Investigation the Effect of Particle Swarm Optimization in Performance of Mixture of Experts

Dimple Rani
University institute of engineering and technology PU Chandigarh-India

Javad Hatami
Human Compute Interaction of Umeå University-Sweden

Diba Meysamiazad
Department of Mathematics, University of Padua-Italy

ABSTRACT
Mixture of experts (ME) is one of the most popular and interesting combining methods, which has great potential to improve performance in machine learning. ME is established based on the divide-and-conquer principle in which the problem space is divided between a few neural network experts, supervised by a gating network. In earlier works on ME, different strategies were developed to divide the problem space between the experts. As result, we have introduced a new method based on the principles of Particle Swarm Optimization (PSO) as a learning step in ME. In this paper, different aspects of the proposed method are compared with the common version of ME. The result carried out from this paper shows that the new method is robust to the variation of ensemble complexity in terms of the number of individual experts, and the number of hidden units.

General Terms
Pattern Recognition, Artificial Intelligence, Neural Networks.

Keywords
Mixture of Experts, Neural Network, Particle Swarm Optimization, Classification

1. INTRODUCTION
Combining classifier is one of the popular approaches in pattern recognition, which leads to having a better classification, increase in the recognition rate and improve in the system reliability. It is usually a good approach in complicated problems due to the small sample size, class overlapping, dimensionality, and the substantial noise in the input samples. Previous experimental and theoretic results have shown the better performance of combining classifier when base classifiers have small error rates, and their errors are different [1]; in other words, the base classifier makes an uncorrelated decision in this case. Generally, classifier selection and classifier fusion are two types of combining classifiers [2]. One of the most popular methods of classifier selection is ME, originally proposed by Jacobs et al. [3]. In the ME, the conditional probability density of the target output is modeled by mixing the outputs from a set of local experts. Each of these, derives a conditional probability density of the target output. The outputs of expert networks are combined using a gating network trained to select the expert(s) that has the best performance in solving the problem [4-7]. In the basic form of ME [3], the expert and gating networks are linear classifiers, however, for more complex classification tasks, the expert and gating networks could be more complicated. Back propagation (BP) algorithm is the most popular technique in neural networks' training. It is an approximation of the Least Mean Square (LMS) algorithm, which is based on the steepest descent method. BP technique follows a straightforward algorithm, but there are some disadvantages to it. Backwards calculating weights method does not seem biologically valid. Neurons do not seem to work backward to adjust their synaptic weights [8]. Furthermore, it contains extensive calculations, and so, often has a slow convergence speed [9]. PSO is an option to solve this problem. It is a population based stochastic optimization technique developed by J. Kennedy and R. Eberhart in 1995. It models the cognitive and social behavior of a flock of birds flying over an area in search of food [10]. PSO has been applied to improve neural networks in various aspects, such as network connection weights, network architecture and learning algorithms. Recently, several papers have been published reporting the replacement of the BP algorithm by PSO for some neural network structures [11-13]. This paper investigates linked references on the efficiency of PSO and BP in terms of the robustness and convergence rate for training a ME. The rest of the paper is stated as follow: In Section 2, ME method is explained. Our proposed method for combining classifier results and experimental results are introduced in Section 4 and 5, respectively. The conclusion is presented in Section 5.

2. Mixture of Experts
MLPs have been used successfully in different regression and classification problems so far. However, for large problems the parameter space of MLPs becomes huge and this leads to the computationally intractability of the training. To tackle this problem, one can take advantage of the "divide and conquer" principle. According to the divide and conquer approach, it is useful to solve a complex task by dividing it into simple solvable tasks and then properly combining the solutions. A well-known method that works based on this principle is ME which was proposed by Jacob et al. [3] for the first time. Their proposed model contains a population of simple linear classifiers (the experts), and a gating network did the mixing of their outputs. Technically the experts perform supervised learning in order to model a combination of the outputs of individual experts. The experts are also self-organized to find a good part of the input space, each expert models its own subspace, and the combination of all experts well models the input space. In order to improve the performance of expert networks, MLPs are applied instead of linear networks; so related revision is necessary in the learning algorithm.
The learning algorithm is modified using an estimation of the posterior probability of desired output by each expert. This way, the gating and expert networks match and improve this model to select the best expert(s). The weights of MLPs expert networks are updated considering those estimations, and the procedure is repeated [14]. Each expert has one hidden layer that computes an output O_i. We assume that each expert specializes in a specific area of the input space.

