

# High Accuracy Arabic Handwritten Characters Recognition Using Error Back Propagation Artificial Neural Networks

Assist. Prof. Majida Ali Abed

College of Computers Sciences & Mathematics,  
University of Tikrit, Tikrit, Iraq

Assist. Prof. Dr. Hamid Ali Abed Alasad

Computers Sciences Department, Education for Pure  
Science College, University of Basra, Basra, Iraq

**Abstract**—This manuscript considers a new architecture to handwritten characters recognition based on simulation of the behavior of one type of artificial neural network, called the Error Back Propagation Artificial Neural Network (EBPANN). We present an overview of our neural network to be optimized and tested on 12 offline isolated Arabic handwritten characters (ا, ب, ج, د, هـ, ط, ظ, ف, ك, م, و, ي) because the similarity of some Arabic characters and the location of the points in the character. Accuracy of 93.61% is achieved using EBPANN which is the highest accuracy achieved during Offline Handwritten Arabic Character Recognition. It is noted that the EBPANN in general generates an optimized comparison between the input samples and database samples which improves the final recognition rate. Experimental results show that the EBPANN is convergent and more accurate in solutions that minimize the error recognition rate.

**Keywords**—Character Recognition; Neural Network; Classification; Error Back Propagation Artificial Neural Network

## I. INTRODUCTION

Many types of Artificial Neural Network have been proposed over the years. In fact, because Artificial Neural Network (ANNs) are studied by many sciences such as Computer Scientists, Electronic Engineers, Biologists and Psychologists, and has many different names such as Artificial Neural Networks (ANNs), Connectionism or Connectionist Models, Multi-layer Perceptron's (MLPs) and Parallel Distributed Processing (PDP). An ANNs defines a mathematical model for the simulation of a network of biological nervous systems and one of the most active learning methods to approximate real-valued, discrete-valued, and vector valued functions. It has been widely and very good used in pattern recognition problems, such as handwritten characters, recognizing spoken words, and face recognition [1]. The Error Back Propagation Artificial Neural Network (EBPANN) is a very common model in artificial neural networks and it does not have feedback connections, but errors propagate backward from the output layer during training. Training a network by Error Back Propagation Artificial Neural Network (EBPANN) includes three steps [2, 3];

- Feed forward of the input training pattern.
- Backward propagation of the associated error.
- Modification of weights.

To modification of connection weights between the pairs of layers, the errors in the output determine measures of hidden layer output errors which are used. The process of modifying the weights between the layers and calculate again the output continues until a stopping criterion is satisfied, such as the total error is reduced and reached to the value of given threshold. After training is completed, the network can be used to find outputs for new inputs [4]. Pattern recognition is important application of artificial neural network which can be applied by using a feed-forward artificial neural network Figure (1) explain this operation that has been trained accordingly.

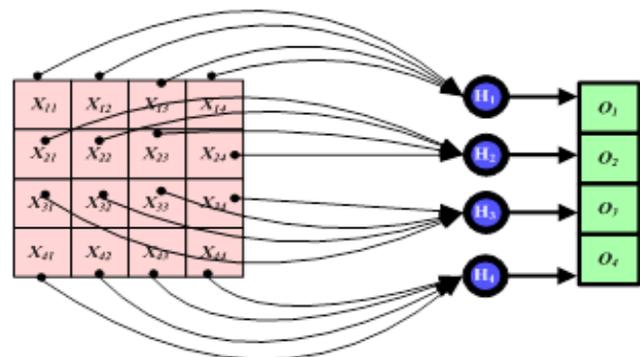


Fig. 1. Feed-forward artificial neural network for pattern recognition

Aim of this manuscript to implement the Error Back Propagation Artificial Neural Network (EBPANN) algorithm to optimize and Recognition the offline isolated Arabic handwritten characters. A brief review of neural network and Error Back Propagation Artificial Neural Network algorithm for training neural networks are presented in section 2. We present our proposed approaches, the pre-processing steps, the feature extraction technique and the classification in section 3. The knowledge base and experimental analysis are made in section 4. Results and discussion is presented in Section 5. Finally the conclusions are given in section 6.

