Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005 (pp152-157)

Ensembles of Multilayer Feedforward: A New Comparison

JOAQUÍN TORRES-SOSPEDRA
CARLOS HERNÁNDEZ-ESPINOSA
MERCEDES FERNÁNDEZ-REDONDO
Dept. de Ingeniería y Ciencia de los Computadores
Universidad Jaume I
Avda. Vicente Sos Baynat s/n 12071 Castellón
SPAIN

Abstract: - As shown in the bibliography, training an ensemble of networks is an interesting way to improve the performance with respect to a single network. However there are several methods to construct the ensemble. In this paper we present some new results in a comparison of twenty different methods. We have trained ensembles of 3, 9, 20 and 40 networks to show results in a wide spectrum of values. The results show that the improvement in performance above 9 networks in the ensemble depends on the method but it is usually low. Also, the best method for a ensemble of 3 networks is called "Decorrelated" and uses a penalty term in the usual Backpropagation function to decorrelate the network outputs in the ensemble. For the case of 9 and 20 networks the best method is conservative boosting. And finally for 40 networks the best method is Cels.

Key-Words: - Neural Networks, Ensembles, Multilayer Feedforward, Combination methods.

1 Introduction

The most important property of a neural network (NN) is the generalization capability. The ability to correctly respond to inputs which were not used in the training set.

One technique to increase the generalization capability with respect to a single NN consist on training an ensemble of NN, i.e., to train a set of NNs with different weight initialization or properties and combine the outputs of the different networks in a suitable manner to give a single output.

It is clear from the bibliography that this procedure in general increases the generalization capability [1,2].

The two key factors to design an ensemble are how to train the individual networks and how to combine the different outputs to give a single output.

Among the methods of combining the outputs, the two most popular are *voting* and *output averaging* [3]. In this paper we will normally use *output averaging* because it has no problems of ties and gives a reasonable performance.

In the other aspect, nowadays, there are several different methods in the bibliography to train the individual networks and construct the ensemble [1-3].

However, there is a lack of comparison among the different methods and it is not clear which one can provide better results.

One comparison can be found in [4], it is a previous work developed by our research group. In paper [4], eleven different methods are compared. Now, we present more complete results by including nine new methods, so we increase the number of methods in the comparison to a total of twenty. The empirical results are quite interesting, one of the new methods analyzed in this paper seems to have the best performance in several situations.

2 Theory

In this section we briefly review the new nine ensemble methods introduced in this paper for comparison. The description of the rest of methods denoted by "Simple Ensemble", "Ola", "Evol", "Decorrelated", "Decorrelated2", "CVC", "Cels", "Boosting", "Bag_Noise" and "Adaboost", can be found in reference [4] and in the references cited there.

CVC version 2: In the usual CVC the available data is divided in training, cross-validation and testing. After that, the data for training is divided by the number of networks giving several subsets. Then, one different subset is omitted for each network and the network is trained with the rest of subsets.

The version 2 of CVC included in this paper is used in reference [5]. The data for training and cross-validation is jointed in one set and with this jointed set the usual division of CVC is performed. In this case, one subset is omitted for each network and the omitted subset is used for cross-validation.

Aveboost: Aveboost is the abbreviation of Average Boosting. This method was propossed in reference [6] as a variation of Adaboost. In Adaboost, it is calculated a probability for each pattern of being included in the training set for the following network. In this case a weighted adaptation of the probabilities is performed. The method is complex and a full description can be found in the reference.

TCA, Total Corretive Adaboost: It was also proposed in reference [6] and it is another variation of Adaboost. In this case the calculation of the probability distribution for each network is treated as an optimization problem and an iterative process is performed. The algorithm is complex and a full description is in [6].

Aggressive Boosting: Aggressive Boosting is a variation of Adaboost. It is reviewed in [7]. In this case it is used a common step to modify the probabilities of a pattern, this common step can increase or decrease the probability of the pattern for being included in the next training set.

Conservative Boosting: It is another variation of Adaboost reviewed in [7]. In this case the probability of the well classified patterns is decreased and the probability of wrong classified patterns is kept unchanged.

