Pakistan Journal of Information and Technology 2(1): 40-43, 2003

Feature Extraction by Using Non-linear and Unsupervised Neural Networks

A. Jalil, I. M. Qureshi, A. Naveed and T. A. Cheema
M.A. Jinnah University, Islamabad, Pakistan

Abstract: Feature extraction is fairly popular in pattern recognition and classification of images. In this paper we propose an unsupervised learning algorithm for neural networks that are used in feature extraction problem. These learning algorithms use genetic algorithm as a searching technique for global minimum of error performance surface and LMS algorithm for final convergence to the global minimum. These learning algorithms used Sammon's stress as criterion for getting feature with maximum inter pattern distances and minimum intra pattern distances. A common attribute of these learning algorithms is that they are adaptive in nature, which makes them suitable in environments when the distribution of patterns in the feature space changes with respect to time.

Key Words: Feature Extraction, Unsupervised Neural Networks, Sammon's Stress, Pattern Recognition, Genetic Algorithm, LMS Algorithm

Introduction

Feature extraction is an important issue in the field of pattern recognition. Different classical feature algorithms are proposed in literature like method of moments(Lhotanzad and Lu, 1990), Zernike moments (Knotanzad and Hong, 1988) and Fourier descriptors (Kim and Nam, 1995). The disadvantage of these algorithms is that they are not adaptive in nature and difficult in real time implementations (Huang and Lippman, 1987) . A large number of artificial neural networks and learning algorithms have also been proposed for feature extraction due to their advantages over the traditional approaches (Abbas and Fahmy, 1992; Baldi and Hornil, 1989). Firstly, most learning algorithms and neural networks are adaptive in nature. Thus they are well suited for many real time environments where adaptive systems are required. Secondly, in real time implementations, neural networks provide good architecture, which is relatively easily implemented using VLSI and optical technologies. Thirdly, neural network implementations can overcome the drawbacks inherent in the classical algorithms (Chien, 1978).

Sammon proposed a non-linear mapping technique for data structure analysis. Sammon's technique attempts to maximally preserve all the inter-pattern distances. He used a gradient descent algorithm to find a configuration that attempts to minimize the error known as Sammon's stress given in Eq. 1.

$$E = \frac{1}{\sum_{\mu=1}^{n-1}\sum_{\nu=\mu+1}^{n} d^{*}(\mu,\nu)} \sum_{\mu=1}^{n-1}\sum_{\nu=\mu+1}^{n} \frac{\left[d^{*}(\mu,\nu) - d\ (\mu,\nu)\right]^{2}}{d^{*}(\mu,\nu)} \quad (1)$$

where $d^{*}(\mu,\nu)$ and $d(\mu,\nu)$ are the distances between pattern μ and ν in the input space and the projected space, respectively. Euclidean distance is commonly used in this projection algorithm.

These techniques of finding a feature vector having maximum inter-pattern distances have serious problems. There are many local minima in the energy surface, and it is unavoidable for the algorithm to get stuck in a local minima. One usually runs the algorithm several times with different random initial configurations and chooses the configuration with lowest stress. Another disadvantage is large amount of computation (Hornik and Kuan, 1992; Jain and Mao, 1992).

Jain and Mao (1992) propose a neural network design for Sammon's non-linear projection algorithm, which offers the generalization ability of projecting new data. Mao and Jain (1992) refined their algorithm by implementing Sammon's projection technique in unsupervised feedforward neural network with learning algorithm based on steepest descent criterion with Sammon's stress taken as error surface (Chien, 1978). The proposed algorithm is close to backpropagation algorithm. The disadvantage is its large number of computations because algorithm requires millions of iterations to train neural network. Another problem is that the algorithm gets stuck in the local minima (Hornik and Kuan, 1992). Different methods have been reported in literature to get out of local minima, like Brains method, adding momentum terms etc. But these methods have not been found satisfactory.

In the present study we proposed unsupervised feedforward neural network model in which we first of all search the global minima by using genetic algorithm and then LMS algorithm to converge the solution to global minimum.

Proposed Neural Network Architecture: The proposed neural network belongs to feedforward neural networks with neurons arranged in layers. Generally, all neurons in layer are connected to all neurons in adjacent layers through uni-directional links. These links are represented by synaptic weights (James and David, 1999). The proposed architecture consists of one input layer, one hidden layer and one output layer as shown in Fig. 1.

