THE COMPLETION OF THE GAME RUBIK'S CUBE USING NEURAL NETWORK BACKPROPAGATION

Taufik Setiyadi

Computer Engineering Informatics University Of Indonesia
Jl.Dipatiukur 112-114 Bandung
E-mail : Taufik.setiyadi@gmail.com

ABSTRACT

Rubik's CUBE is a mechanical puzzle game three dimensions found in 1974 by a sculptor and Professor of architecture from Hungary named Erno Rubik. The first Rubik's CUBE 3x3x3 cube sized small. each small cube side facing out to have different colors. Rubik's CUBE consists of twenty-six small cubes that are United by a small cube in the center of the Rubik's CUBE.Each side of the Rubik's CUBE can be rotated as much and as well as unidirectional counter-clockwise. Rubik's CUBE cube, which consists of 26 small cubes that can be spun on its axis. Each side of the cube has 9 surface which consists of six different colors and resolved by performing a series of twisted columns and rows to uniform the color on each side after being scrambled.

Required algorithm that can provide the solution to be able to complete the game Rubik s CUBE.; There are many algorithms that can be used to provide solutions for searching paths on a game. Based on the results of the study can be summed up the COMPLETION of the GAME Rubik's CUBE USING a BACKPROPAGATION NEURAL NETWORK can provide a solution to complete the game and the test results can be known the value of Backpropagation Algorithm accuracy achieved an average 95% correct in testing and 5% incorrectly implemented the game Rubik's CUBE 2 x 2.

Keywords: Computer Vision, games, Rubik's CUBE, Backpropagation Neural Netowrk

1. INTRODUCTION

Rubik's CUBE is a mechanical puzzle game three dimensions found in 1974 by a sculptor and Professor of architecture from Hungary named Erno Rubik. The first Rubik's CUBE 3x3x3 cube sized small.each small cube side facing out to have different colors. Rubik's CUBE consists of twenty-six small cubes that are United by a small cube in the center of the Rubik's CUBE. Each side of the Rubik's CUBE can be rotated as much and as well as unidirectional counter-clockwise.

Rubik's CUBE cube, which consists of 26 small cubes that can be spun on its axis. Each side of the cube has 9 surface which consists of six different colors and resolved by performing a series of twisted columns and rows to uniform the color on each side after being scrambled. Puzzle Rubik's CUBE is one of the well-known Combinatorial problems due to the irreducibly complex, where there are 43x10^18 different configurations that may be generated with the shuffle it.

Completion of the Rubik's CUBE puzzle by relying on intuition as well as with certain steps memorized humans usually requires more than 40 rare. This has encouraged various studies to find optimal solutions, namely, a series of steps the shortest among the various possibilities of alternative steps to solve a Rubik's CUBE with a random configuration simulate on computer programs and applying artificial intelligence to complete.

*Several algorithms have been proposed in order to find the optimal solution in solving this puzzle, among others, by Morwen Thistlethwaite Thistlethwaite Algorithm in 1981, algorithm of Two-Phase by Herbert Kociemba in 1992, IDA * Algorithm with Pattern Databases by Richard e. Korf in 1997, as well as the development of algorithms for Two-phase by Kociemba and Thomas Rokicky in 2010.*

In research that has been done the game Rubik's CUBE can be solved by Greedy Algorithms and Backtraking, at the conclusion that the research excellence of this method we can apply the algorithm without using computers though. Disadvantages of this method is causing the performance of both the Algorithm because it cannot be tested using a computer so that the advice given is to experiment with the use of other algorithms that may be able to give a better solution.

One of the Algorithms that can be used is the Backpropagation Neural Network Algorithms. This algorithm has never applied to complete search solution on the game Rubik's CUBE and performance in resolving the problem is still unknown. Neural Network Backpropagation algorithm was one of the artificial intelligence that can complete the job with the given training. Neural Network Backpropagation network is a group of pemproses elements, units, or nodes are modeled based on the neurons of the brain.

the ability of the processor is stored in the strength of the connections between neurons or weights (weights), derive from the ability of adaptation to the process or learning from a training set of patterns.

