A Multiple Classifier System for fast an accurate learning in Neural Network context

E. F. Romero[1], R.M. Valdovinos[2], R. Alejo[3], J. R. Marcial-Romero[2], J. A. Carrasco-Ochoa[4]

Universidad Autónoma del Estado de Mexico,
[1]Centro Universitario Valle de Chalco, Hermenegildo Galena #3, Col. Ma. Isabel, Valle de Chalco, Mexico
[2]Facultad de Ingeniería, Ciudad Universitaria, Cerro de Coatepec s/n, Toluca, Mexico.
[3]Tecnológico de Estudios Superiores de Jocotitlán, Carretera Toluca-Atlacomulco km 44.8, Col. Ejido de San Juan y San Agustín, Jocotitlán, Mexico
[4]Instituto Nacional de Astrofísica Óptica y Electrónica
{eliasfranck, li_rmvr, ralejoll}@hotmail.com, jrmarcialr@uaemex.mx, ariel@inaoep.mx

Abstract. Nowadays, the Multiple Classification Systems (MCS) (also called as ensemble of classifiers, committee of learners and mixture of experts) constitutes a well-established research field in Pattern Recognition and Machine Learning. The MCS consists in dividing the whole problem with resampling methods, or using different models for constructing the system over a single data set. A similar approach is studied in the Neural Network context, with the Modular Neural Network. The main difference between these approaches is the processing cost associate to the training step of the Modular Neural Network (in its classical form), due to each module requires to be learned with the whole data set. In this paper, we analyze the performance of a Modular Neural Network and a Multiple Classifier System integrated by small Modular Neural Networks as individual member, in order to identity the convenience of each one. The experiments here were carried out on datasets from real problems showing the effectiveness of the Multiple Classifier System in terms of overall accuracy and processing time respect to uses a single Modular Neural Network.

Keywords: Artificial Neural Networks, Modular Neural Networks, Mixture of Experts, Linear Perceptron.

1 Introduction

The Modular Neural Networks (MNN) presents a new trend in Neural Network (NN) architectural designs. It has been motivated by the highly-modular nature in biological networks and based on the "divide and conquer" approach [1]. The MNN bases its structure on the idea of a cooperative or competitive working, fragmenting the problem into modules where each module is part of the whole problem [10]. Some advantages of this network respect to other models are:

1. Learning speed. The numbers of iterations needed to train the individual modules is less than the number of iterations needed to train a Non-Modular NN for the same task [5].

2. Data processing. MNN is useful when it is working with different data sources [2], or when the data has been preprocessed with different techniques.
3. Knowledge distribution. In a MNN, the network modules tend to specialize by learning from different regions of the input space [5]. And the modules can be trained independently and in parallel.

There exist several implementations of the MNN, although the most important difference among them refers to the nature of the gating network. In some cases, this corresponds to a single neuron evaluating the performance of the other expert modules [5], other are based on a NN trained with a data set different from the one used for training the expert networks [2]. Finally, training all modules, including the integrator module, with the same dataset [6].

On the other hand, currently, the multiple classifier system (MCS) (also known as ensemble of classifiers, committee of learners, etc.) is a set of individual classifiers whose decisions are combined when classifying new patterns. Some reasons for combining multiple classifiers to solve a given learning problem are: First, MCS tries to exploit the local different behavior of the individual classifiers to improve the accuracy of the overall system. Second, in some cases MCS might not be better than the single best classifier but can diminish or eliminate the risk of picking an inadequate single classifier. Finally, the limited representational capability of learning algorithms, it is possible that the classifier space considered for the problem does not contain the optimal classifier

To ensure a high performance of the MCS it is necessary to have enough diversity in the individual decisions, and consider an acceptable individual accuracy of each membership, which constitutes the MCS.

Some aspects of the MCS aim to overcome in comparison when a single classifier is used, are [7]: The MCS takes advantage of the combined decision over the individual classifier decisions, the correlated errors of the individual components can be eliminated when the global decisions is considered, the training patterns cannot provide enough information to select the best classifier, the learning algorithm may be unsuitable to solve the problem and finally, the individual search space cannot contain the objective function.

