A Learning Algorithm For Neural Network Ensembles

H. D. Navone, P. M. Granitto, P. F. Verdes and H. A. Ceccatto

Instituto de Física Rosario (CONICET-UNR)
Blvd. 27 de Febrero 210 Bis, 2000 Rosario. República Argentina.
{navone,verdes,granitto,ceccatto}@ifir.edu.ar

Abstract

The performance of a single regressor/classifier can be improved by combining the outputs of several predictors. This is true provided the combined predictors are accurate and diverse enough, which posses the problem of generating suitable aggregate members in order to have optimal generalization capabilities. We propose here a new method for selecting members of regression/classification ensembles. In particular, using artificial neural networks as learners in a regression context, we show that this method leads to small aggregates with few but very diverse individual networks. The algorithm is favorably tested against other methods recently proposed in the literature, producing equal performance on the standard statistical databases used as benchmarks with ensembles that have 75% less members on average.

Keywords: Neural Networks, Ensemble Learning, Regression, Bias/Variance Decomposition.

1. Introduction

Ensemble techniques have been used recently in regression/classification tasks with considerable success. The motivation for this procedure is based on the intuitive idea that by combining the outputs of several individual predictors one might improve on the performance of a single generic one [1]. However, this idea has been proved to be true only when the combined predictors are simultaneously accurate and diverse enough, which requires an adequate trade-off between these two conflicting conditions. Some attempts [2,3] to achieve a good compromise between these properties include elaborations of *bagging*[4], *boosting*[5] and *stacking*[6] techniques.

Artificial neural networks (ANN) provide a natural framework for ensemble techniques. This is so because ANN are a very unstable learning method, *i.e.* small changes in training set and/or parameter selection can produce large changes in prediction outputs. This diversity of ANN comes naturally from the inherent data and training process randomness, and also from the intrinsic non-identifiability of the model (many different but *a priori* equally good local minima of the error surface). On the other hand, the above mentioned trade-off between the ensemble diversity and the accuracy of the individual members posses the problem of generating a set of ANN with both reasonably good (individual) generalization capabilities and distributed predictions for the test points. This problem has been considered in several recent works in the literature [2,3,7-9].

We provide here an alternative way of generating an ANN ensemble with members that are both accurate and diverse. The method essentially amounts to the sequential aggregation of individual predictors where, unlike in standard aggregation techniques which combine individually optimized ANN [7], the learning process of a (potential) new member is validated by the *overall* aggregate prediction performance. That is, an ANN is incorporated to the

Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No. 12 (2001), pp 70-74.
ISSN: 1137-3601. *© AEPIA (http://www.aepia.dsic.upv.es/.)*

ensemble only when this improves the generalization capabilities of the previous-stage aggregate predictor (see Section 3). The proposed algorithm seeks for a new ensemble member that is at least partially anticorrelated to the current ensemble on some validation set. Here "partially anticorrelated" means that the prediction errors of the potential new ensemble member and the previous aggregate predictor must have opposite signs at least for some validation points, so that by combining their predictions the errors on these points, and eventually the generalization error on the test set, will decrease.

We tested the proposed algorithm in the regression setting by comparing it against a simple early-stopping method and the recently-proposed NeuralBAG algorithm (a description of these two methods can be found in [9]). We compare the performances of the composite regression machines generated by these algorithms using as benchmarks the Ozone and Friedman#1 statistical databases.

The organization of this work is the following: In Section 2 we discuss the so called bias/variance dilemma, which provides the theoretical setting for ensemble averaging. In Section 3 we present our method for selecting the individual members of the ensemble and give some insights to understand why this method should work. Then, in Section 4 we show empirical evidence of its effectiveness by applying it to the Ozone and Friedman#1 databases and comparing the results obtained with other methods. Finally, in Section 5 we draw some conclusions.

2. The Bias/Variance Dilemma

The theoretical framework for ensemble averaging is based on the bias/variance decomposition of the generalization error[10]. Let's consider in the context of regression a set of N noisy data pairs $D=\{(t_i,\mathbf{x}_i), i=1,N\}$ obtained from some distribution P and generated according to

$$t = f(\mathbf{x}) + \varepsilon(\mathbf{x}),$$

where t is the observed target value, $f(\mathbf{x})$ is the true regression and $\varepsilon(\mathbf{x})$ is a random noise with zero mean. If we estimate f using an ANN trained on D and obtain a model f_D, the (quadratic) generalization error on a test point $(t,\mathbf{x})$ averaged on all possible realizations of the data set D and noise ε can be decomposed as:

$$\mathrm{E}\left[(t-f_D(\mathbf{x}))^2|D,\varepsilon\right] = \mathrm{E}[\varepsilon^2|D,\varepsilon] + \left(\mathrm{E}[f_D(\mathbf{x})|D,\varepsilon] - f(\mathbf{x})\right)^2 + \mathrm{E}\left[\left(f_D(\mathbf{x}_i) - \mathrm{E}[f_D(\mathbf{x}_i)|D,\varepsilon]\right)^2|D,\varepsilon\right]$$

