On the Effectiveness of Negative Correlation Learning

Gavin Brown and Xin Yao
School of Computer Science, The University of Birmingham
Edgbaston, Birmingham B15 2TT, U.K.
{g.brown,x.yao}@cs.bham.ac.uk

Abstract

Neural network ensembles are well accepted as a route to combining a group of weaker learning systems in order to make a composite, stronger one. It has been shown that low correlation of errors ("diverse members") will give rise to better ensemble performance. Most techniques for creating diverse ensemble members indirectly affect the learning trajectories, and are built upon heuristics and intuition. Other techniques directly influence the learning trajectory, by altering the training algorithm itself. For a particular direct technique, Negative Correlation Learning, we demonstrate the effectiveness of the algorithm in reducing correlations, as it relates to the size and complexity of the ensemble. We offer some possible research avenues on this class of ensemble methods. This work is a first step towards understanding the effectiveness of explicitly incorporating diversity measures in error functions during ensemble training.

1 Introduction

Neural network ensembles offer a number of advantages over a single neural network system. They have the potential for improved generalization, lower dependence on the training set, and reduced training time (Sharkey [16] provides a excellent summary of the literature).

Training a neural network generally involves a delicate balance of various factors. The *bias-variance decomposition* [2] states that the mean square error of an estimator (in our case, a neural network) is equal to the bias squared plus the variance. There is a tradeoff here — with more training, it is possible to achieve lower bias, but at the cost of a rise in variance. Krogh and Vedelsby [4] extend this concept to ensemble errors, showing how the bias can be seen as the extent to which the averaged output of the ensemble members differs from the target function, and the variance is the extent to which the ensemble members disagree. Ueda and Nakano [18] further provide a detailed proof of how the decomposition can be extended to *bias-variance-covariance*. From this result, one way to decrease the error is clear: decrease the covariance, ideally making it strongly negative - though too large a decrease in covariance can cause a rise in bias and variance. This means that an ideal ensemble consists of highly correct classifiers that disagree as much as possible (balancing the covariance against the bias and variance), empirically verified by Opitz and Shavlik [11] among others.

Ensembles have been successfully applied to both regression and classification problems in varied domains, such as time series prediction [19], robotics [10], medical diagnosis [15], and traffic flow prediction in a telecommunications system [20]. Following from observations in our previous work [20] we investigate the behaviour of the Negative Correlation algorithm [6] as we increase the complexity of the ensemble on a particular problem. The primary aim of this paper is to gain a deeper understanding of how Negative Correlation Learning works and what improvements can be made. A lot of techniques for creating diverse ensemble members are built upon heuristics and intuition. This work is a first step towards understanding the effectiveness of explicitly incorporating diversity measures in error functions during ensemble training.

The rest of this paper is organised as follows. Section 2 briefly reviews methods for creating effective ensembles. Section 3 introduces the basic ideas behind negative correlation learning. Section 4 presents the experimental setup and Section 5 presents the results. Finally, Section 6 concludes the paper.

2 Creating an Effective Ensemble

Various algorithms have been proposed for training ensembles to achieve better generalisation. They can be broadly classified as manipulating the initial conditions, the network architectures, the training data, or the learning algorithm. All but the latter, we refer to as *implicit methods*, in that they alter the learning environment, hoping that indirectly the learning trajectories will emerge diverse. Manipulating the learning algorithm is an *explicit method* for achieving diversity, in that it directly influences the learning trajectories of the networks to this end.

2.1 Implicit Methods

Early work trained networks independently, then averaged the results, hoping to achieve higher performance simply through differences in initial conditions (different weight initializations). The idea was that starting the networks in different areas of the weight space, they would follow different trajectories in the functional space. However, if a random initialization by chance gives a set of weights that are far from a solution, convergence can be exceedingly slow.

Manipulation of training data has been the most widely investigated method. Boosting [1], bagging, disjoint input sources [17], nonlinear transformations of input [17], and noise injection [13] have all proved their worth.

Manipulating the network topologies would mean having hybrid ensembles, consisting of estimators that work in different search spaces entirely. Different areas of functional (solution) space will be more accessible in certain search spaces than in others. Although this at first seems a promising path, not much work seems to have been done in the area. Partridge and Yates [12] present the conjecture that variation in network architectures is, after initial weights, the least useful method of creating diversity, due to the methodological similarities in the supervised learning algorithms. Advantages may be revealed through more detailed investigations of the behaviour of truly hybrid ensembles, i.e. consisting of entirely different learning machines.

