Multistage Neural Network Ensembles

Shuang Yang[1], Antony Browne[1], and Philip D. Picton[2]

[1] School of Computing, Information Systems and Mathematics, London Guildhall University, London EC3N 1JY, UK
Tel: (+44) 0207 320 1705, Fax: (+44) 0207 320 1707
syang@lgu.ac.uk
[2] School of Technology and Design, University College Northampton, Northampton NN2 6JD, UK

Abstract. Neural network ensembles (some times referred to as committees or classifier ensembles) are effective techniques to improve the generalization of a neural network system. Combining a set of neural network classifiers whose error distributions are diverse can lead to generating more accurate results than any single network. Combination strategies commonly used in ensembles include simple averaging, weighted averaging, majority voting and ranking. However, each method has its limitations, dependent either on the application areas it is suited to, or due to its effectiveness. This paper proposes a new ensembles combination scheme called multistage neural network ensembles. Experimental investigations based on multistage neural network ensembles are presented, and the benefit of using this approach as an additional combination method in ensembles is demonstrated.

1 Introduction

Combining the outputs of diverse classifiers can lead to an improved result [1, 2]. Common ensemble combination strategies include simple averaging [8, 9], weighted averaging [10], majority voting [11], and ranking [12,13].

There are no unique criteria on the usage of all the above combination methods. The choice mainly depends on the nature of the application the ensemble is being used for, the size and quality of training data, or generated errors on different regions of the input space. One combination method applied on the ensemble of a regression problem may generate good results, but may not work on a classification problem and vice versa. In addition, different classifiers will have an influence on the selection of the appropriate combination method. However, empirical experiments reported to date cannot find an optimal method for selecting the combination strategy to be used. More theoretical development and experiments are needed to explore in this field. The ensemble combination technique most related to the work reported in this paper is stacking, as the new model outlined here inherits some ideas from stacking and develops them further. In this paper, we will propose a new model for ensemble combination method based on another neural network layer.

F. Roli and J. Kittler (Eds.): MCS 2002, LNCS 2364, pp. 91–97, 2002.

1.1 Stacking

Stacking [6] covers two areas of ensemble construction: preparing data and ensemble combination. Generally, stacking deals with two issues. Firstly, it uses the idea of cross-validation to select training data for ensemble members. Secondly, it explores the notion of using the second level generalizers to combine the results of the first level generalizers (ensemble members). A feature of stacked generalization is that the information supplied to the first-level ensemble members comes from multiple partitioning of the original dataset, which divides that dataset into two subsets. Every ensemble member is trained by one part of the partitions, and the rest of parts are used to generate the outputs of the ensemble members (to be used as the second space generalizers (i.e. combiners) inputs). Then the second level generalizers are trained with the original ensembles outputs and the second level generalizers output is treated as the correct guess. In fact, stacked generalization works by combined classifiers with weights according to individual classifier performance, to find a best combination of ensemble outputs. Based on the idea of the combining method of stacking, we propose a new type of ensemble neural network model called multistage ensemble neural networks.

2 Multistage Neural Network Ensembles

Inspired by stacking, some researchers have realized that it is possible to construct a new combination method using a similar idea.

As early as 1993,some experiments were done in digit recognition [14] by using a single layer network to combine ensemble classifiers. Unfortunately, these experiments did not show any performance gain compared with other combination strategies. It was claimed the failure was due to the very high accuracy of all the classifiers being combined.

In 1995, Partridge and Griffith presented a selector-net approach [5]. The selector-net was defined as a network which used the outputs from a group of different trained nets as its input. The experiments based on this idea delivered that selector-net's performance was better than the populations of networks they were derived from. It clearly confirmed that this kind of ensemble method is better than individual neural networks. But no further exploitation has been done to compare the performance of this strategy with any other ensemble method.

More recently, Kittler [15] stated that:"it is possible to train the output classifier separately using the outputs of the input classifiers as new features".

Very recently, Zeng [7] used a single neural network as an approximator for a voting classifiers. It was claimed that storage and computation could be saved, at the cost of a little less accuracy. However, it is noticed here a neural network being used to approximate the behavior of the ensemble, instead of using it as part of the ensemble components.

This paper extends the idea of stacking and investigates the use of a single neural network model as a combiner to combine the ensemble members results.

The experimental results demonstrate that it is an improved approach which achieves the better generalisation performance of neural network ensembles. The major improvement of this combination method is it offers an alternative neural network ensemble model, which is proved effective after the generalisation improvement obtained, compared with majority voting.

