Buhuth Mustaqbaliya (14) 2006
PP. [45-60]

Recognition of Handwritten Arabic Alphanumeric Characters by Backpropagation Neural Networks

Rafid Ahmed Khalil (1)

الملخص

يقدم هذا البحث النتائج المستخلصة من استخدام نموذج الانتشار العكسي BP للشبكات العصبية في تطبيق تمييز الحروف والأرقام العربية المكتوبة باليد. استخدمت طريقة جديدة في استخلاص الخصائص معتمدة علي إسقاط الظل. وتم تدريب الشبكة باستخدام أنماط من حروف وأرقام مكتوبة من قبل أشخاص مختلفين، هذه الأنماط تدعي (مجموعة التعلم). والمطلوب بعد إتمام هذه العملية، تمييز أنماط جديدة من خارج مجموعة التعلم. كما يتضمن البحث تقييماً لقابلية الشبكة في تمييز 28 حرف ورقم يدوي.

Abstract

This paper presents the results obtained by applying Back propagation (BP) neural network model, to application of handwritten Arabic alphanumeric characters (HAAC) recognition. A novel method for features extraction, based on a shadow projection has been used. The network is trained using alphanumeric character samples written by different people (learning set). They are required, after the learning is over, to recognize alphanumeric characters outside the learning set. Also, the paper includes an evaluation of the recognition capability of the BP model for 28 alphanumeric characters.

(1) Ph.D., College of Engineering, Univ. of Mosul, Iraq.

Received: 4/5/2005
Accepted: 29/6/2005

1. Introduction

Character recognition has been an active area of pattern recognition not only because it improves man-machine communication, but it also provides a solution for editing, sorting, indexing, searching, and other text processing tasks of large volumes of data automatically [1-3].

Much literatures has been published on Arabic characters recognition. There is a big difference between Arabic and Latin alphabet based languages [4]. Gallinari [5] has investigated the recognition of handwritten digit using BP algorithm. Kamel et. al. [6], have proposed multiple classification architecture for handwritten Arabic characters recognition. Rafid A. Khalil [7], and Sharaidah [8], has reported on experiments in handwritten Arabic character recognition using neural networks. Amin [9] has presented a survey of Arabic characters recognition.

This paper is organized as follows: section 2 describes the use of Backpropagation (BP) neural network algorithm in patterns recognition. HAAC recognition procedure is presented in section 3,experimental results of the BP learning algorithm is presented and discussed in section 4, and section 5 is dedicated to conclusions.

2. Learning by Backpropagation (BP) Algorithm

The network typology of BP algorithm is illustrated in Figure1.

It consists of a 3-layer fully connected feed forward network. The algorithm is used to train this network having feature maps as inputs and alphanumeric character classes as outputs.

The nonlinearity most often used is the sigmoid function and can be written as:

$$f(u) = \frac{1}{1+\exp(-\beta x)} \qquad \text{.............................} \; 2.1$$

This function is continuous and varies monotonically from 0 to 1 as u varies from $-\infty$ to ∞. The gain of the sigmoid β, determines the steepness of the transition region of the function [10].

The notation used in the BP algorithm is as follows:

Notation *Meaning*

l.th neuron (node) in layer k .... output of the $y_l(k)$
th output. k …………… desired (target) response of the $t(k)$

th component of the input vector (pattern). i ……..
$x(i) = y_o(i)$
. l th neuron in layer k …………… error of the $\delta_l(k)$
$\vartheta_l(k)$ …………… external input of the k th neuron in layer l .

$w_l(j,i)$ ………… weight which connect the i th neuron in layer l -1 to the j th neuron in layer l .
L .... number of layers.
N_l ……………….. number of neurons in layer l.
m, β ……………... recursion index, and sigmoid gain.
ε, α ……………… updating value, and momentum term.

Where : $0 \le \varepsilon \le 1$, and $0 \le \alpha \le 1$.

The BP learning algorithm applied to the HAAC recognition problem can be stated as follows:

Step 1. (a) specify network architecture: number of layers, and number of neuron in each layer.

(b) set the algorithm parameters: random value generator,maximum number of recursion, maximum tolerance error, $\beta, \varepsilon, and \alpha.$

(c) initialize the weights and external inputs to small random values.

Step 2. select running or learning phase.

Step 3. if learning phase, then:

(a) apply the input patterns and targets to the network as a learning set.