ArcX4: It is another variation of Boosting, it was proposed and studied in reference [8]. The method selects training patterns according to a distribution, and the probability of the pattern depends on the number of times the pattern was not correctly classified by the previous networks. The combination procedure proposed in the reference is the mean average of the different networks. In our experiments we have used this procedure and also voting.

EENCL Evolutionary Ensemble with Negative Correlation: This method is proposed in reference [9]. The ensemble is build as a population of a genetic algorithm, the fitness function is selected to consider the precision in the classification of the individual networks and also to penalize the correlation among the different networks in the ensemble. The details can be found in the reference. In this paper two variations of the method proposed in the reference are used. They are denoted by EENCL UG (the ensemble used in this case is given by the last generation of the evolutionary algorithm) and EENCL MG (in this case the best generation is selected).

3 Experimental Results

We have applied the twenty ensemble methods to ten different classification problems. They are from the UCI repository of machine learning databases. Their names are Cardiac Arrhythmia Database (Aritm), Dermatology Database (Derma), Protein Location Sites (Ecoli), Solar Flares Database (Flare), Image Segmentation Database (Image), Johns Hopkins University Ionosphere Database (Ionos), Pima Indians Diabetes (Pima), Haberman's survival data (Survi), Vowel Recognition (Vowel) and Wisconsin Breast Cancer Database (Wdbc).

We have constructed ensembles of a wide number of networks, in particular 3, 9, 20 and 40 networks in the ensemble. In this case, we can test the results in a wide set of situations.

We trained the ensembles of 3, 9, 20 and 40 networks. We repeated this process of training an ensemble ten times for different partitions of data in training, cross-validation and test sets. With this procedure we can obtain a mean performance of the ensemble for each database (the mean of the ten ensembles) and an error in the performance calculated by standard error theory. The results of the performance are in table 1 for the case of ensembles of three networks, in table 2 for the case of nine and in table 3 for the case of 20. We omit the results of 40 networks by the lack of space and because the improvement of increasing the number of networks is in general low, a resume of the performance for 40 networks can be found below.

By comparing the results of table 1, and 2 with the results of a single network we can see that there the improvement by the use of the ensemble methods depends clearly on the problem.

Table 1. Results for the ensemble of three networks.

	ARITM	**DERMA**	**ECOLI**	**FLARE**
Single Net.	75.6±0.7	96.7± 0.4	84.4±0.7	82.1±0.3
Adaboost	71.8±1.8	98.0± 0.5	85.9±1.2	81.7±0.6
Bagging	74.7±1.6	97.5± 0.6	86.3±1.1	81.9±0.6
Bag_Noise	75.5±1.1	97.6± 0.7	87.5±1.0	82.2±0.4
Boosting	74.4±1.2	97.3± 0.6	86.8±0.6	81.7±0.4
Cels_m	73.4±1.3	97.7± 0.6	86.2±0.8	81.2±0.5
CVC	74.0±1.0	97.3± 0.7	86.8±0.8	82.7±0.5
Decorrelated	74.9±1.3	97.2± 0.7	86.6±0.6	81.7±0.4
Decorrelated2	73.9±1.0	97.6± 0.7	87.2±0.9	81.6±0.4
Evol	65.4±1.4	57± 5	57± 5	80.7±0.7
Ola	74.7±1.4	91.4± 1.5	82.4±1.4	81.1±0.4
CVC version 2	76.1±1.6	98.0± 0.3	86.8±0.9	82.5±0.6
AveBoost	73.4±1.3	97.6± 0.7	85.3±1.0	81.8±0.8
TCA	70.7±1.9	96.1± 0.6	85.4±1.3	81.9±0.7
ArcX4	75.4±0.8	97.8± 0.5	85.3±1.1	78.3±0.9
ArcX4 Voting	73.0±0.8	97.0± 0.5	85.7±1.1	80.6±0.9
Aggressive B	72.3±1.9	97.0± 0.5	85.7±1.4	81.9±0.9
Conservative B	74.8±1.3	96.9± 0.8	85.4±1.3	82.1±1.0
EENCL UG	71±2	96.8± 0.9	86.6±1.2	81.4±0.8
EENCL MG	74.5±1.3	97.2± 0.8	86.6±1.2	81.9±0.5
Simple Ens.	73.4±1.0	97.2± 0.7	86.6±0.8	81.8±0.5

Table 1. (continuation 1) Results for the ensemble of three networks.