The generalized formula for the neuron outputs of any layer is:

$$u_j = \sum_{i=0}^{n_s-1} w_{ji}^{[s]} o_i^{[s-1]} \quad (2)$$

where $w_{ji}^{[s]}$ are the synaptic weights by which the jth neuron multiplies the inputs $x_i^{[s]} = o_i^{[s-1]}$, $o_i^{[0]} = x_i$, $o_i^{[3]} = y_i$ and n_s is the number of neurons in the sth layer. In our network, we used $s=3$ (Davis and Lawrence, 1991). In the proposed algorithm, we have updated the values of synaptic weights by using two learning algorithms, genetic algorithms and LMS algorithm. The Genetic Algorithm is used to search the global minimum, where as LMS algorithm converges to the required minimum.

Genetic Algorithm: One main desire is to find out optimized synaptic weights that will give us maximally desired feature vectors which preserve all the inter-pattern distances (Chambers and Lance, 1995). The Genetic Algorithm is being used to find weights of the feedforward neural network, which are close to the desired solution. The proposed neural network consists of three layers and each layer contains a set of weights $w_{ji}^{[s]}$, s is the layer number 1,2,3.

$$W = [w_{ji}^{[1]} \quad w_{ji}^{[2]} \quad w_{ji}^{[3]}] \tag{3}$$

where $W_{ji}^{[s]}$ is a set of floating point numbers. The ordinary rule of thumb for choosing the population size is to choose it five to six times the length of a chromosome. Fig. 2 is the flowchart of the genetic algorithm.

Initialization: The genes representing weights is produced by using a pseudo random generator. The values of the genes are floating between 0 and 1.

Fitness Criterion: The fitness criterion denoted by F is taken as

$$F = \frac{1}{1+E} \tag{4}$$

where *E* is Sammon's stress.

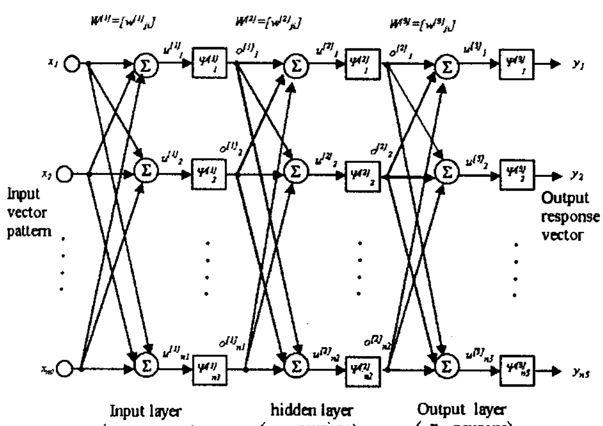

Fig. 1: Proposed Feed Forward Neural Network

Cross-over, Reproduction and Next Generation: The population in a generation is sorted out according to the fitness criterion in a descending order. The top one third were given a chance to produce five children per parent pair. The next one third was given chance to produce 3 children per pair. They were similar to child 1 and child 2 and child 3 as given above. The last one third was given a chance to produce only one child per pair, which is similar to child 1. For selection of a new generation, the rule for the survival of the fittest is being used. All the parents and children are sorted in a descending order of fitness after calculating the fitness of the children. The ones equal to the number of population were chosen for the next generation. So the new population was a blending of the previous and the present with no fixed percentage for either generation (Simon S. Haykin, 1998).

Mutation: Mutation is the process in which genes are randomly mutated (changed). Mutation plays an important role when every new generation does not seem to improve the fitness.

