Face Recognition Based on Neuro-Fuzzy System

Nina Taheri Makhsoos †, **Reza Ebrahimpour** †† **and Alireza Hajiany** †††

Department of Electrical Engineering, Shahid Rajaee University, Tehran, Iran

Summary

This paper investigates the benefits of combining neural networks and fuzzy logic into neuro-fuzzy system, especially for application in face recognition task. The former is the result of combining fuzzy logic and mixture of experts, ME, by multi-layer perceptron networks. In the description of ME, some proposals to improve the performance of model are also described. An analysis comparing the model enhancements proposed in this paper with the corresponding original ME model in face recognition is also presented. Experimental result confirms the importance of combining these two technologies (neural networks and fuzzy logic) in face recognition.

Key words:

Fuzzy logic, Neuro-fuzzy system, Mixture of Experts.

1. Introduction

The combining of the techniques of fuzzy logic and neural networks suggests the novel idea of transforming the burden of designing fuzzy logic systems to the training and learning of connectionist structure and learning to the fuzzy logic systems and the fuzzy logic systems provide the neural networks with a structural framework with high-level fuzzy IF-THEN rule thinking and reasoning. These benefits can be witnessed by the success in applying neuro-fuzzy system in areas like pattern recognition and control.

Fuzzy logic [1,2,3] and artificial neural networks [4,5] are complementary technologies in the design of intelligent system. The combination of these two technologies into an integrated system appears to be a promising path toward the development of intelligent systems capable of capturing qualities characterizing the human brain. However, fuzzy logic and neural networks generally approach the design of intelligent systems from quite different angles. Neural networks are essentially low-level, computational algorithms that sometimes offer a good performance in pattern recognition tasks. On the other hand, fuzzy logic provides a structural framework that uses and exploits those low-level capabilities of neural networks.

Both neural networks and fuzzy logic are powerful design techniques that have their strengths and weaknesses. Neural networks can learn from data sets while fuzzy logic solutions are easy to verify and optimize. Table 1 shows a comparison of the properties of these two technologies. In analyzing this table, it become obvious that a clever combination of the two technologies delivers the best of both worlds.

Evolutionary Artificial Neural Networks have been widely studied in the last few decades. The main power of artificial neural networks lies in their ability to correctly learn the underlying function or distribution in a data set from a number of samples. This ability can be expressed in terms of minimizing the estimation error of the neural network, on previously unseen data. As discussed in the comprehensive review of [6], variety of methods has been applied on different issues of ANNs, such as the architecture and the connection weights to improve their performance. There are several published works in the literature that have shown an ensemble of neural networks demonstrates improved generalization ability in comparison with individual networks [7,8,9,10,11,12]. Most real world problems are too complicated for a single individual network to solve. Divide-and-Conquer approach, which tries to solve a complex problem by dividing it into simple problems, has proved to be efficient in many of such complex situations.

Jacobs et al. [13,14] have proposed an ensemble method called Mixture of Experts, ME, based on the Divide-and-Conquer principle. ME is one the most famous methods in the category of dynamic structures of combining classifiers, in which the input signal is directly involved in actuating the mechanism that integrates the outputs of the individual experts into an overall output [4]. Consider a modular neural network in which the learning process proceeds by fusing self–organized and supervised forms of learning as shown in Figure 1. The experts are technically performing supervised learning in that their individual outputs are combined to model the desired response. There is, however, a sense in which the experts are also performing self–organized learning; that is they self–organize to find a good partitioning of the input space so that each expert does well at modeling its own subspace, and as a whole group they model the input space well. The

Manuscript received April 5, 2009
Manuscript revised April 20, 2009

learning algorithm of the mixture structure is described in [15].

Table 1: Properties of neural networks and fuzzy logic

	Neural Networks	***Fuzzy Logic***
Knowledge Representation	Implicit, the system cannot be easily interpreted or modified	Explicit, verification and optimization are very easy and efficient
Trainability	Trains itself by learning from data sets	None, everything must be defined explicitly

However, in our models, in order to improve the performance of the expert networks, and consequently the whole network performance, we use modified of ME in which MLPs instead of linear networks or experts are used, and is hereafter referred to as mixture of multilayer perceptron experts (MME).