	IMAGEN	IONOS	PIMA
Single Net.	96.3±0.2	87.9±0.7	76.7±0.6
Adaboost	96.8±0.2	88.3±1.3	75.7±1.0
Bagging	96.6±0.3	90.7±0.9	76.9±0.8
Bag_Noise	93.4±0.4	92.4±0.9	76.2±1.0
Boosting	95.0±0.4	88.9±1.4	75.7±0.7
Cels_m	96.8±0.2	91.9±1.0	76.0±1.4
CVC	96.4±0.2	87.7±1.3	76.0±1.1
Decorrelated	96.7±0.3	90.9±0.9	76.4±1.2
Decorrelated2	96.7±0.3	90.6±1.0	75.7±1.1
Evol	77±5	83.4±1.9	66.3±1.2
Ola	95.6±0.3	90.7±1.4	69.2±1.6
CVC version 2	96.9±0.3	89.7±1.4	76.8±1.0
AveBoost	96.8±0.2	89.4±1.3	76.5±1.1
TCA	94.8±0.5	87.9±1.2	75.4±0.8
ArcX4	96.6±0.2	89.4±1.0	76.0±0.8
ArcX4 Voting	96.5±0.2	89.0±1.0	76.3±0.8
Aggressive B	96.6±0.3	90.3±0.9	74.3±1.5
Conservative B	96.5±0.3	89.4±1.0	75.6±1.2
EENCL UG	96.3±0.2	93.0±1.0	74.7±1.0
EENCL MG	96.0±0.2	93.7±0.9	75.3±1.0
Simple Ens.	96.5±0.2	91.1±1.1	75.9±1.2

Table 1. (continuation 2) Results for the ensemble of three networks.

	SURVI	VOWEL	WDBC
Single Net.	74.2± 0.8	83.4±0.6	97.4±0.3
Adaboost	75.4±1.6	88.4± 0.9	95.7±0.6
Bagging	74.2±1.1	87.4± 0.7	96.9±0.4
Bag_Noise	74.6±0.7	84.4±1.0	96.3±0.6
Boosting	74.1±1.0	85.7±0.7	97.0±0.4
Cels_m	73.4±1.3	91.1±0.7	97.0±0.4
CVC	74.1±1.4	89.0±1.0	97.4±0.3
Decorrelated	74.6±1.5	91.5±0.6	97.0±0.5
Decorrelated2	74.3±1.4	90.3±0.4	97.0±0.5
Evol	74.3±0.6	77.5±1.7	94.4±0.9
Ola	75.2±0.9	83.2±1.1	94.2±0.7
CVC version 2	74.1±1.2	89.8±0.9	96.7±0.3
AveBoost	75.1±1.2	88.1±1.0	95.6±0.5
TCA	73.0±1.5	87.5±1.1	91±4
ArcX4	68±2	90.8±0.9	96.3±0.6
ArcX4 Voting	74±2	86.2±0.9	96.1±0.6
Aggressive B	73.8±1.5	86.9±1.2	96.6±0.6
ConservativeB	75.6±1.1	88.8±1.1	97.0±0.6
EENCL UG	73.9±1.2	87.2±0.8	96.2±0.4
EENCL MG	73.9±0.8	87.4±0.7	96.4±0.5
Simple Ens.	74.3±1.3	88.0±0.9	96.9±0.5

For example in databases Aritm (except for the case of CVC version 2), Flare, Pima and Wdbc there is not a clear improvement.

In the rest of databases there is an improvement, perhaps the most important one is in database Vowel.

There is, however, one exception in the performance of the method Evol. This method did not work well in our experiments. In the original reference the method was tested in the database Heart. The results for a single network were 60%, for a simple ensemble 61.42% and for Evol 67.14%. We have performed some experiments with this database and our results for a simple network are 82.0 ± 0.9, clearly different.

Now, we can compare the results of tables 1 and 2 for ensembles of different number of networks. We can see that the results are in general similar and the improvement of training an increasing number of networks, for example 20 and 40, is in general low.