(b) outputs calculation: the forward pass calculate the outputs of all layers from the formula:

$$y_l(k) = f\left(\sum_{j=1}^{N_{l-1}} \left\{ w_l(k,j) * y_{l-1}(j) - \vartheta_l(k) \right\}\right) \quad \text{..............2.2}$$

where $1 \le k \le N_{l,}$ **and** $0 \le l \le L-1.$

(c) error backpropagated: the backward pass if neuron k is an output neuron, then:

$$\delta_{L-1}(k) = y_{L-1}(k) * (1 - y_{L-1}(k)) * (t(k) - y_{L-1}(k)) \quad \text{........2.3}$$

If neuron k is not an output neuron, then:

$$\delta_l(j) = y_l(j) * (1 - y_l(j)) * \sum_{k=1}^{N_{l+1}} \delta_{l+1}(k) * w_{l+1}(k,j) \quad \text{.........2.4}$$

where $L-1 \ge l \ge 1.$

(d) weight updating formulation:

$$w_l(i,j)^{(m+1)} = w_l(i,j)^{(m)} + \varepsilon * \delta_l(j) * y_{l-1}(i) + \alpha(w_l(i,j)^{(m)} - w_l(i,j)^{(m-1)}) \quad \text{...2.5}$$

(e) repeat by going to (a).

Step 4. if running phase, then:

(a) calculate the outputs of all layers repeating step 3 (b) exactly (the forward pass).

(b) select the output with highest value and replace it with one. Replace all other outputs with zero. Point to the class identifying the alphanumeric character.

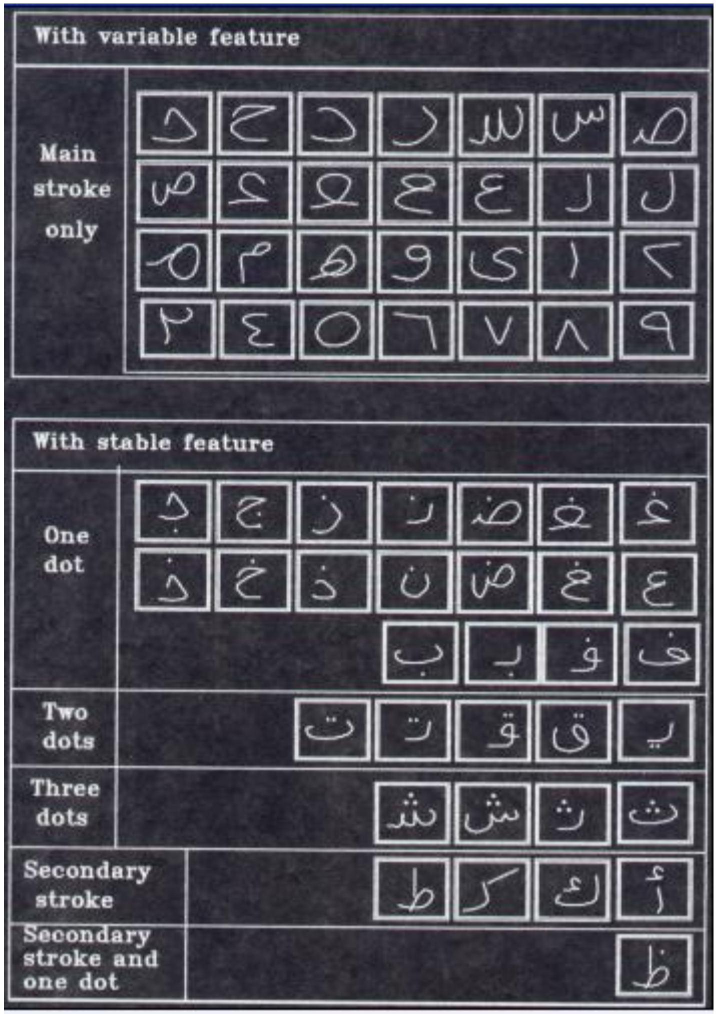
(c) repeat by going to step 2, unless exit is specified.

3. Handwritten Arabic Alphanumeric Characters (HAAC) Recognition Procedure

3.1 Arabic Alphanumeric Characters

The Arabic alphanumeric characters consist of 28 basic characters and 10 numerals. The Arabic characters differ from other systems of characters in their structure and in the way they are connected to form words. The same character may take different shapes according to its position in the word. This feature increases the number of Arabic characters to about 60 different shapes. Table (1) gives a classification of the Arabic alphanumeric characters according to their features. These features are divided into stable and variable categories, and a different representation (consequently a different classification strategy) is needed. In general, the number of secondary strokes and the number of dots and their relative positions are stable features of Arabic characters, while the structure of the main strokes is the most variant portion of the characters [7].