To evaluate the performance of our proposed model, we use ORL dataset in our experiments, and we use Principal Component Analysis, PCA, in the feature extraction phase. PCA is one of the most common methods of dimensionality reduction that is widely used in the literature of face recognition for feature extraction [16].

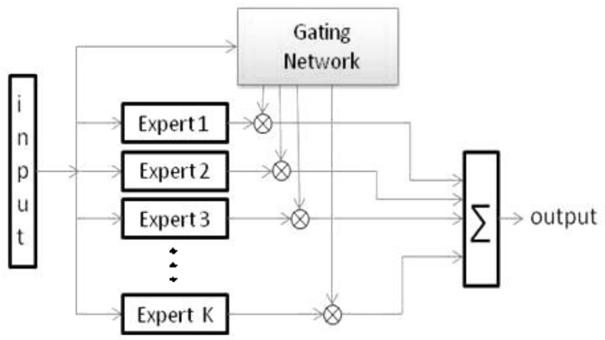

Fig 1. Block diagram of Committee Machine based on Mixture of Experts.

The rest of this paper is organized as follows: In Section 2, we provide a brief description of the Mixture of Experts structures with linear and Multi Layer Perceptrons, MLP, as experts. Section 3 describes the proposed model in which fuzzy rules are employed to learn a MLP, which is then used in ME structure. In section 4 a comparison with previously published methods is provided, along with a discussion on the obtained results. Finally, Section 5, concludes and summarizes the paper.

2. A modified Mixture of Experts

From a computational point of view, according to the principle of divide and conquer, a complex computational task is solved by dividing it into a number of computationally simple tasks and then combining the solutions to those tasks. In supervised learning, computational simplicity is achieved by distributing the learning task among a number of experts, which in turn divides the input space into a set of subspaces. The combination of experts is said to constitute a combination of classifiers.

Mixture of experts is one the most famous methods in the category of dynamic structures of combining classifiers, in which the input signal is directly involved in actuating the mechanism that integrates the outputs of the individual experts into an overall output [4]. Consider a modular neural network in which the learning process proceeds by fusing self–organized and supervised forms of learning as shown in Fig. 1. The experts are technically performing supervised learning in that their individual outputs are combined to model the desired response. There is, however, a sense in which the experts are also performing self–organized learning; that is they self–organize to find a good partitioning of the input space so that each expert does well at modeling its own subspace, and as a whole group they model the input space well. The learning algorithm of the mixture structure is described in [15].

However, in our model, in order to improve the performance of the expert networks, and consequently the whole network performance, we devise a modified version of MoE in which each expert is a MLP, instead of linear networks [17]; furthermore, we make use of a parameter obtained from fuzzy logic in the weight update phase. Using this fuzzy parameter, the degree of ambiguity of an input pattern during the learning phase of MLP experts is obtained.

3. Proposed Fuzzy Mixture of Expert

Our proposed model is designed to achieve robust face recognition with PCA in the feature extraction stage, and a mixture of Fuzzy MLP experts in the classification stage (Fig. 2).

In the conventional MLP, the number of nodes in the output layer corresponds to the number of classes in the task to be performed. The winner-takes-all scheme is applied during the learning process in order to define the desired output vector. In the desired output vector, the class to which the input pattern belongs is assigned the value 1, and other classes are assigned the value 0; this is called a crisp desired output. In real-world problems, however, the data are generally ill-defined, with overlapping or fuzzy class boundaries. That is, there might be some patterns with non-zero similarity to two or more pattern classes. In the conventional MLP, this multiple similarity (or membership value) is not considered with the crisp desired output that is used during its training. In order to consider the membership values of each class, it would seem very promising to incorporate fuzzy concepts in forming the desired output [18]. A common way of forming the fuzzy desired output is proposed in [19], which we use to develop our method by using the two parameters of class prototype and variability.

Prototype of a class points to the distribution of values in that class and is determined on a pixel by pixel basis. Prototype parameter is obtained according to Eq. (1).

$$P_{ik} = \frac{\sum_{j=1}^{L} T_{ij}}{L} \qquad i = 1,\dots,M \tag{1}$$

where P_{ik} is the prototype of the i^{th} pixel of class k, $k = 1,\dots C$ (C is the number of classes), T_{ij} is the i^{th} pixel of the training pattern j, M is the number of pixels of a pattern and L is the number of training patterns.