Based on the background of the problem can be summed up how to implement Neural Network Backpropagation Algorithm on the game Rubik's CUBE for solving them with the aim of finding a solution can solve Rubik's CUBE game and to know the performance (time) Algorithm Neural Network Backpropagation in different settlement game Rubik's CUBE.

1.2 ARTIFICIAL NEURAL NETWORK

Artificial neural network (ANN) United Kingdom (Artificial Neural Network (ANN), also called a Simulated Neural Network (SNN), or generally just called a Neural Network (NN) is a network of a group of small processing unit modelled based on neural network humans. JST is adaptive systems that can change its structure to solve problems based on information of external as well as internal flow through the network. Simply put, the JST is a modeling tool for non-linear statistical data. THEY can be used to model complex relationships between inputs and outputs to find patterns in the data shown in the picture below.

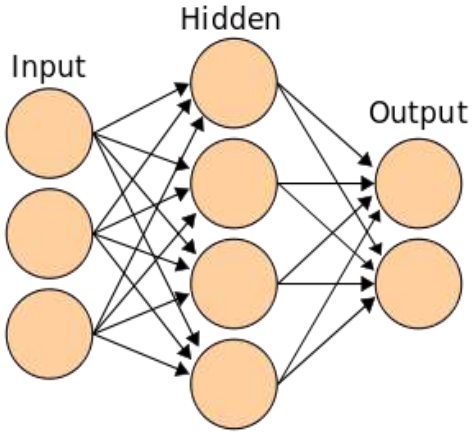


Figure 1.Input and Output patterns JST

1.3 BACKPROPAGATION

One of the artificial neural network architecture that is very popular is a multilayer feedforward networks. In General, such a network consists of a number of units of the neuron as a the input layers, one or more layers of node-node computational layer neurons, and a hidden layer node-node computational output neurons. Dipropagasikan input signals to the next direction (the direction of the output layer), layer upon layer. This type of network is the result of generalizing from a single layer perceptron architecture, so commonly referred to as the multilayer perceptron (MLPs).

Error Backpropagation Algorithm is MLPs that uses the principle of learning including a watch. Back propagation (in the direction of the input layer) occurred after the network generate output containing errors. In this phase the entire synaptic weights (which has no zero activation) in the network will be adjusted to correct/minimize the error occurred (error correction rule). For the training phase, the partners network propagation to the front and back done repeatedly for a set of data, then the exercise is repeated for a number of Epoch (one session through an entire data to exercise in a training process networks) until the error happened reached the limits of tolerance of certain small or zero.

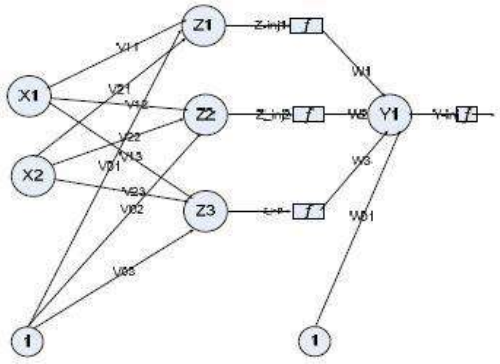


Figure 2.The Form Of Backpropagation Network

It can be seen that the act as a dendrite is X 1 and x 2, i.e. the input data on the network. There are two synapses or weights namely V and W, whereas the Z and Y are part of soma or cell body of the network. And acting as the axon or the data output is Y the following learning Algorithms. the method of Backpropagation.

