The Use of the Ambiguity Decomposition in Neural Network Ensemble Learning Methods

Gavin Brown G.BROWN@CS.BHAM.AC.UK
School of Computer Science, University of Birmingham,
Edgbaston Park Road, Birmingham, B15 2TT

Jeremy Wyatt J.L.WYATT@CS.BHAM.AC.UK
School of Computer Science, University of Birmingham,
Edgbaston Park Road, Birmingham, B15 2TT

Abstract

We analyze the formal grounding behind Negative Correlation (NC) Learning, an ensemble learning technique developed in the evolutionary computation literature. We show that by removing an assumption made in the original work, NC can be seen to be exploiting the well-known *Ambiguity* decomposition of the ensemble error, grounding it in a statistics framework around the bias-variance decomposition. We use this grounding to find bounds for the parameters, and provide insights into the behaviour of the optimal parameter values. These observations allow us understand how NC relates to other algorithms, identifying a group of papers spread over the last decade that have all exploited the Ambiguity decomposition for machine learning problems. When taking into account our new understanding of the algorithm, significant reductions in error rates were observed in empirical tests.

1. Introduction

We study the formal basis behind Negative Correlation (NC) Learning, a successful neural network ensemble learning technique developed in the evolutionary computation literature. NC has shown a number of empirical successes and applications, including regression problems, time-series prediction (Liu, 1998), and classification problems (McKay & Abbass, 2001). It has consistently demonstrated significant performance improvements over a regular ensemble system, showing very competitive results with other techniques like Mixtures of Experts and RBF networks (Liu & Yao, 1997). NC so far has had very little formal analysis to explain why it works when it does; this provides the motivation for our work.

1.1. A Dichotomy of Methods

A *neural network ensemble* is a collection of neural networks. The ensemble as a whole provides an output which is a combination of the individual network outputs; the motivation for this is to further decrease generalisation error. Several theoretical and empirical works have shown that for best performance, the errors of the individuals should exhibit as low correlation as possible (Ueda & Nakano, 1996; Rosen, 1996), whilst maintaining a reasonably high accuracy. Due to the many different forms of network outputs (probabilities, class labels, ranked lists of labels) this has become the amorphous goal of achieving high 'error diversity'. In this work we are concerned with regression-based combinations, enabling a more rigorous treatment of the error diversity.

A number of methods designed to encourage error diversity have matured over the last decade. Our framework for this investigation hinges on regarding these methods as dichotomous: *explicit* and *implicit* diversity methods. Explicit methods measure diversity (correlation) in some manner and directly incorporate this knowledge into the construction or combination of the estimators; for example Input Decimation Ensembles (Oza & Tumer, 2001), which measure

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

correlation between features before assigning them to particular networks, or AdaBoost (Freund & Schapire, 1996) and its many variants, which explicitly modify the distributions of training data fed to each network. Implicit methods utilise purely stochastic perturbations to encourage diversity; for example, Bagging (Breiman, 1996) or similar data resampling techniques. In this paper we are concerned with explicit methods, in particular those which share a common root in the Ambiguity decomposition from (Krogh & Vedelsby, 1995), widely recognised as one of the most important theoretical results obtained for ensemble learning. It states that *the mean-square error (MSE) of the ensemble estimator is guaranteed to be less than or equal to the average mean-square error of the component estimators*; the details of this will be expanded upon later.