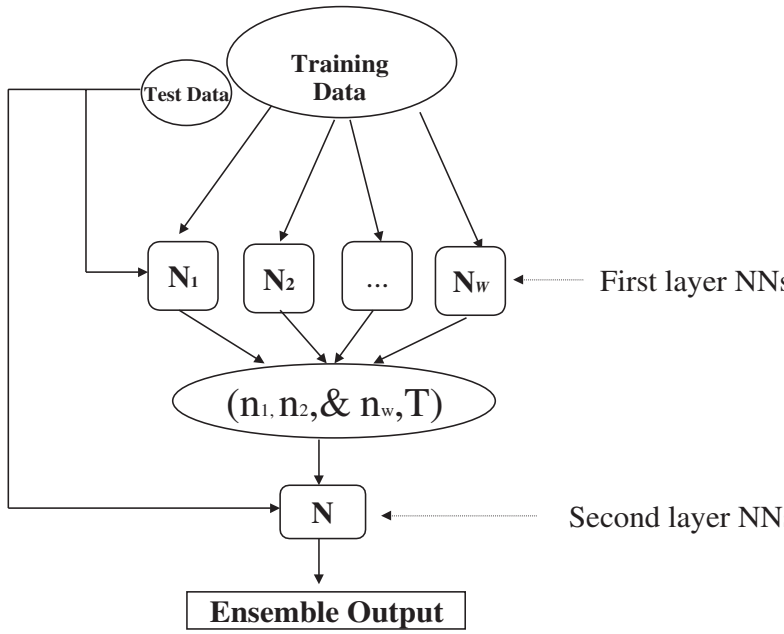

Fig. 1. An illustration of a multistage ensemble neural network.

The experiments on multistage neural network ensembles is based on a well trained group of diverse single neural networks (so called ensemble candidates). A single neural network is trained to combine these well trained neural nets results by concatenating their outputs together as its input. The reason of employing another neural networks to combine ensemble candidates is relying on neural networks capability. A neural network can be trained to prerform complex functions by asjusting the connection weights. Except majority voting, other approaches all adopt weights while combining. If so, why not use neural networks to assign weights to those ensemble members automatically instead of employing some traditional mathematical method manually? The advantage of a neural network is its ability to automatically adjust the connection weights. Therefore, it is very natural to think of using a neural network to combine ensemble results.

Table 1. Summary of UCI machine learning depository data sets, where '*' signifies a multi-class data set.

Data Set	No. of Cases	No. of Input features	No. of Output features
Breast-cancer-w	682	9	2
Bupa-Liver	345	6	2
Glass*	214	9	6
Ionosphere	351	34	2
Iris*	150	4	3
Pima-Diabetes	768	8	2

An illustration of the multistage neural network model described here is shown in Figure 1.

In Figure 1, suppose there is a source data set $S\{s_1, s_2, ...s_n\}$ and its corresponding target data set $T\{t_1, t_2, ...t_n\}$, which are partitioned into two parts: test data and training data. The training data usually will be preprocessed by various methods in order to generate diverse results before they being applied to the first layer's neural network models: $N_1, N_2, ...N_w$. The preprocessing methods on the training data set include distributing sequences randomly, noise injection, bagging, boosting or other methods. After training, the test data set will be applied to these ensemble candidates to access their performance. Afterwards, the whole training data set will be applied and each first layer neural networks' corresponding results $(n_1, n_2, ...n_w)$ are used as the second layer neural network model's inputs. The second layer neural networks, was trained by using the first layers generated results on the whole training data as inputs combined with their target data set. Advantages of multistage ensemble neural networks include:

- Multistage ensemble neural networks can be applied to both classification and regression problems.
- For each ensemble candidate, normal data preparation methods can be applied to them, and ensemble candidates can be trained separately by using various different neural network models and algorithms.
- Generalization of these first layer neural networks when using this model can be tuned to be as diverse as possible, so the choices for the first layer neural network training are very flexible and allow a wide range of selections.

3 Experiments

To investigate the performance of multistage ensembles, six classification data sets are taken from UCI Machine Learning Depository to construct experimental datasets. Details of these data sets are listed in Table 1, where datasets marked with '*' are multi-class datasets.

4 Experiments

4.1 Data Preparation

For each ensemble candidates' training, each data set was randomly partitioned into three parts: training, validation and test data. The sequences of training data were randomly distributed before they were applied to the neural networks model (i.e., steps were taken to prepare the most diverse data among ensemble members for training). There were no overlapping data instances inside the training data sets. For most of the data sets listed in Table 1, this approach was effective in generating diverse training data sets. The exception to this were the Glass and Iris data set, where the bagging [3] method was applied, due to source data's small size.

4.2 Experimental Procedures

Five single hidden layer neural networks trained by the backpropagation algorithm were generated as ensemble candidates for each ensemble. These five candidates were constructed with different sequences of training data, different neural network structures (numbers of hidden neurons) and different initialization. Each neural network model was trained 20 times with random initialization of starting weights. The number of hidden neurons is changed from one to the number of inputs of each data set (time considerations prevented the exploration of networks with hidden layers larger than this). During training, the validation data set was used to prevent overfitting. As the experiments in this paper is concentrated on comparing the performance of two ensemble combination methods. To make the things simpliest, the test data set is applied to the ensemble candidates after their training. Those ensemble candidates with best generalisation performance were kept. Majority voting and multistage neural networks then applied to these same ensemble members to generate the combination results.

After the training of the first layer's ensemble candidates, the whole source data is randomly disturbed again. 10-fold cross-validation [4] is applied to the second layer's neural network training in order to estimate the average performance. First the training data is injected into the ensemble members and their outputs are concatenated together with the corresponding target values as the input of the second layer's neural network. The training procedure and parameter setting for a neural network combiner are the same as an ensemble candidate's training.

