Performance Analysis of Classifier Ensembles: Neural Networks *Versus* Nearest Neighbor Rule

R.M. Valdovinos[1] and J.S. Sánchez[2]

[1] Lab. Reconocimiento de Patrones, Instituto Tecnológico de Toluca
Av. Tecnológico s/n, 52140 Metepec (México)
li_rmvr@hotmail.com
[2] Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I
Av. Sos Baynat s/n, E-12071 Castelló de la Plana (Spain)
sanchez@uji.es

Abstract. We here compare the performance (predictive accuracy and processing time) of different neural network ensembles with that of nearest neighbor classifier ensembles. Concerning the connectionist models, the multilayer perceptron and the modular neural network are employed. Experiments on several real-problem data sets demonstrate a certain superiority of the nearest-neighbor-based schemes, in terms of both accuracy and computing time. When comparing the neural network ensembles, one can observe a better behavior of the multilayer perceptron than that of the modular networks.

1 Introduction

The combination of classifiers (ensemble of classifiers) is now a well-established research line in Pattern Recognition and other related areas. The fundamental idea presents a number of advantages when compared to the use of individual classifiers [10, 15]: the correlated errors of the individual components can be eliminated when considering the total of the decisions, the training patterns cannot often provide enough information to select the best classifier and finally, the individual space search may not contain the objective function.

Let $\mathcal{D} = \{D_1, \ldots, D_h\}$ be a set of h classifiers. Each classifier D_i $(i = 1, \ldots, h)$ gets as input a feature vector $x \in \Re^n$, and assigns it to one of the c problem classes. The output of an ensemble of classifiers is an h-dimensional vector $[D_1(\mathbf{x}), \ldots, D_h(\mathbf{x})]^{\mathrm{T}}$ containing the decisions of each of the h individual classifiers. For combining the individual decisions, the most popular (and simplest) method is the majority voting rule, although there exist other more complex schemes (e.g., average, minority, medium, product of votes) [15, 2].

To ensure the success of the combining systems, two fundamental conditions are considered: the ratio of error in the individual components and the level of diversity in the individual decisions [10]. Two classifiers are said to be diverse if their decisions are different when classifying a same input pattern, that is, if the individual classifiers do not always agree. No benefit arises from combining the predictions of a set of classifiers that frequently coincide in the classifications (strongly correlated classifiers). Although measuring diversity is not straightforward [16], this diversity in combining classifiers

J. Martí et al. (Eds.): IbPRIA 2007, Part I, LNCS 4477, pp. 105–112, 2007.

has been sought through different ways: by manipulating the training patterns (training each classifier on different subsets of the training sample) [7], by using different decision rules, or by incorporating random noise into the feature values or into some parameters of the learning model considered [18], among others.

In the present paper, the performance (in terms of overall predictive accuracy and processing time) of nearest neighbor classifier ensembles and artificial neural network ensembles are comparatively analyzed. This type of evaluation has several precedents in the literature, although all they pursue different goals. For example, Brown and Wyatt [8] carried out a detailed analysis of neural network ensembles and the impact of negative correlated learning on classification performance.

For the neural networks, we here focus on two different models: the multilayer feedforward perceptron and the modular neural network. Diversity and independence are achieved by using two resampling methods: the well-known Bagging scheme [7] and the random selection without replacement [3]. Both are applied by performing a way of class-dependant (or stratified) resampling [20], that is, resampling is done separately over the training instances of each class, thus obtaining the same class distribution in each subsample as that of the original training set.

2 Theoretical Background of the Ensembles

This section provides the main characteristics of the ensembles that will be analyzed in the present work. We describe the particular topology of the neural networks and the basis of the nearest neighbor classifier.

2.1 The Multilayer Perceptron Ensemble

The Multilayer Perceptron (MP) neural network is one of the most popular connectionist models for classification purposes. It organizes the representation of knowledge in the hidden layers and has a very high power of generalization. The typical topology consists of three sequential layers: input, hidden and output [11, 13].

Currently, there does not exist empirical evidence on the optimal number of neurons in the hidden layer. For the ensemble of multilayer perceptrons proposed in this study, all the networks have n input units and c output units corresponding to the number of input features (or attributes) and data classes, respectively. Also, each network has only one hidden layer with two different structures, here called MP1 and MP2. In the former, the amount of neurons in the hidden layer is set to $(n + 1)$ [17] whereas in the latter, it is set to 2. The initial values of the connection weights are randomly picked out in the range $[-0.5, 0.5]$.

