A Neuro-Genetic System for Face Recognition

V. Saishanmuga Raja[1], Dr. S.P. Rajagopalan[2]

[1] Department of Computer Science and Engineering,
Dr. MGR Educational & Research Institute University, Madhuravoyal,
Chennai, Tamilnadu, India

[2] Department of Computer Applications,
Dr. MGR Educational & Research Institute University, Madhuravoyal,
Chennai, Tamilnadu, India

Abstract

Face Recognition has been identified as one of the attracting research areas and it has drawn the attention of many researchers due to its varying applications such as security systems, medical systems, entertainment, etc. A wide variety of systems requires reliable personal recognition schemes to either confirm or determine the identity of an individual requesting their services. Face recognition is the preferred mode of identification by humans: it is natural, robust and non-intrusive. The proposed approach uses Neural and Genetic algorithm that overcomes low recognition rate, low accuracy and increased time of recovery. Optimization of neural network parameters is done using Genetic algorithm. This increases recognition accuracy and reduced training time.

Key words: *Genetic algorithm, Neural Network, Face recognition.*

1. Introduction

Human face detection plays an important role in applications such as video surveillance, human computer interfaces, face recognition, and face image databases. To enable this biometric technology it requires having at least a video camera, PC camera or a single-image camera. Nevertheless, this biometric approach still has to deal with a lot of problems and cannot work with acceptable identification rates unless certain restrictions are being considered. Finding a face in a picture where the position, the orientation, the background and the size of a face is variable is a very hard task and many algorithms have been worked on to solve this problem. Other problems with face detection occur whenever faces are partially covered, as with beards, glasses, hair style or hats, because a lot of information just stays hidden.

It is often useful to have a machine perform pattern recognition. In particular, machines which can read face images are very cost effective. A machine that reads passenger passports can process many more passports than a human being in the same time [1]. This kind of application saves time and money, and eliminates the requirement that a human perform such a repetitive task. This document demonstrates how face recognition can be done with a back propagation artificial neural network. The information inserted is trained to learn the characteristic, behavior and others in order to ensure the system that is able to recognize the input data.

In biological system, the learning process involves adjustments to the synaptic connections that exist between the neurons. Learning is like a starting point to gain knowledge, experience in order to success or failure. For example, during training the classification network learns the association and significance of features.

The flow of the study consists of three major parts.

a) Collecting and analyzing dataset which is Face image
b) Training accuracy (dataset of Face Images)
c) Testing accuracy (dataset of Face Images)

Dataset is obtained from with 150 facial images from 5 different people. The images of face is synchronized and labeled to ease in identifying the face later.

The structure design of the Neural Network mostly relies on the designers experience and repeated experiments. The design efficiency is very low, and it cannot guarantee that design of the network structure is optimal. Instead, it reduces the prediction performance of neural networks. Deng kai [1] fixes the topology structure of the network in advance, and adopts a genetic algorithm rather than randomly set the initial weights of neural networks, and then trains the network precisely in search space defined by the genetic algorithm. Finally the trained model is used for face recognition.

3. Neuro – Genetic Methodology

In neural network training, the most commonly used algorithms are versions of the back propagation algorithms developed by Rummelhart et al. (1986). The well known limitations of gradient search techniques applied to complex nonlinear optimization problems have often resulted in inconsistent and unpredictable performance. They typically start at a randomly chosen point (set of weights) and then adjust weights to move in the direction which will cause the errors to decrease most rapidly. These types of algorithms work well when there is a smooth transition toward the point of minimum error. But the surface of the neural network is not smooth. It is characterized by hills and valleys that cause techniques such as BPN trapped in local minimum.

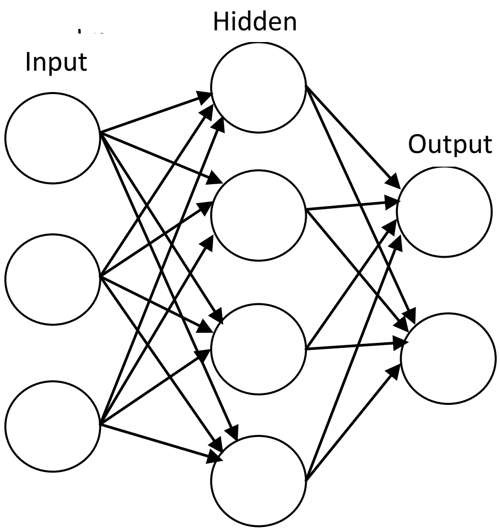

Fig. 1 Three layers in Neural Networks.