## II. ERROR BACK PROPAGATION ARTIFICIAL NEURAL NETWORK

Error Back Propagation Artificial Neural Network was first described by Paul Werbos in 1974, and further developed by David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams in 1986 [5]. It was Common supervised learning technique; it's used for pattern recognition, function

approximation, prediction and Mapping tasks. The Error Back Propagation Artificial Neural Network algorithm has been widely used and a popular technique as a learning algorithm by examples in feed forward with a finite number of pattern pairs consisting of an input pattern and a corresponding (target) output pattern. Multilayer neural networks each layer consists of one or more cells (neurons), each cell (neuron) in layer receives input from each cell (neuron) in the previous layer, and output to each cell (neuron) in the following layer. The Back propagation is applied to feed forward artificial neural network with one or more hidden layers this algorithm is different than others in the way in which the weights are calculated during the learning phase of the network. Zweiri, Y.H. et al [6] minimized the total squared error of the output based on Error Back Propagation Artificial Neural Network. The important points in the Error Back Propagation Artificial Neural Network are:

- Networks in layered structure (multi-layers network).
- The value of all weights are random
- Learns by examples and learning process is done through sequential mode and batch mode
- Error calculated on output layer and propagated back to hidden layers.
- Consists from two parts forward part(calculated output) reverse pass(actual output)
- The input and its corresponding output(target)are called a training Pair

Weight changes calculated using learning rate ( $\eta$ )

Error Back Propagation Artificial Neural Network has some problems which include network paralysis, local minima and slow convergence these problem occur when the network always change the weights, modify value of weights to very large value during the training operation. There are a number of solutions for these problems. The important one is very simple and that is to change the weights to different random numbers and try training again. Another solution is to add “momentum” to the weight Change. This means that a weight changes from iteration to iteration at any given time depends not just on the present error, but also on preceding changes. Further training of the Error Back Propagation Artificial Neural Network is continued till the desired classification performance is reached. The algorithm consist of two parts first part is forward propagation, second part is reverse propagation which explain below the two parts and Multilayer feed-forward networks are trained using Error back propagation Artificial Neural Network learning algorithm as shown in Figure (2) [7].

- Forward Propagation part:

**Step 1:** Compute  $\delta_i^1$  in the output layer ( $o_i = y_i^1$ ).

$$\delta_i^1(c) = g'(h_i^1)[d_i^w(c) - y_i^1(c)] \quad (1)$$

Where  $h_i^1$  represents the net input to the  $i$ th cell in the  $l$ th layer, and  $g'$  is the derivative of the activation function  $g$ .

**Step 2:** The error of a character is calculated

$$E(c) = (1/2) \sum [d_i^w(c) - \delta_i^1(c)]^2 \quad (2)$$

- Reverse Propagation part:

**Step 3:** Compute the deltas for the preceding layers by propagating the errors backwards;

$$\delta_i^l(c) = g'(h_i^l) \sum_j w_j^{l+1} \delta_i^{l-1}(c) \quad (3)$$

Where  $w_j^{l+1}$  the weights and the following equations define the change in the weights

$$w_{kj}^1(t+1) = w_{kj}^0(t) + \eta \delta_{pk}^0 i_{pj}$$

$$w_j^{l+1}(t+1) = w_{ji}^l(t) + \eta \delta_{pj}^1 x_i$$

**Step 4:** Update weights using equation (4);

$$\Delta w_{ji}^l = \eta \delta_i^l y_i^{l-1} \quad (4)$$

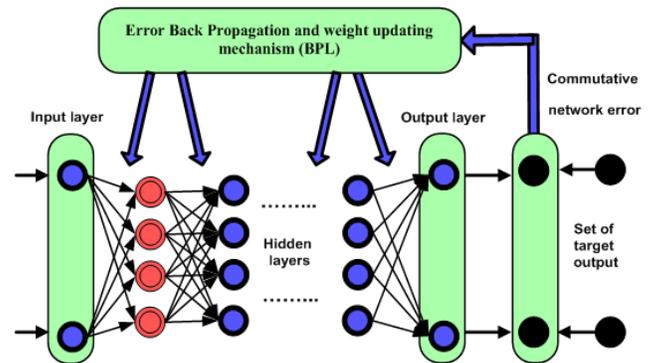


Fig. 2. Multilayer Feed-forward Neural Network.

### III. PROPOSED NEURAL NETWORK STRUCTURE WORK

Our proposed neural network structure work offline Arabic handwritten characters recognition is implemented with the help of Error Back Propagation Artificial Neural Network (EBPANN). Handwritten character recognition system can be divided into four steps and shown in Figure (3):

- 1) Character acquisition.
- 2) Pre-processing.
- 3) Feature extraction.
- 4) Classification and Recognition.