The networks will be here trained by using the backpropagation algorithm with a sigmoidal activation function for both the hidden and output layers. The backpropagation algorithm is simple and easy to compute. It often converges rapidly to a local minimum, but it may not find a global minimum and in some cases, it may not converge at all. To overcome this problem, a momentum term can be added to the minimization function, and a variable learning rate can be applied. In our experiments, the learning rate and momentum will be set to 0.9 and 0.7, respectively, whereas the number of training iterations will be 5000.

2.2 The Modular Neural Network Ensemble

The second neural model used in this paper is the modular neural network (also known as Mixture of Experts, ME) [14]. This is based on the human nervous system, in which each cerebral region has a specific function but at the same time, the regions are interconnected to each other. A modular network solves a complex computational task by dividing it into a number of simpler subtasks and then combining their individual solutions. Thus, a modular neural network consists of several *expert neural networks* (modules), where each expert is optimized to perform a particular task of an overall complex operation. An integrating unit, called *gating network*, is used to select or combine the outputs of the modules in order to form the final output of the network. In the more basic implementation, all the modules are of a same type [4, 12], but different schemes could be used.

There exist several implementations of the modular neural networks, although the main 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 [12]. Others are based on a network trained with a data set different from the one used for training the expert networks [4]. In this work, all modules (both the experts and the gating network) will be trained with a unique data set [14] (see Fig. 1).

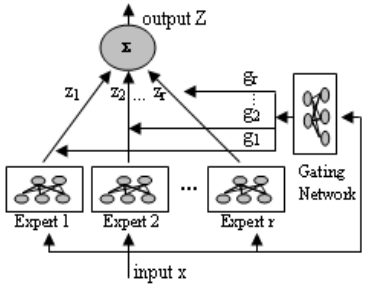

Fig. 1. Representation of the modular neural network architecture. Each module is a feedforward network and receives the same input vector. The final output of the system is the sum of $z_j g_j$.

All modules, including the gating network, have n input units, that is, the number of features. 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, say r. The learning process is based on the stochastic gradient algorithm, being the objective function defined as follows:

$$-\ln\left(\sum_{j=1}^{r} g_j \cdot \exp\left(-\frac{1}{2}\left|\left|s - z_j\right|\right|^2\right)\right) \tag{1}$$

where s is the output desired for input x, $z_j = xw_j$ is the output vector of the j'th expert network, $g_j = \frac{\exp(u_j)}{\sum_i \exp(u_j)}$ is the normalized output of the gating network, u_i is the total weighted input received by output unit j of the gating network, and g_j can be viewed as the probability of selecting expert j for a particular case.

The configuration of the ensembles used in this paper is as follows. Each individual component in the ensemble corresponds to a modular neural network, each one with the same structure: five expert networks and one gating network. As in the case of the multilayer perceptron, the decisions of the base classifiers in the ensemble will be finally combined by simple majority voting.

2.3 The Nearest Neighbor Classifier Ensemble

The Nearest Neighbor (NN) rule [9] constitutes one of the most studied learning algorithms in Pattern Recognition. Interest in this method is mainly due to its conceptual and implementational simplicity along with an asymptotic error rate conveniently bounded in terms of the optimal Bayes error. In its classical manifestation, given a set of N previously labelled instances (or training sample), this classifier assigns any input pattern to the class indicated by the label of the closest example in the training set.

Earlier reported results stimulated research into and applications of ensembles of classification models like neural networks and decision trees, while they discouraged the use of ensembles of NN classifiers. Experiments with Bagging did not show a difference in performance in the built ensemble as compared to the single NN classifier trained with the original learning set. These results led to the conclusion that the NN classifier is a very stable model: small changes in the training sample do not produce serious perturbations in the behavior of the classifier.

Nevertheless, in the last few years, several attempts to create ensembles of NN classifiers have been reported. Bay [5] searches to break down the stability by constructing different nearest neighbor individual classifiers, each learning with a randomly selected subset of features. Skalak [19] and Alpaydin [1] base the destabilization of the NN classifier upon the application of different procedures for reducing the training set size. This approach has the additional advantage of allowing a decrease in the computational burden inherent to the NN classifiers.

