A Novel Handwritten Letter Recognizer Using Enhanced Evolutionary Neural Network

Fariborz Mahmoudi, Mohsen Mirzashaeri, Ehsan Shahamatnia, and Saed Faridnia

Electrical and Computer Engineering Department,
Islamic Azad University, Qazvin Branch, Iran
{Mahmoudi,Mirzashaeri,E.Shahamatnia,SFaridnia}@QazvinIAU.ac.ir

Abstract. This paper introduces a novel design for handwritten letter recognition by employing a hybrid back-propagation neural network with an enhanced evolutionary algorithm. Feeding the neural network consists of a new approach which is invariant to translation, rotation, and scaling of input letters. Evolutionary algorithm is used for the global search of the search space and the back-propagation algorithm is used for the local search. The results have been computed by implementing this approach for recognizing 26 English capital letters in the handwritings of different people. The computational results show that the neural network reaches very satisfying results with relatively scarce input data and a promising performance improvement in convergence of the hybrid evolutionary back-propagation algorithms is exhibited.

Keywords: Handwritten Character Recognition, Neural Network, Hybrid Evolutionary Algorithm, EANN.

1 Introduction

Neural networks are powerful tools in machine learning which have been widely used for soft computing. The very first artificial neuron was introduced in 1943 by Warren McCulloch, a neurophysiologist, and Walter Pits, a logician, but due to the technical barriers no further work was made then. Since that time this topic has been attracted numerous of researchers and enormous improvements have been made to the subject. Artificial neural networks, ANN in short, are data processing techniques inspired from biological neurotic systems ambitiously aiming to model the brain. ANNs are popular within artificial intelligence applications such as function approximation, regression analysis, time series prediction and modeling, data processing, filtering and clustering, classification, pattern and sequence recognition, medical diagnosis, financial applications, data mining and achieving fine tuning parameters e.g. in fault-tolerant stream processing where balancing the trade off between consistency and availability is crucial [1, 2, 3, 4].

Within machine vision and image processing field, ANNs have been mostly applied to classification and pattern recognition [5]. Their special characteristics in being highly adaptive and learning make them suitable for comparing data sets and extracting patterns. Pattern recognition with neural networks includes a wide range

M. Sorell (Ed.): e-Forensics 2009, LNICST 8, pp. 1 – 9, 2009.

from face identification to gesture recognition. This paper focuses on English handwritten recognition. The learning process is implemented using a hybrid back-propagation neural network with genetic algorithm in which the convergence is important in recognizing the pattern.

Genetic algorithms are founded on bases of biological evolution model suggested by Darwin in 1859 under the theory of evolution by natural selection. GA was first introduced by John Holland in 1975 but was not wide spread until the extensive studies of Goldberg in 1989 published. Now, GA is a popular techniques due to its unique properties for complex optimization problems where there is no, or very little, information on the search space [6, 7].

Evolutionary algorithms' key feature is to find near-optimal answers in a complex search spaces. As a general search method they have been applied to many problems including classifiers, training neural networks, training speech recognition systems [8, 9, 10], in all these cases by properly characterizing the problem GA has been successfully employed.

This paper takes advantage of genetic algorithms. First the weights of neural network is generated randomly for a fixed number which is called initial population then by running the algorithm the population will converge to the goal.

The other core of this implementation is a neural network. Feed-forward network has been used for simulation. A typical feed-forward network consists of one input layer, one or more intermediate layer(s) called hidden layer(s) and one output layer. Each node in this network passes its data to the next node by an activation function. Different architectures can be designed for hidden layers but designing a successful architecture is problem dependent. It is known that if a network with several hidden layers can learn some input data, it can also learn those data with a single hidden layer but the time taken may be increased [11]. Our proposed approach addresses this problem.

The next section explains the feature vector extraction from handwritten character images and suggests a novel approach for character input for the neural network feeding. Section 3 describes the architecture of the neural network used and section 4 explores the hybrid genetic algorithm. Computational results and comparison between the proposed approach and conventional neural networks is provided in section 5. Finally, section 6 concludes this paper.

