Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp46-51)

Recognition Persian Handwritten Digits Using Templates and Back-propagation Network with Adaptive Learning's Rate

KAMBIZ RAHBAR[1], SAEID RAHATI QUCHANI[2], MUHAMMAD RAHBAR[3]
[1]Young Researchers Club, Islamic Azad University of Mashhad, Email:
Islamic Republic of IRAN
[2]Computer Department, Islamic Azad University of Mashhad
Islamic Republic of IRAN
[3]Islamic Azad University of Quchan, Islamic Republic of IRAN

***Abstract*: -** This paper studies the recognition of Persian handwritten characters using templates and back propagation networks. The last one is learned by gradient decent learning law which was promoted by adaptive learning rate and momentum. Different technical methods which are often based on artificial neural network or neuro-fuzzy ones are used in recognition characters. Often the whole data is squeezed in the aforementioned networks which are to classify the data to each of existed classes. However, in this paper the templates are used for the primary classification of data. So, using templates leads to the recognition of some squeezed data and classification of remaining one into smaller common classes before feeding them into neural network; then the neural network is used for final classification. The results show that there are some significant improvements on recognition performance. This happens because decreasing input and output space causes to have a simpler mapping between input and output which in turn lead to increasing learning rate and decreasing classification error. Recognition rate on our dataset is almost 100%.

***Keywords*: -** Persian handwritten digits recognition; Templates; Back-propagation neural network; Adaptive learning rate; Momentum; Pattern recognition

1 Introduction

Today character recognition is among the subjects which receive special attention [1, 2, 6, 7]. Different projects were handled on this respect that most of them are based on neural or neuro-fuzzy networks [1, 4]. But since the network input space for characters' classification are often huge, the recognition process can not be performed completely in full scale; even though the learning time is lengthened. Here, we first classify the characters primarily by using templates. In this phase some of them are completely recognized without the need of neural networks and some are placed within a smaller common class. Next, recognition is performed with certainty and high precision due to the reduction of neural network input space (large common class is classified into small classes having more common categories). Consequently, the reduction of input space leads to the reduction of learning time. We applied this method for more than 100,000 Persian handwritten digits; this data was the scores lists of Islamic Azad University of Mashhad which were scanned in the scores' lists of 2338 x 1648 grayscale images. For learning and testing process, 80% of the data was used randomly (40% each). 10% of the remaining dataset was used as checking data. The learning process was accomplished through using less than 4% of the learning data.

The remaining parts of the paper are organized as follows: section 2 explains the methodology. Section 2.1 to 2.5 includes the details. Section 3 shows the experimental results and section 4 gives the conclusions of this paper.

