# Performance Enhancement Of Backpropagation Algorithm Using Momentum and Learningrate With a Case Study On Fingerprint Recognition

Bernadetta Raras I.R.[1], Jimmy Tirtawangsa[2], Thomas Anung[3]

[1]PT Telekomunikasi Indonesia,Tbk, Kebon Sirih 12, Jakarta Pusat 10110, Indonesia

[2]School of Computing, Telkom University Telekomunikasi 1 Bandung 40257, Indonesia

[3]Parahyangan University Jl. Ciumbuleuit 94, Bandung 40141, Indonesia

[1]raras@telkom.co.id, [2]jimmytirtawangsa@telkomuniversity.ac.id, [3]thomasanung@gmail.com

**Abstract— Backpropagation algorithm is very often used for learning process in multilayer networks of Artificial Neural Network (ANN) [7]. One area which oftenly uses ANN is fingerprint recognition. By applying learning rate [1] to backpropagation algorithm, learning process will be more stable and faster in finding the optimal delta (stepsize) on producing error . Momentum in backpropagation algorithm [2] monotonously decreases the errors during the training procedure with weakly convergent.**

**Combination of momentum and learningrate in the backpropagation algorithm during training process may increase the recognition accuracy of fingerprints. Our experiments show that by applying momentum and learningrate gradually to the backpropagation algorithm, recognition accuracy increased to 80.9%, which is 20% better than the standard backpropagation.**

***Keywords—Neural Networks; Fingerprint patterns; Backpropagation; Momentum; Learningrate***

## 1. Introduction

Artificial Neural Network (ANN) is a branch of artificial intelligence theory that has been used in various applications such as pattern recognition. The advantages of ANN as a system is the ability to imitate human thoughts by way of computational intelligence in such pattern recognition that is useful to do modelling prediction, error detection and control systems with artificial intelligence approaches and computational design.

One very popular method for learning in ANN multilayer networks called Backpropagation [7]. Backpropagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks used in conjunction with an optimization method such as gradient descent. The method calculates the gradient of a loss function with respects to all the weights in the network. The gradient is fed to the optimization method which in turn will be used to update the weights, in an attempt to minimize the loss function. Standard backpropagation had few shortfall such as bigger step-size that taken in each iteration produces error rate and oscillation problem when the error rate had a very narrow minimum area.

ANN has many applications and most likely applications for the ANN are Classification, Association, and Reasoning. An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a Backpropagation algorithm the one of ANN's branch that has been trained accordingly. Examples of pattern recognition are fingerprint recognition, optical character recognition, and voices or sound pattern recognition.

Pattern recognition has been used since 1936 [9], compared to conventional fingerprint technology that consist several shortcomings such as:

1) If the finger had injured, cut or burned, it will affect the detection performance.

2) The fingerprints may be affected by chemical substance. Therefore the detection method should be changed not by using conventional authentication mode.

3) Any sweep movement include alteration can leave visible traces on the surface of scanner tool and biased that makes the sensor of tool difficult to recognize fingerprint.

4) It is very difficult to use

5) Possible inaccuracy because of fingerprint altered position in scanner tools.

In standard backpropagation, the weight values are based on changes in gradient that occurs for the patterns. Modification that can be done is to change the weights based on the direction of the resulted gradient pattern and the previous pattern is called momentum. In backpropagation, the learning rate is analogous to the step-size parameter from the gradient-descent algorithm. If the step-size is too high, the system will either oscillate about the true solution, or it will diverge completely. If the step-size is too low, the system will take a long time to converge on the final solution. Finding the best step-size can be performed by learning rate.

In ANN there are 3 commonly used methods to solve problems which are heuristic rule, delta-delta rule, and delta-bar-delta rule [3]. The heuristic rule updates the learning rate in every learning step, but the learning rate in one learning step is the same for all weights in the network. Delta-delta rule in ANN take the partial derivative of the error function in the next iteration to the current iteration. Nevertheless it still has some problems in step-size that is taken in the next iteration. In delta-bar-delta rule that is used by Backpropagation, learning rates increases linearly and decreases exponentially. Linear increment avoids excessive speed to increase. Exponential decrement makes learning rates decrease tremendously fast, but remain positive.