2 Preparing Input Data for Neural Network

As the name suggests, back-propagation network training is based on the propagation of errors to the previous layer. In this method, as data feed in the network, the network weights are accumulated and as the error is back propagated they are updated. Another method in training the network is by using evolutionary algorithms and specifically genetic algorithm for its convenience and suitability. Either of these methods has its own drawbacks. An adeptly designed hybrid approach seems to overcome these limitations and meanwhile exploits the advantages of both methods. The simulation results in section 5 demonstrate the truth of this claim. The back-propagation algorithm, BP in short, is vulnerable to local minima. By using genetic algorithms we

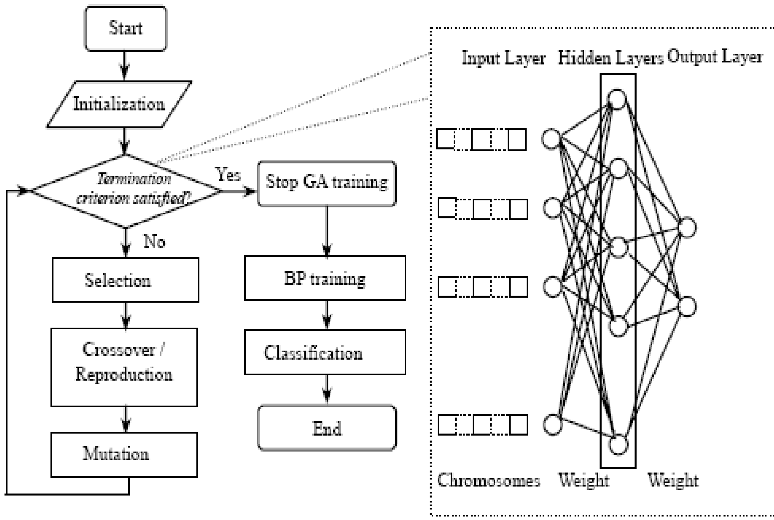

Fig. 1. General scheme of tuning neural network weights with GA

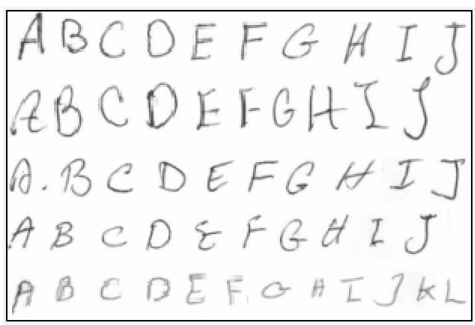

Fig. 2. Sample handwriting of 5 different persons

will overcome this issue; as the genetic algorithm searches fast the entire search space, the back-propagation algorithm is assigned to do the local search. Figure 1 illustrates the general concept of tuning neural network weight with genetic algorithms.

The input of the system is the scanned image of several different persons' handwriting of 26 English capital letters. Figure 2 represents sample different handwritten letters.

For preparing the input of the neural network, first the centroid of the scanned image of the letters is divided to four sections and the density of pixels in each section is calculated. The calculation of centroid and density is provided hereunder.

$$\mathrm{A} = \sum_{i=1}^{n} \sum_{j=1}^{m} B[i, j] \tag{1}$$

$$\bar{x}\sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j]=\sum_{i=1}^{n}\sum_{j=1}^{m}iB[i,j] \tag{2}$$

$$\bar{y}\sum_{i=1}^{n}\sum_{j=1}^{m}B[i,j]=\sum_{i=1}^{n}\sum_{j=1}^{m}jB[i,j] \tag{3}$$

$$\bar{x}=\sum_{i=1}^{n}\sum_{j=1}^{m}iB[i,j]/\mathrm{A} \tag{4}$$

$$\bar{y}=\sum_{i=1}^{n}\sum_{j=1}^{m}jB[i,j]/\mathrm{A} \tag{5}$$

$$COMPACTNESS=\frac{p^2}{\mathrm{A}}\geq 4\Pi \tag{6}$$

Where *A* denotes the area of the image and *B[i,j]* denotes one pixel of the image. *X* and *Y* give the centroid of object. In the equation (6), *p* is the perimeter and *A* is the area of the image, thus the greater this measure is, more compact the object is. According to this equation maximum compactness stands for circle and objects with other shapes are less compact than circle.