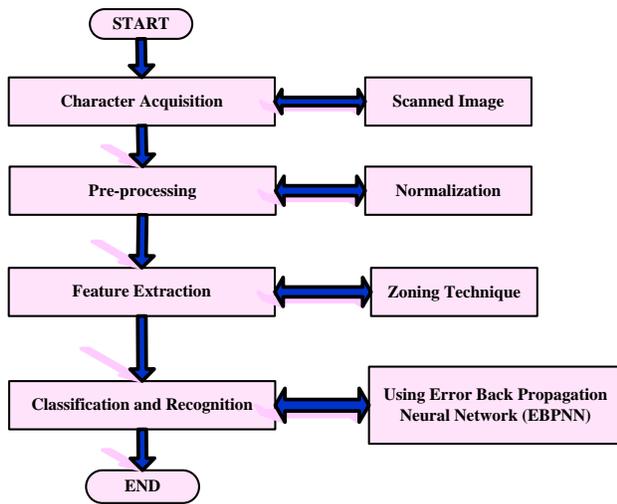


Fig. 3. Our neural network structure

### 1) Character acquisition

Acquisition of offline handwritten Arabic characters is done by using a scanner. In this investigation, some isolated Arabic handwritten characters been employed for image database in BMP type file. The Method of character acquisition in our neural network structure depend on the database used in the experimentation was preprocessed, size normalized images of scanned 252 image of offline handwritten Arabic characters.

### 2) Pre-processing

The process of normalization of characters it is necessary performed in pre-processing operation of our manuscript so that all characters can become of equal dimensions of the matrix, the size of a character varies but is typically around of 30 x30 pixels, since the input of Error Back Propagation Artificial Neural Network (EBPANN) is fixed size, it is necessary to normalize the size of the input characters and made the characters with same size, the normalization performed by using linear transformation to make the size of all characters with equal dimension in 18x18 array of binary pixels image , Figure (4) explain the process of normalization for Arabic character( ﻯ ) [8].

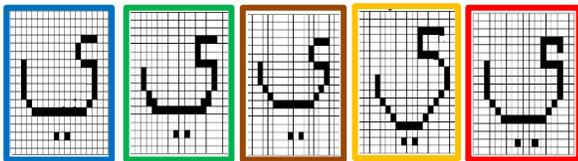


Fig. 4. Input image character after normalization

### 3) Feature Extraction

Feature extraction is the first step in classification and image recognition; it is the process of generating features to be used in the classification task. A method is used in our proposed neural network structure to extract features input Arabic characters based on the zoning technique [9]. The binary image is fixed and stored in a matrix form of size 18 x 18 array of binary pixels image to every Arabic character of twelve characters (ا, ب, ت, ج, هـ, ش, ط, ع, ف, ك, م, و, ي). Figure (5a)

shows the size normalized binary image of Arabic character ( ﻯ ). The binary image fitted in a matrix is divided into nine square sub- matrices of size 6x6 as shown in Figure (5b) some of these sub- matrices have the value "1" when the cell, have one feature values is black and other have the value "0" when the cell is white, have zero feature values.

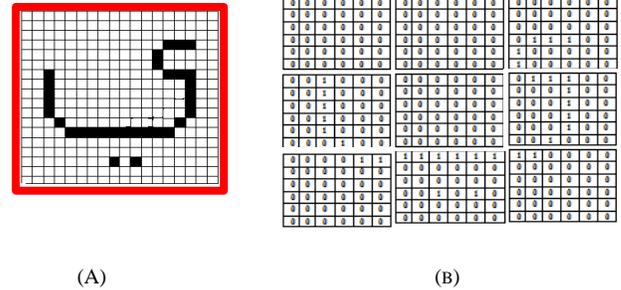


Fig. 5. Normalized image of character "ي", (b) Partitioned into nine square regions of size 6x6

For made the feature extraction simple method for implementation, we can converted the one feature value of each sub-matrix in matrix form into a real number by the equation (5), such that the Feature value of v<sup>th</sup> sub-matrix, f<sub>v</sub> is computed by this equation. The total number of 1s cells in a sub-matrix is divided by the total number of cells in that sub-matrix [10].

$$f_v = (1/pv) \sum C_{ij} \quad (5)$$

Where

f<sub>v</sub>= feature value of v<sup>th</sup> sub-matrix

pv = total number of cells in the sub-matrix v ,

C<sub>ij</sub> = value of (i,j)<sup>th</sup> cell in the sub-matrix

The normalized feature value of v<sup>th</sup> sub-matrix is computed by dividing the sum distances of 1s cells from the sum distances of all cells in that sub-matrix. The feature value of sub-matrix v is given by the equation (6).

$$f_v = \frac{\sum_{ij}^{pv} d_{ij}^v}{\sum_{pv} d_{ij}^v}, \quad d_{ij}^v = \sqrt{i^2 + j^2} \quad (6)$$

Which represent distance of (i,j)<sup>th</sup> cell. The set of features extracted from each sub-matrix is called feature vectors of samples which are used to train the Error Back Propagation Neural Network (EBPNN) [11].

