Cascade ensembles

N. García-Pedrajas[1], D. Ortiz-Boyer[1], R. del Castillo-Gomariz[1], and C. Hervás-Martínez[1]

University of Córdoba
Campus Universitario de Rabanales
Córdoba (Spain)
npedrajas, dortiz, rcastillo, chervas@uco.es

Summary. Neural network ensembles are widely use for classification and regression problems as an alternative to the use of isolated networks. In many applications, ensembles has proven a performance above the performance of just one network.

In this paper we present a new approach to neural network ensembles that we call "cascade ensembles". The approach is based on two ideas: (i) the ensemble is created constructively, and (ii) the output of each network is fed to the inputs of the subsequent networks. In this way we make a cascade of networks.

This method is compared with standard ensembles in several problems of classification with excellent performance.

1 Introduction

Neural network ensembles[11] are receiving increasing attention in recent neural network research, due to their interesting features. They are a powerful tool especially when facing complex problems. Network ensembles are usually made up of a linear combination of several networks that have been trained using the same data (see Figure 1, although the actual sample used by each network to learn can be different. Each network within the ensemble has a potentially different weight in the output of the ensemble. Several works have shown [11] that the network ensemble has a generalisation error generally smaller than that obtained with a single network and also that the variance of the ensemble is lesser than the variance of a single network. If the networks have more than one output, a different weight is usually assigned to each output. The ensembles of neural networks have some of the advantages of large networks without their problems of long training time and risk of over-fitting. For more detailed descriptions of ensembles the reader is referred to [2] [14] [3] [10] [5].

Although there is no clear distinction between the different kinds of multinet networks [7] [1] [6], we follow the distinction of [13]. In an ensemble several redundant approximations to the same function are combined by some

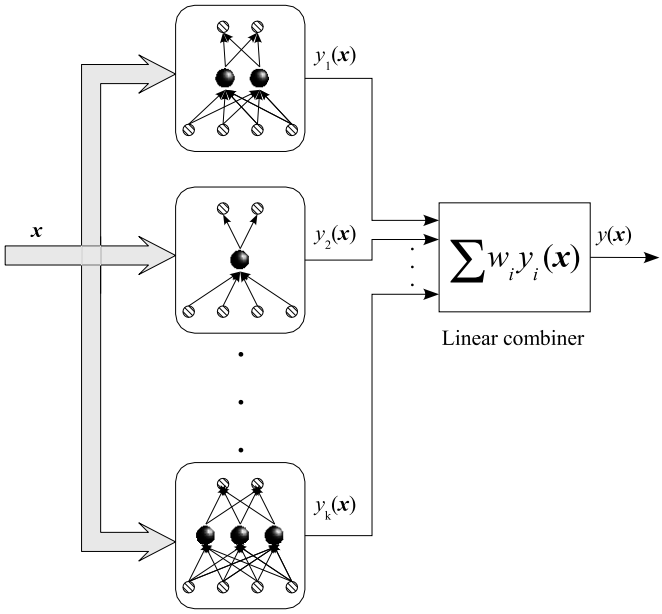

Fig. 1. Standard ensemble of neural networks

method, and in a modular system the task is decomposed into a number of simpler components.

This combination of several networks that cooperate in solving a given task has other important advantages such as [8] [13]:

- They can perform more complex tasks than any of their subcomponents [15].
- They can make an overall system easier to understand and modify, as the whole system is decomposed in smaller parts.
- They are more robust than a single network.

In most cases, neural networks in an ensemble are designed independently or sequentially, so the advantages of interaction and cooperation among the individual networks are not exploited. Earlier works separate the design and learning process of the individual networks from the combination of the trained networks. In this work we propose a framework for designing ensembles, where the training and combination of the individual networks are carried out together, in order to get more cooperative networks and more effective combinations of them.

The design of neural network ensembles implies making many decisions that have a major impact on the performance of the ensembles. The most important decisions that we must face when designing an ensemble are the following:

- The method for designing and training the individual networks.
- The method of combining the individual networks, and the mechanism for obtaining individual weights for each network if such is the case.
- The measures of performance of the individual networks.
- The methods for encouraging diversity among the members of the ensembles and how to measure such diversity.
- The method of selection of patterns that are used by each network to learn.
- Whether to include regularization terms and their form.

Techniques using multiple models usually consist of two independent phases: model generation and model combination[10]. The disadvantage of this approach is that the combination is not considered during the generation of the models. With this approach the possible interactions among the trained cannot be exploited until the combination stage [8], and the benefits that can be obtained from this interactions during the learning stage are lost.

However, several researchers[16][9] have recently shown that some information about cooperation is useful for obtaining better ensembles. This new approach opens a wide field where the design and training of the different networks must be interdependent.