3 Architecture of Neural Network Core

The neural network used in this paper is based on the fully connected feed-forward networks demonstrated in figure 3. The input layer consists of four nodes and the hidden layer is divided in two layers, each with ten nodes. The output layer has 26 nodes, each representing one English capital letter. With the provided settings only a single output node will be active in the network for each input.

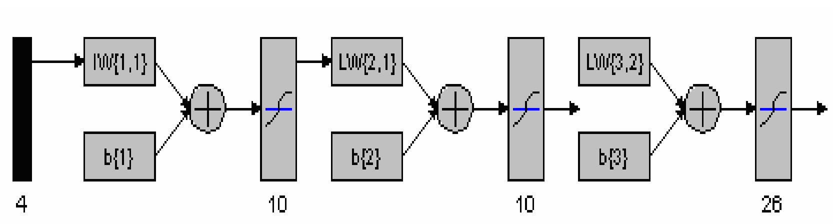

Fig. 3. Structure of artificial neural network core

The training algorithm of the network and weights update procedure and the error calculation is as below:

$$\Delta w_{ho}(s+1)=-\eta\sum_{i=1}^{H}\frac{\partial E}{\partial w_{ho}}+\alpha\Delta w_{ho}(s)+\frac{1}{1-(\alpha\Delta w_{ho}(s))} \tag{7}$$

$$\Delta w_{ih}(s+1) = -\eta \sum_{i=1}^{N} \frac{\partial E}{\partial w_{ih}} + \alpha \Delta w_{ih} + \frac{1}{1-(\alpha \Delta w_{ho}(s))} \tag{8}$$

Where, w_{ih} are the weights of input layer towards hidden layer and w_{ho} are the weights of hidden layer towards output layer. The constant parameter η determines the convergence ratio of the network and in our implementation is set to 0.1. By α parameter, momentum is incorporated into the network which helps the network to escape the local minima. In our implementation α is assigned the value 0.9. E stands for the error of the network and is calculated according to the equation below:

$$E = \frac{1}{2} \sum_{P=1}^{P} \left(O^{o} - T\right)^{2} \tag{9}$$

In the equation (9), O is the output of the network and T is the real output expected. For all input values the square difference of these two parameters is calculated and the overall error of the network is determined.

4 Genetic Algorithm Core

The weights of the neural network core typically are produced by BP algorithm in the first place. But being trapped at local minima is a connate threat of this algorithm. To overcome this issue, in our approach the initial weights of the neural network is obtained by a genetic algorithm which can explore the entire search space fast and then the further improvements are made through BP algorithm.

The structure of genetic algorithm is depicted in figure 1. Individuals in the population of the GA are weights and bias values of the neural network. The initial population is generated under a uniform random distribution. By applying GA operators the population evolves to better fit the optimization criteria, which in our case is the better performance of the neural network. These operators need to be modified to be suitable for the ranges applicable to the ANN core as it is provided in the following parts. Selecting the best population of the weights is done in a way that the least discrepancy between the network output and the real output is resulted. A chromosome in this population is a square matrix of weights. If any element of this matrix is zero, two neurons of the corresponding indices are not connected; otherwise their connection weight is the real number of that gene.

4.1 Mutation

The mutation operator is implemented by randomly choosing a single chromosome and summing it with a uniformly generated random number. The mutation is preformed according to the equation (10).

$$C^{new} = C^{old} \bigcup_{i=1}^{n} \left[C_{j-\lambda} \bigcup \left(C_{\lambda}^{old} + \varepsilon \right) \right] \tag{10}$$

In the equation (10), C^{old} denotes the current chosen chromosome for mutation which has j genes. ε is a random number in the range [-1, 1]. λ represents a randomly selected gene from current chromosome that is to be modified. C^{new} represents the next generation of chromosome.