The gating network assigns a weight g_i to each of the experts' output O_i. The gating network determines g_i, and linear activation is used in the output layer to avoid range constraints. Note that the softmax function is applied to the gating network to produce more diverse output signals. The g_i can be interpreted as an estimate of the probability of expert i generating the desired output y for input x. The experts compete to learn the training patterns, and the gating network mediates the competition. Thus, the gating network computes O_g which is the output of the MLPs layer of the gating network, then applies the softmax function to get:

$$g_i = \frac{\exp(O_{gi})}{\sum_{j=1}^{N} \exp(O_{gj})} \quad i = 1,..., N$$

Where N is the number of expert networks, so g_i are non-negative and sum to 1. The final mixed output of the entire network is:

$$O_T = \sum_i O_i g_i \quad i = 1,..., N$$

The classifier weights are updated for each expert i according to the following rules:

$$\Delta w_y = \eta_e h_i (y - O_i)(O_i(1 - O_i))O_{hi}^T$$
$$\Delta w_h = \eta_e h_i w_y^T (y - O_i)(O_i(1 - O_i))Oh_i(1 - Oh_i)x_i$$

In equation 3 and 4, η_e is the learning rate for the expert. w_h and w_y are the parameters of experts describing weights of input to hidden and hidden to output layer, respectively. Similarly, O_{hi}^T is the output of the hidden layer of expert, and h_i is an estimation of the posterior probability that expert i generates the desired output y:

$$h_i = \frac{g_i \exp\left(-\frac{1}{2}(y - O_i)^T (y - O_i)\right)}{\sum j g_j \exp\left(-\frac{1}{2}(y - O_j)^T (y - O_j)\right)}$$

As pointed out in [14], in the network's learning process, the experts "compete" for explaining the input while the gate network rewards the winner of each competition using larger feedbacks. So the gate divides the input space according to the performance of experts.

3. Training Algorithms

In ME, the weight assigned to output of each expert is expected to increase the classification accuracy rate if a weight could be designated to each pattern class of experts. In other words, assigning a weight to each output neuron of experts could enhance the accuracy rate of the ensemble system. Figure 1 shows that the structure of the ensemble model applied in this paper is able to enhance the accuracy rate of the ensemble system.

Figure 1 shows the structure of the ensemble model applied in this paper.

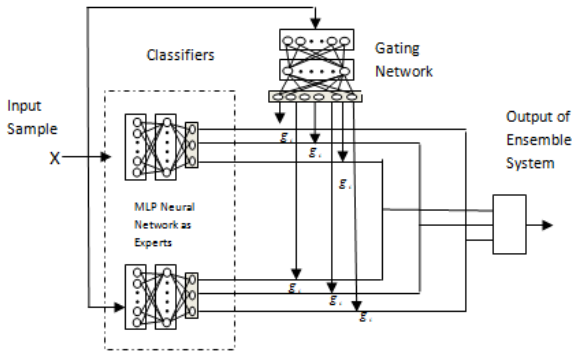

Fig 1: A weight is assigned to each output neuron of the classifier

Two different methods are compared for training the gating network: BP and PSO. Both methods are applied online and are compared in terms of their respective Mean Square Error (MSE).

3.1. Backpropagation Algorithm

The BP algorithm was developed by Paul Werbos in 1974. Based on LMS algorithm, BP applies a weight correction to the neural network connection weights which is proportional to the partial derivative of the error function [13]. This adjustment to the weights is in the negative direction of the gradient of the error (steepest descent). The error function is defined as:

$$E(t) = \frac{1}{2}.|e(t)|^2$$

where $e(t)$ is the error value, i.e. the difference between the target and the estimated output. The gating network weights are adjusted according to:

$$W(t+1) = W(t) - \eta \frac{\partial E(t)}{\partial W(t)}$$

where η is the learning rate parameter. A large learning rate might lead to oscillations in the convergence trajectory, while a small learning rate provides a smooth trajectory at the cost of slow convergence speed.