Variability of a class has a fundamental role in the process of calculating the fuzzy desired output and is concerned with the variations of prototype values for each pixel in the set of the training patterns. Variability for each is calculated considering the prototype of that class and it is defined according to Eq. (2).

$$V_k = \frac{\sum_{i=1}^{M} (0.5 - P_{ij})^2}{\frac{M}{4}} \tag{2}$$

Prototype and variability parameters are obtained considering the whole training set before starting the training process. During the training phase, for each input pattern, a comparison is made between the input and the parameters of its corresponding class in order to derive its fuzzy desired output.

The weighted distance between an input and all classes is calculated considering the squared error between the input pattern and the prototype of a class by Eq. (3).

$$D_k = \left(\frac{\frac{1}{E_k}}{\sum_{l=1}^{c} \frac{1}{E_l}}\right)^f \tag{3}$$

$$E_k = \sum_{i=1}^{M} (X_i - P_{ik})^2 \tag{4}$$

where X_i is the i^{th} pixel of the input pattern, f is the fuzziness similarity which determines the rate of decrease of the weighted distance of the input pattern according to its error.

For each input pattern, class membership value is calculated according to the weighted distance of the closest class, other weighted distances and its corresponding variability Eq. (5).

$$Z_k = D_k \times V_k \quad , \quad \mu_k = \left(\frac{Z_k}{Z}\right)^{\exp} \tag{5}$$

where $\exp$ is a fuzzy parameter which defines the width of the membership function, Z is the similarity of the closest class calculated for the input pattern.

In the MLP weight update process, all training patterns have the same impact on adjusting the weights. However, there are patterns that are in overlapping areas and that are mainly responsible for erratic behavior. One way to improve the performance of the weight update algorithm is to consider the amount of correction in the weight vector produced by the input pattern. Here, the amount of correction is defined by the degree of ambiguity of a pattern in which the more ambiguous an input pattern is, the less correction in the weight is made. The degree of ambiguity is an additional parameter to be used in the proposed fuzzy MLP which was not employed in the model proposed in [18]. The degree of ambiguity is defined according to Eq. (6).

$$A = (\mu_i(x) - \mu_j(x))^m \tag{6}$$

where $\mu_i(x)$ is the membership value of the top class i, $\mu_j(x)$ is the membership value of the second highest one and m is an enhancement/reduction fuzzy parameter.($m < 1$ enhances, $m = 1$ maintains and $m > 1$ reduces the influence of the ambiguity of a training pattern). The degree of ambiguity is, then, used as a parameter in the experts weight updating. The new weight updating equation for each of expert will be in 3.2 Section.

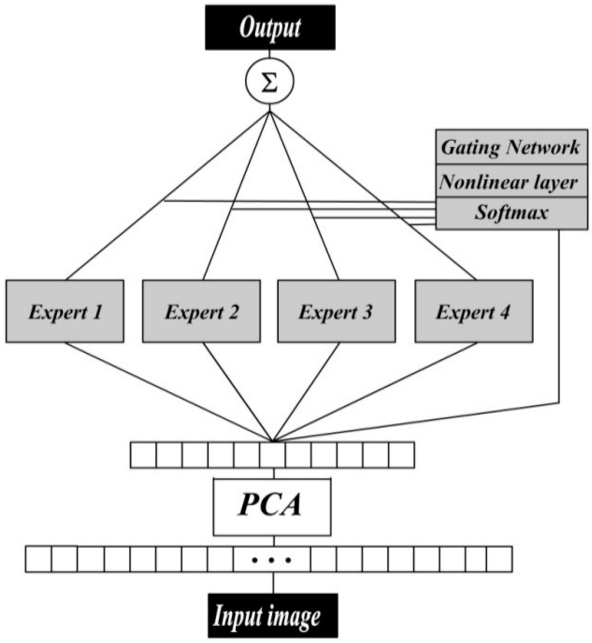

Fig. 2. Sketch of the proposed models.

Each expert is a one-hidden-layer Fuzzy MLP, with Momentum constant, that computes an output vector Oi as a function of the input stimuli vector x and a set of parameters such as weights of hidden and output layer and a sigmoid function as the activation function. It is assumed that each expert specializes in a different area of the face space. The gating network assigns a weight gi to each of the experts' outputs, Oi.