4.2 Recombination

The recombination operator is responsible for making diversity in the population of answers while keeping an eye on the better chances of suitability. This operator is applied by equation (11). In this recombination first a chromosome is selected, and then two random genes of this chromosome are swapped.

$$C^{next} = C^{sel} \bigcup_{i=1}^{n} \left[\left(C^{sel} - C_{\alpha}^{sel} - C_{\beta}^{sel} \right) \bigcup \left(C_{\alpha}^{sel} \longleftrightarrow C_{\beta}^{sel} \right) \right] \tag{11}$$

In the equation (11), α and β denotes the locus of the randomly selected genes in the chosen chromosome from the previous step.

4.3 Fitness Evaluation Function

Fitness function must be able to evaluate the suitability of the weights, individuals of population, for our neural network. To this end we calculate the total sum of network square errors. As the input data are fed to the network this measure is calculated and chromosome with the smallest total sum of square errors is appointed the maximum fitness. This leads the GA to find the most suitable set of weights and bias values for the neural network with least errors.

5 Simulation and Computational Results

The performance of the proposed approach has been evaluated by simulating with MATLAB. In [3] it is shown that training the neural network only by BP algorithm is very prone to be tangled in local minima. There have been several techniques suggested to overcome this drawback; one of the most successful ones is by using evolutionary algorithms. In this approach a customized genetic algorithm has been utilized in hybrid evolutionary feed-forward neural network which is responsible for searching entire search space while BP algorithms is responsible for local search.

The simulation results are obtained by feeding the neural network with the scanned image of 26 English capital letters in the handwritings of different people. Five different handwriting data sets have been used. The output of the system is the classification of letters independent of the specific writers' handwriting styles.

Further contribution is made in feeding the neural network with scanned character images input. For each letter the image centroid is calculated and accordingly the image is divided into four subsections, then these subsections are fed into the network.

Table 1. Numbers of Epochs Required for Network Convergence within Same Setting

Sample Letters:	A	I	E	O	U
Proposed Approach with Centroid:	2200	950	4600	-	-
Without Centroid:	2600	-	-	-	-

Table 2. Network Error Comparison for Some Sample Letters

Sample Letters:	A	I	E	O	U
Proposed Approach with Centroid:	0	0	0	0.19	0.2
Without Centroid:	0	0.4	0.19	0.2	0.2

Table 1 provides the comparison between the numbers of epochs required for convergence of the network in the proposed approach by computing the image centroid and in the case that image centroid is not taken into account, as in [3]. Table 2 represents the networks' errors. These tables are provided for sample letters. The simulation results showed that the proposed approach is promisingly successful in letter recognition. As shown in table 1 both algorithms are not converged with specified setting, but within the same settings the proposed approach converges with fewer epochs and according to table 2 with fewer errors. Finally, with 50000 epochs the algorithm is run for all alphabets.