1. Determine the value Weights

Step 0:

The awarding of the weighting value initialization (rated small randomly)

Step 1:

If the condition stops not reached, repeat steps 2 through 9 until the end of the iteration condition is fulfilled (condition stops if it reaches maximum looping looping or Mean Square Error (MSE) is smaller than the ratio study)

Step 2:

For each pair of data training (Training data) do steps 3 through 8

2.Feedforward

Step 3:

Each unit of input (Xi, I = 1, ..., n) receives the signal input signal the Xi and disseminated to the units the next section (hidden layer units)

Step 4:

Calculate the value of the Hidden and the Net, to calculate the Net value, the input must be normalized using the formula used to be bipolar, sigmoid should be normalized using the equation

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

Each hidden layer units multiplied by the factor weighing and aggregated and added with his bias:

$$Z_{in_j} = V_{oj} + \sum_{I=1}^{n} X_i V_{ij} \quad \textbf{(2-2)}$$

Then calculate in accordance with the activation function is used:

$$Z_j = f(Z_{in_j}) \quad \textbf{(2-3)}$$

When used is the sigmoid function then form the function is

$$Z_j = \frac{1}{1+exp^{-z_in_j}} \quad \textbf{(2-4)}$$

Each output unit (Yk, k = 1,2,3, ... m) multiplied by a factor of weighing and added:

$$Y_{in_k} = W_{ok} + \sum_{j=1}^{p} Z_j W_{jk} \quad \textbf{(2-5)}$$

Count back in accordance with the function activation

$$y_k = f(y_{in_k}) \quad \textbf{(2-6)}$$

3.Backpropagation and the error

Step 5:

Each output unit (Yk, k = 1, ..., m) receives the target pattern in accordance with the input patterns while training and calculated error:

$$\delta_k = (t_k - y_k) f'(y_in_k) \quad \textbf{(2-7)}$$

Because $f'(y_in_k) = y_k$ *the using a sigmoid function, then:*

$$f'(y_in_k) = f(y_in_k)(1 - f(y_in_k)) = y_k(1 - y_k) \quad \textbf{(2-8)}$$

Calculate the weighting value factor (later improvements to improve

$$w_{jk}\ \Delta W_{kj} = \alpha . \partial_k . Z_j \quad \textbf{(2-9)}$$

Calculate the correction fixes:

$$\Delta W_{0j} = \alpha . \partial_k \quad \textbf{(2-10)}$$

And use the value δ _ k on all units of the previous layer.

Step 6:

Each penimbang connecting the units l; apisan exodus with units on a hidden layer Z_j, j = 1, ..., p) multiplied by delta and added as input to the next layer of units

$$.\delta_in_j = \sum_{k=1}^{n} \delta_k W_{jk} \quad \textbf{(2-11)}$$

Further multiplied by the derivative of the function aktifasinya to calculate the error.

$$\delta_j = \delta_in_j\ f'(y\ _in_j) \quad \textbf{(2-12)}$$

Then count the repair weighting value (used to fix ***V_ij****)*

$$\Delta V_ij = \alpha \delta_j\ X_i \quad \textbf{(2-13)}$$

Then calculate the refractive repair (fix for ***V_oj****)*

$$\Delta V_oj = \alpha \delta_j \quad \textbf{(2-14)}$$

Fix weighing and bias:

Step 7:

Each output unit (yk, k = 1, ..., m) fixed bias and weighting value (j = 0, ..., p)

$$W_jk\ (baru) = W_jk\ (lama) \Delta W_jk \quad \textbf{(2-15)}$$

Each hidden unit (Zj, j:1, ..., p) fixed bias and weighting value (j = 0, ..., n)

$$V_jk\ (new) = V_jk\ (old) \Delta V_jk \quad \textbf{(2-16)}$$

Step 8:

Test conditions of dismissal (the end of the iteration) testing is done by calculating the dismissal of the mean Square Error (MSE). Where MSE will take the average squared error between the output and the target chain, here is the formula of MSE

$$\textbf{MSE} = \frac{\sum_{i=1}^{n} e_j^2}{n} \quad \textbf{(2-17)}$$

1. CONTENT OF THE STUDY

Rubik's CUBE 2 × 2 in accordance with the configuration provided by the application. In this case the large number of rows and columns on the Rubik's is 2 × 2. Analysis of data input is performed to obtain a value for the input that will be used for the process of classifying in the method of Backpropagation. The matrix form of the input Data Set is retrieved from the specified application, is the result of the extraction of the characteristics of each step. Following is the display of any 2 x 2 Rubik's.