The gating network determines the gi as a function of the input vector x and a set of parameters such as weights of the hidden layer, the output layer and a sigmoid function as the activation function. The gi can be interpreted as estimates of the prior probability that expert i can generate the desired output y. The gating network is composed of two layers: the first layer is an MLP network, and the second layer is a softmax nonlinear operator as the gating network's output. The gating network computes Og , which is the output of the MLP layer of the gating network, then applies softmax function to get:

$$g_i = \frac{\exp(O_{gi})}{\sum_{j=1}^{N} \exp(O_{gi})} \tag{7}$$

So the g_i is nonnegative and sum to 1. The final mixed output of the entire network is:

$$O_T = \sum_i O_i g_i \qquad i = 1,2,3,4 \tag{8}$$

The "normalized" exponential transformation of Eq. (7) may be viewed as a multi-input generalization of the logistic function. It preserves the rank order of its input values, and is a differentiable generalization of the "winner-takes-all" operation of picking the maximum value, so referred to as softmax.

The weights of MLPs are learned using the back-propagation, BP, algorithm, in order to maximize the log likelihood of the training data given the parameters.

Assuming that the probability density associated with each expert is Gaussian with identity covariance matrix, MLPs obtain the following online learning rules:

$$\Delta w_y = \mu_e h_i (y - O_i)(O_i(1 - O_i))Oh_i^T \tag{9}$$

Our method of increasing the learning rate, and also avoiding the danger of instability, is to modify the fuzzy of Eq. (9) by including a Momentum term [17] and degree of ambiguity of Eq.(6):

$$\begin{aligned}\Delta w(n)_y &= \alpha \Delta w(n-1)_y + A\mu_e h(n)_i (y - O(n)_i) \\ &\quad (O(n)_i(1 - O(n)_i))Oh(n)_i^T\end{aligned} \tag{10}$$

$$\begin{aligned}\Delta w(n)_h &= \alpha \Delta w(n-1)_h + A\mu_e h(n)_i\, w(n)_y^T \\ &\quad (y - O(n)_i)(O(n)_i(1 - O(n)_i)) \\ &\quad Oh(n)_i\,(1 - Oh(n)_i)\end{aligned} \tag{11}$$

$$\begin{aligned}\Delta w(n)_{yg} &= \alpha \Delta w(n-1)_{yg} + \mu_g (h(n) - g(n)) \\ &\quad (O(n)_g(1 - O(n)_g))Oh(n)_g^T\end{aligned} \tag{12}$$

$$\begin{aligned}\Delta w(n)_{hg} &= \alpha \Delta w(n-1)_{hg} + \mu_g w(n)_{yg}^T \\ &\quad (h(n) - g(n))(O(n)_g(1 - O(n)_g)) \\ &\quad Oh(n)_g(1 - Oh(n)_g)x_i\end{aligned} \tag{13}$$

where μ_e and μ_g are learning rates for the experts and the gating network, respectively, Oh_i is the output of expert network's hidden layer, and h_i is an estimate of the posterior probability that expert i can generate the desired output y:

$$h_i = \frac{g_i \exp(-\frac{1}{2}(y-O_i)^T(y-O_i))}{\sum_j g_j(-\frac{1}{2}(y-O_i)^T(y-O_i))} \tag{14}$$

This can be thought of as a softmax function computed on the inverse of the sum squared error of each expert's output, smoothed by the gating network's current estimate of the prior probability that the input pattern was drawn from expert i's area of specialization. As the network's learning process progresses, the expert networks "compete" for each input pattern, while the gating network rewards the winner of each competition with stronger error feedback signals. Thus, over time, the gate partitions the face space in response to the expert's performance.

Fig. 3. Samples of facial variations of the ORL dataset.

The inclusion of Momentum term in the back-propagation algorithm tends to accelerate descent in steady downhill directions and has a stabilizing effect in directions that oscillate in sign. The incorporation of Momentum term in the back-propagation algorithm represents a minor modification to the weight update process, yet it may have some beneficial effects on the learning behavior of the algorithm. The Momentum term may also have the benefit of preventing the learning process from terminating in a shallow local minimum on the error surface [17].