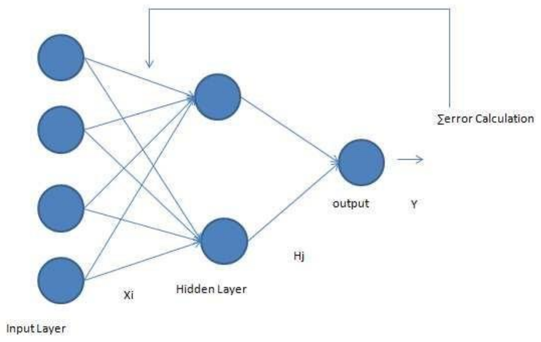


Fig. 1. Backpropagation Architect

Fingerprint is a pattern of streaks on the finger. Fingerprint recognition can be perform using various methods, one of which is a biometric method. Fingerprints have a feature pattern and minutiae that is different for every individual, therefore it is usefull to get to know a person's unique identity through fingerprint pattern. Fingerprint recognition is already widely known and used in various electronic applications. The recognition of fingerprint patterns can be perform using ANN that can recognize various pattern from the provision of past experience.

The research target in this study is to show that the combination of momentum and learning rate in Backpropagation Algorithm can improve learning performance on fingerprint recognition. The accuracy of standard backpropagation networks by Yoon, Feng, and Jain [5] which is 66.4% recognition accuracy. using NIST Fingerprint Image Quality (NIST), could be improved by this approach.

## 2. Background Study

### 2.1. Pattern Recognition with Backpropagation Method

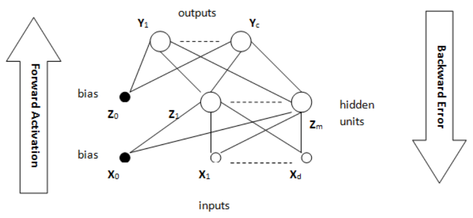


Fig. 2. Feed-forward Network Architecture

Backpropagation method consists of two stages of advanced methods of propagation (feed-forward) and backward propagation (back-propagation). Feed-forward networks can be seen as a graphical representation of a function that locates a group into the value of a certain output value [8]. Figure above shows an example of a feed-forward network that has been widely used in practical applications.

The nodes in the figure shows the input, output, and hidden variables, while the arcs are connectors. Network function can then be derived as follows, counting all output variables on the hidden layer Z_netj:

$Z_net_j = V_{j0} + \sum_{i=1}^{n} X_i V_{ji}$n layer then generated by converting the results Z_netj into Zj using the activation function f(Z_netj) :

$Z\_net_j$

$Y\_net_k = W_{k0} + \sum_{j=1}^{n} Z_j W_{kj}$ is then converted by generating a combination of hidden layer variables:

### 2.2 Backpropagation Algorithm

Here is the ANN with standard backpropagation algorithm to be used as a research method which illustrates the existing method:

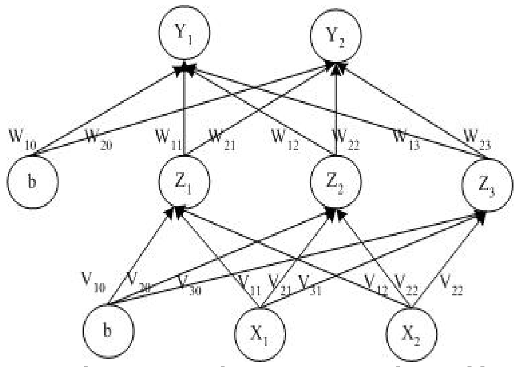


Fig. 3. Architecture Backpropagation with 1 Hidden Layer

**Step 0:** Initialize all weights and biases in small random numbers using formula 1 and 2. The weight of the input layer to hidden layer= $V_{ji}$, the weights of the hidden layer to output layer = $W_{kj}$.

$$|V_j| = \sqrt{V^2_{j1} + V^2_{j2} + \cdots + V^2_{jn}}$$

$$V_{ji} = \frac{\beta\, V_{ji}}{|v_j|}$$ the termination condition (is closed to or equals to the specified targets) has not been met, do steps 2-9.

**Step 2:** For each pair of training data or each group of data, do steps 3-8.

**Phase I: Weight Selection Algorithm and Initial Bias**

**Step 3:** Each input unit = Xi receives the signal and passes it to the hidden unit

**Step 4:** Calculate all output in the hidden units (Zj) (j = 1,2,3,..., p). Z_netj is the result of the calculation of the value of the input with weights that connect the hidden layer to the biased input layer (Vj0). Backpropagation sigmoid function (f (Z_netj)) is used to obtain the value of the hidden unit (Zj).