3.2. Particle Swarm Optimization

Nowadays, the applications of metaheuristic in solving problems are increased dramatically [15-18]. Among the all metaheuristic algorithms, PSO is a population (swarm) based optimization tool. The particles are evaluated using a fitness function to see how close they are to the optimal solution [9]. Particles have a tendency to duplicate their individual past behavior that has been successful (cognition) as well as to follow the successes of the other particles (socialization). The neural network weight matrix is rewritten as an array to form a particle. Particles are initialized randomly and updated afterwards according to:

$$\Delta W(t+1) = w.\Delta W(t) + C_1.(lBest(t) - W(t)) + C_2(gBest(t) - W(t))$$
$$W(t+1) = W(t) + \Delta W(t+1)$$

where, w and C_1. and C_2 Care inertia, cognitive and social acceleration constants, respectively. For every specific particle, *lBest* is the best solution that the particle has achieved so far and indicates the tendency of the individual particles to replicate their corresponding past behaviors that have been successful. *gBest* is the global best solution so far, i.e. the best solution that any particle (in the whole population) has achieved so far. This quantity indicates the tendency of the particles to follow the success of others. Another important parameter associated with PSO is the maximum velocity $V_{\max}$ which determines the resolution or fineness that the search space is searched with. Using a large value might cause the particles to fly past good solutions, while a small number can trap particles in the local optima. Selection of the constant parameters, population size, neighborhood size and suchlike depend on the problem and for this specific problem will be explained in the next section. In our proposed method, a collection of several experts is created using ME, and then, optimal weights for linear combination of experts output are found using PSO.

4. EXPERIMENTAL RESULTS

PSOME is verified on six standard datasets taken from the UCI Machine Learning Repository as summarized in table I. These datasets were downloaded from www.ice.uci.edu

Table 1. Six Datasets from the UCI machine learning repository

Dataset	Class	Size	Attributes
Sonar	208	60	2
Breast Cancer	569	32	2
Pima Indian Diabetes	768	8	2
Glass	214	10	7
Vehicle	946	18	4
Iris	150	4	3
Pima Diabetes	768	8	2

In the experiments, we used a ten-fold cross validation for each dataset. The dataset is divided into ten disjoint subsets using stratified sampling. Each subset is, in turn, taken as the test set, altogether making ten trials. In each experiment, a remaining subset is randomly chosen as the validation set, while the eight other remaining subsets are combined to form the training set. A different prefixed random seed is used to generate the required random numbers. For each fold, the system is trained using the training set, stopped by one of the different criteria using the validation set, and the ensemble obtained by combining the population at the stopping point is tested on the test set. The test set error is treated as the primary result of each trial. In this paper, the following architecture is used for the ME model: four experts, each of which is MLPs with one hidden layer consisting of three hidden nodes. Nodes in the experts are all sigmoidal. The gating network is a MLPs, with four linear output nodes corresponding to the experts and five hidden nodes, but in PSOME it has a number of output neuron equal to the number of dataset classes for each expert. The outputs of the gate network are passed through a softmax function to obtain probability-like values. The ME model is trained with the BP algorithm for 400,000 forward propagations with three sets of different learning rates for the experts and the gate:

1. $\eta_{\text{expert}} = \eta_{gate} = 0.1$
2. $\eta_{\text{expert}} = 0.1, \eta_{gate} = 0.01$
3. $\eta_{\text{expert}} = 0.01, \eta_{gate} = 0.1$

For PSOME, The settings of the PSO algorithm are as follows: $V_{\max}$ was set to 5, swarm size was set to 10, and C_1 and C_2 were both set to 2, and the inertia weight was linearly decreased from 0.7 to 0.4. The maximum number of epochs was limited to 40,000 (number of forward propagations = swarm size × maximum number of epochs). Thus, the overall computational cost (number of evaluations) is consistent between the experiment with BP alone and the experiments with PSO. The MSE is reported as the performance of the ensemble. An ANOVA test is used and plotted to show comparison. In the ANOVA plot, the group mean and error bars are shown; "Two means are significantly different if their intervals are disjoint, and are not significantly different if their intervals overlap" (MATLAB manual). We compare PSOME against the ME. Figure 2 shows an ANOVA significance test of PSOME vs. ME performance, taken over the six datasets and the range of learning rates used. The disjoint intervals imply a statistically significant difference in the performance of ME and PSOME. This provides a strong evidence of the advantage gained by using PSO in conjunction with the ME model.