1.2. Negative Correlation Learning

After this initial branching, both explicit and implicit methods can be further divided as manipulating either: the initial weights of the networks, the network architectures, the training data, or the learning algorithm. Some authors, taking the latter approach, have found benefit from using a regularisation term in the learning. Negative Correlation[1] (NC) Learning (Liu, 1998), an extension of Rosen's decorrelated networks (Rosen, 1996), is an ensemble learning technique which incorporates such a regularisation term into the backpropagation error function. The regularisation term is meant to quantify the amount of error correlation, so it can be minimised explicitly during training—as such, it is an *explicit* diversity method. In NC the error ϵ_i of network i is:

$$\epsilon_i = \frac{1}{2}(f_i - d)^2 + \lambda p_i \qquad (1)$$

where f_i is the output of the i^{th} network on a single input pattern, d is the target, and λ is a weighting parameter on the penalty function p_i. Strictly, this notation should include input, so $f_i(n)$ and $d(n)$ for the n^{th} input pattern, but we omit this for notational simplicity. The output of the ensemble is a simple average:

$$\bar{f} = \frac{1}{M}\sum_{i=1}^{i=M} f_i \qquad (2)$$

The λ parameter in equation (1) controls a trade-off between objective and penalty functions; with $\lambda = 0$ we have an ensemble with each network training independently of the others, using vanilla backpropagation. NC has a penalty function of the form:

$$p_i = (f_i - \bar{f})\sum_{j \neq i}(f_j - \bar{f}) \qquad (3)$$

where $\bar{f}$ is the output of the whole ensemble of M networks at the previous timestep. Since this penalty is meant to quantify error correlation in some manner, the λ parameter can be seen as managing the balance between *accuracy* and *diversity*; too much emphasis on diversity (large λ) and the networks will sacrifice their accuracy for the sake of just being "different" from one another. NC has seen a number of empirical successes (described in the introduction to this paper), consistently *outperforming* a simple ensemble system, but so far has had very little formal analysis to explain why it works when it does; this leads naturally to our first question.

1.3. Why does the algorithm work?

The MSE of an ensemble system can be decomposed into bias, variance and covariance (Ueda & Nakano, 1996). The strength parameter λ in NC provides a way of controlling the trade-off between these three components: a higher value encourages a decrease in covariance, as has been demonstrated empirically (Liu, 1998). Liu also observes that too high a λ value can cause a rapid increase in the variance component, causing overall error to be higher; no theoretical explanation was given for this behavior, and as such we do not yet have a clear picture of the exact dynamics of the algorithm.

When $\lambda = 1$, we have a special situation; this was described by Liu to show a theoretical justification for NC-Learning. It should be noted that, in the calculation of the derivative, Liu has: *"... made use of the assumption that the output of the ensemble $\bar{f}$ has constant value with respect to f_i"* (Liu, 1998, p.29).

When $\lambda = 1$ we have:

$$\begin{aligned}\epsilon_i &= \frac{1}{2}(f_i - d)^2 + \lambda(f_i - \bar{f})\sum_{j \neq i}(f_j - \bar{f}) \\ \frac{\partial \epsilon_i}{\partial f_i} &= f_i - d + \sum_{j \neq i}(f_j - \bar{f}) \\ &= f_i - d - (f_i - \bar{f}) \\ &= \bar{f} - d\end{aligned}$$

However, although the assumption of constant $\bar{f}$ is used, so is the property that $\sum_{j \neq i}(f_j - \bar{f}) = -(f_i - \bar{f})$,

[1] So-called because it has demonstrated on a number of occasions that it is able to generate estimators with *negatively correlated* errors.

the sum of deviations around a mean is equal to zero; obviously the sum of deviations around a constant does not have this property. Using this apparently contradictory assumption, and the fact that the overall ensemble error function is defined as $\epsilon = \frac{1}{2}(\bar{f} - d)^2$, it was stated:

$$\frac{\partial \epsilon}{\partial f_i} = \frac{1}{M}\Big[\frac{\partial \epsilon_i}{\partial f_i}\Big] \tag{4}$$

showing that the gradient of the individual network error is directly proportional to the gradient of the ensemble error. Though this is obviously a useful property, the justification for the assumption is unclear. The remainder of this work will illustrate the benefits that be gained from removing this assumption. Before we embark on this, it would be useful to first understand a framework into which NC can fit. Can we find a more solid theoretical grounding to NC?