∑ **(3)**

——— **(4)**

**Step 5:** Calculate all the network in the output unit Yk (k = 1,2,3,..., m). Y_netk is the result of the calculation of the hidden unit with weights connected to the output layer that added to the biased hidden layer (Wk0). Backpropagation sigmoid function (f (Y_netk)) is used to obtain the value of output unit (Yk).

∑ **(5)**

**Step 6:** Calculate δ (error) factor output units based on each unit of output error Yk (k = 1,2,..., m)

! " 1 **(7)**

δ k is a unit of error that will be used in the layer underneath the weight change, while tk is the targeted network. Calculate new WKJ weight change rate (which will be used later to change the weights WKJ) with acceleration rate of δ

Δ % ; k=1,2,…,m ; j=0,1,2,…,p

**(8)**

**Step 7:** Calculate δ factor hidden units based on errors in each hidden unit Zj (j = 1,2,3,..., p)

∑& **(9)**

Factors δ hidden units:

!1 " **(10)**

Calculate new VJI weight change rate (which will be used later to change the weights VJI)

Δ % ;

j=1,2,…,p; i=1,2,…,n **(11)**

**Phase III: Changes in weight**

**Step 8:** Calculate the weights of all the changes. Change the line weight to the output unit:

'()* +(‸( Δ ;

k=1,2,…,m; j=0,1,…,p **(12)**

'()* +(‸( Δ ;

j=1,2,…,p; i=0,1,…,n **(13)**

**2.3. Feature Extraction**

A fingerprint is the pattern of ridges and valleys on the finger tip. A fingerprint is thus defined by the uniqueness of the local ridge characteristics and their relationships. Minutiae points are these local ridge characteristics that occur either at a ridge ending or a ridge bifurcation. A ridge ending is defined as the point where the ridge ends abruptly and the ridge bifurcation is the point where the ridge splits into two or more branches as said by Barham [4]. This is diagram of

minutiae extractor that will be used in this paper :

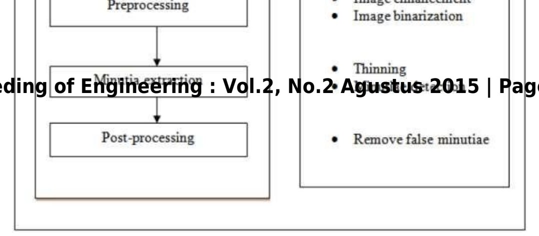


Fig. 4. Feature Extraction Diagram

**(6)**

**Phase II: backpropagation**

Step 1 : Preprocessing
Step 1.1 : Fingerprint Image Enhancement

1) Histogram Equalization, Histogram equalization is to expand the pixel value
2) Fourier Transform

Step 1.3 : Fingerprint Image Segmentation

In this technique only Region of Interest is usefull for recognition purposes. Area of the image without ridge which doesnt contain usefull information will be discarded and saved as background or noise information. There are two steps for ROI extraction which are :

i. Block Direction Estimation

For every block of image (16x16 pixel per block), this following algorithm is applied : Calculate the gradient values along x-direction and y-direction for each pixel of the block. Two Sobel filters [12] are used to fulfill the task. For each block, use the following formula to get the Least Square approximation of the block direction for all the pixels in each block by regarding gradient values along x-direction and y-direction as cosine value and sine value.

$$\tan 2\theta = 2\sin\theta \cos\theta /(\cos^2\theta - \sin^2\theta) \quad (14)$$

After finishing determining direction of each block in the image, blocks without significant information will be regarded as background by measuring E level (if value of E under threshold then the block is regarded as background).

$$E = \{2 \Sigma \Sigma (g_x * g_y) + \Sigma \Sigma (g_x^2 - g_y^2)\} / W*W*\Sigma \Sigma (g_x^2 + g_y^2) \quad (15)$$

ii. Morphological Operation

Morphological operation has two values, OPEN and CLOSE. OPEN can expand an image and remove peaks that shows by background noise. CLOSE operation will shrink the image and eliminate cavities.

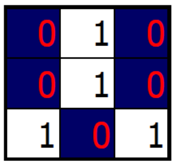
Step 2 : Minutiae Extraction [4]

Step 2.1 : Fingerprint Ridge Thinning

Purposes of ridge thinning is to eliminate the redundant pixels of ridges till the ridges are just one pixel wide. In each scan of fingerprint image, the algorithm will marks down redundant pixels in each small image window (3x3) and removes all marked pixels after several scans. In this step, any single points, whether they are single-point ridges or single-point breaks in a ridge are eliminated and considered processing noise.