2 Handwritten characters recognition using templates and back-propagation network

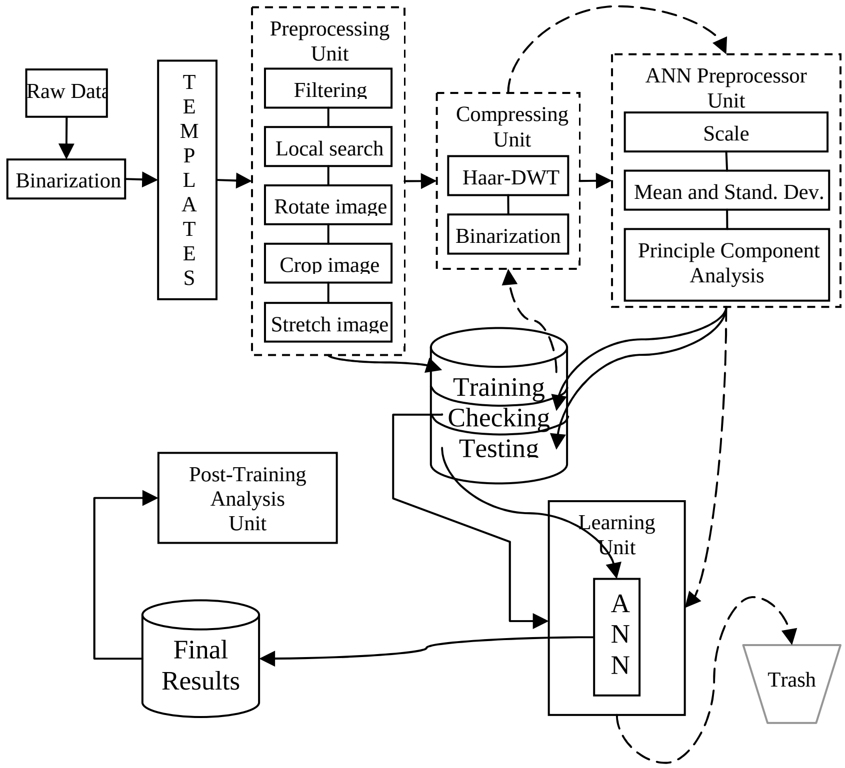

Fig.1. System block diagram

Fig. 1 shows the general diagram of character recognition. Here the recognition features of character binary images are under the consideration. For recognition, first of all, scores' lists should be in the binary format, then, Arabic numbers should be extracted from them. Pre-process phases including images noise elimination and scores' lists correction in the way that figures placed on the predetermined areas, is the pre-requisites of this process. Here 40% of data is transferred to database randomly to be used for framing the learning network process. The remaining figures are classified by templates and sent to compressing unit. In this unit images are reduced by using the Haar wavelet transformation to $2/43$ of its original size (i.e. from 30×40 to 7×8); and then is transferred to ANN pre-process unit. To make neural network training more efficient data is corrected in this unit through performing certain processing steps on the network input and targets. Finally training units perform the training process by training data using data check. (Since it was not clear how much training data was needed, it didn't pass through the pre-determined phases. So it is necessary to pass through template, compressing, and ANN pre-process phases before entering the network.) After accomplishing the training test, data fed into the network and consequently recognition process begins. When recognition process is done, post-training analyses unit analyzes the network efficiency.

2.1 Primary pre-processing

2.1.1 Filtering

By considering the data qualities, complex filters are not needed. Only the filter mask (Fig. 2) is used for eliminating the individual spots [3].

-1	-1	-1
-1	8	-1
-1	-1	-1

Fig.2. filter mask

2.1.2 Local search

Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp46-51)