Termination Criterion: When the fitness of the best chromosome in the new population is almost the same as that of the best chromosome in the previous population, the program is terminated. Quantitatively the criterion for termination was:

$$|F_{new} - F_{prev}| < 0.01 \tag{5}$$

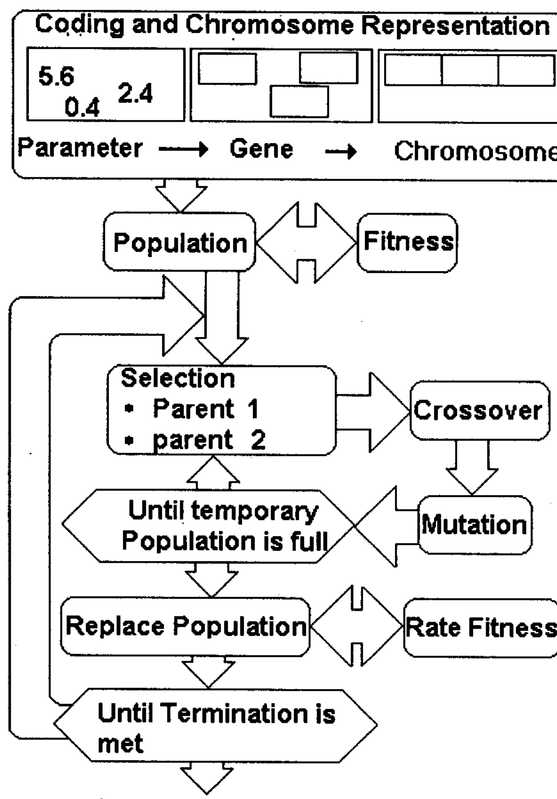

Fig. 2: Flowchart of Genetic Algorithm

LMS Algorithm: A steepest descent method was proposed to update the synaptic weights of the neural network used for the global convergence of the neural networks (Chambers and Lance, 1995). The synaptic weights of any layer are updated according to the Eq. 5

$$w_{ji}^{[l]new} = w_{ji}^{[l]old} + \eta \frac{\partial E_{\mu\nu}}{\partial w_{ji}^{[l]}} \tag{5}$$

where l is the layer number (1,2,3) and the mapping error $E_{\mu\nu}$, also called Sammon's stress. The Euclidean distance $d(\mu,\nu)$ used in this projection algorithm is defined as

$$d(\mu,\nu)=\left\{\sum_{i}^{n_3}\left[y_i^{[3]}(\mu)-y_i^{[3]}(\nu)\right]^2\right\}^{1/2} \quad (6)$$

The weights of the third layer of the neural network were updated using Eq. 7

$$w_{ji}^{[3]new}=w_{ji}^{[3]old}+\eta[\Delta_{ji}^{[3]}(\mu)o_j^{[2]}(\mu)-\Delta_{ji}^{[3]}(\nu)o_j^{[2]}(\nu)] \quad (7)$$

where

$$\Delta_{ji}^{[3]}(\mu)=\delta_i^{[3]}(\mu,\nu)[1-y_i(\mu)]y_i(\mu)$$

$$\delta_i^{[3]}(\mu,\nu)=-2\lambda\left(\frac{[d^*(\mu,\nu)-d(\mu,\nu)]^2}{d^*(\mu,\nu)}\right)\frac{y_i(\mu)-y_i(\nu)}{d(\mu,\nu)}$$

The weights of the second layer of the neural network were updated using Eq. 8

$$w_{ji}^{[2]new}=w_{ji}^{[2]old}+\eta[\Delta_{ji}^{[2]}(\mu)o_j^{[1]}(\mu)-\Delta_{ji}^{[2]}(\nu)o_j^{[1]}(\nu)] \quad (8)$$

where

$$\Delta_{ji}^{[2]}(k)=\delta_i^{[2]}(k)[1-o_i^{[2]}(k)]o_i^{[2]}(k)$$

$$\delta_j^{[2]}(k)=\sum_{k=1}^{m}\Delta_{jk}^{[3]}(k)w_{jk}^{[3]} \qquad k=\mu \text{ or } \nu$$

The weights of the first layer of the neural network were updated using following equation:

$$w_{ji}^{[1]new}=w_{ji}^{[1]old}+\eta[\Delta_{ji}^{[1]}(\mu)x_j(\mu)-\Delta_{ji}^{[1]}(\nu)x_j(\nu)] \quad (9)$$

where

$$\Delta_{ji}^{[1]}(k)=\delta_i^{[1]}(k)[1-o_i^{[1]}(k)]o_i^{[1]}(k)$$

$$\delta_j^{[1]}(k)=\sum_{k=1}^{m}\Delta_{jk}^{[2]}(k)w_{jk}^{[2]} \qquad k=\mu \text{ or } \nu$$

The unsupervised algorithm is summarized as below:

1. First of all create different layers of neural network.
2. Create a population of parents containing weights of neural network three times greater than the number of weights.
3. Initialize all parents randomly using different seeds.
4. Calculate the fitness of all parents and arrange them in descending order
5. Produce five children per pair from the first one-third best-fit parents. The next one third was given chance to produce 3 children per pair. The last one third was given a chance to produce only one child per pair.
6. Calculate fitness of the children.
7. Align parents and children in descending order.
8. Choose the best-fit among them equal to the number of parents.
9. Repeat steps 4 to 6 for a certain number of generations.
10. Use these weights to initialize the neural network weights while using backpropagation algorithm.
11. Run backpropagation algorithm and update weights of different layers of neural network for different patterns.
12. Terminate the algorithm when neural network converges to a solution.

Results and Discussion

The proposed neural network was tested on a data set of three characters, which are shown in Fig. 1. The data set consists of three sixteen dimensional patterns AOC from three classes "A", "O" and "C". All the patterns had the same dimensionality but different characteristics. The performance of the proposed neural network was evaluated based on visual judgment of features and some numerical (Simon S. Haykin, 1998).

The numerical criteria are i) Sammon's stress ii) Nearest neighbor classification error rate P_e^{NN} on projected data, iii) Minimum distance classification error rate P_e^{MD} of projected data. Since The classification error of the projected data P_e *(projection)* depends on the classification error of the original data P_e *(original)* so we used the normalized ratio:

$$R_e=\frac{1+P_e(projection)}{1+P_e(orignal)} \quad (10)$$

The value of the R_e is within range (0.5, 2.0). When R_e=1.0, it means that the extracted features have same discriminatory power as original features for a given classifier. If R_e <1.0, the extracted features have a better performance in terms of classification accuracy. This could happen if the effect of the "Curse of dimensionality" is eliminated by the feature extraction (Simon S. Haykin, 1998). The Sammon's stress measures how well the extracted features preserve inter pattern distances. The nearest neighbor classification error rate indicates how well the extracted features are projection preserve the local category structure of the date, while the minimum-distance classification error rate provides some information on the linear reparability in the new feature space (Biswa *et al.,* 1981).

The pattern from three classes was presented to the neural network. The optimum set of weight was found using genetic algorithm. The error performance surface of genetic algorithm is shown in Fig. 4(a). We used 100 generations of the chromosomes representing weights but the algorithm find global minimum in less than ten generations. There was employed LMS algorithm to converge to the global minimum. The graph of LMS algorithm is shown in Fig. 4(b) in which error is minimized in approximately three thousand iterations and Sammon's stress of 0.07 was achieved. Such Sammon's stress requires thousand of iterations with many random initial configurations when neural network uses LMS algorithm only. Therefore the proposed neural network reduces the computational complexity.

The features obtained from the three classing using proposed neural network are shown in Table 1. The results also show that the net preserved all the inter pattern distances. The inherent problems of the method were also solved due to the usage of genetic algorithms, which finds the optimal initials weights for the gradient algorithm.

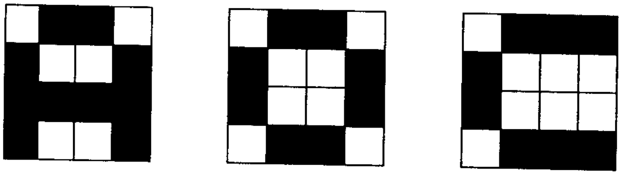

Fig.3: The patterns used to Train Neural Network

Table 1: Features of the pattern which Belong to three Different Classes

Classes	Features			
	1	2	3	4
A	0.600313	0.513414	0.510628	0.969344
O	0.541961	0.491416	0.453723	0.8306
C	0.491094	0.444814	0.415031	0.773331