The first term on the RHS is simply the noise variance; the second and third terms are, respectively, the squared bias and variance of the estimation method. From the point of view of a single estimator $f_D(\mathbf{x})$, we can interpret this equation by saying that a good method should be roughly not biased and have as little variance as possible between different realizations. It is in general believed that the first condition is reasonably well met by ANN; however, as stated in the introduction, the second one is in general not satisfied since, even for a particular data set D, different training experiments will reach distinct local minima of the error surface (non-identifiability of the ANN model).

A way to take advantage of this apparently weakness of ANN is to make an aggregate of them. If we rewrite the error decomposition in the form:

$$\mathrm{E}\left[\left(t - \mathrm{E}[f_D(\mathbf{x})|D,\varepsilon]\right)^2|D,\varepsilon\right] = Bias^2 + \sigma_\varepsilon^2 = Error - Variance,$$

we can reinterpret this equation in the following way: using the average $\Phi \equiv \mathrm{E}[f_D|D,\varepsilon]$ as estimator, the generalization error can be reduced if we are able to produce fairly accurate models f_D (small *Error*) while, at the same time, allowing them to produce the most diverse predictions at every point (large *Variance*). Of course, there is a trade off between these two conditions, but finding a good compromise between accuracy and diversity seems particularly feasible for largely unstable methods like ANN. Several ways to generate an ensemble of models with these characteristics have been discussed in the literature [2,3,7-9]. In the next section we propose a new method to perform this task and give some arguments that suggest why it should be effective; these ideas are later supported by empirical evidence on benchmark data.

3. The Algorithm for Ensemble Learning

As suggested in the previous section, in order to improve the generalization capabilities of the aggregate predictor one must generate diverse individual predictors retaining only those that perform reasonably well on validation data. This can be accomplished by the following procedure:

Step 1: Generate a training set T_1 by a bootstrap re-sample[11] from dataset D, and a validation set V_1 collecting all instances in D that are not included in T_1. Generate a model f_1 by training a network on T_1 until a minimum e_1 in the generalization error on V_1 is reached.

Step 2: Generate new training and validation sets T_2 and V_2, respectively, using the procedure described in step 1. Produce a model f_2 by training a network until the generalization error on V_2 of the *aggregate* predictor $\Phi_2 = \frac{1}{2}(f_1+f_2)$ reaches a minimum $e_{\Phi_2}(V_2) < e_1$. In this step the parameters in model f_1 remain constant and the model f_2 is trained with the usual (quadratic) cost function on T_2.

Step 3: If the previous step cannot be accomplished after a maximum N_E of training epochs, disregard the current training and validation sets and start again with step 2 using new random sets T_2 and V_2

Step 4: If a model f_2 is found, incorporate it to the ensemble and proceed again with steps 2 and 3 seeking for a model f_3 such that $e_{\Phi_3}(V_3)$, the minimum generalization error on V_3 of the aggregate $\Phi_3=(f_1+f_2+f_3)/3$, becomes smaller than $e_{\Phi_2}(V_2)$.

Step 5: Iterate the process until N_A models are trained (including accepted and rejected ones). The individual networks collected up to this point constitute the final ensemble.

A few comments are in order at this point. First, the algorithm incorporates a new member to the ensemble only when it helps in improving the *aggregate* generalization performance on a particular validation set; in practice we found useful to check that the performance on the *whole* data set D also improves or otherwise to stop the algorithm. This helps in reducing the computational time since for large values of N_A the procedure will first terminate by this second condition. On the other hand, this condition seems to lead to final ensembles with nicer generalization capabilities on the test set (see next section). Second, compared to other methods the algorithm produces smaller ensembles although with members that are more efficient in reducing the generalization error.