Table 1. Arabic alphanumeric character classification



3.2 The Database

In our experiments of HAAC recognition, 28 alphanumeric characters are used. These characters are of the main stroke only (see Table 1). Ten different samples of HAAC written by 10 different people each of which of consists of 28 alphanumeric characters were collected. The resulting database has 280 alphanumeric patterns. These patterns are used as learning set. It is necessary to perform some additional preprocessing to normalize the shape of the alphanumeric characters. The bitmaps are scaled to fit into a square window. Some scaled samples taken from the database are shown in figure 2.

The database is used to generate epochs for the learning phase of the BP, where an epoch is defined as "one presentation of the entire set of 280 learning patterns". In epochs, the patterns from each person may be presented in a random ordering, otherwise the system may tend to memorize the output for latest cases forgetting some of the previously given patterns.

In this experiment, the collected patterns are photographed by digital scanner in order to display it on the screen. The object here was to change the representation from bitmaps to feature maps. This is achieved after scaling process of the bitmaps to fit into fixed size window. The scaled alphanumeric characters are then processed by software to obtain a shadow projection value on segments bar mask (i.e., feature representation).

3.3. Shadow Codes for HAAC

Handwritten pattern recognition systems are known to perform more accurately and efficiently when input features are well selected and encoded. A particular encoder which satisfies these requirements is based on a shadow projection of 16-segment mask array as shown in Figure 3(a).

An input alphanumeric character is first normalized so that it extends the full height and width of the bar mask. A shadow projection operation which is defined as simultaneously projects a point into its three closest vertical, horizontal, and diagonal bars. A projected shadow turns on a set of bits distributed uniformly along the bar. After all the points on a character are projected, the number of on bits in each bar is counted. The character is represented by these 16 numbers, which we refer to as the shadow codes.

An example showing the encoding of the character "hah ح" is illustrated in figure 3(b). Code values ranging from 0 to 100 are normalized to the range (0, 1) for the neural network.

4. Experimental Results

4.1. Recognition Rate

The HAAC recognition problem outlined in the previous sections is implemented on a microcomputer based system. The behaviors of BP are simulated on P4, 2.4 GHz full cache processor PC using C^{++} language program. The program runs under MS-DOS and was compiled with Borland C^{++}. After the training is completed, the network is required to recognize any input pattern (alphanumeric character) out of the learning set patterns. Figure 4, shows four different samples of alphanumeric characters written by four different people used as test set. Then the recognition rate of the system is given as :

Recognition rate $\Re = \frac{\Im}{\aleph} * 100$ 4.1

Where $\Im$ the number of is correct recognition, and $\aleph$ is the total number of recognition.

4.2 Algorithm Performance

The BP algorithm is used to train classification network with single hidden layer for HAAC. The network has 16 input neurons corresponding to 16 elements input (feature) vectors, and 28 output neurons, one for each alphanumeric class. The desired (target) vector contains exactly 15 zeros and a single one, whose location is a key to the recognized character. Output values in the range (0- 1) are obtained experimentally. Therefore, the output with the largest value is chosen as a key to the recognized alphanumeric character.

This approach is based on a gradient search that may find a local minimum in the error surface instead of the desired global minimum in the error surface instead of the desired global minimum [10]. This problem corresponds to identifying two or more alphanumeric characters as one, and can be minimized using multiple starts with different random weights and a low updating function to adapt weights.

In the learning phase, approximately 100 epochs are applied (see figure 2), for which the error is calculated and weights are adjusted repeatedly until the error is reduced to 0.0001. The random weights and external input are typically initialized to small random values. This start the search in a relatively "safe" position. The random value generator is set in the interval (0.01 – 0.1) to prevent overflows. Furthermore, the sigmoid function uses a steeper with a multiplicity of $\beta = 3$, which results in a better distinguishment of the highest output than the other outputs, producing output values in the range : 0.01 – 0.97.

Another important point is the number of neurons in the hidden layer. When 48 neurons are used instead of 64 neurons in the hidden layer, oscillation may occur at 60-th epochs during learning phase. Note that the number of hidden layer neurons should always be less than the number of learning patterns, otherwise the network will simply memorize the learning patterns resulting in poor generalization. Finally, the system is very sensitive to input pattern characteristic. If the examples of the learning set have many patterns with similar feature, the satisfactory recall rate degrades.