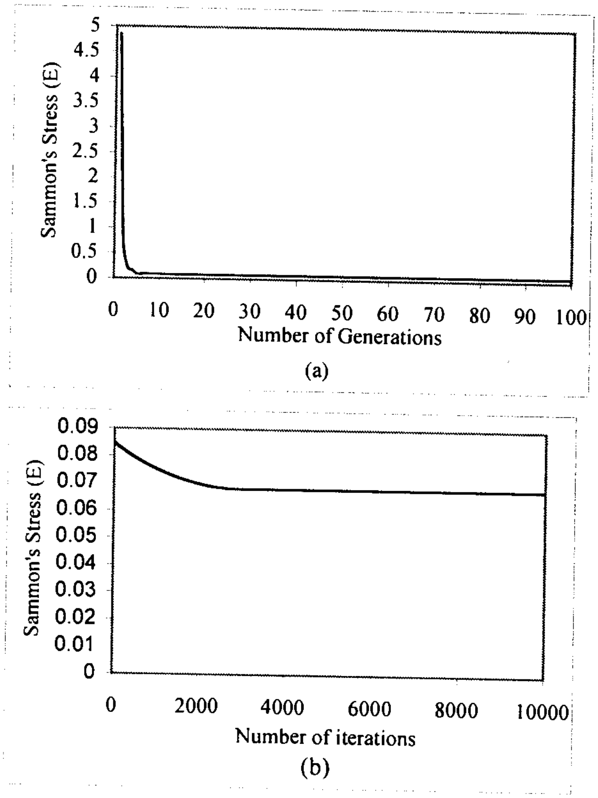

Fig. 4 (a): Graph Between Number of Generations and Sammon's Stress(E) of the Genetic Algorithm Employed on Neural Network. (b) Graph Between Number of Iterations and Sammon's Stress of the LMS Algorithm Employed on Neural Networks

Conclusion

We have designed an unsupervised neural network algorithm for feature extraction of different patterns, which belong to the different classes. The net used Sammon's stress which optimally preserves the inter pattern distances. The net consisted of two parts; the first part uses genetic algorithms to find out optimal initial weights of the net. This technique was used to search the global minima of the error performance surface. The second parts used gradient-based algorithm in order to converge to the solution or global minima or the error performance surface. The net was used to find out features of patterns, which belong to the three different classes, and the results showed that the obtained features formed the three clusters in the feature space. The proposed net can provide better results when used on hand written character.

References

Abbas, H. M. and M. M. Fahmy, 1992. "A neural model for adaptive Karhunen Loeve Transform (KLT)", In Proc. Int. Joint Conf. Neural Networks, Baltimore, MD, Vol. 2, 975-980.

Baldi, P. and K. Hornil, 1989. "Neural networks and principal component analysis: Learning from examples without local minima", IEEE Trans. Neural Networks, Vol. 2, 53-58.

Biswa, G., A. K. Jain and R. C. Dubes, 1981. "Evaluation of projection algorithms," IEEE Trans., Patt. Anal. Mach. Intell., Vol. PAMI-3, no. 6, 701-708.

Chien, Y., 1978. Interactive Pattern Recognition. NY: Marcel Dekker, Inc.

Chambers and Lance (editor), 1995. Practical Handbook of Genetic Algorithms: Applications: Vol. I. Boca Raton, FL: CRC Press.

Davis and Lawrence (editor), 1991. Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold.

Huang, W. Y. and R. P. Lippman, 1987. "Comparisons between neural net and traditional classifiers," in IEEE 1st Int. Conf. Neural Networks. San Diego, CA, IV-485-IV-493.

Hornik, K. and C. M. Kuan, 1992. "Convergence Analysis of local feature extraction algorithm," neural networks, Vol. 5, 229-240.

Jain, A. K. and J. Mao, 1992. "Artificial neural network for nonlinear projection of multivariate data", In Proc. Int. Joint Conf. Neural Networks, Baltimore, MD, Vol. 3, 335-340.

James, A. Freeman and M. David, 1999. Skapura Addison-Wesley.

Knotanzad, A. and Y. H. Hong, 1988. "Rotation invariant pattern recognition using Zernike moments", in Proc., 9th ICPR (Rome, Italy), Nov. 14-17, 326-328.

Kim, H. and K. Nam, 1995. "Object recognition of One-DOF tools by a back propagation neural net", IEEE Trans. Neural Networks, Vol. 6, no. 2.

Lhotanzad, A. and J. H. Lu, 1990. "Classification of invariant image representations using neural networks", IEEE Trans. Acoust., Speech, Signal Processing, Vol. 38, no. 6.

Simon S. Haykin, 1998. Neural Networks: A Comprehensive Foundation, Prentice Hall.