2. Formalising NC-Learning

In this section we show how NC can be related to the ambiguity decomposition (Krogh & Vedelsby, 1995) which showed that the mean-square error of the ensemble estimator is guaranteed to be less than or equal to the average mean-square error of the component estimators. They showed the ensemble error could be broken down into two terms, one of which is dependent on the correlations between network outputs; the exact nature of this result will be given in the next section.

2.1. NC uses the Ambiguity Decomposition

We note that the penalty function, equation (3), can be broken down into a sum of pairwise correlations between the networks. If we remember again that the MSE of an ensemble decomposes into bias plus variance plus covariance (Ueda & Nakano, 1996), then including some measure of correlation to be minimised seems like an intuitive thing to do, first noted by (Rosen, 1996). However this intuition is not enough. We note that the penalty function can be rearranged to:

$$p_i = -(f_i - \bar{f})^2 \tag{5}$$

which is again due to the property that the sum of deviations around a mean is equal to zero. This rearrangement is only possible if we remove Liu's assumption of constant $\bar{f}$. As can be seen, each network minimises its penalty function by moving its output away from the ensemble output, the mean response of all the other networks.

So, why should increasing distance from the mean, or optimising equation (1), necessarily lead to a decrease in ensemble error? An examination of the proof by Krogh and Vedelsby can answer this question, and also raise some new questions on the setting for the λ parameter. Their work showed that the following statement about ensemble error was true:

$$(\bar{f} - d)^2 = \sum_i w_i(f_i - d)^2 - \sum_i w_i(f_i - \bar{f})^2 \tag{6}$$

where w_i is the weighting on the i^{th} network. This says that the *squared error of the ensemble estimator is equal to the weighted average squared error of the individuals, minus a term which measures average correlation.* This allows for non-uniform weights (with the constraint $\sum_i w_i = 1$) so the general form of the ensemble output is $\bar{f} = \sum_i w_i f_i$. This result in equation (6) stems from a number of definitions, one of which is the *ambiguity* of a single member of the ensemble:

$$v_i = (f_i - \bar{f})^2 \tag{7}$$

Remembering that the individual networks in NC-learning minimise the penalty function, and looking at equations (5) and (7) we can see $p_i = -v_i$ and so the networks are in fact maximising this ambiguity term, equation (7). This in turn of course affects the total ensemble error.

To understand this further we take equation (6), multiply through by $\frac{1}{2}$ and rearrange slightly assuming our ensemble is uniformly weighted, we then have:

$$\frac{1}{2}(\bar{f} - d)^2 = \frac{1}{M}\sum_i \left[\frac{1}{2}(f_i - d)^2 - \frac{1}{2}(f_i - \bar{f})^2\right] \tag{8}$$

We see that the MSE of an ensemble can be decomposed into a weighted summation, where the i^{th} term is the backpropagation error function plus the NC-Learning penalty function with the strength parameter set at 0.5.

Now, since we have removed the constraint of assuming constant $\bar{f}$ to allow a link to the ambiguity decomposition, it seems more rigorous to differentiate the network error again without this assumption. What happens in this case? We have a partial derivative:

$$\frac{\partial \epsilon_i}{\partial f_i} = f_i - d - \lambda\Big[2\frac{M-1}{M}(f_i - \bar{f})\Big] \tag{9}$$

where M is the number of networks in the ensemble. Keeping the assumption of constant $\bar{f}$ causes this term $2\frac{M-1}{M}$ to disappear. However, it does seem sensible to retain this, as it takes account of the number of networks. In all of the previous work on NC (Liu, 1998;

Liu & Yao, 1997; McKay & Abbass, 2001), the λ parameter was thought to be problem dependent. Now we understand that it has a deterministic component, this $2\frac{M-1}{M}$. To avoid confusion, from this point on, we shall refer to the λ parameter in the following context, where γ is still a problem-dependent scaling parameter:

$$\lambda = \gamma \left[2\frac{M-1}{M} \right] \qquad (10)$$

In understanding the role of the strength parameter a natural question to ask is, what are the bounds?