In order to demonstrate the ability of the system, the four different samples of HAAC shown in figure 4 are used as the test set. Table (2) shows the experimental results for BP obtained by applying the above four samples and the recognition rate calculation described in Esq. (4.1). The parameter setting that we have used in these experiments are as follows:

Neurons in the input layer : = 16

Neurons in the hidden layer: = 64

Neurons in the output layer: = 28

Updating value: = 0.02

Sigmoid gain: = 3

Random generator: 0.01-0.1

Momentum term: = 0.8

Number of weights: = 2944

Patterns processed in the learning phase: = 28,000

The 28,000 patterns of the HAAC are needed approximately during the learning phase (as a rule of thumb [10]) to achieve acceptable level of recognition rate as shown in Table (2).

Table 2. Experimental results of BP algorithm applied to HAAC recognition problem.

Test Set HAAC Samples	Recognition Rate %
Sample A	80
Sample B	72
Sample C	92
Sample D	33

4.2 Evaluation of Performance

Depending on the experimental results and statistics gathered by different handwritten styles, the BP algorithm produced satisfactory results. The algorithm consumes approximately 92 neurons. For the number of operations per pattern, the BP takes 3000 interconnects per pattern for the running phase and approximately 8816 interconnects per pattern for the learning phase. From time consumption point of view, the algorithm spends 600 second in learning all 28 HAAC. The overall throughput including image acquisition is approximately 10 patterns per second. This value is limited mainly by the scaling and shadow projection steps.

When the recognition rate is considered, the system produced a satisfactory and approximately the same result in samples A,B, and C of the test set (see figure 4). The low recognition rate result in sample type D is not unexpected, since we have supplied the networks with such information that the system could not distinguish the HAAC out of so many similar features.

5. Conclusions

This paper has successfully applied BP algorithm to a large, complex task. The results appear to be the state of the art in HAAC recognition. It has showed that the networks can be learned on a shadow codes of HAAC that have minimal preprocessing (as opposed to other elaborate feature extraction methods [11][12]). Because of the redundant nature of the data and the constraints imposed on the networks, the learning time has been relatively short considering the size of the learning set. Scaling properties have been far better than the expected just from extrapolating results of BP network on smaller artificial problems.

References

1. Amin A., (1997), Handbook of Character Recognition and Document Image Analysis". World Scientific Pub. Co.
2. Al-Ohali Y. et. al., (2000), Arabic Character Recognition: Progress and Challenges. Journal of King Saua University. 12, Comp & Inf. Sci. 85-116.
3. Khedher Z., Abandah G., (2002), Arabic Character Recognition using Approximate Stroke Sequence. URL: http://www.al-mishkat.com/.
4. Cowell J., 2000, Character Recognition of Arabic and Latin Scripts. Proceeding of the conference on Information Visualization.
5. Gallinari P. , (2003), An MLP architecture for offline handwritten digit recognition. Int'l J. Doc. Anal. Rec. 5(4).
6. Kamel M. S., 1999, Multiple Classification Hierarchical Architecture for Handwritten Arabic Character Recognition. Pattern Analysis 6(5) pp. 2834-2838.
7. Khalil R. A. , (1997), A unified systolic architecture of artificial neural networks and applications to handwritten Arabic alphanumeric character recognition, Ph. D. Thesis, Univ. of Technology, Baghdad, Iraq.
8. Sharaidah M. A., 2000. Recognition of Handwritten Arabic Characters via Neural Networks, M. Sc. Thesis, Jordan.
9. Amin, 1997, Off line Arabic character recognition – a survey. In ICDAR 97 pp. 596-599.
10. Hush D. , and Horne B. , 1993, "Progress in Supervised neural networks". IEEE Signal processing Magazine.
11. Baasli H. , 2003, A noval feature extraction technique for the recognition of segmented handwritten characters. In ICDAR pp. 137,141.
12. Amin, 1998, Preprocessing and Structural Feature Extraction for a Multi-font Arabic/ Persian OCR. Proceedings of 5th International Conference of Document Analysis and Recognition.

Figure 1. The network topology used in BP algorithm.