In this paper, a comparative study that aims to display the advantages of both methods, the MNN and a MCS are used for classification task. In the first method (MNN) each member corresponds to a linear perceptron, and in the MCS each individual classifier corresponds to a single MNN, that is to say, the MCS is a neural network made with MNN.

2. Modular Neural Network

MNN called as systems committee, Hierarchical Mixture of Experts or Hybrid Systems [6], bases its structure (modular) on the modularity of the human nervous system, in which each brain region has a specific function, but in turn, the regions are interconnected. Therefore, we can say that an ANN is modular if the computation performed by the network can be decomposed into two or more modules or subsystems that work independently on the same or part of the problem. Each module corre-

sponds to a feed forward artificial neural network, and can be considered as neurons in the network as a whole.

In its most basic implementation, all modules are of the same type [5], [2], but different schemes can be used. In the classical architecture, all modules, including the gating module, have *n* input units, that is, the number of features in the sample. The number of output neurons in the expert networks is equal to the number of classes *c*, whereas in the gating network it is equal to the number of experts *r* [6] (Fig. 1).

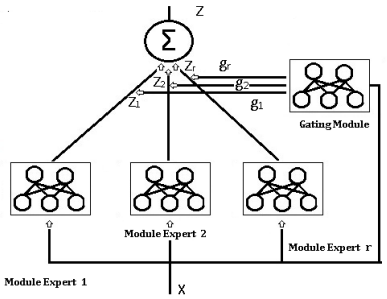

Fig. 1. Representation of the MNN architecture [6].

In the learning process, the network uses the stochastic gradient function:

$$-\ln\left(\sum_{i=1}^{r} g_i * \exp\left(-\frac{1}{2}\|s - Z_i\|^2\right)\right) \quad \text{(eq.1)}$$

where *s* is the desired output for the input *x*, and Z_i is the output vector of the *j'th* expert network, g_j is the output of the gating network, u_i is the total weighted input received by output neuron *j* of the gating network.

Given a pattern *x* *n*-dimensional as input, the overall learning process of the MNN considers the following steps:

1. Random initialization of the synaptic weights for the different networks with small values uniformly distributed. Henceforth, we will consider w_{ji} as weights of the expert network and w_{ti} as the integrating network.
2. The pattern *x* is presented to each and every one of the networks (experts and integrating network) so, the output of the knowledge network is given by:

$$Z_i^m = x * \mathrm{w}_{\mathrm{ji}}^{\mathrm{m}} \quad \text{(eq.2)}$$

where *x* is the input vector, and the superscript *m* is indicative of module. Similarly, the output of the gating network is obtained by, where $u_i = x * w_{ti}$:

$$g_i = \frac{\exp(u_i)}{\sum_{j=1}^{r} \exp(u_j)} \quad \text{(eq.3)}$$

3. Adjusting the weights of the expert networks and the gating network: To adjust weights, two criteria are taken into account.
   a. From expert networks: $w_{ji}^m(I+1) = w_i^m(I) + \eta * h_i(s - Z_i^m)x$ (eq.4)
   b. For the gating network: $w_{ti}(I+1) = w_{ti}(I) + \eta(h_i(I) - g_i(I))x$ (eq.5)
   where:

$$h_i = \frac{g * \exp\left(-\frac{1}{2}\left\|s - Z_i^m\right\|^2\right)}{\sum_{j=1}^{r} g * \exp\left(-\frac{1}{2}\left\|s - Z_i^m\right\|^2\right)} \quad \text{(eq.6)}$$

4. Finally, the network decides how the modules outputs will be combines to obtain the final output of the MNN by: $Z = \sum_{i=1}^{r} gi * z$ (eq.7)

3 Multiple Classifier System

Let $D = \{D_1,..., D_h\}$ be a set of h classifiers. Each classifier D_i $(i = 1,..., h)$ gets as input a feature vector $x \in R^n$, and assigns it to one of the c problem classes. The output of the MCS is an h-dimensional vector $[D_1(x), . . . , D_h(x)]^T$ containing the decisions of each h individual classifiers. After that the individual decisions are combined by some strategy [9], [8] in order to obtain a final decision.

For constructing a MCS it is based on two aspects: the diversity in the individual decisions and the accuracy of the single classifiers. The methods used to achieve diversity can be described in five groups [4]: Pattern manipulation, attribute manipulation, tags manipulation, using different classification algorithms and use randomness.