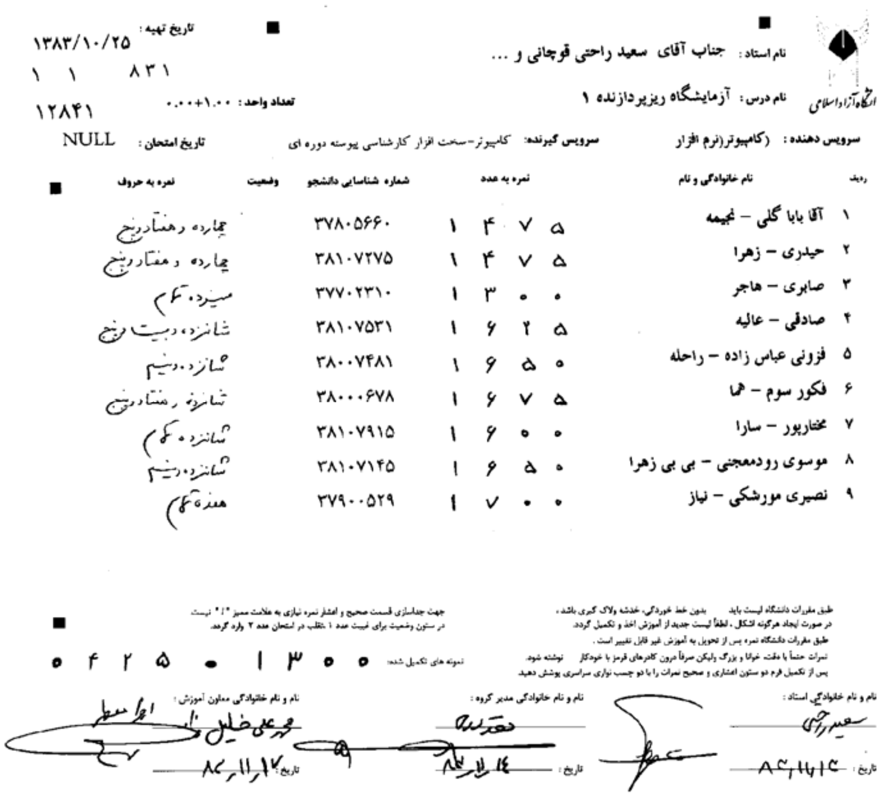

Fig.3. Islamic Azad University of Mashhad Sample form

Four squares sized 30 x 40 pixels are printed on different parts of the scores' lists (Fig. 3). Local search is identified based on the pre-determined sample square. A search is done on the scores' lists in an area where the existence of one of the squares is expected. This search selects and identifies the coordinates of square that has the most correlation with pre-determined one based on two-dimensional correlation.

2.1.3 Scores' lists correction and extracting numbers

Based on the obtained information from the position of existed squares, scores' lists is rotated and moved in the way that squares are positioned in the pre-determined coordinates. Now it is expected that numbers are located in pre-identified areas. Numbers are extracted in binary format and in the form of 30 x 40 pixel images. Then become stretch so that its expansion fills the whole space of matrix. Stretch is performed based on cube interpolation. Fig. 4 shows the image of the number before and after stretch.

(a) (b)

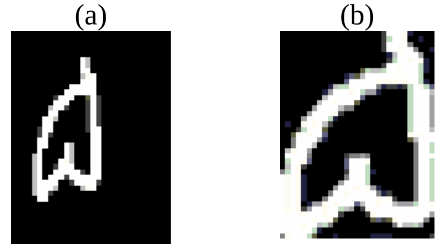

Fig.4. (a) Before stretch (b) After stretch

2.2 Templates

The new idea is that if the pre-classified learning data piling on each other for each class independently then we will have a cube that each of its surface belongs to the image of written number by individuals. If we consider the number of corresponded pixels of the images of demonstrated number as the corresponded cost for its column, then a two-dimension matrix is produced which shows that many people at the time of writing that number, move their pencils on the same path. This matrix can be taken as a template after thresholding (Fig. 5).

(a) (b)

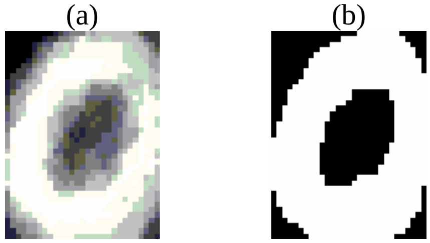

Fig.5. (a) Before thresholding
(b) After thresholding

Now the numbers images could be classified based on the idea that whether they are placing on the template or not. Notice that some different numbers could be placed in the common template after scaling. For example both number five and zero in Persian handwriting, which is shown in Fig. 6 after scaling, could be classified in the common class.

(a) (b)

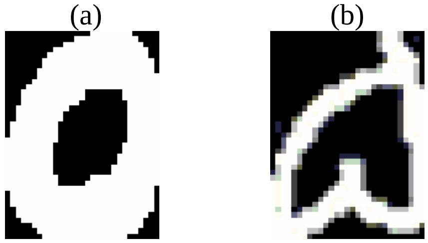

Fig.6. (a) Persian zero Template (b) Persian number 5

In this way numbers are classified in smaller common classes before a Neural Network classifies them, therefore, the network is only used for classifying each common smaller class members. This process increases the final answers certainty. Meanwhile by reducing the input and output of network space, mapping between input and output which is to be identified by network, becomes simpler. This simplicity in turn will increase the network learning rate and reduce network error. It should be mentioned that some numbers such as seven or eight in Persian handwriting will be recognized completely in this phase. Fig. 7 shows the sample of produced templates and the number which is recognized completely.