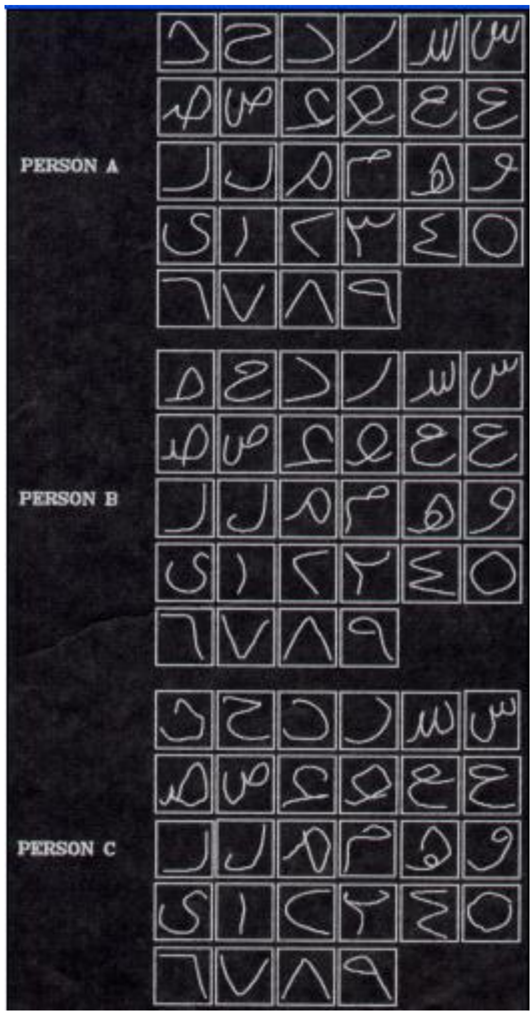
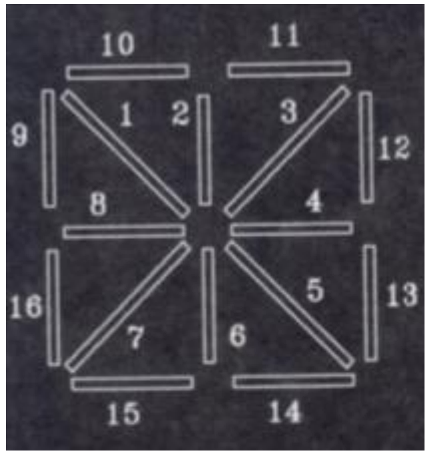
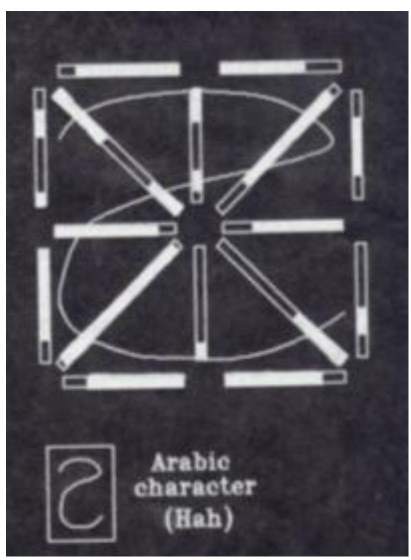


Figure 2. Some samples of Arabic alphanumeric character written by Different people used to train BP network.



(a)



(b)

Figure 3 (a) 16-segment bar mask used for HAAC recognition.
(b) Illustration of encoding the character (Hah ح), with shadow projections shown in white.

- **The shadow code obtained by scanning bars in sequence is (69, 43, 71, 78, 42, 20, 94, 86, 40, 86, 75, 56, 26, 80, 80,76).**
- **Each number represents total length of the corresponding Shadow.**

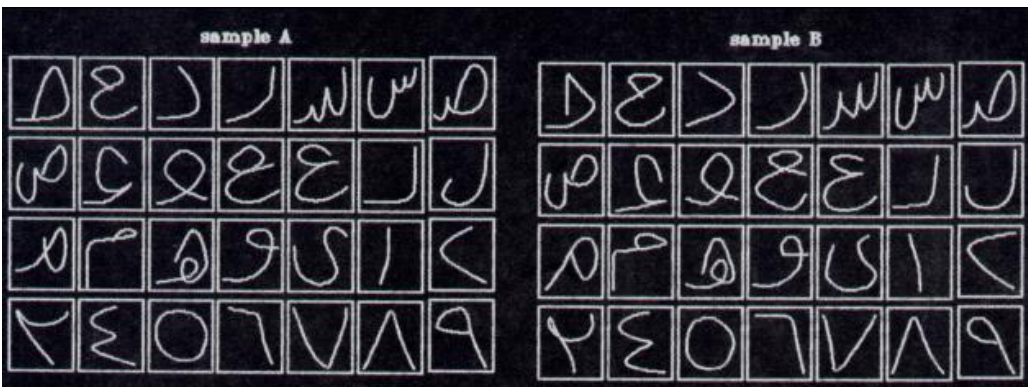


Figure 4. Four different samples of HAAC written by four people
(samples A-D) used as test set for BP algorithm.