2.2 Explicit Methods

As mentioned, the previous three methods (manipulating initial weights, training data and architectures) are all implicit methods for achieving diverse errors; the networks may still converge to be highly correlated, regardless of your efforts. Explicit methods manipulate the training algorithm itself to produce decorrelated errors. Rosen [14] used a regularisation term, training an ensemble sequentially, to decorrelate nets from ones that had been trained before, although this did not guarantee negative correlation of all the networks. A recent advancement, *Negative Correlation Learning* [6], trained the networks in parallel and negatively correlated the network errors. This had the advantage of removing any bias in manipulation of the training data, as well as elimination of the need for a gating network, inherent in the Mixtures-of-Experts architecture [3]. The Negative Correlation (NC) learning algorithm has shown marked improvements over other ensemble learning algorithms [6, 7, 8, 21],

3 Negative Correlation Learning

NC-learning [6] is an efficient ensemble training method which can easily be implemented on top of standard backpropagation in feedforward networks. It incorporates a measure of ensemble diversity into the error function of each network: thus each network not only decreases its error on the function, but also increases its diversity from other network errors. The procedure has the following form: take a set of neural networks N and a training pattern set P, each pattern in P is presented and backpropagated on, *simultaneously*, by the networks.

In the standard backpropagation algorithm, the error function for the output layer nodes is

$$\frac{1}{2}(F_i(n) - d(n))^2,$$

where $F_i(n)$ is the output of network i on pattern n, and $d(n)$ is the desired response for that pattern. In NC-learning, the error function becomes

$$\frac{1}{2}(F_i(n) - d(n))^2 + \lambda p_i(n), \tag{1}$$

where $p_i(n)$ is

$$(F_i(n) - F(n)) \sum_{j \neq i} (F_j(n) - F(n)), \tag{2}$$

and λ is an adjustable strength parameter for the penalty. $F(n)$ is the output of the ensemble on pattern n. A common ensemble output function is a simple average of the networks in the ensemble, i.e.,

$$F(n) = \frac{1}{N} \sum_{i=1}^{N} F_i(n) \tag{3}$$

In this case we have an overall error function of

$$\frac{1}{2}(F_i(n) - d(n))^2 - \lambda (F_i(n) - F(n))^2 \tag{4}$$

As can be seen from (4), each network receives lower error for moving its response closer to the target response, and away from the mean response of all the other networks — this is a trade-off, controlled by the penalty strengh parameter, λ. When $\lambda = 0.0$, the networks ignore the other errors, and this is termed *independent training*, equivalent to not using NC at all.

The dynamics of an algorithm incorporating such a diversity measure, and how to set its strength parameter, λ, are not well understood. In previous work [20] we presented an evolutionary approach, essentially *evolving the diversity* of the ensemble. We unexpectedly found situations where positive values for λ were preferred. Here we do not attempt to explain the negative λ phenomenon, but attempt to understand the general behaviour of the algorithm more fully, as it relates to ensemble size and complexity.

4 Experimental Setup

4.1 Dataset

The data was generated by the function:

$$f(x) = \frac{1}{13}\left[10sin(\pi x_1 x_2) \right.$$
$$\left. +20\left(x_3 - \frac{1}{2}\right)^2 + 10x_4 + 5x_5\right] - 1 \qquad (5)$$

where $x = [x_1, .., x_5]$ is an input vector whose components lie between zero and one. The value of $f(x)$ lies between -1 and $+1$. The data consisted of input/output patterns with the input vectors sampled uniformly at random from the interval (0,1). Training pattern set size was 500, testing set size was 1024, as used by Liu [6] and previously Jacobs et al [3]. The difference between their use of the data and the use here, is that we use only 1 training set, whereas they used 25 trials. The point of their investigations was to gain an accurate measure of their method's performance on the function approximation - here we are not concerned with the fit, but a better understanding of how the error varies with alterations to the NC-learning algorithm and the ensemble topology.

4.2 Ensemble setup

In the first part of the investigation we use an ensemble architecture with 4 networks, each with the same number of hidden nodes, H. We observe the performance gain for the ensemble using NC-learning, compared to independent training, as we vary H from 2 to 20. In the second part, we use an ensemble architecture consisting of N networks with 4 hidden nodes. We observe the performance gain for the ensemble using NC-learning, compared to independent training, as we vary N from 2 to 15. We then repeat this part, using ensembles with 2 and then 6 hidden nodes, again observing the gain over independent training. As mentioned, the λ parameter is the emphasis put on achieving negative correlations. In the third part of the investigation we observe how the value of λ relates to the actual correlations achieved, when we scale up the complexity of the ensemble: firstly with more hidden nodes, and secondly with more networks. The measure used is Pearson's correlation coefficient, averaged over all possible pairings of networks in the ensemble as in [6].