To integrate the MCS, in this study we use subsamples which consider patterns manipulation, such that the resulting subsets have a proportional size to the number of classifiers that integrate the MCS. Thus, in the experiments here reported the MCS was integrated with 7 and 9 classifier each one, according to [11], this means that the subsample only includes seven or nine percent of the samples included in the original training dataset.

To obtain the subsamples, we use the random selection without replacement of patterns [12] and Bagging [3]. In the first method, the random selection is performed without replacement of patterns in which a certain pattern cannot be selected more than once, thereby reducing the redundancy patterns. On the other hand, Bagging produces subsamples called Bootstrap, where each subsample has the same size than the original dataset. For each subsample obtained with Bagging, each pattern has a probability of $1-(1/m)^m$ of being selected at least once between the m times that is selected with , that is to say, each pattern has approximately 63% chance of appearing in the subsample.

When the subsamples are integrated using some resampling method, each one is presented to the MCS (Fig. 2). After that, for combining the individual classifier decisions in the literature two strategies are proposed: Fusion and selection. In classifier selection, each individual classifier is supposed as an expert in a part of the feature space and correspondingly, only one classifier is selected to label the input vector. In classifier fusion, each component is supposed to have knowledge of the whole feature space and thus, all individual classifiers are taken into account to decide the label for the input vector.

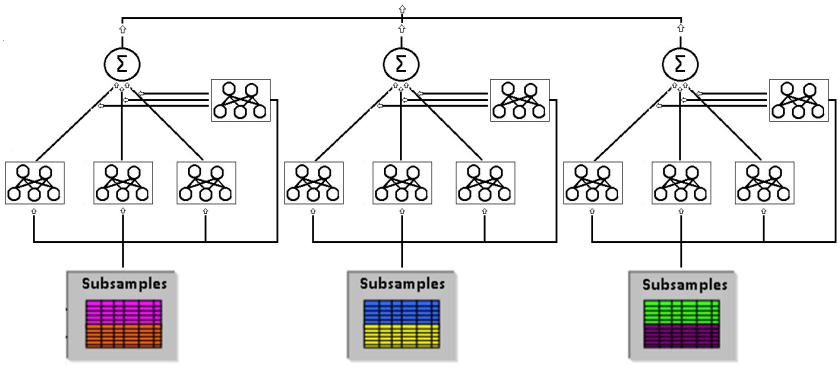

Fig. 2. MCS of MNN.

4 Experimental Results

The results correspond to the experiments carried out over 12 real data sets taken from the UCI Machine Learning Database Repository (http://archive.ics.uci.edu/ml/).

Table 1. A brief summary of the UCI databases used in this paper.

Dataset	Classes	Features	Training samples	Test Samples
Cancer	2	9	546	137
German	2	24	800	200
Heart	2	13	216	54
Iris	3	4	120	30
Liver	2	6	276	69
Phoneme	2	5	4322	1082
Pima	2	8	615	153
Satimage	6	36	5147	1288
Segment	7	19	1848	462
Sonar	2	60	167	41
Vehicle	4	18	678	168
Waveform	3	21	4000	1000

For each database, we estimate the average predictive accuracy and processing time by 5-fold cross-validation, considering the 80% as the training set and the remaining as the test set (20%). According to the scheme of MNN and the MCS, some specifications are as follows:

1. Topology. Each expert in the MNN corresponds to a linear perceptron, in which the number of nodes in the input layer corresponds to the number of attributes in the input pattern. For the experts network, the number of neurons in the output layer is equal to the number of categories in the problem, while for the integrating network is equal to the number of experts used.

2. Connection weights. The connection weights where initialized to random values in the range between -0.5 and 0.5.
3. Each MNN consists of 5 modules and an gating network.
4. For the final decision on the MCS, simple majority voting was used.

Only the result of the best technique on each database has been presented. Analogously, for each database, related to the number of subsamples to induce the individual classifiers, that is, the number of classifiers in the MCS, we have experimented with 7 and 9 elements, and the best results have been finally included in Table 2. Besides, single MNN classification accuracy for each original training set is also reported as the baseline classifier.