Table 2. Results for the ensemble of nine networks.

	ARITM	DERMA	ECOLI	FLARE
Adaboost	73.2±1.6	97.3±0.5	84.7±1.4	81.1±0.7
Bagging	75.9±1.7	97.7±0.6	87.2±1.0	82.4±0.6
Bag_Noise	75.4±1.2	97.0±0.7	87.2±0.8	82.4±0.5
Cels_m	74.8±1.3	97.3±0.6	86.2±0.8	81.7±0.4
CVC	74.8±1.3	97.6±0.6	87.1±1.0	81.9±0.6
Decorrelated	76.1±1.0	97.6±0.7	87.2±0.7	81.6±0.6
Decorrelated2	73.9±1.1	97.6±0.7	87.8±0.7	81.7±0.4
Evol	65.9±1.9	54±6	57±5	80.6±0.8
Ola	72.5±1.0	86.7±1.7	83.5±1.3	80.8±0.4
CVC version2	76.1±1.6	98.0±0.3	86.8±0.9	82.5±0.6
AveBoost	73.4±1.3	97.6±0.7	85.3±1.0	81.8±0.8
TCA	70.7±1.9	96.1±0.5	85.4±1.3	81.9±0.7
ArcX4	75.4±0.8	97.8±0.5	85.3±1.1	78.3±0.9
ArcX4 Voting	73.3±0.8	97.6±0.5	84.9±1.1	80.1±0.9
Aggressive B	72.3±1.9	97.0±0.5	85.7±1.4	81.9±0.9
ConservativeB	74.8±1.3	96.9±0.8	85.4±1.3	82.1±1.0
EENCL UG	71±2	96.8±0.9	86.6±1.2	81.4±0.8
EENCL MG	74.5±1.3	97.2±0.8	86.6±1.2	81.9±0.5
Simple Ens.	73.8±1.1	97.5±0.7	86.9±0.8	81.6±0.4

Table 2. (continuation 1) Results for the ensemble of nine networks.

	IMAGEN	IONOS	PIMA
Adaboost	97.3±0.3	89.4±0.8	75.5±0.9
Bagging	96.7±0.3	90.1±1.1	76.6±0.9
Bag_Noise	93.4±0.3	93.3±0.6	75.9±0.9
Cels_m	96.6±0.2	91.9±1.0	75.9±1.4
CVC	96.6±0.2	89.6±1.2	76.9±1.1
Decorrelated	96.9±0.2	90.7±1.0	76.0±1.1
Decorrelated2	96.8±0.2	90.4±1.0	76.0±1.0
Evol	67±4	77±3	66.1±0.7
Ola	96.1±0.2	90.9±1.7	73.8±0.8
CVC version 2	96.9±0.3	89.7±1.4	76.8±1.0
AveBoost	96.8±0.2	89.4±1.3	76.5±1.1
TCA	94.8±0.5	87.9±1.2	75.4±0.8
ArcX4	96.6±0.2	89.4±1.0	76.0±0.8
ArcX4 Voting	97.2±0.2	91.3±1.0	76.3±0.8
Aggressive B	96.6±0.3	90.3±0.9	74.3±1.5
Conservative B	96.5±0.3	89.4±1.0	75.6±1.2
EENCL UG	96.3±0.2	93.0±1.0	74.7±1.0
EENCL MG	96.0±0.2	93.7±0.9	75.3±1.0
Simple Ens.	96.7±0.3	90.3±1.1	75.9±1.2

Table 2. (continuation 2) Results for the ensemble of nine networks.

