001	-0.0001	0.0000	-0.0012	0.0019	0.0005	0.0013
3	-0.0227	-0.0050	0.0010	-0.0128	-0.0013	0.0002	-0.0125	0.0498	0.0066	0.0056
4	0.0007	0.0025	0.0008	-0.0022	-0.0005	0.0006	-0.0132	0.0237	0.0057	0.0073
5	-0.0382	-0.0002	-0.0012	0.0002	-0.0014	-0.0005	-0.0147	0.0205	0.0083	0.0119
6	-0.0012	-0.0015	0.0000	-0.0084	-0.0015	0.0004	-0.0082	0.0480	0.0058	0.0147
7	0.0145	0.0063	0.0023	-0.0067	-0.0019	0.0007	-0.0144	0.0822	0.0054	0.0233
8	-0.0521	-0.0097	0.0036	0.0051	-0.0032	-0.0008	-0.0534	0.0830	0.0192	0.0452
9	-0.0153	-0.0042	0.0012	-0.0005	-0.0005	-0.0002	-0.0077	0.0003	0.0018	0.0071
10	-0.0225	0.0115	-0.0041	-0.0075	-0.0030	0.0000	-0.0034	0.0385	0.0143	0.0152
11	-0.0310	-0.0082	-0.0001	0.0090	0.0051	-0.0015	0.0115	-0.0159	0.0087	0.0376
12	-0.0216	-0.0053	0.0028	0.0055	-0.0009	-0.0009	0.0123	-0.0179	0.0029	0.0190
13	0.0067	0.0100	-0.0025	0.0005	-0.0001	-0.0003	-0.0072	-0.0190	-0.0027	-0.0146
14	-0.0060	-0.0017	0.0002	-0.0015	0.0002	0.0005	-0.0059	-0.0356	-0.0007	-0.0034
15	0.0043	0.0021	0.0019	0.0032	0.0005	-0.0005	0.0070	0.0248	0.0038	0.0085
16	-0.0777	-0.0240	0.0043	-0.0046	-0.0020	0.0036	-0.0759	0.1893	0.0018	0.0155
17	-0.0294	-0.0183	-0.0039	-0.0297	-0.0025	0.0058	-0.1299	0.4385	0.0026	0.0231
18	-0.0164	-0.0142	0.0017	-0.0011	-0.0017	-0.0009	-0.0595	0.1667	0.0118	0.0461
19	-0.0350	-0.0073	0.0017	-0.0052	-0.0004	-0.0008	-0.0397	0.0087	0.0248	0.0273
20	-0.0197	0.0027	0.0016	-0.0183	-0.0012	0.0024	-0.0425	-0.0001	0.0110	0.0056
21	-0.0096	0.0011	-0.0030	-0.0007	0.0000	0.0000	-0.0025	-0.0005	0.0021	0.0012
22	-0.0001	0.0002	-0.0005	0.0002	0.0000	0.0000	0.0005	-0.0007	0.0003	0.0009
23	0.0058	0.0041	0.0020	0.0022	-0.0014	-0.0004	0.0055	0.0055	0.0010	0.0134
24	0.0273	0.0083	0.0047	0.0000	-0.0011	0.0061	-0.0239	0.2281	-0.0006	0.0310
25	-0.0449	-0.0210	0.0006	0.0038	0.0021	-0.0002	-0.0335	0.0780	-0.0010	0.0116
26	-0.0921	-0.0223	-0.0042	0.0034	-0.0020	-0.0014	-0.0136	0.0954	0.0038	0.0150
27	-0.0054	-0.0006	0.0013	0.0019	-0.0001	-0.0003	-0.0037	0.0007	0.0038	0.0090
28	0.0051	-0.0003	-0.0001	0.0002	0.0002	0.0000	0.0007	-0.0007	-0.0009	-0.0012
29	-0.0015	0.0020	-0.0041	-0.0019	0.0005	0.0001	0.0024	0.0313	0.0022	0.0053
30	0.0068	0.0033	0.0005	0.0027	-0.0007	-0.0004	0.0045	0.0227	0.0010	0.0113
31	-0.0115	-0.0033	0.0046	0.0047	-0.0036	-0.0003	-0.0173	0.1429	0.0074	0.0368
32	-0.0070	-0.0114	0.0035	-0.0041	-0.0006	0.0016	-0.0487	0.1050	0.0005	0.0231
33	-0.0461	-0.0003	-0.0009	0.0001	-0.0023	-0.0006	-0.0357	0.0802	0.0071	0.0172
34	-0.0244	-0.0062	-0.0003	0.0022	0.0002	-0.0005	-0.0030	0.0001	0.0044	0.0097
35	-0.0065	0.0061	0.0020	0.0010	-0.0017	-0.0007	0.0091	-0.0120	0.0054	0.0090
36	-0.0042	0.0017	0.0001	0.0012	0.0005	0.0004	-0.0013	0.0101	0.0033	0.0094