In all experiments, performance is measured at $\lambda = 0.0, 0.3, 0.6$ and 0.9. A higher value of λ means more emphasis on decorrelating the errors, in preference to each network just fitting the objective function; a value of zero means NC-learning was not used at all. These settings are not meant to be optimal, but were chosen to demonstrate a wide range of performance.[1] All networks have a single hidden layer, and use the logistic activation function for all nodes, learning rate 0.1, with no momentum term. The ensembles were trained for 2000 iterations, and the error averaged over 30 random weight initialisations. The ensemble is combined by a uniform average of the outputs of the individual networks.

5 Results

Figure 1 shows the percentage gain over independent training that NC-learning provided to the ensemble. It is clear that as we increase the complexity of the component networks, NC-learning is of less and less utility. In some cases, when the number of hidden nodes was very large, a *decrease* in ensemble performance is observed with NC-learning, as can be seen at $H > 12$. One possible explanation for this is that as individual networks become more and more capable, they can approximate the whole function by themselves, and so an ensemble approach in general has less utility. Figure 2 supports this, showing a plot of the actual MSE of a

[1] During initial experiments, particularly high error was obtained at $\lambda = 1.0$, this phenomena is under investigation, but for this work, was avoided.

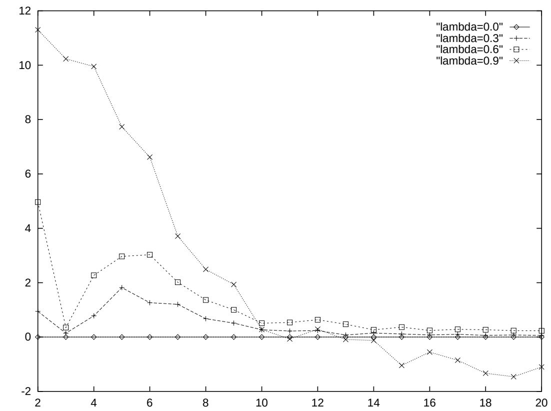

Figure 1: Percentage performance gain of NC-learning over independent training ($\lambda = 0.0$), varying the number of hidden nodes, H, in a 4-network ensemble.

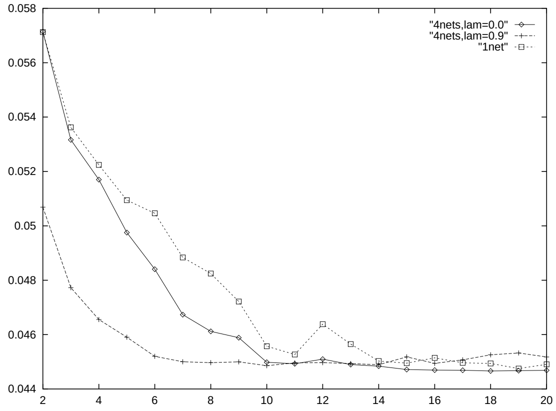

Figure 2: Mean squared error, varying the number of hidden nodes in a 4-network ensemble

single net, compared to an ensemble with and without NC-learning. The ensemble error is immediately very low with 6 hidden nodes, and $\lambda = 0.9$; the single net approaches the same level of performance around 11 hidden nodes, at which point the the NC-learning cannot provide any significant gains. Furthermore, when using NC-learning and networks which can approximate the whole function by themselves, the risk may be that since the networks cannot fit the objective function any more accurately, they will choose to overfit to the penalty function, causing a decrease in testing generalisation.