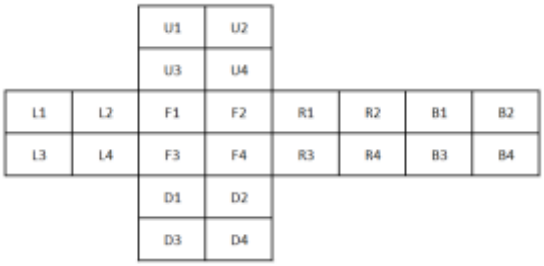


Figure 3.Rubik's display 1

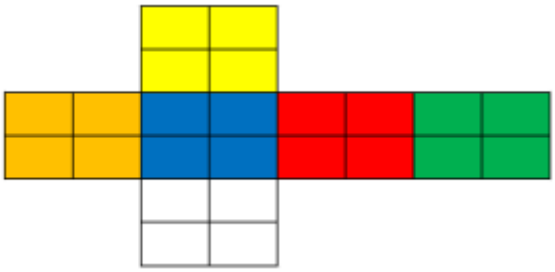


Figure 4.Rubik's display 2

Table 1.Color conversion Rubik's

No	Color	Description
1	RED	R
2	GREEN	G
3	BLUE	B
4	YELLOW	Y
5	WHITE	W
6	ORANGE	O

2.1 Conversion Code Notation Rubik's

The terminology commonly used to describe an algorithm Rubik's CUBE as follows:

1. *Rubik's CUBE consists of six fruit sides:front, back, left, right, up, and down.*
2. *Twist the sides of the Rubik's CUBE clockwise symbolized by the first letter of the name of that side in the language of United Kingdom. For the opposite direction with the needle, added quotation marks ('). For example, F means twisting the front side (front) clockwise, while L ' means turning the left side (left) in the opposite direction with the clock.*

On the application to be designed the notation code changes to the algorithm that is used in the application. This change aims in order to make the system easier in reading command to run code written notation. The change can be seen in the following table :

Table 2.Rubik's Notation conversion

No	Description		Symbols
1	*UP*	*Rotate the top clockwise to*	U
2	*UP'*	*Rotate counter-clockwise up to the top*	U'
3	*Down*	*Rotate downward and clockwise*	D
4	*Down'*	*Rotate counter-clockwise down*	D'
5	*FRONT*	*Turning to the next clockwise*	F
6	*FRONT'*	*Turning to the future fight clockwork*	F'
7	*BACK*	*Play back clockwise*	B
8	*BACK'*	*Rotate counter-clockwise to back*	B'
9	*LEFT*	*Rotate to the left and clockwise*	L
10	*LEFT'*	*Rotate to the left counterclockwise*	L'
11	*RIGHT*	*Turning to the right clockwise*	R
12	*RIGHT'*	*Turning to the right clockwise*	R'

2.2 Analysis method of Backpropagation Training

The steps by using the Backpropagation Algorithm will be described in section later analysis methods. This value will be a reference in conducting the Testing. In searching for the features of each class then the need for training. For more details, architectural analysis can be seen in the following image.