4. Experimental Results

In this section, we describe the three experiments that were carried out to evaluate the performance of our proposed face recognition method. In order to select the best candidates for constructing the Mixture of Experts, we first performed two preliminary experiments on three committee machines and two individual classifiers. The third experiment is designed to test our combining classifier model.

The investigated aspects of the experiments are, first, finding the optimum number of principal components for the feature extraction phase, second, selection of a suitable neural network topology to form the experts, third, finding the optimum architectural parameters of each individual neural classifier, and forth, Selection of the optimal training parameters, such as the number of training epochs and testing the individual and the ensemble structures on an unseen set of 200 test faces.

Fig. 4. Sample of ORL images resized to 48×48 pixel.

As mentioned before, we test our model using the ORL dataset. The images are grey scale with a resolution of 92×112 (Fig. 3). In our experiments, 200 images are chosen for training and the remaining 200 for testing. For implementation convenience, all images were first resized to 48×48 pixels (Fig. 4).

PCA is a well-known statistical technique for feature extraction. Each $M \times N$ image in the training set is row concatenated to form $MN \times 1$ vectors x_k. Given a set of N_T training images $\{x_k\}_{k=0,1,\ldots,N_T}$ the mean vector of the training set is obtained as:

$$\overline{x} = \frac{1}{N_T}\sum_{k=1}^{N_T} x_k \tag{15}$$

A $N_T \times MN$ training set matrix $X = [x_k - \overline{x}]$ can now be built. The basis vectors are obtained by solving the eigenvalue problem:

$$\Lambda = V^T \sum_x V \tag{16}$$

where $\sum_x = XX^T$ is the covariance matrix, V is eigenvector matrix of $\sum_x$ and Λ is the corresponding diagonal matrix of eigenvalues. As the PCA has the property of packing the greatest energy into the least number of principal components, eigenvectors corresponding to the largest eigenvalues in the PCA are selected to form a lower-dimensional subspace. It is proven that the residual reconstruction error generated by discarding the $N_T - m$ components is low even for small m . Figure 5 illustrates the largest and smallest eigenfaces generated by PCA on ORL dataset used in our experiment in the training set.

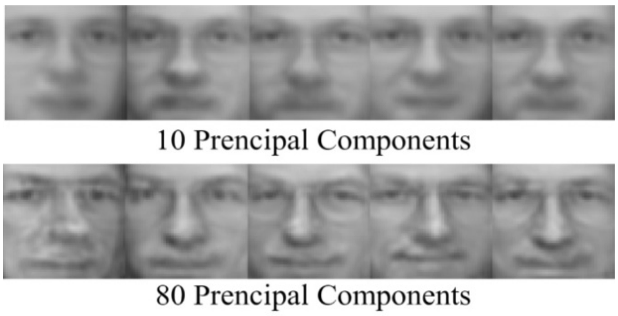

Fig. 5. Two examples of eigenfaces that include the largest and smallest principal components used in our experiment.

One purpose of the experiments conducted in this paper is to evaluate the effect of different learning algorithms on the number of required hidden neurons. We investigate the training speed and, in general, the recognition rate. Finally, Mixture of FMLP Experts method is compared with other face recognition systems.

We performed the above mentioned experiment with MLP, fuzzy MLP, Mixture of MLP Experts and Fuzzy Mixture of Experts. The experiment was repeated for 20 times by randomly choosing different training and testing sets form the ORL dataset. The number of principal components representing the eigenface was set to different values of 10, 15,20,...,80. A total of 20 runs were executed for each learning algorithm. In Fig. 6 the average performance with respect to the number of principal components for the above mentioned algorithms in plotted. As shown in Fig. 6, the first 40 principal components yields the best average performance.

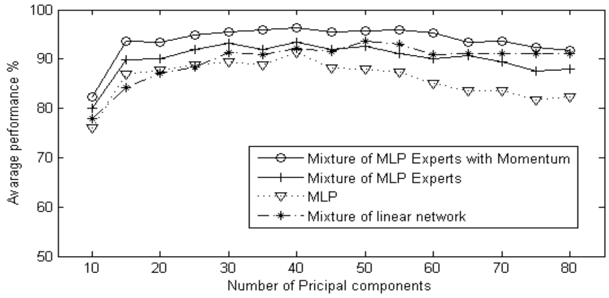

Fig. 6. Average performance with respect to principal components for different learning algorithms on the ORL dataset.