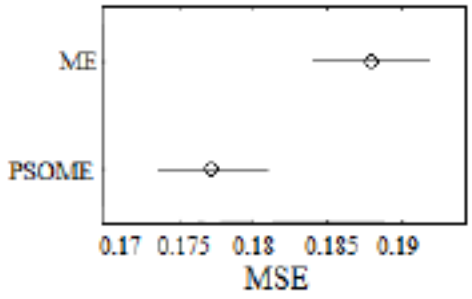

Fig. 2. ANOVA test on the generalization errors for the PSOME vs. ME for six datasets across

Moreover, we test the effect of the ensemble size on the PSOME model on different ensemble sizes, to see how robust the results are to different ensemble sizes. The PSOME model with seven different ensemble sizes of 3, 6, 9, 12, 15, 17 and 19 experts is compared on these datasets. The ANOVA test is presented in figure 3. It is clear that the PSOME model is quite robust to variations in the ensemble size, i.e. the model is not sensitive to different ensemble sizes. The ANOVA plot shows that the ensemble size has no significant effect on the performance of the ensemble.

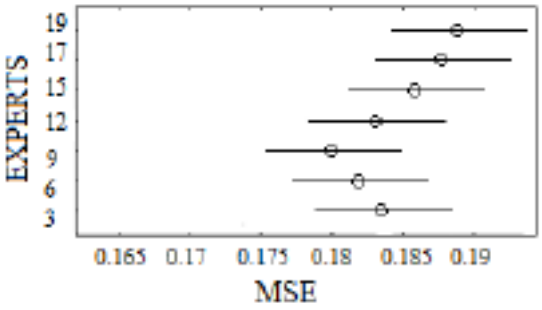

Fig. 3. ANOVA test for the ensemble size factor in PSOME

In addition, we test the PSOME model with different network complexities (represented by the number of hidden units in each expert). PSOME models with six different numbers of hidden nodes: 3, 5, 7, 9, 11 and 13 are compared in terms of performance. ANOVA tests are presented in figure 4. The plots indicate the robustness of PSOME model to the network complexity, i.e. performance is not significantly changed with a different number of hidden units per expert. Although the plot suggests that too simple and too complex (in term of number of hidden units) MLPs are not desirable, the lack of a significant difference suggests that choosing the right complexity is not too important.

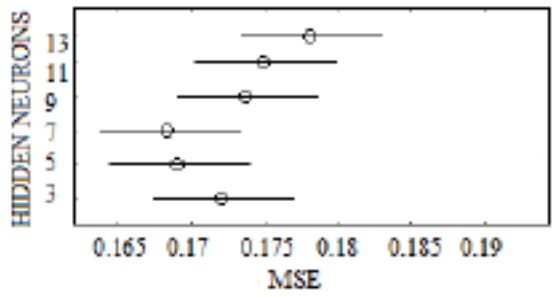

Fig. 4. ANOVA test for the network complexity factor in PSOME

In the last experiment, we compare PSOME and ME in the view of time to show which of them converge faster. Therefore, Iris dataset is used to carry out this experiment. Figure 5 shows that PSOME is much faster than ME.