### 4) Classification

Classification is performed after pre-processing step to the values of the resulting features with Error Back Propagation Artificial Neural Network (EBPANN), from this process will be obtained calculation of recognition, all the data which will then determine the percentage of success of this method [12]. The tests in our neural network consists image processing phase, feature vector extraction phase, and three network phases that is Neural network structure phase, Error Back Propagation algorithm phase, Running neural network structure

phase, and final the results phase presented in Figure (6). Our neural network structure optimized and tested 12 isolated Arabic handwritten characters (ا،ت،ج،ذ،ش،ط،غ،ف،ك،م،و،ي) because the similarity of some Arabic characters and the location of the points in the character. four characters without pointes (ا،ط،م،و)، three characters with one point above (ذ،غ،ف)، one character with one point below (ج)، one character with three points above(ش)، one character with hamaza (ك)، one character with two points below(ي). Each character is represented as a 18x18 array of binary pixels image, with 0 representing “off” and 1 representing “on”. 252 example images were used, and these were divided into two groups: 100 images were used as training data, 240 images were used as validation data after each epoch of training data, and finally 12 images were used to test the resulting network architecture. The images represented 12 isolated Arabic handwritten characters (ا،ت،ج،ذ،ش،ط،غ،ف،ك،م،و،ي).

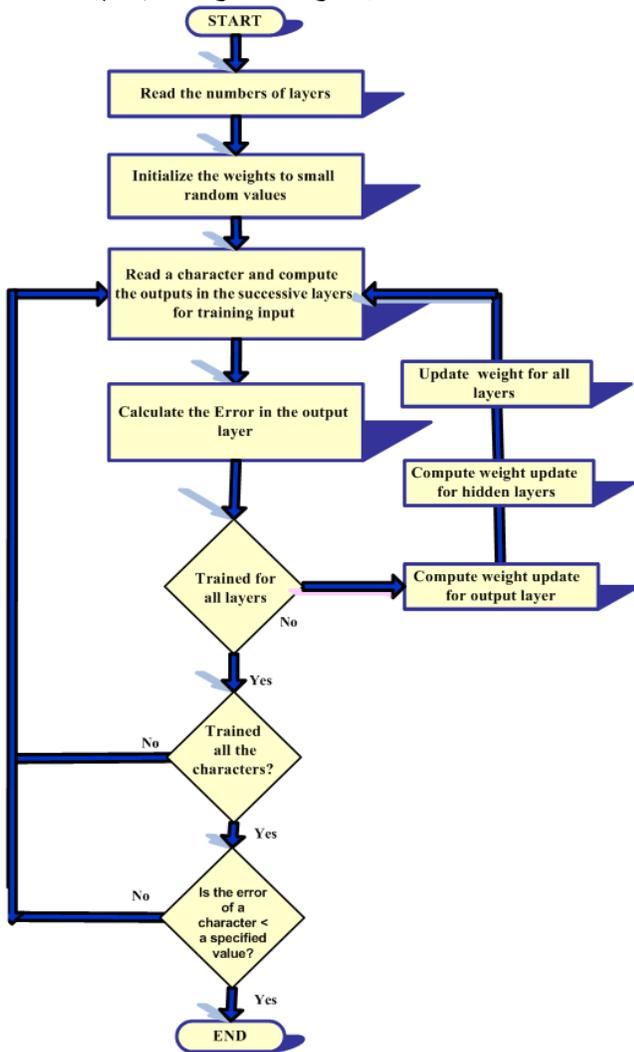


Fig. 6. The flowchart of Error Back Propagation Artificial Neural Network training architecture

#### IV. KNOWLEDGE BASE AND EXPERIMENTAL ANALYSIS

##### A. Knowledge Base

The knowledge base contains the Arabic characters in our neural network structure. It is basically a knowledge base of isolated Arabic handwritten characters, Individual character images in the knowledge base are used to generate the values for the input character Size of each character 30x30 pixels. Knowledge base consist 12 Characters only characters (ا،ت،ج،ذ،ش،ط،غ،ف،ك،م،و،ي) because the similarity of some Arabic characters and the location of Points in the character. four characters without pointes (ا،ط،م،و)، three characters with one point above (ذ،غ،ف)، one character with one point below (ج)، one character with three points above(ش)، one character with hamaza (ك)، one character with two points below(ي), in our neural network each character has different shapes written by hand arrange in separate file for each character. The training knowledge base consists of 240 samples for 12 Arabic characters, 252 characters with different shapes input samples. We use Matlab to extract the handwritten character for each file. Some training samples are shown in Figure (7).