Setting the number of principal components to 40, we evaluate and compare the performance of fuzzy Mixture of Experts with other networks. The single MLP has one hidden layer, and it is trained using the back–propagation algorithm,. To have a reasonable evaluation, we compare the single MLP with the mixture model, consisting of four simple MLPs. Experimental results support our claim that four simple MLPs in the mixture architecture perform the recognition task much better than a single MLP.

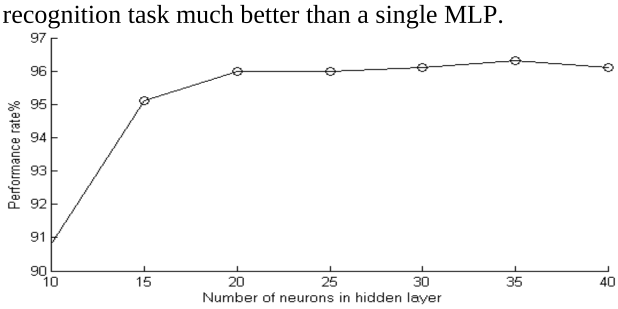

Fig. 7. Performance rate versus number of neurons in hidden layer for experts of our model.

All the networks were trained and tested on the same training and test sets. From the results, it is evident that the performance of the Fuzzy Mixture of Experts network is superior to that of others. Table 2 lists the details of the training parameters and the best structures, in terms of higher recognition rate, for the experts and gating network.

To find the sufficient number of neurons in hidden layer of experts, we experiment it with different number of neurons. As shown in Fig. 7, experts with 20 neurons in their hidden layer reveal the best performance. To find the required number of epochs for reaching the highest recognition rate in both Mixture of MLP Experts and Fuzzy Mixture of Experts, we repeated the same experiment with different epochs in training the networks and observed their performance.

Table 2: The details of the training and fuzzy parameters in as well as the recognition rates of the MLP, Fuzzy MLP, Mixture of MLP Experts and Fuzzy Mixture of Experts networks on the set test.

Network Model	MLP	Fuzzy MLP	Mixture of MLP Experts	**Fuzzy Mixture of Experts**
Topology	40:40:40	40:20:40	Experts: 40:20:40 Gating: 40:10:4	**Experts: 40:20:40 Gating: 40:10:4**
Learning rate & fuzzy parameters	0.1	0.1 Momentum: 0.5 f: 0.9 exp: 0.5 m: 0.9	Experts: 0.1 Gating: 0.5	**Experts: 0.1 Gating: 0.5 Momentum: 0.5 f: 0.9 exp: 0.5 m: 0.9**
Max percentage	91	93.5	96	**98.9**

As shown in Fig. 8, Fuzzy Mixture of Experts needs less epochs to reach its best result, in other words, it converges faster in comparison with the same network which is trained without Fuzzy MLP.

5. Conclusion

This paper presented the use of a modified ME structure to improve human face recognition. Our ME is composed of four modified experts (which were fuzzy MPLs) and a gating network (a MLP). In fuzzy MLP, the training rules are modified such that the weights are updated considering the degree of ambiguity of each training sample. That is, in case of an ambiguous training sample (a sample that is likely to be misclassified) no weight update is applied. Our proposed ME was trained and tested on the ORL dataset. The recognition rates achieved by the modified ME turned out to be higher than that of a single MLP, Fuzzy MLP, and the ME with MLPs as experts trained without the fuzzy system.

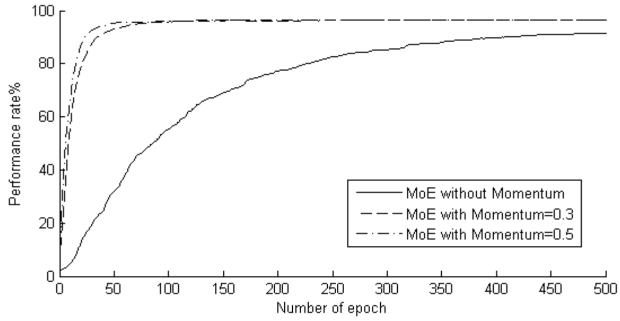

Fig. 8. Performance rate versus number of epoch for Mixture of FMLP Experts and Mixture of FMLP Experts.