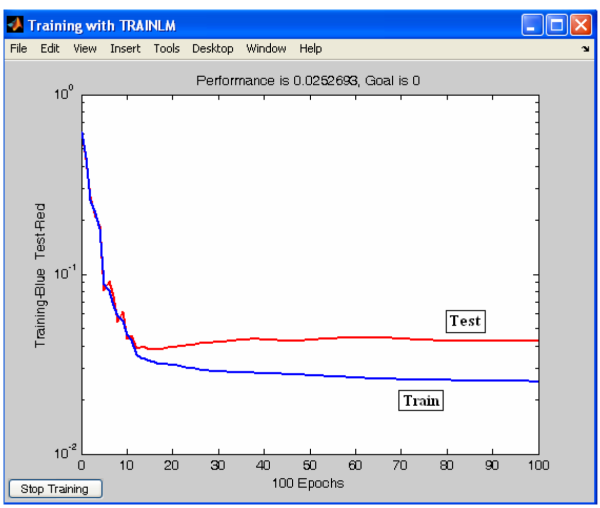

Fig. 4. Neural network output

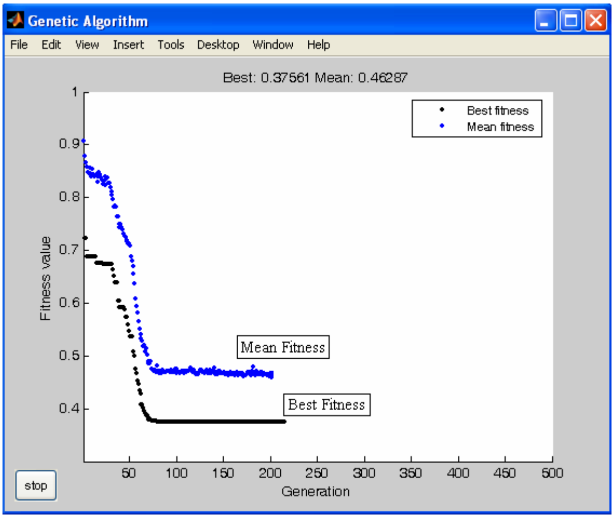

Fig. 5. Evolution of neural network weights

Figure 4 represents the proposed neural network output. As it is shown the network errors is reduced below 0.05 and hence the termination criteria is met and the algorithm stops. Figure 5 demonstrates the evolution of neural network weights with genetic algorithm.

There is a limitation of 50000 iterations on training phase. The network training is by entering all samples of one set handwritten letters in one step and the entire set in next steps. It should be noted that in training every step the order in which the letters are fed into the network must be different from the order of entered letters in the previous training step. Simulations are based on division of data set as 70% of all data used for training and the rest 30% is used for testing.

6 Conclusion

This paper aims at the problem of recognizing single alphabetical letters in the various handwriting styles of different people. We have opted to test the proposed algorithm on English capital letters due to their wide application in filling forms, and their intrinsic feature of preserving their block style. This approach is also applicable to learn and recognize the Farsi language alphabets written in various handwriting styles, but in block separate letters. The application of this system is in properly converting scanned images of official forms into text files.

The simulation results indicate that the proposed hybrid evolutionary feed-forward neural network with enhanced image feeding to the network outperforms the conventional approaches. The advantage is better performance of the network in training and correct classification of letters. Moreover, by using image centroid in dividing network input image into subsections, the whole system is invariant to translation, rotation, and scaling of input letters. Since these deformities are very common in handwritten texts, this approach demonstrates a promising property in real world applications.

References

1. Mahmoudi, F., Parviz, M.: Visual Hand Tracking Algorithms. In: IEEE Proc. Geometric Modeling and Imaging–New Trends, pp. 228–232 (July 2006)
2. Meinagh, M.A., Isazadeh, A., Ayar, M., Mahmoudi, F., Zareie, B.: Database Replication with Availability and Consistency Guarantees through Failure-Handling. In: IEEE Proc. International Multi-Conference on Computing in the Global Information Technology (ICCGI 2007), p. 14 (2007)
3. Mangal, M., Singh, M.P.: Handwritten English Vowels Recognition Using Hybrid Evolutionary Feed-Forward Neural Network. Malaysian Journal of Computer Science 19(2), 169–187 (2006)
4. Mangal, M., Singh, M.P.: Patterns Recalling Analysis of Hopfield Neural Network with Genetic Algorithms. International Journal of Innovative Computing, Information and Control (JAPAN) (2007) (accepted for publication)
5. Mahmoudi, F., Shanbehzadeh, J., Eftekhari, A., Soltanian-Zadeh, H.: Image retrieval based on shape similarity by edge orientation autocorrelogram. Journal of Pattern Recognition 36(8), 1725–1736 (2003)
6. Gao, W.: New Evolutionary Neural Networks. In: Proc. of International Conference on Neural Interface and Control, May 26-28 (2005)
7. Goldberg, D.: Genetic Algorithms. Addison-Wesley, Reading (1989)
8. Pal, S.K., Wang, P.P.: Genetic Algorithm for Pattern Recognition. CRC Press, Boca Raton (1996)
9. Gelsema, E.S.: Editorial Special Issue On Genetic Algorithms. Pattern Recognition Letters 16(8) (1995)
10. Auwatanamongkol, S.: Pattern recognition using genetic algorithm. In: IEEE Proc. of the 2000 Congress on Evolutionary Computation (2000)
11. Murthy, B.V.S.: Handwriting Recognition Using Supervised Neural Networks. In: Joint Conference on Neural network vol. 4 (1999)