The result of the majority voting applied to 3 and 5 ensemble members are then compared with the performance of multistage neural networks by averaging over 10-fold cross-validation.

5 Results

Table 2 shows the results for these different combination strategies when combining the best three ensemble candidates and all five ensemble candidates. In this

Table 2. Percentage correct performance on test set of voting versus multistage networks on UCI datasets. S1..S5 signify single networks, where each '#' indicates one of the three ensemble members selected for three-member combination. V1 and V5 signify combination by voting with three and five ensemble members respectively, whilst M3 and M5 signify combination by multistage neural network with three and five ensemble members respectively.

Data	S1	S2	S3	S4	S5	V3	M3	V5	M5
Breast-cancer-w	97.73#	98.48#	95.45	96.21	96.97#	97.21	97.35	97.35	97.5
Bupa-Liver	75.0#	78.0#	77.0#	77.0	78.0	72.65	73.82	73.53	73.82
Glass	42.19	46.88#	50.0#	45.31#	45.31	58.17	58.50	54.29	54.76
Ionosphere	85.53	94.74#	93.42#	86.84#	86.84	96.0	96.29	96.0	96.29
Iris	93.33#	92.0#	94.67#	90.67	90.67	97.60	98.0	97.6	98.0
Pima-Diabetes	72.02	73.81#	72.02#	75.60#	69.64	74.47	75.13	73.95	74.61

table, the actual ensemble candidates selected for ensembles of three networks are marked with '#'.

From these results, it can be seen that multistage neural networks always perform better than majority voting based on the same ensemble members, regardless the number of ensemble members used in combination is 3 or 5.

6 Conclusions

This paper has demonstrated that, on a wide range of datasets (including simple categorization and multiple categorization), multistage neural network ensembles offer improved performance when compared with majority voting as an ensemble combination method. The experimental results clearly show that statistic improvement can be made by using this new ensemble model on this range data of sets, when compared with another widely used ensemble technique such as majority voting. Currently, experiments are being carried out to investigate the exact theoretical reason for this performance improvement offered by multistage neural networks. The reason probably lies within differences between the kinds of decision surfaces a second layer network can model when compared to those decision surfaces that can be produced by majority voting. However, a clearer analysis and description of this area needs to be developed. In the future the intention is to develop and implement further experiments to investigate if the performance of multistage neural networks can be enhanced by using more ensemble members in the first layer, choice of training, validation and test datasets, and choice of neural network for the second layer combiner. It may be that these factors interact, and will allow this research to push the performance of such ensembles even further.

References

1. Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connection Science: Special Issue on Combining Artificial Neural Ensemble Approaches, 8(3&4), 385-404, 1999.
2. Sharkey, A. J. C., Sharkey, N. E., Chandroth, G. O.: Neural nets and diversity. Neural Computing and Applications, 4, 218-227, 1996.
3. Breiman, L.: Bagging predictors. Machine learning, 24 123-140, 1996.
4. Krogh, A., Vedelsby, J. : Neural Network Ensembles, Cross Validation, and Active Learning. Advances in Neural Information Processing Systems, 7, MIT press, Editors: Tesauro, G., Touretzky, D.S. and Leen, T.K. pp.231-238, 1995.
5. Partridge, D., Griffith, N. : Strategies for Improving Neural Net Generalisation. Neural Computing and Applications, 3, 27-37, 1995.
6. Wolpert, D. H.: Stacked generalization. Neural Networks, 5, 241-259, 1992.
7. Zeng, X., Martinez, T. R.: Using a Neural Network to Approximate an Ensemble of Classifiers. Neural Processing Letters, 12, 225-237, 2000.
8. Tumer, K., Ghosh, J.: Order statistics combiners of neural classifiers. In Proceedings of the World Congress on Neural Network, INNS press, Washington DC, 31-34, 1995.
9. Lincoln, W., Skrzypek, J.: Synergy of clustering multiple back propagation networks. Advances in Neural Information Processing Systems-2, Touretzky, D., (ed.), Morgan Kaufmann, 650-657, 1990.
10. Jacobs, R. A.: Methods for combining experts' probability assessments. Neural Computation, 7, 867-888, 1995.
11. Hansen, L., Salamon, P.: Neural network ensembles. IEEE Trans. Pattern Analysis and Machine Intell, 993-1001, 1990.
12. Al-Ghoneim, K., Kumar Vijaya B. V. K.: Learning ranks with neural networks. In Applications and Science of Artificial Neural Networks: Proceedings of the SPIE, 2492, 446-464, 1995.
13. Ho, T. K., Hull, J. J., Srihari, S. N.: Decision combination in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(1), 66-76, 1994.
14. Lee, D. S., Srihari, S. N.: Handprinted Digit Recognition: A Comparison of Algorithms. Pre-Proc. 3RD International Workshop On Frontiers In Handwriting Recognition, Buffalo, USA, 153-162, 1993.
15. Kittler, J.: Combining Classifiers: A Theoretical Framework. Pattern Analysis and Applications, 1, 18-27, 1998.