Step 2.2 : Minutiae Marking

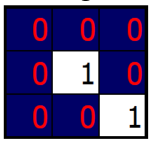
This step using Crossing Number concept to solve the minutiae marking, for each 3x3 window, if the central pixel is 1 and has exactly 3 one-value neighbors, then the central pixel is a ridge branch. Example for ridge branch :



| | | |
|---|---|---|
| 0 | 1 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Fig. 5. Ridge Branch

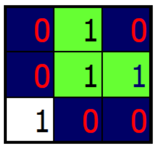
If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending. Example :



| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

Fig. 6. Ridge Ending

And for special case that a genuine branch is triple counted. Example :



| | | |
|---|---|---|
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

Fig. 7. triple counted

At this point average inter-ridge width D (variable of information) is estimated. All step above (ridge thinning and minutiae marking) will labeled with unique ID for next operation.

Step 3 : Post Processing : False Minutiae Removal

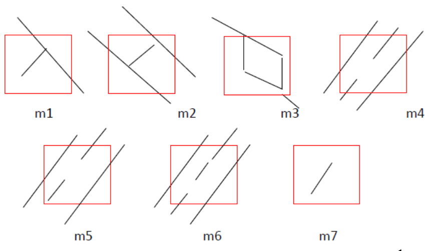


Fig. 8. Seven types of false minutiae[1]

The procedure for the removal of false minutiae are :

1) If the distance between one bifurcation and one termination is less than D and the two minutiae are in the same ridge (m1 case). Remove both of them. Where D is the average inter-ridge width representing the average distance between two parallel neighboring ridges.
2) If the distance between two bifurcations is less than D and they are in the same ridge, remove the two bifurcations. (m2, m3 cases).
3) If two terminations are within a distance D and their directions are coincident with a small angle

[1] Zain S. Barham, 2011

variation. And they suffice the condition that no any other termination is located between the two terminations. Then the two terminations are regarded as false minutiae derived from a broken ridge and are removed. (Cases m4,m5 & m6).

4) If two terminations are located in a short ridge with length less than D, remove the two terminations (m7).

The advantage of removing false minutiae is distinguish minutiae by seven types of false minutiae and seconds is the order of removal procedures is well considered to reduce complexity of computation because its utilize the relation among the false minutiae types.

## 3. Proposed Method

To Improve the existing method, the addition of momentum and learningrate change is intended to avoid the smaller weight as a result of the differences in the data. If the data supplied to the network has a similar pattern, the weight changes quickly. However, if the latest data entered has a very different pattern from previous patterns, then the weight change slowly. With the addition of momentum, new weights at time (t +1) are based on the weights at time t and (t-1). The formula should add two new variables to record the amount of momentum for the last two iterations. If $\mu$ is a constant ($0 \leq \mu \leq 1$) then the momentum of the new weight parameters as a new proposed method is calculated by the equation:

$$W_{kj}(t+1) = W_{kj}(t) + \alpha\, \delta_k Z_j + \mu\, (W_{kj}(t) - W_{kj}(t-1)) \quad \textbf{(12)}$$
$$V_{ji}(t+1) = V_{ji}(t) + \alpha\, \delta_k X_i + \mu\, (V_{ji}(t) - V_{ji}(t-1)) \quad \textbf{(13)}$$

This algorithm is a one-time training (training set), the training set is continued until the error value (factor $\delta$) is close to 0 or the specified target. After the training completes, the network is ready for pattern recognition.

## 4. Data Presentation

This research used two dataset to show the impact of small alteration on the experiment accuracy:

1. Dataset with position and alteration between $0^o$ and $45^o$ of angle, 240 fingerprint for training and 60 fingerprint for testing. This is example of 10 times captured on one finger :
2. Dataset with position and alteration between $0^o$ and $30^o$ of angle, 240 fingerprint for training and 60 fingerprint for testing. Below is an example of 10 images of one finger :

The details of training and testing datasets:

1. Training dataset contains 80% of fingerprint images from onedataset, while testing dataset contains the other 20% of the fingerprint images [11].
2. For cross validation data, as proposed by Rodriguez, Perez, & Lozano [6] k-fold validation is used with k=5 variants of data from one dataset for 80%:20% division of training and testing datasets

## 5. Experimental Evaluation

The experiment conducted in this research is limited to 30 classes of left thumbs that have only low risk of chemical substance. The proposed method is to optimize identification of fingerprints affected only by small position alteration.