	SURVI	VOWEL	WDBC
Adaboost	74.3±1.4	94.8±0.7	95.7±0.7
Bagging	74.4±1.5	90.8±0.7	97.3±0.4
Bag_Noise	74.8±0.7	85.7±0.9	95.9±0.5
Cels_m	73.4±1.2	92.7±0.7	96.8±0.5
CVC	75.2±1.5	90.9±0.7	96.5±0.5
Decorrelated	73.9±1.3	92.8±0.7	97.0±0.5
Decorrelated2	73.8±1.3	92.6±0.5	97.0±0.5
Evol	74.8±0.7	61±4	87.2±1.6
Ola	74.8±0.8	88.1±0.8	95.5±0.6
CVC version 2	74.1±1.2	89.8±0.9	96.7±0.3
AveBoost	75.1±1.2	88.1±1.0	95.6±0.5
TCA	73.0±1.5	87.5±1.1	91±4
ArcX4	68±2	90.8±0.9	96.3±0.6
ArcX4 Voting	73.9±1.0	94.6±0.9	96.6±0.6
Aggressive B	73.8±1.5	86.9±1.2	96.6±0.6
ConservativeB	75.6±1.1	88.8±1.1	97.0±0.6
EENCL UG	73.9±1.2	87.2±0.8	96.2±0.4
EENCL MG	73.9±0.8	87.4±0.7	96.4±0.5
Simple Ens.	74.2±1.3	91.0±0.5	96.9±0.5

Taking into account the computational cost, we can say that the best alternative for an application is an ensemble of three or nine networks.

We have also calculated the percentage of error reduction of the ensemble with respect to a single network. We have used equation 1 for this calculation.

$$PorError_{reduction} = 100 \cdot \frac{PorError_{\text{single network}} - PorError_{ensemble}}{PorError_{\text{single network}}} \quad \textbf{(1)}$$

Table 3. Results for the ensemble of twenty networks.

	ARITM	DERMA	ECOLI	FLARE
Adaboost	71.4±1.5	97.5±0.6	86.0±1.3	81.1±0.8
Bagging	75.9±1.7	97.6±0.6	87.1±1.0	82.2±0.5
Bag_Noise	76.0±1.1	97.3±0.6	87.4±0.8	82.1±0.5
Cels_m	75.4±1.2	93.9±1.4	86.3±1.3	81.5±0.4
CVC	74.8±1.3	97.3±0.6	86.5±1.0	81.7±0.7
Decorrelated	76.1±1.1	97.6±0.7	87.1±0.7	81.3±0.5
Decorrelated2	73.9±1.1	97.6±0.7	88.1±0.7	81.6±0.5
Evol	65.9±1.9	47±5	55±4	81.2±0.5
Ola	72.5±1.1	87.0±1.4	84.3±1.2	80.7±0.4
CVC version2	74.3±1.2	97.5±0.6	86.6±1.1	81.8±0.4
AveBoost	75.5±1.1	97.9±0.5	86.2±1.2	82.4±0.7
TCA	71.6±1.8	92±2	85.4±1.5	79.7±0.9
ArcX4	74.4±1.4	97.8±0.6	85.6±0.8	78.4±1.4
ArcX4 Voting	75.1±1.4	97.3±0.6	86.0±0.8	78.6±1.4
Aggressive B	74.8±1.5	97.0±0.6	87.1±1.1	82.0±0.5
Conservative	74.7±0.9	97.9±0.6	86.9±1.2	82.8±0.6
EENCL UG	72.9±0.9	95.1±1.1	87.2±0.7	82.0±0.8
EENCL MG	73.5±1.6	96.2±0.9	87.7±1.0	81.4±0.6
Simple Ens.	73.8±1.1	97.3±0.7	86.9±0.8	81.5±0.5

Table 3. (continuation 1) Results for the ensemble of twenty networks.

	IMAGEN	IONOS	PIMA
Adaboost	97.29±0.19	91.4±0.8	74.8±1.0
Bagging	97.0±0.3	89.6±1.1	77.0±1.0
Bag_Noise	93.3±0.3	92.7±0.6	76.3±0.8
Cels_m	95.74±0.19	93.3±0.7	75.4±1.0
CVC	96.8±0.2	89.6±1.3	76.2±1.3
Decorrelated	96.9±0.2	91.1±0.9	76.1±1.0
Decorrelated2	96.8±0.2	90.9±0.9	76.1±1.0
Evol	63±5	66.1±1.2	65.2±0.9
Ola	96.4±0.2	69.4±1.2	74.2±1.1
CVC version 2	97.03±0.17	91.0±0.9	76.7±0.8
AveBoost	97.3±0.3	91.4±1.0	76.0±1.1
TCA	95.7±0.3	86.1±1.0	73.5±0.9
ArcX4	97.38±0.19	92.0±0.9	72.7±1.1
ArcX4 Voting	97.3±0.2	92.6±0.9	75.0±1.1
Aggressive B	97.2±0.3	91.6±0.9	75.5±1.3
Conservative B	97.2±0.3	92.4±1.0	76.7±1.2
EENCL UG	96.9±0.3	92.3±1.1	75.2±0.8
EENCL MG	96.6±0.3	92.3±1.0	76.2±1.3
Simple Ens.	96.7±0.2	90.4±1.0	75.9±1.2