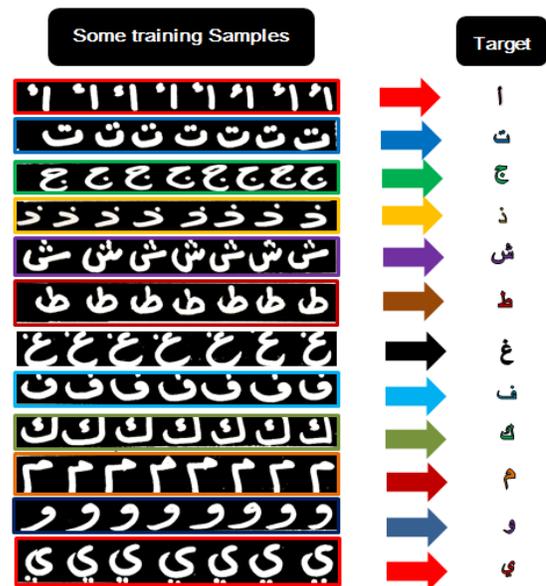


Fig. 7. Some samples of scanned offline handwritten Arabic character knowledge base

Each character is normalized into 18x18 array of binary pixels image, during preprocessing, along with noise/spurs removal. We have been extracted feature values by zoning technique for these normalized images. Thus for 18x18 array of binary pixels image, number of input nodes to the neural network are 252 and learning has been implemented using Error Back Propagation Artificial Neural Network. For training purpose we have used 20 samples of each class from 12 classes and testing applied to all images. The character ا is assigned as class 1, character ت as class 2 and so on up to character ي as class 12.

B. Experimental Analysis

Experimental Analysis in our proposed neural network structure consists from three parts,

- Neural network structure.
- Error Back propagation algorithm.
- Running neural network structure.

1) Neural network structure

Our proposed neural network structure consist multi-layer, feed forward network with error back propagation algorithm is given in Figure (8). The output from one layer feeds forward into the next layer of cells, cells of each layer connect with the cells of next layer i.e. our neural network are fully connected with various weights. The obtained error is determined by the difference of outputs from output layer and target outputs. Then errors are feed backward to modify weights, which are then applied to next iteration to generate new network outputs [13].

1) *Input Layers:* Data sets in our this work is arranged 18x18 array of binary pixels image, the white Pixel ("off" pixel) is represented by 0, black pixel ("on" pixel) is represented by 1.

2) *Hidden Layers:* In our system the operation of recognition handwritten Arabic characters used three hidden layers. However, choosing the number of hidden layers accurately is deterministic in a neural network structure. If neural network contain too many hidden layers classification presentation do not agreement higher, and too little hidden layers may fail in complete high recognizing rate.

3) *Output Layers:* Our neural network structure applied on 12 Arabic characters for this reason it has 12 output layers sequence represent each character giving to the necessity of character recognition target.

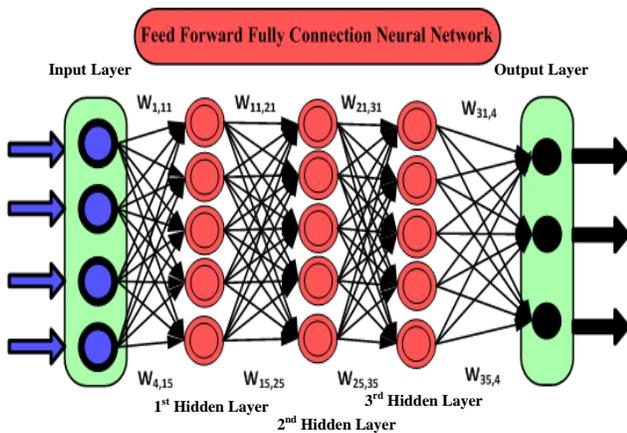


Fig. 8. Our proposed neural network structure

2) Back propagation algorithm

we give the step of Error Back Propagation Artificial Algorithm in section (4) used in this neural network structure in this section explain the equations compute the details ( $\delta$ ), the change of weights and updating weights is that after training stopping, In last iteration's weights should be stored as neural

network weights, because it is having the lowest error which is needed to stop the training [14].

3) Running Neural Network Structure

For running the error back propagation learning process, always need set of training characters, input, and output(target), learning rate value, condition that stopping the algorithm, updating weights, nonlinearity function, in our neural network structure ,the activation sigmoid function sigmoid (bounded, monotonically increasing and differentiable) of hidden and output layers( neurons)are used. Each time all the characters in our problem have been used once in the network during training is called an epoch, small random values for Initial weight .Suppose we wanted to train a network to recognize for example the four Arabic characters (ت، غ، ك، م) from 12 Arabic characters (ي، م، ه، و، ف، ك، م، ي، ا، ت، ج، ذ، ش، ط، غ، ف، ك، م، ي) as shown in below Figure (9).

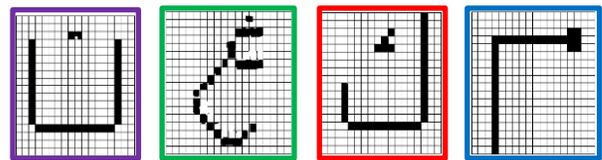


Fig. 9. the four Arabic characters (ت، غ، ك، م)

To train our neural network apply the first character (ت) and change all the weights in the neural network once, which explain in Figure (5). Next apply the second character (غ) and do the same, then the third character (ك) and so on. Once we have done all four characters above, return to the first one again and repeat the process until the error becomes minim which means that it is recognizing all the characters. Figure (10) explain how our neural network should work.