3 Resampling Methods

In this work, two different methods for designing the training sample are used: Bagging [6] and random selection without replacement [3]. These are here implemented by means of the class-dependent strategy [20], which consists of picking instances up in a way that the initial class distributions are preserved in each of the h subsamples generated. The rationale behind this strategy is to reduce the possibly high computational complexity (computing time and storage loads) of the base classifiers induced by each of the subsamples.

Briefly, in a class-dependent Bagging ensemble, each base classifier is trained on a set of N/h training examples randomly drawn, by replacement, from the original training set (of size N). Such a set is called a bootstrap replicate of the original training sample. By this technique, many examples may appear multiple times, while others may be left out. This negative effect can be overcome by using the random selection without replacement method, in which each example can be selected only once.

4 Experiments and Results

The results here reported correspond to the experiments over eight real-problem data sets taken from the UCI Machine Learning Database Repository (`http://www.ics.uci.edu/~mlearn`), whose main characteristics are summarized in Table 1. For each data set, 5-fold cross-validation was used to estimate the average predictive accuracy and processing time: 80% of the patterns for training and 20% for the test set.

Table 1. A brief summary of the UCI databases

	No. Classes	No. Features	No. Patterns
Cancer	2	9	685
Pima	2	8	770
Iris	3	4	150
Vehicle	4	18	848
Phoneme	2	5	5406
Waveform	3	21	5001
Segment	7	19	2212
Satimage	6	36	6437

In all experiments, the ensembles consist of nine individual classifiers. For each ensemble, the corresponding training samples have been designed by using the class-dependent Bagging and random selection without replacement techniques, as described in Sect. 3. Four different configurations for the base classifiers have been tested: NN, MP1, MP2, and ME. The results (predictive accuracy and processing time) for each single classifier (i.e., with no combination) has also been included as a baseline. The experiments have been carried out using a personal computer with Intel Centrino 1.3 GHz and 512 MB of RAM.

From results in Table 2, some initial comments can be drawn. Firstly, for all data sets there exists at least one ensemble whose classification accuracy is higher than that obtained when using the single classifier (training sample). On the other hand, while the NN ensembles show a favorable and uniform behavior on all data sets, accuracy of the neural network ensembles can strongly differ from one problem to another depending on each particular database. For example, the MP1 ensemble is better than the respective single classifier for the Phoneme database, but it clearly gives poorer results in the case of Vehicle and Satimage.

When comparing the overall predictive accuracies of the ensembles, in most cases the NN-based solutions provide better results than the neural networks. The most significant gains are in Iris, Vehicle, Phoneme, and Satimage; in these cases, the NN ensemble is about 12% superior to the best connectionist model (MP2). Concerning the neural network ensembles, one can observe that the multilayer perceptrons generally yield better results than the ME. Also, note that the behavior here discussed seems to be independent of the resampling method used, that is, both random selection without replacement and Bagging give very similar accuracy rates in all data sets.

Table 2. Overall classification accuracies (second row of each database corresponds to standard deviations)