Since the accuracies are very different for the distinct data sets, using these results across the data sets will be inadequate. Instead we calculate ranks for the methods. For each data set, the method with the best accuracy receives rank 1, and the worst receives rank 5. If there is a tie, the ranks are shared. Thus the overall rank of a method is the averaged rank of this method across the 12 data sets. The smaller rank indicates the better method.

In Table 2 are two sections; the first one includes the MNN results. The second section shows the results when the MCS is used with 7 and 9 classifiers. In this case, the corresponding capital letter identifies the resampling method used for obtaining the subsamples: randomly without replacement (A) and Bagging (B). The results correspond to the overall accuracy and the standard deviation included in parentheses and values in bold type indicate the highest accuracy for each database.

Table 2. Overall Accuracy results.

Dataset	MNN	MCS 7 Classifiers		MCS 9 Classifiers	
		A	B	A	B
Cancer	**88.4** (4.6)	**88.4** (3.1)	87.9 (3.0)	87.1 (4.7)	86.5 (4.2)
Heart	73.7 (8.6)	**81.5** (5.4)	**81.5** (4.5)	78.9 (4.7)	80.4 (7.4)
Liver	63.5 (5.4)	54.8 (8.1)	62.9 (6.9)	62.0 (4.9)	**67.0** (3.8)
Pima	66.5 (1.6)	**68.0** (1.8)	67.6 (3.2)	66.1 (2.3)	67.8 (2.4)
Sonar	65.9 (6.2)	73.7 (3.2)	67.8 (4.7)	**77.1** (12.2)	70.7 (7.1)
Iris	80.7 (11.4)	78.0 (6.9)	**82.0** (8.0)	78.0 (7.7)	80.0 (6.7)
Vehicle	36.4 (7.1)	**47.1** (3.7)	42.8 (10.9)	42.2 (4.0)	43.5 (3.8)
German	61.8 (18.0)	**73.7** (1.3)	72.4 (4.5)	73.2 (1.9)	72.7 (4.1)
Phoneme	67.9 (4.5)	67.2 (5.5)	**68.9** (3.5)	67.7 (4.2)	68.1 (4.2)
Waveform	77.2 (2.7)	81.6 (1.7)	**82.0** (2.6)	79.2 (3.8)	80.2 (3.3)
Segment	**78.2** (5.6)	75.0 (2.2)	74.9 (2.2)	76.9 (2.4)	74.5 (1.8)
Overall Rank	46.5	34.0	28.5	37.5	33.0

From results shown in Table 2, some comments may be drawn. First, except with Cancer and Segment data set, it is clear that some MCS schemes leads to better performance than the MNN. This is confirmed by the general basis of the MNN, which clearly corresponds to the poorer. Second, comparing the MCS using 7 or 9 classifiers, it is possible to observe that when we use a MCS with 7 classifiers we can find some results with a precision greater than (or equivalent) rating when nine classifiers are used. Finally, to compare different resampling methods, the A method (random

selection without replacemnt) behave generally better performance than the B method (Bagging), using MCS with 7 classifiers on 5 datasets. In fact, for best results, the details are still very close to the winner.

The Vehicle data set is a special case due to the poor performance, regardless of the scheme used. In this case, a thorough analysis of the data distribution is necessary in order to identify the reason why the MNN and the MCS are not able to recognize the kinds of problem which is required.

Another aspect to be analyzed is the computational cost associated with each model. To this end, Table 3 shows the time required in minutes during the training and the classification process by each classifier model.

Table 3. Training time (in minutes).

Dataset	**MNN**	**MCS with**			
		7 Classifiers		**9 Classifiers**	
		A	**B**	**A**	**B**
Cancer	11.3	9.4	9.2	9.5	9.2
Heart	5.5	4.3	4.1	3.7	2.5
Liver	3.3	1.5	1.3	1.7	1.7
Pima	9.8	9.8	9.9	9.6	9.8
Sonar	9.1	0.3	0.4	0.3	0.3
Iris	2.3	2.3	2.2	1.8	1.8
Vehicle	0.3	0.3	0.3	0.3	0.3
German	31.4	21.6	22.0	25.2	22.9
Phoneme	66.8	61,6	55.6	57.4	59.0
Waveform	174.5	131.5	124.7	133.0	127.2
Segment	1.4	2.0	2.1	2.1	2.1

Results in the Table 3 clearly show large differences between the processing times obtained by the three models used. It is interesting to note that in the majority of cases, the time required by the MNN is almost two times more than the required by any MCS. For example, with Sonar dataset the MNN requires nine times more than the MCS. These differences could be because the MCS uses small subsamples in the training process m/L, where m, is the number of patterns of training and L the number of subsamples [12], reducing the computational cost in terms of runtime. In fact, using 9 classifiers requires less time in most cases, because the subsamples are smaller.