2.2. What are the bounds of λ and γ?

Liu stated that the bounds of λ should be $[0, 1]$, based on the following calculation:

$$\begin{aligned} \frac{\partial \epsilon_i}{\partial f_i} &= f_i - d + \lambda \sum_{j \neq i} (f_j - \bar{f}) \\ &= f_i - d - \lambda(f_i - \bar{f}) \\ &= (1-\lambda)(f_i - d) + \lambda(\bar{f} - d) \end{aligned}$$

He states: *"the value of parameter λ lies inside the range $0 \leq \lambda \leq 1$ so that both $(1-\lambda)$ and λ have non-negative values"* (Liu, 1998, p.29). However this justification is questionable, and again here we see the assumption of constant $\bar{f}$ is violated.

We therefore have to ask, why would it be a problem if either $(1-\lambda)$ or λ took negative values? Maybe the bounds of λ should not be $[0, 1]$. How can we determine what the true bounds should be? The NC penalty term is a regularisation term—it 'warps' the error function of the network, making the global minimum easier to find. If, due to this warping, the second derivative of this function becomes negative, then our landscape contains only local maxima or points of inflexion and we have lost any useful gradient information from our original objective function. We have the second partial derivative of ϵ_i with respect to f_i:

$$\frac{\partial^2 \epsilon_i}{\partial f_i^2} = 1 - \lambda(1 - \frac{1}{M})$$

So, we would like the following inequality to hold:

$$0 < 1 - \lambda(1 - \frac{1}{M})$$

Rearranging this we have an upper bound for λ and also γ:

$$\lambda_{upper} = \frac{M}{M-1} \qquad \gamma_{upper} = \frac{M^2}{2(M-1)^2}$$

Figure 1 plots λ_{upper} and the equivalent γ_{upper} for different numbers of networks. We see that as the number of networks increases, λ_{upper} asymptotes to 1, and γ_{upper} to 0.5. It should be noted that with $M \leq 10$ we have λ_{upper} considerably greater than 1.0.

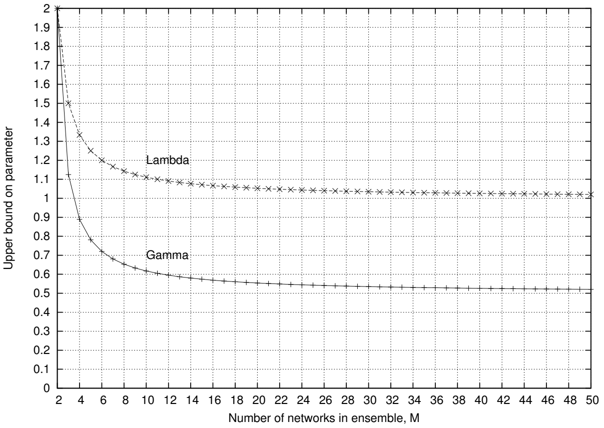

Figure 1. The Upper bound on γ and λ

2.3. An Empirical Study

With our new understanding of the grounding behind NC, we now perform an empirical evaluation, and show that it is critical to consider values for the strength parameter *outside* the originally specified range.

Table 1 shows the classification error rates of two empirical tests, on the Wisconsin breast cancer data from the UCI repository (699 patterns), and the heart disease Statlog dataset (270 patterns). An ensemble consisting of two networks, each with five hidden nodes, was trained for 2000 iterations using NC. We use 5-fold cross-validation, and 40 trials from uniform random weights in $[-0.5, 0.5]$ for each setup; in total 200 trials were conducted for each experimental configuration. It should be noted that with 2 networks, $\gamma = \lambda$. The λ values tested are those considered in the original work on NC: 0.0, 0.5 and 1.0. When λ was set appropriately, results on the heart data showed NC significantly better than a simple ensemble (equivalent to $\lambda = 0$) at $\alpha = 0.05$ on a two-tailed t-test. On the breast cancer data, although the mean was lower, it was not statistically significant.