The algorithm above described seems to directly reduce the ensemble generalization error without paying much attention to whether this improvement is related to enhancing ensemble diversity or not. We can see that it actually finds diverse models to reduce the aggregate error as follows. Let's assume that after n successful iterations we have an aggregate predictor Φ_n which produces an average error $e_{\Phi_n}(V_n)$ on the validation set V_n. When we try to find model f_{n+1}, the average validation error on V_{n+1} of the aggregate predictor Φ_{n+1} will be

$$e_{\Phi_{n+1}}(V_{n+1}) = (n+1)^{-2}\{e_{n+1} + n^2 e_{\Phi_n}(V_{n+1}) + 2n\mathrm{E}[(t - f_{n+1}(\mathbf{x}))(t - \Phi_n(\mathbf{x}))|(t,\mathbf{x})\in V_{n+1}]\}.$$

Since to keep f_{n+1} in the ensemble we require $e_{\Phi_{n+1}}(V_{n+1}) < e_{\Phi_n}(V_n)$, in general we will have $e_{\Phi_{n+1}}(V_{n+1}) < e_{\Phi_n}(V_{n+1})$. Furthermore, due to overtraining of model f_{n+1} we expect $e_{\Phi_{n+1}}(V_{n+1}) < e_{n+1}$. Then,

$$\mathrm{E}[(t - f_{n+1}(\mathbf{x}))\,(t - \Phi_n(\mathbf{x})) \mid (t,\mathbf{x})\in V_{n+1}] < e_{\Phi_{n+1}}(V_{n+1}),$$

which is only possible if f_{n+1} is at least partially anticorrelated to the aggregate Φ_n. This analysis shows that at every stage the algorithm is seeking for a new diverse model anticorrelated with the current ensemble. In the next section we will show how this heuristic works on real and synthetic data corresponding respectively to the Ozone and Friedman#1 statistical databases.

4. Evaluation

We have used two regression problems to evaluate the algorithm described in the previous section: the real-world Ozone and the synthetic Friedman#1 data sets. We compared our method against a simple early-stopping method and the recently-proposed NeuralBAG algorithm [9]. In order to compare only the networks selection methods, we used the same training process of individual networks for all of them, and changed only the selection criteria.

Dataset	Simple	NBAG	This Work
Ozone	19.81 ± 3.33	19.70 ± 3.21	19.72 ± 3.14
Friedman#1	2.39 ± 0.47	2.26 ± 0.39	2.25 ± 0.44

Table 1: Mean-squared generalization errors averaged over 20 runs corresponding to three different algorithms for ensemble learning.

4.1. Ozone

The Ozone data correspond to meteorological information (humidity, temperature, etc.) related to the maximum daily ozone (regression target) at a location in the Los Angeles basis. Removing missing values one is left with 330 training vectors, containing 8 inputs and 1 target output in each one. The data set can be downloaded by ftp (ftp://ftp.stat.berkeley.edu/pub/users/breiman) from

the Department of Statistics, University of California at Berkeley.

Like in [9], we have considered ANN architectures with 5 hidden units trained by the backpropagation rule with learning rate η=0.1 and momentum p=0.2. Furthermore, we performed also the same (random) splitting of the data in training and test sets containing, respectively, 125 and 80 patterns. The parameters that characterize the algorithm discussed in Section 3 were set to N_E = 10,000 and N_A = 30. Notice that according to [9] these parameter values are nearly the optima for the NeuralBAG algorithm. The results given in Table 1 correspond to the average over 20 runs of the whole procedure. The average mean-squared error corresponding to our method is practically the same error produced by NeuralBAG, and is slightly smaller than the error produced by the simple method. All runs lead to fairly small ensembles containing a minimum of 2 to a maximum of 7 ANN, which should be compared with the ensembles of 30 ANN generated by the other two methods. In Fig. 1 we plot the evolution in one of the experiments of the mean-squared errors corresponding to the validation (whole data) and test sets, as a function of the number of networks incorporated to the ensemble. Notice that, in spite of the small number of validation and test examples, the minima of these errors occur for slightly different numbers of ANN in the ensemble. Actually, in 40% of the experiments these minima coincide.

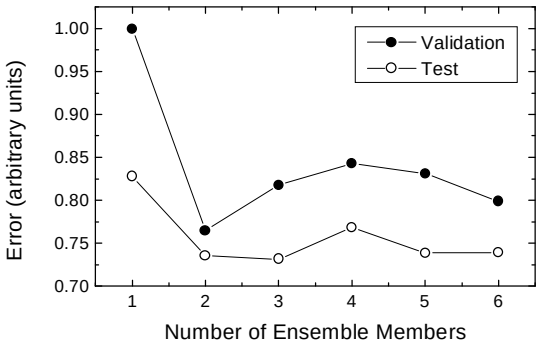

Fig. 1: Evolution of the errors corresponding to the validation and test sets as a function of the number of networks incorporated to the ensemble for the Ozone problem.

4.2. Friedman#1

The Friedman#1 synthetic data set corresponds to training vectors with 10 input and 1 output variables generated according to

$$t = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon,$$

where ε is Gaussian noise with distribution $N(0,1)$ and $x_1,..,x_{10}$ are uniformly distributed over the interval [0,1]. Notice that $x_6,..,x_{10}$ do not enter in the definition of t and are only included to check the prediction method's ability to ignore these inputs.