References

[1] Zadeh, L., Fuzzy sets. Inf Cont, No.8, pp.338-353, 1965.

[2] Ruspini, E., Bonissone, P., and Pedrycz, W., Handbook of Fuzzy Computation. Ed. Iop Pub/Inst of Physics. 1998.

[3] Cox, E., The Fuzzy Systems Handbook. AP Professional – New York, 1994.

[4] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, USA, 1999.

[5] Mehrotra, K., Mohan, C. K., and Ranka, S., Elements of Artificial Neural Networks, The MIT Press, 1997.

[6] X. Yao, “Evolving artificial neural networks”, IEEE 87, 1999, pp.1423–1447.

[7] R. Avnimelech and N. Intrator, “Boosted mixture of experts: An ensemble learning scheme”, Neural Computation, 11(2), 1999, pp.483 –497.

[8] D. Jimenez and N. Walsh, “Dynamically weighted ensemble neural networks for classification”, International Joint Conference on Neural Networks, 1998, pp.753–756.

[9] Y. Liu and X. Yao, “Evolving modular neural networks which generalize well”, IEEE International Conference on Evolutionary Computation, USA, 1997, pp.605–610.

[10] Y. Liu and X. Yao, “Ensemble learning via negative correlation”, Neural Networks, 12, 1999, pp.1399–1404.

[11] Y. Liu, X. Yao, and T. Higuchi, “Evolutionary ensembles with negative correlation learning”, IEEE Trans Evolutionary Computation, 4(4), 2000, pp.380–387.

[12] A. Sharkey, “Combining Neural Nets Ensemble and Modular Multi-Net Systems”, Springer, New York, 1998.

[13] R.A Jacobs, M.I Jordan, and A.G. Barto, “Task decomposition through competition in a modular connectionist architecture: what and where vision tasks”, Technical report, University of Massachusetts, 1991.

[14] R.A. Jacobs and M.J. Jordan, “Adaptive mixtures of local experts”, Neural Computation 3, 1991, pp.79 – 87.

[15] Jacobs, R., Jordan, M., Nowlan, S., and Hinton, G., “Adaptive Mixtures of Local Experts”, Neural Comput3, 1991, pp. 79–87.

[16] M. Turk and A. Pentland, “Eigenfaces for Recognition”, *Journal of Cognitive Neuroscience*, Vol.3 (1), pp. 71–861.

[17] R. Ebrahimpour, E. Kabir and M.R. Yousefi, “Face Detection Using Mixture of MLP Experts”, *Neural Processing Letters*, 2007, pp. 69-82.
[18] Pal, S. K., Mitra, S., “Multilayer perceptron, fuzzy sets and classification”, IEEE Transactions on Neural Networks, 3(5),1992, pp. 683–697.
[19] Canuto, A., Howells, G., and Fairhurst, M.“An investigation of the effects of variable vigilance within the repart neuro-fuzzy network”, Journal of Intelligent and Robotics Systems, Special on Neural-Fuzzy Systems, 29(4), 2000, pp. 317–334.
[20] N. Taheri M., A. Hajiany, R. Ebrahimpour G. Sepidnam, “A Modified Mixture of MLP Experts for face recognition”, IPCV 2008, Las Vegas, Nevada, USA, pp.740-745.

Nina Taheri Makhsoos received the B.S. degree in Computer Engineering from Najaf Abad Azad University of Esfahan in 2000 and 2004and M.S. degree in Software Engineering from Ferdowsi of Mashhad University in 2005 and 2008, respectively. Her research interest includes image processing, pattern recognition, fuzzy logic, neural networks and their application to face recognition.

Reza Ebrahimpour was born in Mahallat, Iran, in July 1977. He received the BS degree in electronics engineering from Mazandaran University, Mazandaran, Iran and the MS degree in biomedical engineering from Tarbiat Modarres University, Tehran, Iran, in 1999 and 2001, respectively. He received his PhD degree in July 2007 from the School of Cognitive Science, Institute for Studies on Theoretical Physics and Mathematics, where he worked on view-independent face recognition with Mixture of Experts. His research interests include human and machine vision, neural networks, and pattern recognition.

Alireza Hajiany BS student, Electrical Engineering Department of Shahid Rajaee University Tehran, Iran from 2004 up now. His research interest includes signal processing, ensemble neural networks, face recognition.