3.1 Population Division

Assume that the neural network to be optimized includes N_{p+2} layers, that is, it has an input layer, N_p hidden layers and an output layer. This vertical segmentation of the neural network is shown in the figure:2 layers named with A,B,C…respectively. The part from the first layer(A) to the third layer(C) is the first module; the part from the second layer(B) to the fourth layer(D) is the second module. For a module, evolve sub-population to optimize the structure and connection weights. The evolution sub populations in total, that is, Neural Genetic Algorithm contains N_p sub-populations.

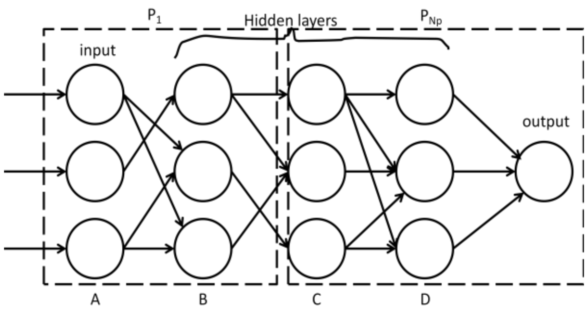

Fig.2 Population Segmentation

3.2 Structure Coding

Suppose that the P_{th} module has M_p neurons; the serial number is ordered from the 1_P to the M_P input layer(p layer), hidden layer (P+1 layer), the output layer node (P+2 layer).The connection relation of the P_{th} module is denoted by matrix, that is S_{MP} x $_{MP}$. In the matrix S(i,j), the elements stand for the connection relationship from the i_{th} neuron to the j_{th} neuron. "0" indicates no connection; "x" indicates no association and "1" indicates a connection. Therefore we can use the matrix to represent the topology structure coding of the P_{th} module individual.

3.3 Weights coding

If there is no connection between two nodes, its connecting weight must be 0. Therefore there is no need to consider the encoding of the connecting weights. In this way, connection weights are controlled by structure coding. Only when the structure coding is 1, we need to consider the encoding of connection weights and the length of the connecting weights encoding is equal to the number of 1 in the structure coding. With the change in the structure coding, the length of the connecting weights encoding will also change. This dynamic method can effectively reduce the computational complexity.

4. Genetic Operation

4.1 Crossover Operation

Since the individual coding of the evolutional sub-populations in this paper contains the structure and connecting weights, and different parts take different encoding methods, it is necessary to design corresponding genetic operations. For structural parts, we use binary coding and the crossover operation of the standard genetic algorithm, such as single point crossover, multi point crossover or same crossover and so on. As for the connection weights part, we use real coding, and the length of the real coding is equal to the number of 1 in the structure coding. Therefore the intersection of real-coded string should depend on the location of the intersection in structural part.

4.2 Mutation Operation

Let us consider the structural parts and connection weights parts of evolutional individuals respectively. For structural parts, use the mutation operation of standard genetic algorithm, such as single point mutation, multi point mutation, or same mutation. For the connection weight parts, since its code is influenced by structure parts, when the code of a certain structural part changes from 0 to 1, the corresponding code of connecting weights parts should be generated randomly within a certain range.

4.3 Fitness Function Evaluation

We set the input and output as (X_i, t_i), i=1,2….M_r, and if input is X_i, the actual output of neural network is y_i . Now consider the P_{th} evolution sub-population. The j_{th} evolution individual is $x_j^{\,p}$, others are $x_r^{\,q}$, q=1,2,…p-1, p+1, N_p , then the individual fitness calculation formula Eq.(1)[6] is given below.

$$F = \frac{1}{\frac{1}{M_r}\sum_{i=1}^{M_r}\left(t_i - y_i(x_r^1, x_r^2, \ldots, x_r^{p-1}, x_j^p, x_r^{p+1}, \ldots, x_r^{N_p})\right) + c} \quad (1)$$

Where c is an appropriate constant

4.4 Collecting and analyze dataset

Dataset is obtained from with 150 images from 5 different people. The face image is synchronized and labeled to ease in identifying the face later. For example, from I until 30 is referring to person 1 and so on. Then, the pre processing is based on utilizing Otsu's method which permits us to determine the necessary grey threshold value to carry out the binarization of the sample [5].

The segmentation is used in order to remove the noise from an image. Then, the similar characteristic from the image will be identified and grouped together. This can be used in statistical classification, edge and region detection, or any of these techniques. This is where the image is changed into grayscale format. The process is about binarising of the grey level to a black and white image which is focusing at each pixel of the image and deciding whether it should be converted to black (0) or white (255).

Then, it will make a comparison between each numeric pixel of grey level image with a fixed number. This is called threshold level to do the decision. The pixel value will be set to 0 when the pixel is less than the threshold level and vice versa. This can be expressed in equation Eq.(2).