Figures 2 and 3 show the results of repeating our experiments, but illustrating the full range of the strength parameter. Mean error rate over the 200 trials is plotted, and 95% confidence intervals shown. We see that performance on the breast cancer data *can be improved significantly* by considering the up-

Table 1. Mean classification error rates (200 trials) using NC on two UCI datasets

	$\lambda = 0$	$\lambda = 0.5$	$\lambda = 1$
Breast cancer	0.0408	0.0410	0.0383
Heart disease	0.2022	0.1995	**0.1802**

per bounds beyond those previously specified; on the heart disease data, stable performance is observed beyond $\lambda = 1$.

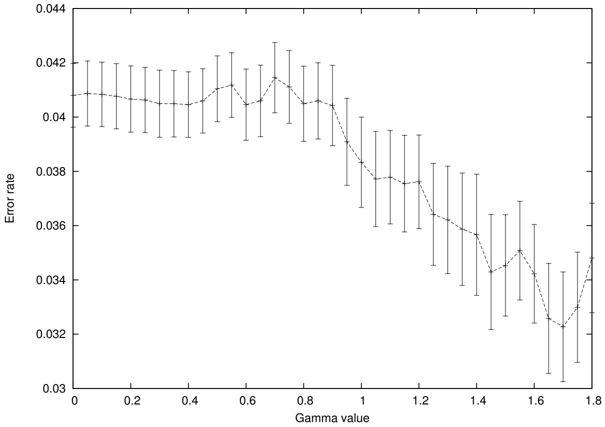

Figure 2. Breast cancer dataset results

As a further measure of comparison, we calculated the percentage reduction in the mean error rate, in relation to when $\lambda = 0$. On the breast cancer data, using $\lambda = 1$ gave a 6% reduction, but using the optimum value at $\lambda = 1.7$ gave a 21% reduction.

We have shown a significant performance improvement by reconsidering the bounds of the strength parameters. It should be noted that, even though the theoretical upper bound is known, in practise it seems error can rise rapidly long before this bound is reached. On the breast cancer data, error became uncontrollable beyond $\lambda = 1.8$, and on the heart disease data at $\lambda = 1.45$; it remains to be seen if it is possible to empirically characterise when this rapid increase will occur.

We know from figure 1 that the upper bound reduces as we add more networks; from this it is reasonable to assume that the optimal value (in the sense of minimising error rate) would follow a similar trend. We sampled the γ value at a resolution of 0.05, and plotted the optimal value found (over 200 trials, as previously described) as we increase the number of networks used for the Breast cancer and Heart disease tasks. Figure 4 shows the optimal parameter decays to 0.5 as we add more networks. But why? What role does γ play?