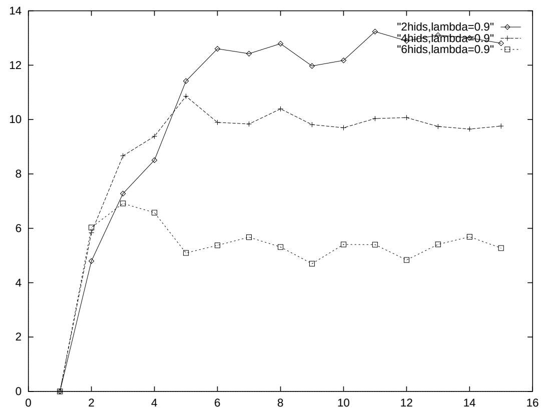

Figure 3: Percentage performance gain of NC-learning over independent training ($\lambda = 0.0$), varying the number of networks, N

We can see in figure 3 the percentage performance gain of NC-learning over independent training, comparing the gain for ensembles using 2, 4 and 6 hidden nodes, as we increase the number of networks. We observed that the largest percentage gain provided by NC-learning was on 2 hidden node networks, with $\lambda = 0.9$, allowing at best a 13 percent increase. An interesting point shows up here - the groups of 2 hidden node networks showed the smallest percentage gain for using NC, until the size of the group was increased. At 5 or more networks in the group, a gain of at least 12 percent was consistently observed. There is a larger percentage gain in performance when the ensemble consists of a large number of less complex networks.

NC-learning is an algorithm to decrease the correlations between the networks. We observed its ability to do this, as we vary the number of hidden nodes in a four network ensemble. Figure 4 shows the mean correlation coefficient on the Y-axis, against the hidden nodes from 2 to 20. This shows that the complexity of the component networks did not have a large effect on the eventual correlations achieved. Indeed, without NC ($\lambda = 0.0$), the correlations show large fluctuations with respect to the complexity. When NC is used at $\lambda = 0.9$, it seems to stabilise the correlation, making it virtually invariant with respect to the network complexity.

Figure 5 shows the mean correlation against the number of networks. It shows that for a given value of $\lambda = 0.9$, it becomes progressively harder to achieve negative correlations with more networks in the group.

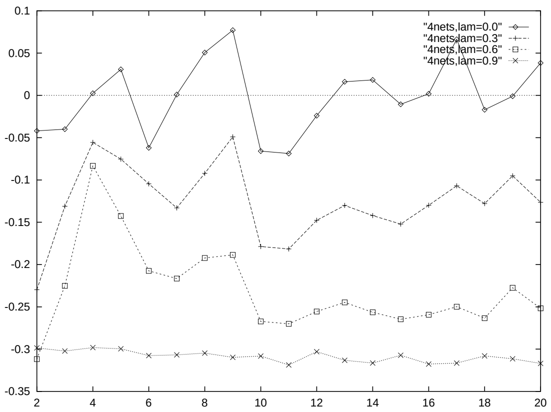

Figure 4: Y-axis is the mean correlaton coefficient after training, X-axis is the number of hidden nodes.

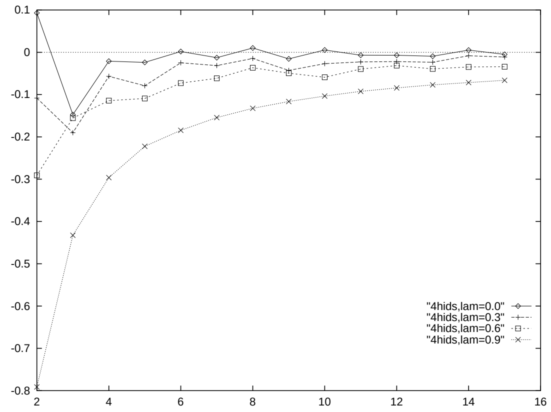

Figure 5: Y-axis is the mean correlaton coefficient after training, X-axis is the number of networks.

6 Conclusions & Further Work

We investigated the behaviour of one explicit method for creating diverse neural ensembles, Negative Correlation Learning, as we scale up the complexity of the ensemble. We observed a decrease in the utility of NC-learning as we increased the size of the networks within the ensemble. It is shown that more complex networks did not lead to more negative correlations. In fact, correlation appeared to be invariant with respect to network complexity.

We observed an increase in the utility of NC-learning when we increased the number of networks with a relatively small number of hidden nodes, for a particular setting of the strength parameter. It appears that the number of networks in the ensemble is more important than individual network complexity. The benefits of the NC technique are best shown when combining relatively weak estimators - when each individual network is powerful enough to approximate the function itself, NC-learning can do little for the group. We observed the mean correlation coefficient of the ensemble, as it relates to ensemble complexity: a higher value of λ stabilised the correlations, while it was still not highly affected by the number of hidden nodes; with a large number of networks it proved harder to achieve negative correlations.