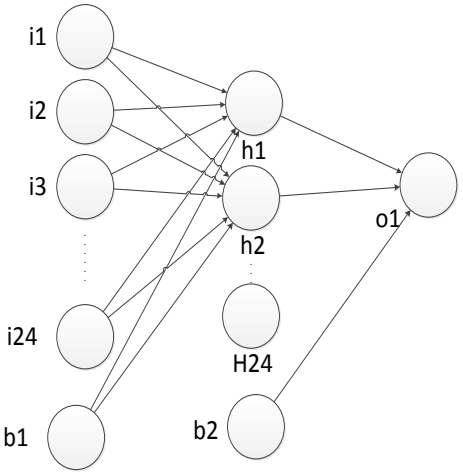


Figure 5.Network Architecture

Parameters to calculate Backpropagation network architecture are as follows:

1. *i = example input value 24 characteristics, derived from the sum of all the boxes contained on rubik as much as 24.*
2. *h = example Hidden Layer 36. Hidden Layer is layer is used as a comparison. Therefore the function of the Hidden Layer is add instruction logic, so the more logic is added or the more Hidden Layer then the memory that is used more and more and will result in a longer process.*
3. *b = bias. While bias is used to input the value remains the same i.e. 1, the use of biased to speed up training to correct against the sufficiency of the input variables that are set.*
4. *O = output. On the value of the output, there is value to determine the shift character until it reaches the solve.*

Then look for the value of the normalization value shift, normalization is intended to find the threshold which is used as a condition in the process of Backpropagation. Threshold itself is the distance between the values of the normalization, while normalization was the deciding edge threshold value. The value obtained using normalization equation as follows:

$$v' \frac{\textit{the value of the character} - \textit{the value of the smallest character}}{\textit{the largest character value} - \textit{the value of the smallest character}}$$

For example using the example at shift F (Front), obtained by the process and the results are as follows:

$$V' = \frac{1-0}{12-0}$$
$$V' = 0,083$$

Once counted all the values to data to 12, then the existing values can be seen in the following table :

Table 3.The Movement Of The Character

Shift	Value normalization shift	Threshold
F (Front)	0,083	0 - 0.083

Shift	Value normalization shift	Threshold
B (Back)	0,16	0.084 - 0.16
D (Down)	0,25	0.17 - 0.25
U (Up)	0,33	0.26 - 0.33
R (Right)	0,42	0.34 – 0.42
L (Left)	0,50	0.43 - 0.50
F'	0,58	0.51 – 0.58
B'	0,67	0.59 – 0.67
D'	0,75	0.68 – 0.75
U'	0,83	0.76 – 0.83
R'	0,92	0.84 – 0.92
L'	1	0.93 - 1

In this case the value used for the Learning Rate is 0.5. Because of the Backpropataion Algorithm can only accept input from 0 – 1 the value of inputs and outputs that are not in accordance with the rules of the Backpropagation. So it could be written down value normalization as follows:

Table 4.Normalized Value

No	*The Value Of The*
0	0
1	0,083
2	0,16
3	0,25
4	0,33
5	0,42
6	0,50
7	0,58
8	0,67
9	0,75
10	0,83
11	0,92
12	1

Of the value of the already established above are then used for the calculation Algorithm Backpropagation. Rare-step calculations are as follows. Neural Network Backpropagation algorithm it is kind of supervised learning Algorithms, so the Algorithm Backpropagation would optimally when data has a lot of Training. Of Data Sets that are already formed above is then used for the calculation of the Backpropagation Algorithm on the input. Rare-step calculations are as follows.

Data input

To make representing Rubik's in calculating settlement to note however is still the stiker Shaper Rubik's in a State of random i.e. the bottom, back, left side of the cube. It means that Rubik's white with stickers, red and blue colors on the data input does not move when manipulating the cube because the scale is already done when performing calculations. A Rubik's done calculations to the program by giving the sequence of 24 space separated characters that correspond to the colors.