Finally, regarding the performance of the schemes used, we can note that the best classification results was obtained with an MCS with 7 classifiers requiring less time processing respect to the single MNN and short differences respect to an MCS with 9 members.

5 Concluding Remarks and Future work

Designing a MCS with MNN as individual classifiers has been here analyzed. Two MCS were used, with 7 and whit 9 classifiers. For the single MNN architecture, we have employed five network experts and one gating network. The experimental re-

sults allow comparing these models, in terms of processing time and predictive accuracy. From this, it has been possible to corroborate that in general, the MCS clearly outperforms the classifier obtained with the MNN.

In addition, when comparing the behavior of the resampling methods, it has been empirically demonstrated, that to use the random selection without replacement offers the best performance: with greater precision and lower computational cost.

Finally, by comparing the results of the classification and the processing time required for each model, the use of the MCS provides the best performance, being the best option to improve the binomial time-accuracy.

As a future work to expand this research, aimed mainly at the improvement the single MNN performance. In this context, other architectures with different parameters and possible mechanisms such as regularization/Cross-validation must be analyzed. Also, it should be further investigated the relationship between the individual classifiers and the resampling methods in order to determine the "optimal" scenario.

Acknowledgements. This work has partially been supported by grants 3834/2014/CIA project, from the Mexican UAEM.

References

1. Alejo, R. "Análisis del Error en Redes Neuronales: Corrección del error de los datos y distribuciones no balanceadas". *Tesis Doctoral*. Universitat Jaume I, Castelló de la Plana, España (2010).
2. Bauckhage, C. Thurau, C.: "Towards a Fair'n Square aimbot Using Mixture of Experts to Learn Context Aware Weapon Handling", *in Proccedings of (GAME-ON'04),* Ghent, Belgium, pp. 20-24 (2004).
3. Breiman, L.: "Bagging predictors", *Machine Learning* 26 (2), pp.123 – 140, (1996).
4. Dietterich, T. G. "Ensemble methods in machine learning," Lecture Notes in Computer Science, vol. 1857, pp. 1–15 (2000).
5. Hartono, P., Hashimoto S.: "Ensemble of Linear Perceptrons with Confidence Level Output", *in Proceedings of the 4th International.* (USA), pp. 97 -106, (2000).
6. Kadlec, P. and Gabrys B. "Learnt Topology Gating Artificial Neural Networks". IJCNN, pp. 2604-2611. IEEE (2008).
7. Kuncheva, L. I.: "Using measures of similarity and inclusion of multiple classifier fusion by decision templates", *Fuzzy Sets and systems, 122 (3),* pp. 401 -407 (2001).
8. Kuncheva, L.I. Bezdek, J.C. Duin R.P.W.: "Decision templates for multiple classifier fusion". *Pattern Recognition,* 34, pp. 299–314 (2001).
9. Kuncheva L.; Roumen I. Kountchev K.: "Generating classifier outputs of fixed accuracy and diversity", *Pattern Recognition letters,* 23*,* pp. 593 -600 (2002).
10. Martínez L. M., Rodríguez P. A. "Modelado de sus funciones cognitivas para entidades artificiales mediante redes neuronales modulares". Tesis doctoral, Universidad Politécnica de Madrid. España (2008).
11. Valdovinos R.M., Sánchez J.S.: "*Sistemas Múltiples de Clasificación, preprocesado, construcción, fusión y evaluación"*. Académica Española: Alemania (2011).
12. Valdovinos, R.M., Sánchez J.S.: "Class-dependant resampling for medical applications", *In: Proc. 4th Intl. Conf. on Machine Learning and Applications*, pp. 351–356 (2005).