Kuncheva and Whitaker [5] have recently presented work relating various diversity measures to the majority vote, on classification problems. It seems plausible that similar relationships might hold for other combination methods with the particular diversity measure implemented by NC-learning. Liu et al [9] have found links between information theory and diversity in an ensemble with an evolutionary approach.

It is important to note two families of explicit diversity methods. The first we term *error dependency methods*: those which evaluate pairwise dependency on a single pattern, such as NC-learning, using a form like (2). The second is *error coincidence methods*: those which are evaluated over a training dataset, taking into account the coincidence of errors on various patterns. Our future work consists of the use of different measures, as well as different methodologies for incorporating them.

A lot of techniques for creating diverse ensemble members are built upon heuristics and intuition. This work is a first step towards understanding the effectiveness of explicitly incorporating diversity measures in error functions during ensemble training.

7 Acknowledgements

We would like to thank Dr. Jeremy Wyatt for numerous useful comments on methodology and content.

References

[1] H. Drucker, C. Cortes, L. Jackel, Y. LeCun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6:1289–1301, September 1994.

[2] Stuart Geman, Elie Bienenstock, and Rene Doursat. Neural networks and the bias/variance

dilemma. *Neural Computation*, 4:1–58, 1992.

[3] Robert A. Jacobs, Michael I. Jordan, and Andrew G. Barto. Task decomposition through competition in a modular connectionist architecture - the what and where vision tasks. *Cognitive Science*, 15:219–250, 1991.

[4] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation and active learning. *Advances in Neural Information Processing Systems (NIPS-7)*, 7, 1995.

[5] L.I. Kuncheva and C.J. Whitaker. Measures of diversity in classifier ensembles. *Machine Learning (submitted)*, 2001.

[6] Yong Liu. *Negative Correlation Learning and Evolutionary Neural Network Ensembles.* PhD thesis, University College, The University of New South Wales, Australian Defence Force Academy, Canberra, Australia, 1998.

[7] Yong Liu and Xin Yao. Negatively correlated neural networks can produce best ensembles. *Australian Journal of Intelligent Information Processing Systems*, 4(3/4):176–185, 1997.

[8] Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404, 1999.

[9] Yong Liu, Xin Yao, Qiangfu Zhao, and Tetsuya Higuchi. Evolving a cooperative population of neural networks by minimizing mutual information. In *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 384–389. IEEE Press, May 2001.

[10] M. Meng and A. C. Kak. Mobile robot navigation using neural networks and nonmetrical environment models. *IEEE Control Systems*, pages 30–39, October 1993.

[11] David Opitz and Jude Shavlik. Generating accurate and diverse members of a neural-network ensemble. *Advances in Neural Information Processing Systems*, 8, 1996.

[12] D. Partridge and W. B. Yates. Engineering multiversion neural-net systems. *Neural Computation*, 8(4):869–893, 1996.

[13] Yuval Raviv and Nathan Intrator. Bootstrapping with noise: An effective regularisation technique. *Connection Science*, 8:355–372, 1996.

[14] Bruce E. Rosen. Ensemble learning using decorrelated neural networks. *Connection Science - Special Issue on Combining Artificial Neural Networks: Ensemble Approaches*, 8(3 and 4):373–384, 1996.

[15] A. Sharkey, N. Sharkey, and S. Cross. Adapting an ensemble approach for the diagnosis of breast cancer. pages 281–286, 1998.

[16] Amanda Sharkey. *Multi-Net Systems*, chapter Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems, pages 1–30. Springer-Verlag, 1999.

[17] Amanda Sharkey, Noel Sharkey, and Gopinath Chandroth. Diverse neural net solutions to a fault diagnosis problem. *Neural Computing and Applications*, 4:218–227, 1996.

[18] N. Ueda and R. Nakano. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks (ICNN96)*, pages 90–95, 1996.

[19] Andreas S. Weigend and Morgan Mangeas. Nonlinear gated experts for time series: discovering regimes and avoiding overfitting. *International Journal of Neural Systems*, 6:373–399, 1995.

[20] Xin Yao, Manfred Fischer, and Gavin Brown. Neural network ensembles and their application to traffic flow prediction in telecommunications networks. In *Proceedings of International Joint Conference on Neural Networks*, pages 693–698. IEEE Press, 2001. Washington DC.

[21] Xin Yao and Yong Liu. Neural networks for breast cancer diagnosis. In *Proceedings of the 1999 Congress on Evolutionary Computation*, volume 3, pages 1760–1767. IEEE Press, July 1999.