P (ij) ={ 0
255 if I(i,j) > T
if I (i,j) ≤ T
(2)

Where;
I (i, j) indicate the original image,
P (i, j) indicate the output binary image,
T is the threshold level,
And (i = 0 ... N, j= 0 ... , M) represent the image size.

The comparison and classification process are done in order to identify the person. This is done when each data of face image is inserted randomly during training. The instruction will be based on the number that had been labeled earlier.

4.5 Training accuracy

Training process is similar with a learning process. The step consists of input, hidden and output layer. The process is that the input of a facial image is inserted to train the network and at the same time recognize each of the image characteristic in order to differentiate with others input given. However, if more input is inserted the stronger weight of face image characteristic the network learns. This is like our brain whereas the more new knowledge the brain gains, the neurons bond will become stronger. But, the input must be not too little or many because if too little input, the network cannot learn and recognize properly and if it too many, the network will tend to memorize instead of learning.

4.6 Testing accuracy

This part is on testing the data which had been trained earlier. This is done to determine whether the network learn properly or not. This data can be used either the new set of data or data from the training process. For example, if the data have about 100 samples, 60 to 70 samples are used for training process and the remaining is used in testing process.

5. Algorithm Process

In Neuro-genetic approach, the learning of a neural network is formulated as a weights optimization problem. Unlike BPN, GAs performs a global search and is thus not easily fooled by local minimum.

5.1 Optimization Using Genetic Algorithm

The algorithm to obtain the optimal weight and optimal learning algorithm is given below.

1. Randomly generate an initial population
   $P^0=(a_1^0 , a_2^0,\ldots, a^0_\lambda)$
2. Compute the fitness $f(a_i^t)$ of each chromosome a_i^t in the current population P^t .
3. Create new chromosome P' of mating current chromosomes, applying mutation and recombination as the parent chromosomes mate.
4. Delete numbers of the population to make room for new chromosomes.
5. Compute the fitness of $f(a_i^t)$ and insert these into population.
6. Increment number of generation, if not (endtest) go to step 3, or else stop return the best chromosome.

5.2 Learning Algorithm

5.2.1 First Learning Stage

Step 1: Initialize 'fitness value' to zero and 'child chromosome' as null.

Step 2: Specify the objective function (system error for the neural network).
Step 3: Evaluate the child chromosome as subset chromosome.

Step 4: Compare the child chromosome with parent chromosome.

Step 5: Select parent using roulette wheel parent selection. Apply single-point crossover and Mutate child chromosome to the parent chromosome.

Step 6: Check for befitting of child chromosome with the objective function (system error for Neural Network).Replace the old generation by the new generation and name it best chromosome WHILE (stopping criteria).

5.2.2 Second Learning Stage

Step 1: Set the best chromosome as the initial weight vector or learning rate, set of topological structure of neural network.

Step 2: Compute the actual output of the neural network.

Step 3: Compute the error between desired output and actual output.

***Step 4*:** Update the weights or learning rate using neural network algorithm.

6. Simulation Results

For test from the people recognition process, images were taken from the ORL database of AT&T Laboratories Cambridge [3], which presents variations in facial gestures and positions. In Fig. 3 shows some examples of face images. To train and test the classifier we used 10 images of each person to identify, for the recognition of 20 people. As the number of samples per class is smaller than the number of classes, then 10 pictures of each individual have been taken to perform validation, including images used in training.

As it has taken 70% of the images for training, 7 neighbors have been used in the classifier. In this case, the genetic algorithm was set the following parameters: population size: 80, number of generations limit: 100, elite population: 2, crossing factor: 0.2.

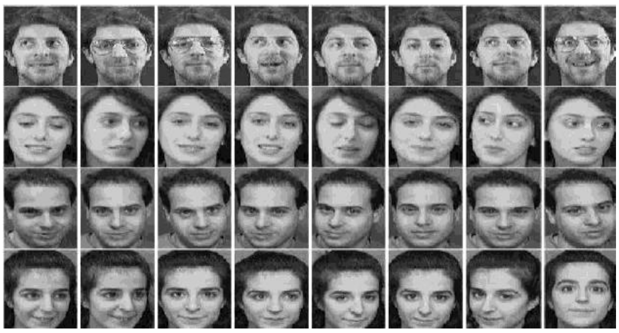

Fig.3 Examples of faces dataset

The work is implemented using MATLAB V7.9 on windows 7 operating system. The performance of different stages of image enhancement, and genetic algorithm for optimizing the neural parameters and recognition of images are evaluated and presented in the following sections.

6.1 Recognition of Facial Images By Neural Network With Back Propagation Algorithm (Bpa)

A back-propagation is one of many different learning algorithms that can be applied for neural network training. It belongs to a category of so called learning with the teacher. For every input vector that is presented to the neural network there is predefined desired response of the network in a vector t (the teacher). The desired output of the neural network is then compared with the real output by computing an error of vector t and neural network output vector y. The correction of the weights in the neural network is done by propagating the error backward from the output layer towards the input layer, therefore the name of the algorithm.