In this paper, we present a new model for constructively making the ensemble. Our basic aim is improving the combination of networks. In this way we created an ensemble where the i-th network receives as inputs the outputs of the $i-1$ already trained networks. This ensemble is so-called *cascade ensemble*. In this way, some of the ideas of the constructive cascade-correlation networks are applied to ensemble construction [4].

This paper is organised as follows: Section 2 describes our model of cascade ensembles; Section 3 shows the experimental results of our model and its comparison with standard ensembles; finally Section 4 states the conclusions of our work.

2 Cascade ensembles

The main advantage of cascade ensembles is that each time a new network is added to the ensemble, that network knows the outputs of the previously trained networks. In this way, the construction of the ensemble is not separated in two stages: training and combination of the networks.

In the proposed model, each network tries to refine the classification carried out by the previous models. For a pattern $\mathbf{x}$ the input to k-th network is $(\mathbf{x}, y_1, y_2, \ldots, y_{k-1})$, where y_i is the output of network i (see Figure 2).

Cascade ensembles can also be defined in other ways that are under development. The main advantages of this approach are:

1. The ensemble is made constructively. In this way, the complexity of the ensemble can match the problem, as the addition of new networks can be stopped when the required performance is achieved.

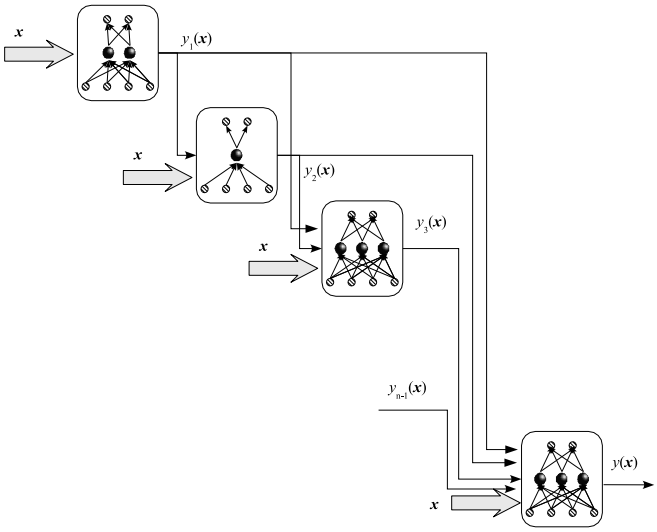

Fig. 2. Cascade ensemble of neural networks

2. The stages of training and combination are not separated. The new networks have the knowledge acquired by previous networks.
3. There is no need of an additional combination scheme and the subsequent optimisation algorithm needed to set its parameters.

3 Experiments

The experiments were carried out with the objective of testing our model against an standard ensemble. We have applied our model and standard ensemble method to several real-world problems of classification. These problems are briefly described in Table 1. These nine datasets cover a wide variety of problems. There are problems with different number of available patterns, different number of classes, different kind of inputs, and of different areas of application. Testing our model on this variety of problems can give us a clear idea of its performance.

The tests were conducted following the guidelines of L. Prechelt [12]. Each set of available data was divided into three subsets: 50% of the patterns were used for learning, 25% of them for validation and the remaining 25% for testing the generalization error. There is an exception, Sonar problem, as the patterns of this problem are prearranged in two subsets due to their specific features. The ensembles are made up by 10 networks, each one with 10 hidden nodes with logistic transfer function. The training parameters for the backpropagation algorithm are $\eta = 0.01$ and $\alpha = 0.01$.

The training and generalisation results are shown in Table 2. The table shows also the p-values of paired t-test for generalisation error. The results

Table 1. Summary of data sets. The features of each data set can be C(continuous), B(binary) or N(nominal). The Inputs column shows the number of inputs of the network as it depends not only on the number of input variables but also on their type.

Data set	Cases		Classes	Features			Inputs
	Train	Test		C	B	N	
Anneal	674	224	5	6	14	18	59
Glass	161	53	6	9	–	–	9
Heart	202	68	2	6	1	6	13
Hepatitis	117	38	2	6	13	–	19
Horse	273	91	3	13	2	5	58
Pima	576	192	2	8	-	-	8
Sonar	104	104	2	60	–	–	60
Promoters	80	26	2	–	–	57	114
Vehicle	635	211	4	18	–	–	18

show that the cascade algorithm performs better than standard ensembles in 7 problems (in three of them with statistical significance). This is interesting as the cascade models avoids the use any method for combining the outputs.

4 Conclusions and future work

In this paper we have introduced a new approach to neural network ensembles called *cascade ensembles* where the ensemble is created constructively. The preliminary results showed in this paper are very promising and open a wide field of study for this approach.

As work for the future, we are working in two ideas. Firstly, the application of sampling methods, such as bagging and boosting, to our model. Secondly, the modification of the cascade inputs, as there different ways for feeding the outputs of the trained networks to the new networks.