	Single				Random w/o replacement				Bagging			
	NN	MP1	MP2	ME	NN	MP1	MP2	ME	NN	MP1	MP2	ME
Cancer	95.6	94.6	97.1	88.4	96.2	96.3	96.6	87.1	96.4	95.6	96.6	86.5
	0.025	0.023	0.021	0.038	0.029	0.025	0.021	0.047	0.028	0.034	0.017	0.042
Pima	65.9	71.1	75.2	66.5	72.8	72.2	72.7	66.1	72.7	72.2	73.1	67.8
	0.052	0.027	0.012	0.025	0.049	0.031	0.017	0.023	0.012	0.035	0.016	0.024
Iris	96.0	94.7	94.7	80.7	98.0	95.3	96.0	78.0	94.0	97.3	97.3	80.0
	0.015	0.022	0.028	0.076	0.018	0.038	0.037	0.077	0.044	0.015	0.028	0.067
Vehicle	64.2	57.5	46.7	36.4	61.4	44.1	51.3	42.2	60.6	47.3	43.0	43.5
	0.018	0.019	0.021	0.025	0.019	0.103	0.106	0.040	0.023	0.033	0.048	0.038
Phoneme	76.1	68.7	69.7	67.9	75.0	71.7	70.9	67.7	75.0	72.6	70.9	68.1
	0.080	0.066	0.031	0.036	0.099	0.067	0.025	0.042	0.094	0.057	0.038	0.042
Waveform	78.0	80.7	82.0	77.2	82.7	83.6	83.6	79.2	83.2	83.6	84.1	80.2
	0.029	0.015	0.020	0.028	0.018	0.017	0.008	0.038	0.014	0.010	0.014	0.033
Segment	94.8	94.1	73.3	78.2	87.6	93.8	80.1	76.9	87.7	93.5	81.4	74.5
	0.014	0.017	0.020	0.019	0.026	0.018	0.022	0.024	0.021	0.015	0.016	0.018
Satimage	83.6	70.4	70.4	34.9	82.9	42.9	70.5	58.9	82.9	44.1	71.9	48.9
	0.116	0.112	0.105	0.048	0.145	0.160	0.095	0.053	0.143	0.273	0.080	0.075

The Satimage database constitutes a very particular case, in which differences are especially important. The accuracy obtained by the MP2 ensemble is 28% higher than that given by MP1, whereas the average predictive accuracy of the ME ensemble is 24% higher than that of the single classifier. Analogously, the NN ensemble presents an accuracy 40% higher than MP1, 24% higher than ME, and about 12% higher than the MP2 ensemble.

Apart from accuracy, the time required for training a classification system is another important factor to take into account when analyzing the performance. This is particularly interesting in the case of neural network ensembles because for several models, training may suppose a very important time consuming process. Thus, a certain trade-off between predictive accuracy and processing time should be achieved to decide which classifier ensemble has to be used. Correspondingly, Table 3 reports the processing times, relative to training and classification phases.

The significant differences in processing time between the three ensembles and the single neural networks result evident (see Table 3). For example, in the case of the Iris data, time required by the single MP1 is one minute, whereas the corresponding ensembles (Bagging and random selection without replacement) need only 10 seconds for training and classification. Focusing on the neural network ensembles, it is clear enough that, as expected, the MP2 configuration is faster than MP1 and ME. In fact, remind that the MP1 structure consists of $(n+1)$ neurons in the hidden layer, while the MP2 scheme is formed by only two neurons. Similarly, each individual component in the ME ensemble corresponds to a modular neural network, each one consisting of five expert networks and one gating network. Obviously, these differences in the structures imply considerable differences in computational cost. In the case of NN, we did not

Table 3. Processing times for the different learning schemes (in minutes)

	Single				Random w/o replacement				Bagging			
	NN	MP1	MP2	ME	NN	MP1	MP2	ME	NN	MP1	MP2	ME
Cancer	0.1	4.5	5.2	11.3	0.1	0.4	0.3	9.5	0.1	0.3	0.4	9.2
Pima	0.1	8.5	3.6	9.8	0.9	5.7	4.1	9.6	0.1	6.8	3.9	9.8
Iris	0.0	1.0	0.4	2.3	0.0	0.1	0.1	1.8	0.0	0.1	0.1	1.8
Waveform	4.5	437.9	65.1	174.5	4.2	167.3	44.8	133.0	4.2	131.2	41.1	127.2
Vehicle	0.3	60.5	10.7	0.3	0.3	36.3	7.4	0.3	0.3	32.4	7.0	0.3
Satimage	8.3	134.4	134.5	593.2	10.4	913.9	88.3	420.6	10.5	829.4	82.4	409.0
Phoneme	1.7	60.5	21.3	66.8	1.0	39.4	22.6	57.4	1.0	41.9	23.7	59.0
Segment	1.1	210.9	35.0	1.4	1.6	81.8	24.4	2.1	1.7	63.1	23.0	2.1

find significant differences between times required by the ensembles and the single classifier. This is due to the resampling method described in Sect. 3, since the ensemble and the single classifier process the same amount of patterns.

5 Concluding Remarks and Further Extensions

Design of ensembles with NN classifiers, multilayer perceptrons and modular neural networks has been here analyzed. All ensembles consist of nine individual classifiers. In the case of perceptrons, the number of neurons in the hidden layer has been determined according to two criteria. In the first case, this has been set to $(n+1)$, while in the second configuration it has been set to 2. For the modular architecture, we have employed five expert networks and one gating network.