Table 3. (continuation 2) Results for the ensemble of twenty networks.

	SURVI	VOWEL	WDBC
Adaboost	74.3±1.5	96.1±0.7	96.3±0.5
Bagging	74.6±1.7	91.3±0.6	97.5±0.4
Bag_Noise	74.6±0.7	86.7±0.7	96.1±0.5
Cels_m	64±3	87.5±0.8	96.5±0.5
CVC	73.8±0.9	91.9±0.5	97.4±0.4
Decorrelated	74.1±1.4	93.3±0.6	97.0±0.5
Decorrelated2	74.3±1.3	93.3±0.5	97.0±0.5
Evol	74.8±0.7	60±3	78±3
Ola	74.1±0.7	88.7±0.8	95.3±0.6
CVC version 2	73.6±1.0	93.3±0.6	95.9±0.6
AveBoost	74.8±1.2	95.8±0.6	95.8±0.6
TCA	71.3±1.8	85±3	94.4±0.7
ArcX4	69±2	96.6±0.5	96.4±0.6
ArcX4 Voting	73.8±1.9	96.1±0.5	96.6±0.6
Aggressive B	73.9±1.7	96.9±0.6	96.8±0.6
ConservativeB	72.8±1.3	96.6±0.6	96.4±0.6
EENCL UG	72.5±1.5	88.2±0.9	95.8±0.4
EENCL MG	74.1±1.0	88.3±0.9	96.5±0.4
Simple Ens.	74.3±1.3	91.4±0.8	96.9±0.5

The value of the percentage of error reduction ranges from 0%, where there is no improvement by the use of a particular ensemble method with respect to a single network, to 100%. There can also be negative values, which means that the performance of the ensemble is worse than the performance of the single network.

This new measurement is relative and can be used to compare more clearly the different methods. Furthermore we can calculate the mean performance of error reduction across all databases this value is in table 4 for ensembles of 3, 9, 20 and 40 networks.

Table 4. Mean percentage of error reduction for the different ensembles.

	Ensambles of			
	3 Nets	**9 Nets**	**20 Nets**	**40 Nets**
Adaboost	1.33	4.26	9.38	12.21
Bagging	6.86	12.12	13.36	12.63
Bag_Noise	-3.08	-5.08	-3.26	-3.05
Boosting	-0.67	--	--	--
Cels_m	9.98	9.18	10.86	14.43
CVC	6.18	7.76	10.12	6.48
Decorrelated	9.34	12.09	12.61	12.35
Decorrelated2	9.09	11.06	12.16	12.10
Evol	-218.23	-297.01	-375.36	-404.81
Ola	-33.11	-36.43	-52.53	-47.39
CVC version 2	10.25	10.02	7.57	7.49
AveBoost	1.13	10.46	9.38	10.79
TCA	-9.71	-25.22	-43.98	-53.65
ArcX4	1.21	2.85	7.85	10.05
ArcX4 Voting	-2.08	9.73	10.76	11.14
Aggressive B	1.22	7.34	13.03	13.54
Conservative	4.45	13.07	14.8	14.11
EENCL UG	0.21	-3.23	-3.59	1.10
EENCL MG	3.96	1.52	2.84	7.89
Simple Ens	5.89	8.39	8.09	9.72

According to this global measurement *Ola*, *Evol* and *BagNoise* performs worse than the *Simple Ensemble*. The best methods are *Bagging*, *Cels*, *Decorrelated*, *Decorrelated2* and *Conservative Boosting*. In total, there are around ten methods which perform better than the *Simple Ensemble*.