6.2 Recognition of Facial Images by Neuro- Genetic (NG) Algorithm

Training of neural networks depends on various sensitive parameters like Learning rate, Initial weight, Number of epochs and number of input and output neurons. Using genetic algorithm, the parameter weight is optimized so that the neural network training is started with the optimized weight.

6. 3 Comparing Neural Network With And Without Genetic Algorithm

Table 2 compares the training and testing time of neural network using back propagation algorithm and neuro-genetic algorithm.

TABLE 1: Training and Testing Of Neural Network with and Without Genetic Algorithm

No of Samples	No. of Epochs		Testing Time (Sec)		Training Time (sec)	
	BPA	NG	BPA	NG	BPA	NG
15	24	7	00.07	00.04	3.42	2.34
30	50	28	01.85	01.02	14.24	11.45
60	90	52	02.59	02.11	19.56	15.11
120	165	85	03.75	03.04	33.21	23.15

Figure 4 shows that the training time has been reduced if the neural network is optimized by genetic algorithm.

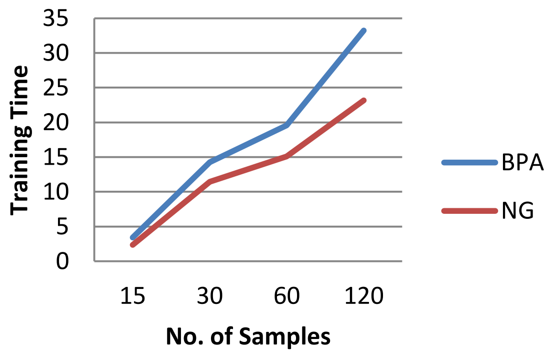

Fig.4 Comparison of BPA and NG

As shown in the table 1, it is observed that training time of the neural network is reduced when it is optimized using genetic algorithm when compared with neural network without genetic algorithm. Also it is observed that the time taken by the neural network with genetic algorithm takes less time to recognize a facial image when compared with neural network without genetic algorithm.

7. Conclusion

An efficient Neuro-Genetic method for Face recognition has been proposed in this paper. From the above results it can be seen that A typical neural network can work efficiently when it is optimized using genetic algorithm. It is also clear that the proposed method reduces the training time of the neural network. The chance for the network to learn is increased rather than memorizing. The Neuro-Genetic method reduces the computational complexity by using genetic algorithm to decide the weights of the neurons in the neural network. Thus the proposed method is an efficient face recognition process.

References

[1] Gamarra A., Christian G. Quintero M. *"Intelligent Approaches for Effective Feature Selection in Image Pattern Recognition",* Universidad Del Norte Universidad Del Norte Barranquilla, Colombia.

[2] Pu Xingcheng, Sun Pengfie, "*A cooperative Co-Evolutionary Genetic Neural Network and itsb application",* Chongqing University of Post & Telecommunications, Chongqing, China.

[3] The ORL database of faces. http://www.cl.cam.ac.uk/research/dtg/attarchive/face database.html, 1992-1994

[4] *Face Recognition using Neural Networks,* Nazish Jamil, Fatima Jinnah Women University, Rewalpindi, Pakistan.

[5] Otsu N , "*A threshold selection method from grey level histograms*" IEEE Transaction on Systems, Man and Cybernetics 9(1):62-66, 1979.

[6] D.W.Gu,X.Y.Sun ,"*Cooperative evolutionary genetic algorithm theory and its application*", [M] Science press, 2009.

V. Saishanmuga raja is a Research Scholar at Dr. MGR Educational & Research Institute University in the Department of Computer Science and Engineering, Chennai, Tamilnadu. He is doing his research under the guidance of Dr. S.P. Rajagopalan. He received his B.E., degree in 2006 and ME degree in 2008 in Computer Science and Engineering from Anna University Chennai. He is presently working as Assistant Professor at Dr. MGR Educational & Research Institute University Chennai. His research interests are Data mining, Neural Networks, and Genetic Algorithm. He is a member of IEEE.

Dr.S.P. Rajagopalan received his M.Sc from IIT Madrs, M.Phil and Ph.D from Madras University, Chennai, India. He is presently working as a Professor (Emeritus) in Department of Computer applications in Dr. M.G.R. Educational and Research Institute, Chennai. He has 40 years of experience in Teaching and Research. He has published 4 Books. He has about 100 publications in International Journals and National Journals. His special fields of interest include Quantitative Techniques, Data Processing and Project Management, Management Information System, Programming Languages, Simulation, Text generation, Cryptography and Data Mining.