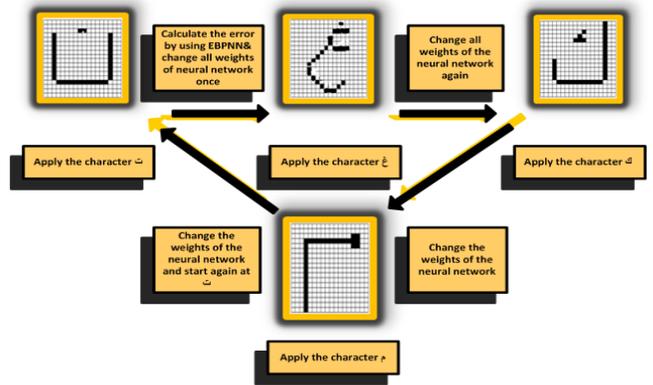


Fig. 10. Running our Neural Network Structure for four Arabic characters (ت، غ، ك، م)

V. RESULTS AND DISCUSSION

Training with validation According to the above observation, we add the restraint condition, that if error of validation is lower than 0.05, any increasing after will terminate training process. With this constrain, hopefully higher successful rate will be obtained, along with training time saving. We run the neural network for10 times, on12 Arabic characters (أ، ت، ج، ذ، ش، ط، غ، ف، ك، م، ي، م، ه، و، ي) showing data in Figure (11) and Figure (12). The average recognition rate is 93.608%,

with average training time 36.18 seconds. Average training epoch is less than 50 iterations. Validation gives us better neural network successful rate as well as optimal training time. For each character, the neural network calculate its recognition rate, also we are showing from the Figure (13), that some characters have recognizing rate as high as 100%, but the worst case character ط can just achieve 75.520%.

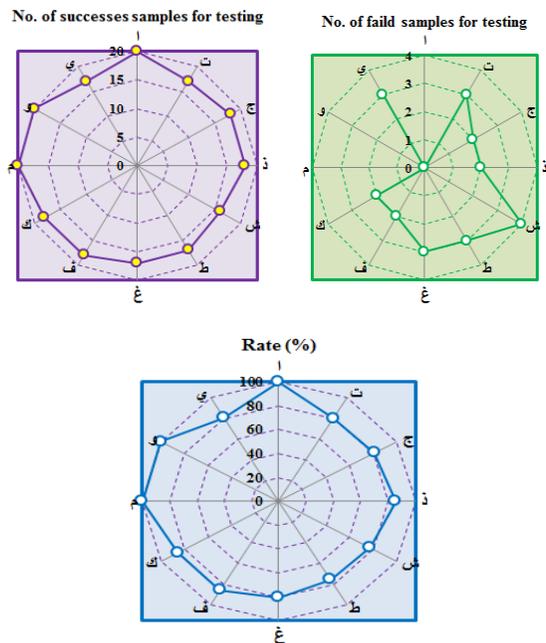


Fig. 11. Final recognition rate obtained from testing our neural network structure

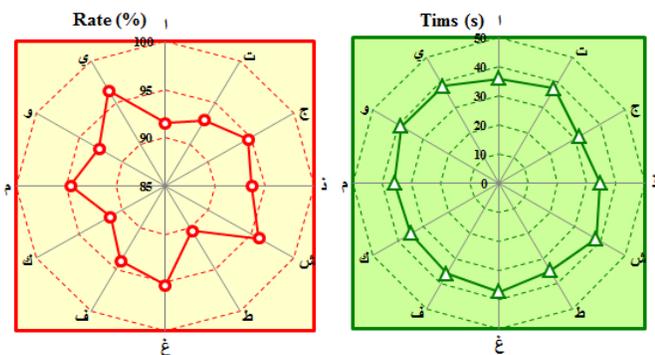


Fig. 12. Successful Classification Rate and Training Time

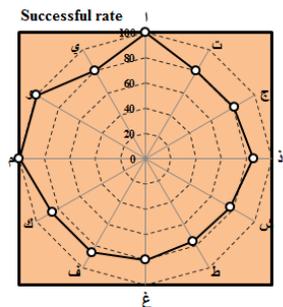


Fig. 13. Successful rate for each character using trained neural network

### A. Effect of hidden layers number

The most commonly structure of any neural network consist of an input layer, hidden layer (layers), output layer. Number of input and output layer depend of the number of characters we want to process by our neural network and the number of characters we want to recognize and how we want to code these characters appear ,for this we must have 13 inputs characters to train our neural network because we have 12 classes for 252 Arabic characters each class contain 20 sample for every character from 12 these characters (ا،ت،ج،ذ،ش،ط،غ،ف،ك،م،و،ي) and one class for Twelve Arabic character used for matching the recognize character when applied our neural network. for the number of hidden layers and number of their cells play important role in neural network structure because the hidden layer(s) and its (their) cells afford the network with its ability to generalize, there are no hard and fast rules for this to make the network typically works well. Theoretically neural network with one hidden layer with a sufficient number of hidden cells is capable of approximating any continuous function. Practically neural network with one and sometimes two or three hidden layers are commonly used and have well preformation. For example if we have three layer network with n input and m output cells can calculating number of hidden cells by geometric pyramid rule , described form below [15]:

$$N = \sqrt{n + m} \tag{7}$$

Where:

N: Number of cells in hidden layer.

Our neural network structure which explain in section (4.2) by Figure (8) when we want to recognize characters on 18x18 array of binary pixels image , we have 12 output cells (one for each characters from 12 Arabic characters we want to recognize). The network will train to recognize all 12 Arabic characters with anywhere between 2 and 7 hidden layer, 2 layers and the network hasn't got enough weights to store all the characters, and more hidden layers need more weights in between each two layers. So updating each weight will take more time. As a result, 3-4 hidden layers may have both good performance and good converging time, above 7 layers the network becomes inefficient and doesn't perform as well. So, the number of hidden layers and its cells needs to be experimented with for the best results. We do experiments with 2, 3, 4,5,6 and 7 hidden layers, from Figure (14) , we can see that large hidden layers number leads to long training time, but did not improve identifying performance significantly. Small hidden layers number also extend the training period, and lower successful rate as well. 2 hidden layers takes more than 60 iterations to converge, while the other situations having less than 40 iterations. 4, 5, 6 and 7 hidden layers have very similar rate as 3 hidden layers but takes more time to train. It is easy to understand that, more hidden layers need more weights in between each two layers. So updating each weight will take more time. As a result, 3-4 hidden layers may have both good performance and good converging time.

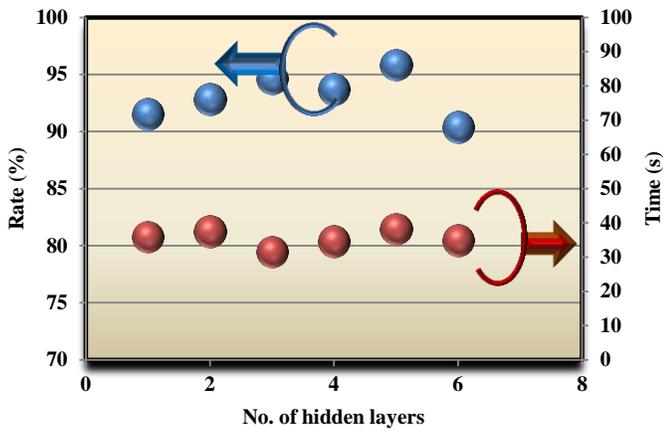


Fig. 14. Effect of hidden Layers number

### B. Effect of learning rate

The amount of corrective modifications applied to weights in EBPANN is called the learning rate which representation by the symbol  $\eta$  and is usually fixed a value between 0 and 1, so that large value of learning rate indicates large delta weights. Usually in our neural network structure choose 0.1 as the default value. And try learning value range from 0.05, 0.08, 0.1, 0.3, 0.5 and 0.7 to evaluate its effect in recognition rate and training time. After run this experiment for a number of times, we find that the minimal error of justification set is larger than 0.05, so 0.05 cannot agreement convergence. Figure (15) shows, smaller value of learning rate 0.08 has very similar act with 0.1, but large learning rate 0.3 and 1.0 will increase training time validation and decrease successful recognition rate. When learning rate equals 0.05, 0.08 and 0.1 the minimal error values are very close, also show the successful recognition rates of those three are also close. Therefore, 0.1 is appropriate rate for this knowledge base and neural network training.

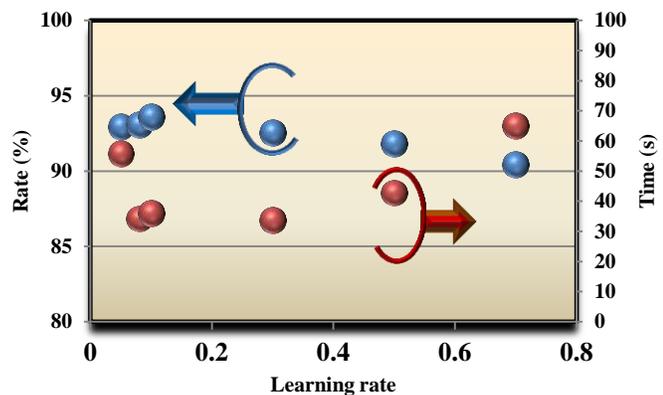


Fig. 15. Effect of Learning Rate

From testing with learning rate 0.1, hidden layers three pieces and about 252 feature vectors, this system had a level of accuracy of 100% for learning data, 80% for learning and non-learning data, and 75.520% for non-learning data