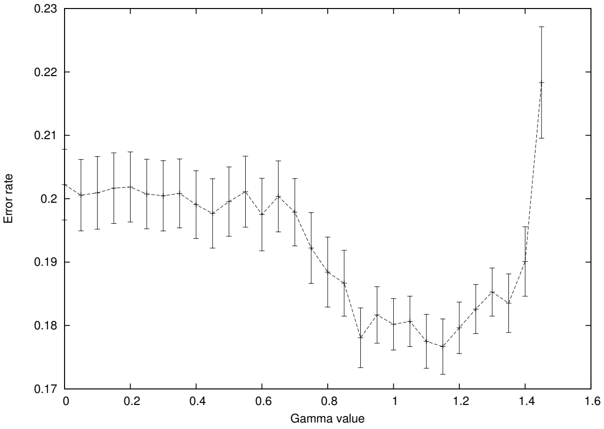

Figure 3. Heart disease dataset results

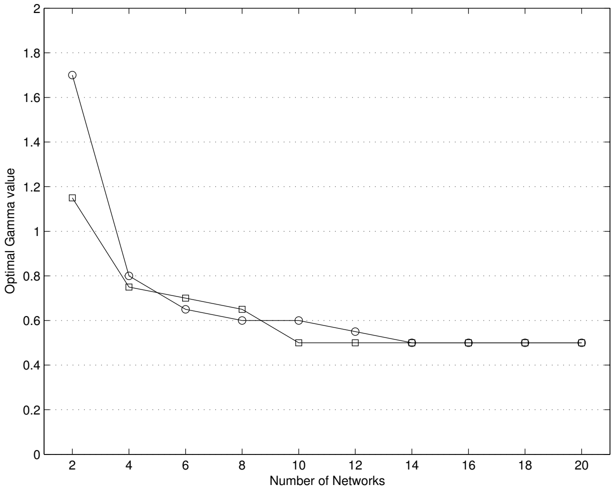

Figure 4. The Optimal γ parameter as we increase the number of networks on the Breast Cancer (circles) and Heart disease (squares) tasks

3. A Statistical Interpretation of NC

In this section we give a statistical interpretation of NC Learning; we use this to form a hypothesis as to why it works and what role the strength parameter plays.

First, as an illustrative scenario, consider a single neural network approximating a sine wave; our network has been supplied with a limited set of datapoints to train on, the inputs chosen randomly at uniform from $[-\pi, \pi]$. Now, consider a single testing datapoint, to find the value of $sin(2)$. The true answer is ~ 0.909, yet we know our network may over- or under-predict that value. The way in which it makes errors will follow a distribution dependent on the random training data sample it received, and also on the random initialisation of the weights. The mean of this distribution is the expectation value $\mathcal{E}_{T,W}\{f\}$, where T and W are the distributions defining these random variables, and f is a network trained with a particular dataset and a particular weight initialisation. Figure 5 illustrates a typical error distribution, with the target value d shown. The four crosses marked are estimates of $sin(2)$ from a hypothetical network; the estimates differ only in that the network was supplied with a different training sample each time.

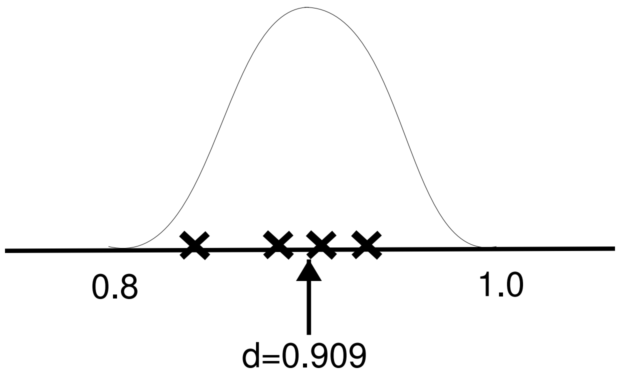

Figure 5. Typical error distribution of an estimator approximating $sin(2)$

From this viewpoint we can immediately see the parallels to an ensemble system. Each member of an ensemble is a realisation of the random variable defined by this distribution over all possible training datasets and weight initialisations. The ensemble output is the weighted average of this set of realisations; all our diversity promoting mechanisms are there to encourage our sample mean $\bar{f}$ to be a closer approximation to $\mathcal{E}_{T,W}\{f\}$. If we have a large ensemble, we have a large sample from this space; consequently with a large sample we can expect that we have a good approximation to the mean, $\mathcal{E}_{T,W}\{f\}$. If we have a smaller ensemble, we cannot expect this: our sample mean may be upward or downward biased. In order to correct this, some methods, such as Bagging, construct our networks from different training datasets, allowing us to sample a more representative portion of the space.