The Goal of experiment are to compare accuracy of standard Backpropagation versus using combined learning rate and momentum, and to show small position alteration impact accuracy. Experiment will produce accuration value and learning time. The following is the experiment detail that will be performed :

1. First experiment used standard Backpropagation Algorithm without momentum and learning rate which are done on both dataset on five mixed variations of data as explain in k-fold validation. The result will be used as the base for the next experiments

TABLE 1
Result of Training and Testing Dataset Experiment 1
(Dataset alteration max $45^o$)

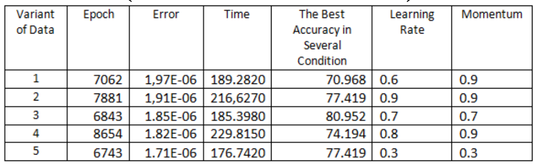
| Variant of Data | Epoch | Error | Time | Accuracy |
|---|---|---|---|---|
| 1 | 9703 | 1.87E-06 | 260.7720 | 48.387 |
| 2 | 11254 | 1,89E-06 | 315.4750 | 57.143 |
| 3 | 9414 | 1,82E-06 | 254.6330 | 64.516 |
| 4 | 9639 | 1,90E-07 | 281.9740 | 57.143 |
| 5 | 9813 | 1,84E-06 | 263.2960 | 57.143 |

TABLE 2
Result of Training and Testing Dataset Experiment 1
(Dataset alteration max $30^o$)

| Variant of Data | Epoch | Error | Time | Accuracy |
|---|---|---|---|---|
| 1 | 8645 | 1,76E-06 | 234,7260 | 67.742 |
| 2 | 7540 | 1,82E-06 | 208,7060 | 64.516 |
| 3 | 9196 | 1,94E-06 | 251,0130 | 74.194 |
| 4 | 7345 | 1,71E-06 | 200,8640 | 70.968 |
| 5 | 7978 | 1,69E-06 | 233,7950 | 57.143 |

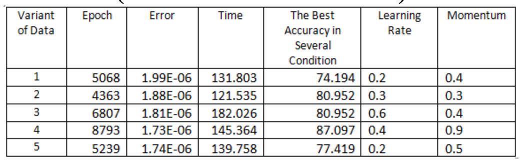
2. Second experiment is implementation of Backpropagation Algorithm with momentum and learning rate in several planned conditions which are done in both dataset in five mixed variation of data as explain in k-fold validation :

TABLE 3
Result of Training and Testing Dataset Experiment 2 (Dataset alteration max 45$^{o}$)



| Variant of Data | Epoch | Error | Time | The Best Accuracy in Several Condition | Learning Rate | Momentum |
|---|---|---|---|---|---|---|
| 1 | 7062 | 1,97E-06 | 189.2820 | 70.968 | 0.6 | 0.9 |
| 2 | 7881 | 1,91E-06 | 216,6270 | 77.419 | 0.9 | 0.9 |
| 3 | 6843 | 1.85E-06 | 185.3980 | 80.952 | 0.7 | 0.7 |
| 4 | 8654 | 1.82E-06 | 229.8150 | 74.194 | 0.8 | 0.9 |
| 5 | 6743 | 1.71E-06 | 176.7420 | 77.419 | 0.3 | 0.3 |

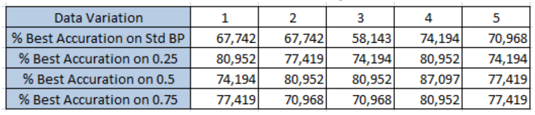
TABLE 4
Result of Training and Testing Dataset Experiment 2 (Dataset alteration max 30$^{o}$)



| Variant of Data | Epoch | Error | Time | The Best Accuracy in Several Condition | Learning Rate | Momentum |
|---|---|---|---|---|---|---|
| 1 | 5068 | 1.99E-06 | 131.803 | 74.194 | 0.2 | 0.4 |
| 2 | 4363 | 1.88E-06 | 121.535 | 80.952 | 0.3 | 0.3 |
| 3 | 6807 | 1.81E-06 | 182.026 | 80.952 | 0.6 | 0.4 |
| 4 | 8793 | 1.73E-06 | 145.364 | 87.097 | 0.4 | 0.9 |
| 5 | 5239 | 1.74E-06 | 139.758 | 77.419 | 0.2 | 0.5 |

For comparison, implementation of all experiment above with different initial weight value (0.25 and 0,75) conducted :