Character color:
y: yellow
g: Green
w: white
b: Blue
r: Red
o: Orange
Stickers that mentioned (1-9 and A-Q) such as using this opens the cube 2-d map as follows:

Table 5.Data Input Rubik's

		1	**2**		
		3	**4**		
H	**I**	**5**	**6**	**N**	**O**
J	**K**	**7**	**8**	**P**	**Q**
		9	**A**		
		B	**C**		
		D	**E**		
		F	**G**		

So, any colour stickers can be encoded as follows:
White = 000
Yellow = 001
Blue = 010
Green = 011
Red = 100
Orange = 101

Later seen from the 6-sided Rubik's, each side can be changed by giving a total of 12 possibilities and can be encoded into the output in the same way, as the following example of four movements:
R = R ' = 0000 1000 L = 0001 L ' = 1001

After determining the intended movement, can do the calculation of the value of the Hidden and the Net where there should be normalized using the formula first sigmoid bipolar using equations (2-1) as follows:

Table 6.Value movements of Rubik's

The Name Of The Movement	Binary Form	The Form Of Value
R	0000	0
R'	1000	8
L	0001	1
.....	.....	.....
L'	1001	9

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

$$\mathbf{W1} = f(R) = \frac{1-e^{-0}}{(1+e^{-0})} = \mathbf{0,732756}$$

$$\mathbf{W2} = f(R) = \frac{1-e^{-8}}{(1+e^{-8})} = \mathbf{0,126772}$$

$$\mathbf{W3} = f(R) = \frac{1-e^{-1}}{(1+e^{-1})} = \mathbf{0,754919}$$

Dan selanjutnya sampai W12,

$$\mathbf{W12} = f(R) = \frac{1-e^{-9}}{(1+e^{-9})} = \mathbf{0,002581}$$

1. *Determine the value Weights*

This first process also aims to optimize band selection merging characters when the data is used for Training are still a bit. Calculate the value of the

weight, i.e., the parameters w and v as weights, these results are obtained from the results of the learning data sets are random. Example of the data already entered as follows:

Table 7.Weighting Value W

w1	0,732756
w2	0,126772
w3	0,754919
w4	0,111275
w5	0,499606
w6	0,294177
w7	0,556117
w8	0,884137
w9	0,357525
w10	0,851517
w11	0,162618
W12	0,002581

After getting the results of weight W of random sets of Data, and then determine the weight of V is obtained when identified as a Hidden Layer with a value as follows:

Table 8.Hidden Layer Weights Of V

v1	**0,489456**
v2	**0,616634**
.	.
.	.
.	.
V12	**0,840474**

2. *Feedforward*

a. *Calculate The Value Of The Net*

To calculate the net value, the input must be normalized using the formula used to be bipolar, sigmoid should be normalized using equations (2-1), where the parameters to be used are:

Learning Rate = 0.5

Target error = 0.1

Before finding the net value, determine the value of the random activation of the unit number is equal to the value of W that is as much as 162. But in an example of the calculation of the value of 5 will be used as follows:

Table 9.Activation of Unit 1

type	I1	I2	I3	I4	I5
the value of	**1**	**0,646082206**	**0,956740301**	**0,516962148**	**0,995244031**

After getting the value of I, then the next step is to find the value of the net, the equation to be used is equation (2-2). With the case this analysis explained that X_i = W_i and and V_ij = I_ij

b. *The Activation Value Net*

To activate the value net already resulting from the calculations above, in the get the value 2.091677417 with value h1 results

c. *Calculate Output*

The value of the output here is the value of the classification, the equation will be used to get the value of the is equation (2-2) so that it can be written as:

*Output = v1 * h1*

*Output = 0.489456 * 2.091677417*

Output = 1.023784061

d. *The Value Of The Target Classification*

The value of the target classification is used to determine the classification of the input that will be klasifikasikan. With the classification in the following table:

Table 10.The Value Of The Target Classification

Shift	Treshold
F (Front)	0 - 0.083
B (Back)	0.084 - 0.09
D (Down)	0.10 - 0.1
U (Up)	0.12 - 0.111
R (Right)	0.112 – 0.125
L (Left)	0.126 - 0.143
F’	0.144 – 0.167
B’	0.168 – 0.2
D’	0.21 – 0.25
U’	0.26 – 0.33
R’	0.34 – 0.5
L’	0.6 - 1

e. *Calculate The Value Of The Error*

The value of the error to determine classification, if the value already meets the criteria of the target classification will then be done shift, rather then do the process next Backpropagation. Because the output value gained from 1.023784061, so it is known to exceed target classification, then carried out the process of Backpropagation.