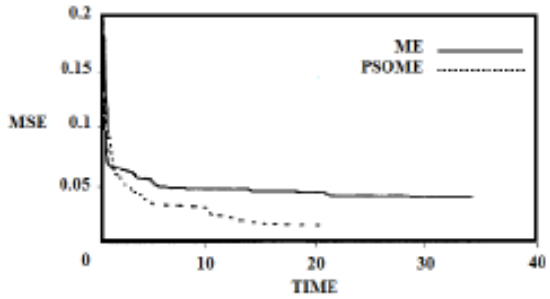

Fig. 5. MSE of Iris dataset plotted over time averaged over 10 trials

5. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a method based on the principles of Particle Swarm Optimization and Mixture of Experts. We have investigated different aspects of the proposed Particle Swarm Optimization Mixture of Experts model. The results of the experiments can be summarized as follows: PSOME is robust to varying ensemble complexity in terms of the number of individual experts, and PSOME is robust to varying MLPs complexity in terms of the number of hidden units. When comparing PSOME and ME, the results show that PSOME is more efficient and faster in comparison with ME on a number of classification problems. The proposed system can contribute to answering many fundamental and practical problems such as self-organization, automatic identification of building blocks and automatic problem decomposition.

6. REFERENCES

[1] Kuncheva, L., M. Skurichina, and R.P. Duin, *An experimental study on diversity for bagging and boosting with linear classifiers.* Information fusion, 2002. **3**(4): p. 245-258.

[2] Woods, K., K. Bowyer, and W.P. Kegelmeyer Jr. *Combination of multiple classifiers using local accuracy estimates*. in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*. 1996. IEEE.

[3] Jacobs, R.A., et al., *Adaptive mixtures of local experts.* Neural computation, 1991. **3**(1): p. 79-87.

[4] Salimi, H., et al., *Extended Mixture of MLP Experts by Hybrid of Conjugate Gradient Method and Modified Cuckoo Search.* International Journal of Artificial Intelligence & Applications, 2012. **3**(1).

[5] Chen, K., L. Xu, and H. Chi, *Improved learning algorithms for mixture of experts in multiclass classification.* Neural networks, 1999. **12**(9): p. 1229-1252.

[6] Hong, X. and C.J. Harris, *A mixture of experts network structure construction algorithm for modelling and control.* Applied intelligence, 2002. **16**(1): p. 59-69.

[7] Güler, İ. and E.D. Übeyli, *A modified mixture of experts network structure for ECG beats classification with diverse features.* Engineering Applications of Artificial Intelligence, 2005. **18**(7): p. 845-856.

[8] Kartalopoulos, S.V. and S.V. Kartakapoulos, *Understanding neural networks and fuzzy logic: basic concepts and applications*. 1997: Wiley-IEEE Press.

[9] Haykin, S., *Neural networks: a comprehensive foundation*. 1994: Prentice Hall PTR.

[10] Hu, X., Y. Shi, and R. Eberhart. *Recent advances in particle swarm*. in *IEEE congress on evolutionary computation*. 2004. Portland.

[11] Gudise, V.G. and G.K. Venayagamoorthy. *Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks*. in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. 2003. IEEE.

[12] Yao, X., *Evolving artificial neural networks.* Proceedings of the IEEE, 1999. **87**(9): p. 1423-1447.

[13] Van den Bergh, F. and A.P. Engelbrecht, *Cooperative learning in neural networks using particle swarm optimizers.* South African Computer Journal, 2000(26): p. p. 84-90.

[14] Dailey, M.N. and G.W. Cottrell, *Organization of face and object recognition in modular neural network models.* Neural Networks, 1999. **12**(7): p. 1053-1074.

[15] Aflakparast, M., et al., *Cuckoo search epistasis: a new method for exploring significant genetic interactions.* Heredity, 2014.

[16] Salimi, H. and D. Giveki, *Farsi/Arabic handwritten digit recognition based on ensemble of SVD classifiers and reliable multi-phase PSO combination rule.* International Journal on Document Analysis and Recognition (IJDAR), 2013. **16**(4): p. 371-386.

[17] Giveki, D., et al., *Detection of erythemato-squamous diseases using AR-CatfishBPSO-KSVM.* Signal & Image Processing, 2012. **2**(4): p. 57-72.

[18] Giveki, D., et al., *Automatic detection of diabetes diagnosis using feature weighted support vector machines based on mutual information and modified cuckoo search.* arXiv preprint arXiv:1201.2173, 2012.