(a) (b)

Fig.7. (a) Template (b) Persian number 7

2.3 Compressing

Compressing is done based on Haar wavelet transformation [3]. From the transformation result only approximation is kept and details are discarded. By twice repetition of the activity, the 30 x 40 pixel images are decreased to 7 x 8. Note that because the result of transformation is in the grayscale format it should be returned to binary format again. Fig. 8 shows input and produced image after transformation is done.

(a) (b)

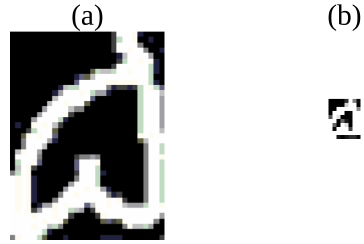

Fig.8. (a) Before transformation (50% of real size) (b) After transformation (200% of real size)

2.4 ANN pre-processing

Before squeezing the features into network, it is tried to correct them in a way that training algorithm can train the network more efficiently [5, 8, 10]. Corrections on data are done in the way that:

- Data is scaled between the ranges of ± 1.
- Having standard deviation and mean of normal distribution.
- Their sizes are to decrease.
- Is to be orthogonal for entering into the network.
- Arranged in the way that there is the least variability in their sequence.

2.5 Network training

Here a back-propagation network with 7 x 8 input space is used which has 40 neuron in hidden layer with tan-sigmoid transfer function (Fig. 9 - a) and N output neuron with linear transfer function (Fig. 9 - b).

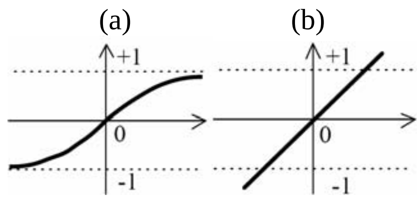

Fig.9. Neural network transfer function

Learning is based on gradient decent learning law which is speeded by adaptive learning rate and momentum. Here if the learning rate is made too large, the algorithm becomes unstable and if the learning rate is set too small, the algorithm takes a long time to converge. In this learning method as with momentum, if the new error exceeds the old error by more than a predefined ratio (typically 1.04), the new weights and biases are discarded. In addition, the learning rate is decreased (typically by multiplying by 0.7). Otherwise, the new weights, etc., are kept. If the new error is less than the old error, the learning rate is increased (typically by multiplying by 1.05). This procedure increases the learning rate, but only to the extent that the network can learn without large error increases. Thus, a near-optimal learning rate is obtained for the local terrain. When a larger learning rate could result in stable learning, the learning rate is increased. When the learning rate is too high to guarantee a decrease in error, it gets decreased until stable learning resumes. The learning algorithm combines adaptive learning rate and momentum. Momentum allows the network to ignore small features in the error surface. If the new performance function on a given iteration exceeds the performance function on a previous iteration by more than a predefined ratio (typically 1.04), the new weights and biases are discarded, and the momentum coefficient is set to zero. Note that if the number of parameters in the network is much smaller than the total number of points in the training set, then there is little or no chance of over-fitting. For this reason we were not worried for over-fitting in this project. Learning is preformed based on the aforementioned law to meet at least one of the stopped conditions. Stopped conditions are as follows:

- The number of iterations exceeds epochs.
- The performance function drops below goal.
- The magnitude of the gradient is less than predetermined value.
- If the training time is longer than predetermined time.

3 Experimental results

The usage data is 2338×1648 pixel images of Islamic Azad University score list. The leaning process is complete by using less than 4% of learning data (1600 data). Fig. 10 shows the network error during the learning process.

The training process is stopped after 3500 epochs. We found that the network error for testing data is almost 0% and 0% for total. The following table shows the classification results with and without templates. Comparing the two tables, it is observable that there are some significant improvements on recognition performance.