Following [9], we generated 1400 sample vectors and randomly split the data in training and test sets containing, respectively, 200 and 1000 patterns. Furthermore, we considered ANN with 6 hidden units and the same learning rate η=0.1 and momentum p=0.2 for the backpropagation rule used in [9]. Again, we set N_E = 10,000 and N_A = 30, and performed 20 runs of the whole process. The corresponding mean-squared error on the test set is given in Table 1, where we again include the performances of the other methods considered in [9] for comparison. This time the ensembles contain 4 to 10 ANN, and give the same average error as the 30 ANN ensembles produced by NBAG.

Fig. 2 shows the evolution of the validation and test set errors as a function of the number of ANN in the ensemble for a typical experiment. Unlike the previous case of the Ozone dataset, here the minima of these curves coincide in more than 75% of the runs.

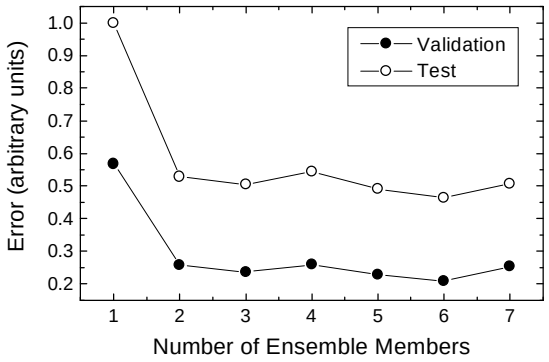

Fig. 2: Evolution of the errors corresponding to the validation and test sets as a function of the number of networks incorporated to the ensemble for the Friedman#1 dataset.

4.3. Discussion

For both the Ozone and Friedman#1 datasets our algorithm shows some improvement over the simple early-stopping method, which can only be produced by the selection procedure. More importantly, it produces ensembles with an average of 4 members for the Ozone dataset and 7 members for Friedman#1, showing the same performance of NeuralBAG that uses 30 members. It has also the advantage over this last method of avoiding the computational burden of keeping all the intermediate networks generated in the training process. Furthermore, the individual networks do not need to be trained to convergence like in

NeuralBAG, since the training process terminates by validation like in the early-stopping method.

6. Conclusions

We proposed a new method for selecting diverse members of ANN ensembles. At every stage, the algorithm seeks for a new potential member that is at least partially anticorrelated with the previous-stage ensemble estimator. By comparison against other methods proposed in the literature, we showed that this strategy is effective and generates small ensembles with very diverse members. This was exemplified by applying our method to two standard statistical benchmarks, the Ozone and Friedman#1 datasets. The results are encouraging and we are presently performing a more extensive check of the algorithm on new databases.

REFERENCES

[1] A. Krogh and J. Vedelsby, "Neural network ensembles, cross-validation and active learning", in G. Tesauro, D. Touretzky and T. Lean, eds., *Advances in Neural Information Processing Systems* **7**, 231-238 (MIT Press, 1995)

[2] J. G. Carney and P. Cunningham, "The NeuralBAG algorithm: Optimaizing generalization performance in bagged neural networks", in M. Verleysen, ed., *Proceedings of the 7th European Symposium on Artificial Neural Networks*, 35-40 (D-Facto, Brussels, 1999)

[3] H. Drucker, R. Schapire and P. Simard, "Improving performance in neural networks using a boosting algorithm", in S. J. Hanson, J. D. Cowen and C. L. Giles, eds., *Advances in Neural Information Processing Systems 5*, 42-49 (Morgan Kaufman, 1993)

[4] L. Breiman, "Bagging predictors", *Machine Learning* **24**, 123-140 (1996)

[5] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", in *Proceedings of the Second European Conference on Computational Learning Theory, 23-37* (Springer Verlag, 1995)

[6] D. Wolpert, "Stacked generalization", *Neural Networks* **5**, 241-259 (1992)

[7] U. Naftaly, N. Intrator and D. Horn, "Optimal ensemble averaging of neural networks", *Network: Comput. Neural Syst.* **8**, 283-296 (1997)

[8] D. Opitz and J. Shavlik, "Generating accurate and diverse members of a neural network ensemble", in D. Touretzky, M. Mozer and M. Hasselmo, eds., *Advances in Neural Information Processing Systems* **8**, 535-541 (MIT Press, 1996)

[9] J. Carney and P. Cunningham, "Tuning diversity in bagged ensembles", *International Journal of Neural Systems* **10**, 267-280 (2000)

[10] S. Geman, E. Bienenstock and R. Doursat, "Neural Networks and the Bias/Variance Dilemma", *Neural Computation* **4**, 1-58 (1992)

[11] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap* (Chapman and Hall, London, 1993)