The ambiguity decomposition shows us that each realisation i contributes two components of error to the overall ensemble generalisation error; the first component is $(f_i - d)^2$, and the second is $(f_i - \bar{f})^2$. Normally in an ensemble, each network would be trained with just the first component. NC works well because it explicitly includes this second component of error; in addition it can use a larger γ to over-emphasise the second component when we have a small number of realisations, causing them to spread more widely, giving us a more representative sample from the true distribution of the estimator. This provides a useful interpretation of the role of γ; however, generalisation error is normally expressed in terms of the statistical concepts of bias and variance. How can NC be understood in this context?

Ensembles are well-known as a *variance-reduction* technique. The variance of the ensemble will be lower than the average variance of the components; this can be understood by remembering that with independent random variables X and Y, and a constant weighting term a, we have $Var[aX] + Var[aY] = a^2Var[X + Y]$. If the variables are not independent, an additional covariance term is introduced, illustrating why we strive for lower correlation when constructing an ensemble. The second term on the right handside of equation 6 is the *ensemble ambiguity*; this is maximised when training an ensemble with NC. When $w_i = \frac{1}{M}$ for all i, it can be shown that the expected value of this term is an approximation to the average covariance of the ensemble members:

$$\mathcal{E}\Big\{\frac{1}{M}\sum_i (f_i - \bar{f})^2\Big\} = -\frac{1}{M}\sum_i \sum_{j\neq i} \mathcal{E}\Big\{(f_i - \bar{f})(f_j - \bar{f})\Big\} \tag{11}$$

It is an approximation because with a finite number of networks, $\bar{f} \neq \mathcal{E}\{f\}$, and also because the sum is multiplied by $\frac{1}{M}$ instead of $\frac{1}{M(M-1)}$. We can see that when we are increasing ambiguity, we are reducing this covariance term. When training an ensemble with NC, we use the γ parameter, directly attempting to reduce covariance by over-emphasising this component. A larger γ parameter will be needed when our approximation is not very good: this is will most likely occur when we have a small number of networks, but it could also be due to noise in the training data. It is hoped that with further analysis we will be able to mathematically characterise this, and provide further guidelines for setting the strength parameter.

4. Related Work: The Ambiguity Family

In this section we briefly review some other techniques which have exploited the ambiguity decomposition in some way, either to create or combine a set of predictors.

In the last few years, the ambiguity decomposition has quietly been utilised in almost every aspect of ensemble construction. Krogh and Vedelsby themselves developed an active learning scheme (Krogh & Vedelsby, 1995), based on the method of query by committee, selecting patterns to train on that had a large ambiguity; this showed significant improvements over passive learning in approximating a square wave function.

(Opitz, 1999) selected feature subsets for the ensemble members to train on, using a genetic algorithm with an ambiguity-based fitness function; this showed gains over Bagging and Adaboost on several classification datasets from the UCI repository. A precursor to this work was Opitz and Shavlik's Addemup algorithm (Opitz & Shavlik, 1996), which used the same fitness function to optimise the network topologies composing the ensemble. Interestingly, both these GA-based approaches also used a strength parameter, λ, to vary the emphasis on diversity. The difference between their work and NC is that NC incorporates ambiguity into the backpropagation weight updates, while Addemup trains with standard backpropagation, then selects networks with a good error diversity.

The original ambiguity paper (Krogh & Vedelsby, 1995) also used an estimate of ambiguity to optimise the ensemble combination weights, showing in some cases it is optimal to set a network weight to zero—essentially removing it from the ensemble. In (Carney & Cunningham, 1999) bootstrap resamples of training data are used to estimate ambiguity, in order to approximate the optimal training time; this minimises the overall ensemble generalisation error.

We can see that ambiguity has been utilised in many ways: pattern selection (Krogh & Vedelsby, 1995), feature selection (Opitz, 1999), optimising the topologies (Opitz & Shavlik, 1996) of networks in the ensemble, optimising the combination function (Krogh & Vedelsby, 1995), and also optimising training time (Carney & Cunningham, 1999). NC fits neatly into the gap as the first technique to directly use ambiguity for network weight updates.