Table 1. Classification results with and without templates

	Test data error rate	**Reject rate**	**Learning time**	**Classification rate**
without templates	0%	12.5%	650 min	87.5%
with templates	0%	0%	65 min	100%

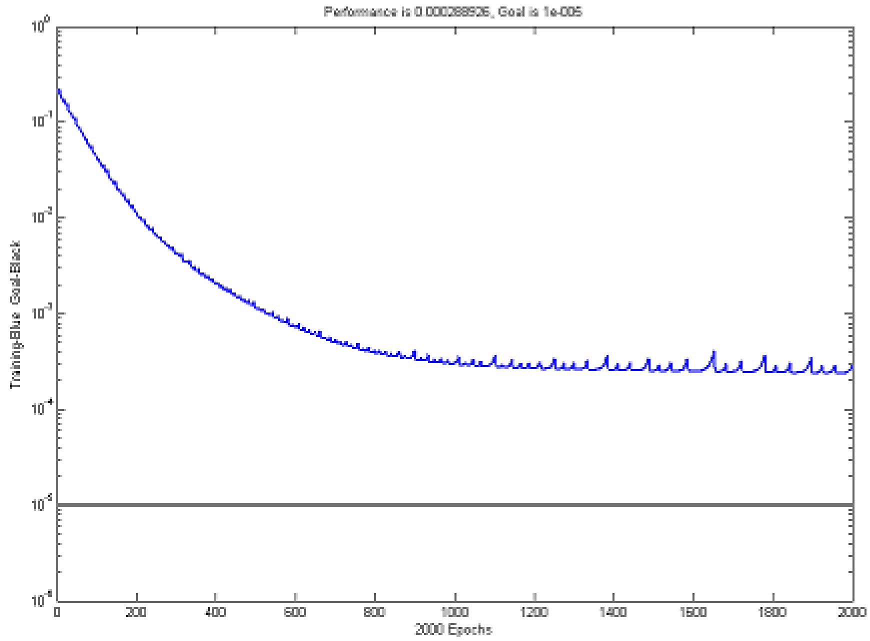

Fig.10. Learning error

4 Conclusion

In this paper the recognition of handwritten Persian digits were studied through introducing templates and using back-propagation neural network. By using this method, all characters are classified in smaller common classes before applying neural networks classifiers; then the neural network is employed for final classification. The results show that there are some significant improvements on recognition performance. This happens because decreasing input and output space causes to have a simpler mapping between input and output which in turn lead to increasing learning rate and decreasing classification error.

References:

[1] Aliev, R.A., Guirimov, B.G., Aliev, R.R., A Neuro-Fuzzy Graphic Object Classifier with Modified Distance Measure Estimator. Iranian Journal of Fuzzy System, Vol. 1, No. 1, (2004) 5-15.

[2] Chen, G.Y., Bui, T.D., Krzyzak, A. Contour-based handwritten numerical recognition using multiwavelets and neural networks. Pattern Recognition 36 (2003) 1597-1604.

[3] Gonzalez, R.C., Woods,R.E., Digital Image Processing. 2nd ed. USA: Prentice-Hall (2002).

[4] Hariri, M., Baradaran Shekohi, Sh, The application of neuro-fuzzy systems for pattern recognition. 5th Iranian CIS (2003).

[5] Le Cum, Y., A Theoretical Framework for Back-Propagation, Proceedings of 1988 connection 1st models summer school, Carnegie-Mellon University (1989).

[6] Le Cun, Y., et. al. Handwritten Digits Recognition with a Back-Propagation Network, AT&T Bell Laboratories, Holmdel, N.J. 07733

[7] Maryaz, G., Hinton, G.E., Recognition Handwritten Digits Using Hierarchical Product of Experts. IEEE Trans. Pattern Analyses and Machine Intelligent. Vol. 24, No. 2 (FEBRUARY 2002).

[8] Menhaj, M.B., Fundamental of Neural Networks. Iran: Politechnique press (2001).

[10] Orr, G., Muller, K. Neural Networks: tricks and trade, Springer, 1998.