Acknowlegments

This work has been financed in part by the project TIC2002-04036-C05-02 of the Spanish CICYT and FEDER funds.

References

1. R. Avnimelech and N. Intrator. Booested mixture of experts: an ensemble learning scheme. *Neural Computation*, 11(2):483–497, 1999.

Table 2. Results for the classification of the described problems for cascade and standard ensemble. For each problem we show the averaged training and test error, the standard deviation, and the best and worst result.

Problem	Model	Training				Generalisation				t-test
		Mean	SD	Best	Worst	Mean	SD	Best	Worst	
Anneal	Casc	0.0113	0.0140	0.0000	0.0460	0.0250	0.0122	0.0134	0.0580	–
	Std	0.0064	0.0056	0.0015	0.0179	0.0106	0.0064	0.0015	0.0223	0.0000
Glass	Casc	0.1201	0.0183	0.0870	0.1553	0.2692	0.0325	0.2075	0.3396	–
	Std	0.1660	0.0147	0.1429	0.1925	0.2943	0.0236	0.2453	0.3396	0.0011
Heart	Casc	0.1229	0.0102	0.0990	0.1436	0.1373	0.0156	0.1029	0.1618	–
	Std	0.0726	0.0056	0.0545	0.0792	0.1539	0.0120	0.1324	0.1912	0.0000
Hepatitis	Casc	0.0499	0.0129	0.0342	0.0855	0.1140	0.0222	0.0789	0.1842	–
	Std	0.0051	0.0043	0.0000	0.0085	0.1289	0.0174	0.1053	0.1579	0.0054
Horse	Casc	0.0223	0.0054	0.0110	0.0366	0.2934	0.0252	0.2527	0.3516	–
	Std	0.0289	0.0036	0.0220	0.0330	0.3018	0.0160	0.2637	0.3297	0.1274
Pima	Casc	0.2215	0.0086	0.2066	0.2500	0.2061	0.0196	0.1667	0.2552	–
	Std	0.2218	0.0048	0.2118	0.2326	0.2049	0.0137	0.1823	0.2396	0.7819
Promoters	Casc	0.0000	0.0000	0.0000	0.0000	0.1526	0.0189	0.1154	0.1923	–
	Std	0.0000	0.0000	0.0000	0.0000	0.1538	0.0000	0.1538	0.1538	0.7109
Sonar	Casc	0.0295	0.0182	0.0000	0.0577	0.1766	0.0137	0.1442	0.2115	–
	Std	0.0000	0.0000	0.0000	0.0000	0.1779	0.0090	0.1635	0.2019	0.6703
Vehicle	Casc	0.1546	0.0136	0.1276	0.1858	0.1919	0.0212	0.1517	0.2417	–
	Std	0.1796	0.0072	0.1669	0.2016	0.1962	0.0190	0.1611	0.2464	0.4157

2. T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000.
3. S. Dzeroski and B. Zenko. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54:255–273, 2004.
4. S. E. Fahlman and C. Lebiere. The cascade-correlation architecture. In D. S. Touretzky, editor, *Advances in Neural Information Systems 2*, pages 524–532, San Mateo, CA, 1990. Morgan Kauffman.
5. A. Fern and R. Givan. Online ensemble learning: An empirical study. *Machine Learning*, 53:71–109, 2003.
6. G. Giacinto and F. Roli. Dynamic classifier selection. In *Multiple Classifier Systems 2000*, volume 1857 of *Lecture Notes in Computer Science*, pages 177–189, 2000.
7. J. Hansen. Combining predictors: Comparison of five meta machine learning methods. *Information Science*, 119(1–2):91–105, 1999.
8. Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380–387, November 2000.
9. Y. Liu, X. Yao, Q. Zhao, and T. Higuchi. Evolving a cooperative population of neural networks by minimizing mutual information. In *Proc. of the 2001 IEEE Congress on Evolutionary Computation*, pages 384–389, Seoul, Korea, May 2001.
10. C. J. Merz. Using correspondence analysis to combine classifiers. *Machine Learning*, 36(1):33–58, July 1999.
11. M. P. Perrone and L. N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In R. J. Mammone, editor, *Neural Networks for*

Speech and Image Processing, pages 126–142. Chapman – Hall, 1993.

12. L. Prechelt. Proben1 – A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, Karlsruhe, Germany, September 1994.
13. A. J. C. Sharkey. On combining artificial neural nets. *Connection Science*, 8:299–313, 1996.
14. G. I. Webb. Multiboosting: A technique for combining boosting and wagging. *Machine Learning*, 40(2):159–196, August 2000.
15. X. Yao and Y. Liu. Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 28(3):417–425, June 1998.
16. Z-H. Zhou, J. Wu, and W. Tang. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137(1–2):239–253, May 2002.