The best method for 3 networks in the ensemble is *Decorrelated*, the best method for the case of 9 and 20 networks is *Conservative Boosting* and the best method for the case of 40 networks is *Cels* but the performance of *Conservative Boosting* is also good.

So, we can conclude that if the number of networks is low it seems that the best method is *Decorrelated* and if the number of network is high the best method is in general *Conservative Boosting*.

Also in table 4, we can see the effect of increasing the number of networks in the ensemble. There are several methods (*Adaboost*, *Cels*, *ArcX4*, *ArcX4 Voting*, *Aggressive Boosting* and *Conservative Boosting*) where the performance seems to increase slightly with the number of networks in the ensemble. But other methods like *Bagging*, *CVC*, *Decorrelated*, *Decorrelated2* and *Simple Ensemble* does not increase the performance beyond 9 or 20 networks in the ensemble. The reason can be that the new networks are correlated to the first ones or that the combination method (the average) does not exploit well the increase in the number of networks.

4 Conclusion

In this paper we have presented experimental results of twenty different methods to construct an ensemble of networks, using ten different databases. We trained ensembles of 3, 9, 20 and 40 networks in the ensemble to cover a wide spectrum of number of networks in the ensemble. The results showed that in general the improvement by the use of the ensemble methods depends clearly on the database, in some databases there is an improvement but in other there is not improvement at all. Also the improvement in performance from three or nine networks in the ensemble to a higher number of networks depends on the method. Taking into account the computational cost, an ensemble of nine networks may be the best alternative for most of the methods. Finally, we have obtained the mean percentage of error reduction over all databases. According to the results of this measurement the best methods are *Bagging*, *Cels*, *Decorrelated*, *Decorrelated2* and *Conservative Boosting*. In total, there are around ten methods which perform better than the *Simple Ensemble*. The best method for 3 networks in the ensemble is *Decorrelated*, the best method for the case of 9 and 20 networks is *Conservative Boosting* and the best method for the case of 40 networks is *Cels* but the performance of *Conservative Boosting* is also good. So we can conclude that if the number of networks is low it seems that the best method is *Decorrelated* and if the number of network is high the best method is in general *Conservative Boosting*.

Acknowledgments

This research was supported by the project MAPACI TIC2002-02273 of CICYT in Spain.

References:

[1] Tumer, K., Ghosh, J., “Error correlation and error reduction in ensemble classifiers”, *Connection Science*, Vol. 8, Nos. 3 & 4, 1996, pp. 385-404.

[2] Raviv, Y., Intrator, N., “Bootstrapping with Noise: An Effective Regularization Technique”, *Connection Science*, Vol. 8, No. 3 & 4, 1996, pp. 355-372.

[3] Drucker, H., Cortes, C., Jackel, D., et alt., “Boosting and Other Ensemble Methods”, *Neural Computation*, Vol. 6, 1994, pp. 1289-1301.

[4] Fernandez-Redondo, Mercedes, Hernández-Espinosa, Carlos, Torres-Sospedra, Joaquín, "Classification by Multilayer Feedforward ensambles", *Internacional Symposium on Neural Networks, Lecture Notes in Computer Science*, Vol. 3173, 2004, pp. 852-857.

[5] Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., Gelzinis, A., "Soft Combination of neural classifiers: A comparative study", *Pattern Recognition Letters*, Vol. 20, 1999, pp 429-444.

[6] Oza, N.C., "Boosting with Averaged Weight Vectors", *Multiple Classifier Systems, Lecture Notes in Computer Science*, Vol. 2709, 2003, pp. 15-24.

[7] Kuncheva, L.I., "Error Bounds for Aggressive and Conservative Adaboost", *Multiple Classifier Systems, Lecture Notes in Computer Science*, Vol. 2709, 2003, pp. 25-34.

[8] Breiman, L., "Arcing Classifiers", *Annals of Statistic*, Vol. 26, No. 3, 1998, pp. 801-849.

[9] Liu, Y., Yao, X., Higuchi, T., "Evolutionary Ensembles with Negative Correlation Learning", *IEEE Trans. On Evolutionary Computation*, Vol. 4, No. 4, 2000, pp. 380-387.