In additional, related work, not strictly members of the Ambiguity family, (McKay & Abbass, 2001) analyzed an alternative NC penalty function, named Root Quartic NC Learning. Their method showed better performance than standard NC when using large ensembles, and is in need of a theoretical analysis to explain why that is so. In this paper, NC uses its regularization term on a uniformly weighted combination of the networks. It would be interesting to extend it to a non-uniformly weighted combination, and also to understand the dynamics of a voted combination; AnyBoost (Mason et al., 2000) uses regularization techniques to optimize the convex combination weights in a weighted voting ensemble, and will be the subject of future study.

5. Conclusions

We analyzed an ensemble technique, Negative Correlation (NC) Learning (Liu, 1998), that extended from (Rosen, 1996), and developed in the evolutionary computation literature. We showed a link to the Ambiguity decomposition, and explained the success of NC in terms of sampling statistics and the bias-variance decomposition. This formalisation allowed us to define bounds on the parameters, and provide insights into the optimal settings for these parameters. These observations allow us understand how NC relates to other algorithms, identifying a group of papers spread over the last decade that have all exploited the Ambiguity decomposition for machine learning problems. When taking into account our new understanding of the parameters, significant reductions in error were observed in empirical tests.

Domingos (Domingos, 2000) showed a unified bias-variance decomposition for squared-loss and 0/1-loss functions. The existence of a bias-variance decomposition for 0/1 loss indicates an ambiguity decomposition should exist as well. The properties of this and whether it could be utilised for a 0/1 loss version of NC-Learning will be the subject of future work.

References

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Carney, J., & Cunningham, P. (1999). *Tuning diversity in bagged neural network ensembles* (Technical Report TCD-CS-1999-44). Trinity College Dublin.

Domingos, P. (2000). A unified bias-variance decomposition and its applications. *Proc. 17th International Conf. on Machine Learning* (pp. 231–238). Morgan Kaufmann, San Francisco, CA.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the 13th International Conference on Machine Learning*. Morgan Kaufmann.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *NIPS*, *7*, 231–238.

Liu, Y. (1998). *Negative correlation learning and evolutionary neural network ensembles*. Doctoral dissertation, University College, The University of New South Wales, Australian Defence Force Academy, Canberra, Australia.

Liu, Y., & Yao, X. (1997). Negatively correlated neural networks can produce best ensembles. *Australian Journal of Intelligent Information Processing Systems*, *4*, 176–185.

Mason, L., Baxter, J., Bartlett, P. L., & Frean, M. (2000). *Advances in large margin classifiers*, chapter Functional gradient techniques for combining hypotheses, 221–246. Cambridge, MA: MIT Press.

McKay, R., & Abbass, H. (2001). Analyzing anticorrelation in ensemble learning. *Proceedings of 2001 Conference on Artificial Neural Networks and Expert Systems* (pp. 22–27). Otago, New Zealand.

Opitz, D. (1999). Feature selection for ensembles. *Proceedings of 16th National Conference on Artificial Intelligence (AAAI)* (pp. 379–384).

Opitz, D. W., & Shavlik, J. W. (1996). Generating accurate and diverse members of a neural-network ensemble. *NIPS*, *8*, 535–541.

Oza, N. C., & Tumer, K. (2001). Input decimation ensembles: Decorrelation through dimensionality reduction. *LNCS*, *2096*, 238–247.

Rosen, B. E. (1996). Ensemble learning using decorrelated neural networks. *Connection Science - Special Issue on Combining Artificial Neural Networks: Ensemble Approaches*, *8*, 373–384.

Ueda, N., & Nakano, R. (1996). Generalization error of ensemble estimators. *Proceedings of International Conference on Neural Networks* (pp. 90–95).