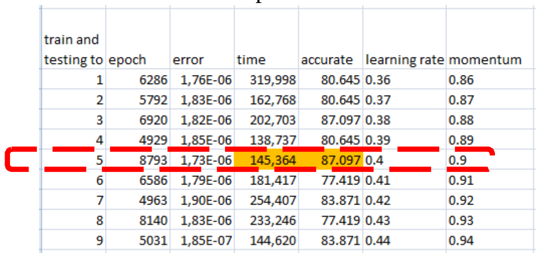
TABLE 5
Enhancement Performance of Backpropagation Base on different initial weight value



| Data Variation | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| % Best Accuration on Std BP | 67,742 | 67,742 | 58,143 | 74,194 | 70,968 |
| % Best Accuration on 0.25 | 80,952 | 77,419 | 74,194 | 80,952 | 74,194 |
| % Best Accuration on 0.5 | 74,194 | 80,952 | 80,952 | 87,097 | 77,419 |
| % Best Accuration on 0.75 | 77,419 | 70,968 | 70,968 | 80,952 | 77,419 |

3. Third experiment is fine tuning experiment which showed that momentum value of 0.9 and learning rate of 0.4 is the best value to represent the best accuracy on dataset 2 and variant 4 :

TABLE 6
Result of Momentum and Learningrate Fine Tuning Experiment



| train and testing to | epoch | error | time | accurate | learning rate | momentum |
|---|---|---|---|---|---|---|
| 1 | 6286 | 1,76E-06 | 319,998 | 80.645 | 0.36 | 0.86 |
| 2 | 5792 | 1,83E-06 | 162,768 | 80.645 | 0.37 | 0.87 |
| 3 | 6920 | 1,82E-06 | 202,703 | 87.097 | 0.38 | 0.88 |
| 4 | 4929 | 1,85E-06 | 138,737 | 80.645 | 0.39 | 0.89 |
| 5 | 8793 | 1,73E-06 | 145,364 | 87.097 | 0.4 | 0.9 |
| 6 | 6586 | 1,79E-06 | 181,417 | 77.419 | 0.41 | 0.91 |
| 7 | 4963 | 1,90E-06 | 254,407 | 83.871 | 0.42 | 0.92 |
| 8 | 8140 | 1,83E-06 | 233,246 | 77.419 | 0.43 | 0.93 |
| 9 | 5031 | 1,85E-07 | 144,620 | 83.871 | 0.44 | 0.94 |

## 6. Conclusion

The standard Backpropagation algorithm yielded 66.91% average accuracy which in par with 66.4% accuracy given by Yoon, Feng, and Jain [5]. Backpropagation using combination of momentum and learningrate increased the average accuracy to 80,9%. Using combination of momentum and learningrate also improved the average training time to 144 seconds from 225 seconds in standard Backpropagation. The most effective learning time show value on 0.9 as momentum and 0.4 as learningrate.

When the average of position alteration was increased up to 45$^{o}$ the recognition rate only reduced to 76.19%, which is only 5% decreased in accuracy.

## 7. Future Work

For further studies in this field, analysis of fingerprint image quality suitable or convenient to be treated as an input should be conducted. Another research on efectiveness of Backpropagation algorithm performance in another case may be conducted, to show and compare the impact of Momentum and Learning rate in pattern recognition.

## References

[1] Mandic P Danillo, Chambers A Jonathon, Towards the Optimal Learning Rate for Backpropagation, Springer, 2000

[2] Hongmei Shao, Gaofeng Zheng, Convergence analysis of a back-propagation algorithm with adaptive momentum, Science Direct, 2011

[3] Huijuan FANG, Jiliang LUO, Fei WANG, Fast Learning in Spiking Neural Networks by Learning Rate Adaptation, Science Direct, 2012

[4] Zain S. Barham, Fingerprint Recognition Using Matlab, Graduation Project Supervised by Dr. Allam Mousa, 2011

[5] Soweon Yoon, JianJiang Feng, Anil K. Jain, Altered Fingerprints: Analysis and Detection, IEEE, 2012

[6] Rodriguez, Perez, Lozano, Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation, IEEE, 2010

[7] Stuart J Russel, Peter Norvig, Artificial Intelligence A Modern Approach, Prentice Hall, 1995

[8] Christoper M. Bishop, Pattern Recognition and Feed-forward Networks, The MIT Encyclopedia of the Cognitive Sciences, 1999

[9] Christoper M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

[10] Saurabh Karsoliya, Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture, International Journal of Engineering Trends and Technology, 2012

[11] McCaffrey James, Understanding and Using K-Fold Cross-Validation for Neural Networks, visualstudiomagazine, 2013

[12] Matlab Documentation (edge), MathWorks, 2015