The experimental results allow to compare these ensemble models, in terms of processing time and predictive accuracy. From this, it has been possible to corroborate that in general, an ensemble of classifiers clearly outperforms the single classifier. Also, when comparing the four ensembles, it has been empirically demonstrated that the employment of a NN classifier results in the best performance: higher accuracy and lower computational cost.

Focusing on the results given by the three neural network ensembles, in most cases the MP2 configurations consume much less processing time than the other two schemes. It is due to the fact that the MP2 schemes are formed by only two neurons in the hidden layer, thus allowing considerable savings in computing time. Also, it is important to remind the classification results obtained with the three ensembles. From these results, it seems possible to conclude that in general, the MP2 ensemble provides a well-balanced trade-off between time requirements and predictive accuracy, becoming somewhat superior to MP1 and ME.

Future work is primarily addressed to improve the performance when using ensembles of neural networks. Within this context, other architectures, different parameters, and possible regularization/cross-validation mechanisms have to be analyzed in the future. 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 10007-2006-01 (ref. 51626) and SEP-2003-C02-44225 from the Mexican CONACYT, and DPI2006-15542 from the Spanish CICYT.

References

1. Alpaydin, E.: Voting over multiple condensed nearest neighbors. Artificial Intelligence Ressearch 11, 115–132 (1997)
2. Bahler, D., Navarro, L.: Methods for combining heterogeneous sets of classifiers, In: Proc. 17th Natl. Conf. on Artificial Intelligence, Workshop on New Research Problems for Machine Learning (2000)
3. Barandela, R., Valdovinos, R.M., Sánchez, J.S.: New applications of ensembles of classifiers. Pattern Analysis and Applications 6, 245–256 (2003)
4. Bauckhage, C., Thurau, C.: Towards a fair'n square aimbot — Using mixture of experts to learn context aware weapon handling, In: Proc. GAME-ON. Ghent, Belgium, pp. 20–24 (2004)
5. Bay, S.: Combining nearest neighbor classifiers through multiple feature subsets, In: Proc. 15th Intl. Conf. on Machine Learning. Madison, WI, pp. 37–45 (1998)
6. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
7. Breiman, L.: Arcing classifiers. Annals of Statistics 26, 801–823 (1998)
8. Brown, G., Wyatt, J.: Negative correlation learning and the ambiguity family of ensemble methods, In: Proc. Intl. Workshop on Multiple Classifier Systems. Guilford, UK, pp. 266–275 (2003)
9. Dasarathy, B.V. (ed.): Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamos, CA (1991)
10. Dietterich, G.T.: Machine learning research: four current directions. AI Magazine 18, 97–136 (1997)
11. Funahashi, K.: On the approximate realization of continuous mapping by neural networks. Neural Networks 2, 183–192 (1989)
12. Hartono, P., Hashimoto, S.: Ensemble of linear perceptrons with confidence level output, In: Proc. 4th Intl. Conf. on Hybrid Intelligent Systems. Kitakyushu, Japan, pp. 186–191 (2004)
13. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366 (1989)
14. Jacobs, R., Jordan, M., Hinton, G.: Adaptive mixture of local experts. Neural Computation 3, 79–87 (1991)
15. Kuncheva, L.I.: Using measures of similarity and inclusion for multiple classifier fusion by decision templates. Fuzzy Sets and Systems 122, 401–407 (2001)
16. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles. Machine Learning 51, 181–207 (2003)
17. Pao, Y.H.: Adaptive Pattern Recognition and Neural Networks. Addison-Wesley, Reading, MA, London, UK (1989)
18. Raviv, Y., Intrator, N.: Bootstrapping with noise: an effective regularization technique. Connection Science 8, 356–372 (1996)
19. Skalak, D.B.: Prototype Selection for Composite Nearest Neighbor Classification. Ph.D. Thesis, University of Massachusetts (1996)
20. Valdovinos, R.M., Sánchez, J.S.: Class-dependant resampling for medical applications, In: Proc. 4th Intl. Conf. on Machine Learning and Applications. Los Angeles, CA, pp. 351–356 (2005)