## VI. CONCLUSION AND FUTURE WORK

In this neural network structure error back propagation artificial neural network is implemented to recognize 12 offline isolated Arabic handwritten characters. As discussed above, successful recognition rate is about 93.61%, we observed in the experiments. The use of Error Back Propagation Artificial Neural Network (EBPANN) proved to be as one method to recognizer Arabic characters based on texture features with accuracy percentage at 93.61%. Determination of the learning rate value at 0.1, three hidden layers, and 132 feature vectors, all the weights are initialized to be small random number between -0.01 and 0.01, Each Arabic characters, has 20 samples for training, and testing, respectively. Can be stated that our neural network structure has been successfully calculate the learning data and less successfully for non-learning data and easy to implement and easy to compute learning rate for both hidden and output layer which modifies the values of weights and increases the convergence speed. We would like to improve the recognition accuracy percentage rate for handwritten Arabic text, which differ from the machine printed case and is more similar to cursive Latin handwritten text and cursive words or any other type of handwritten words, in our future work. By using a good feature extraction system along with other neural network classifier performs better than a single classifier.

### REFERENCES

- [1] M. K. Mohammed Altahaf, M. B. Begum, "Handwritten Characters Pattern Recognition Using Neural Networks", International Conference on Computing and Control Engineering, 2012.
- [2] Detlef Nauck. Neuro-Fuzzy Systems: Review and Prospects. In Proc. Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), pp. 1044-1053, Verlag und Druck Mainz GmbH, Aachen, 1997.
- [3] Nauck, D and Kruse, R., Neuro-fuzzy systems for function approximation, Proceedings of 4th International Workshop Fuzzy-Neuro Systems, 1997.
- [4] Anita Pal, Dayashankar Singh "Handwritten English Character Recognition Using Neural Network" International Journal of Computer Science and Communication , Vol. 1, No. 2, pp 141-144, 2010.
- [5] Sumit Goyal and Gyandera Kumar Goyal, "Cascade and Feedforward Backpropagation Artificial Neural Network Models For Prediction of Sensory Quality of Instant Coffee Flavoured Sterilized Drink". Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition Vol. 2, No. 6, August 2011.
- [6] Zweiri, Y.H., Seneviratne, L D., Althoefer, K.: Stability Analysis of a Three-term Back-propagation Algorithm. J. Neural Networks. Vol 18, pp. 1341--1347, 2005.
- [7] Efe MO , " Novel neuronal activation functions for feed forward neural networks," Neural Process Lett , Vol. 28, pp. 63--79, 2008.
- [8] A. Shawkat, K.A. Smith-Miles, "Improved Support Vector Machine Generalization using Normalized Input Space," LNAI, Springer-Verlag, Heidelberg, pp.362-371.
- [9] M. Kumar, R. K. Sharma, M. K. Jindal, "SVM based offline handwritten Gurmukhi Character Recognition," in Proc. of SCAKD, pp. 52-63, 2011.
- [10] J. Pradeep, E. Srinivasan and S. Himavathi, " Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network", International Journal of Computer Science & Information Technology , Vol 3, No. 1, 2010.
- [11] Ashoka H N, Manjaiah D H, Rabindranath Bera " Feature extraction Technique for neural network based pattern recognition" International journal of computer science and engineering , Vol. 4, No. 3, pp 331-339, 2012.

- [12] Kundu, A., McLean MITRE Corp., T. Hines, J. Phillips and B.D. Huyck., "Arabic Handwriting Recognition Using Variable Duration HMM." ICDAR. Ninth International Conference on Document Analysis and Recognition, 2007.
- [13] Dayashankar Singh, Sanjay Kr.Singh, Dr.Maitreyee Dutta. "Handwritten character recognition using twelve directional feature input and neural network". International Journal of Computer Applications, Vol 1-No.3, pp.82-85, 2010.
- [14] M. Z. Rehman , N. M. Nawi , " Improving the Accuracy of Gradient Descent Back Propagation Algorithm (GDAM) on Classification Problems," International Journal on New Computer Architectures and Their Applications, Vol. 1, No. 4, pp.838-847, 2011.
- [15] S.N. Nawaz, M. Sarfraz, A. Zidouri, W.G. Al-Khatib, "An Approach to Offline Arabic Character Recognition using Neural Networks", in Proc. of 10th IEEE International Conference on Electronics, Circuits and Systems, 3, pp. 1328-1331, 14th-17th Dec. 2003.