3. *Backpropagation the error*

Backpropagation method will be applied to the analysis of the problems in the quest to develop color mixed reviews on the game Rubik's CUBE, this process is done to change the weights so that it gets an error that the approaching target Error. The stage will be in the doing :

a. *Looking for a delta value 1*

Delta 1 is used for the calculation of the value of the delta v
The calculation will be done
*Delta 1 = (target – output1) * output1 * (1-output1)*
*Delta 1 = (0-1.023784061) * 1.023784061 * (1-1,023784061)*
Delta 1 = 0.024928878
Then determine the delta W with coaching the output with the hidden steps being performed.

b. *Looking for a delta value w*
Delta w used to get the value of the delta v with the calculation below:
*Delta w = Learning Rate * 1 * output hidden delta i*
*Delta w = 0.5 * 0.024928878 * 1.023784061*
Delta w = 0.012760893

c. *Looking for a delta value net*
Delta net used to get the calculations on the value of search output. With the following calculation:
*Net Delta = delta w * v random values*
*Delta net = 0.012760893 * 0.489456*
Delta net = 0.006245895

d. *Looking for a delta value output*
Delta output or i used to get perhitungna in finding a value weighting w. Calculations to be performed as follows:
*Delta delta net i = output hidden * i * (1-output hidden i)*
*Delta i = 0.006245895 * 1.023784061 * (1-1,023784061)*
Delta i =-0.000152085

e. *Find the value of weight w*
Weight w is used to get the value of new or update weights weights. The calculation is as follows:
*W = weights Learning Rate * delta i * input i*
*Weight w = 0.5 * (0,000152085) * 1*
Weight w =-0.000076042

f. *Update the weights*
Update bobo is used to determine the value of the new bobo.
Update the weights = old + weights weights wi
Update the weights = 0.732756 + (-0,000076042)
Update the weights = 0.732679958

From the results of the update the weighting was done to obtain the value of the error is smaller than the results of the previous step. And with a value of 0.732679958 so that it meets the target criteria D '.

So the next iteration of the movement is done repeatedly to produce the solve. Below are the results of the end of each perhitung the rubik's turnaround:

Table 11.The Results Of Calculations

No	*The Value Of The Results Of Weighting*	*Movement Notation*	*Description*
1	0,934345	L’	Rotate to the left counterclockwise
2	0,656757	B’	Rotate counter-clockwise to back
3	0,773435	U’	Rotate counter-clockwise up to the top
4	0,854656	R’	Rotate to the right counter-clockwise
5	0,548986	F’	Play it forward counterclockwise

3. Conclusion

Based on the results obtained in the research and drafting of this thesis as well as adapted to the purpose of the research is then retrieved the following conclusions:

1. *application s can determine the stage of action research using the Backpropagation Algorithm.*
2. *The accuracy of the testing Algorithm Backpropagation of timing the fastest open apply with an average time of 0.3275 seconds while the accuracy of the test results the longest time resolved applies with average time 2,321 seconds.*

Bibliography

[1]. Rubik's Official Website, "Rubik's World Sube Fact", http://www.Rubik ' URs.com/world/URCUBE_facts.php (accessed on September 22, 2015).

[2]. Richard e. Korf, Larry a. Taylor, "Pruning Duplicate Nodes in Depth-First Search", Proceedings of the Eleventh National Conference on Artificial Intelligence, Washington, d. c., pp. 753-761, July 1993.

[3]. Tomas Rokicki Herbert Kociemba, "God's Number is 20", http://www.CUBE20.org (accessed on September 22, 2015).