BIMODAL DISTRIBUTION REMOVAL

P. Slade & T.D. Gedeon

School of Computer Science and Engineering,
The University of New South Wales, Australia

Abstract:

A number of methods for cleaning up noisy training sets to improve generalisation have been proposed recently. Most of these methods perform well on artificially noisy data, but less well on real world data where it is difficult to distinguish between noisy data points from valid but rare data points.

We propose here a statistically based method which performs well on real world data and also provides a natural stopping criterion to terminate training.

Background:

Some researchers (White, 1989, Geman et al, 1992) have compared neural networks to non-parametric estimators. As such, the limitations of neural networks can be explained by a well understood problem in non-parametric statistics, namely the 'bias and variance' dilemma. Basically, the dilemma is that to obtain a good approximation of an input-output relationship using some form of *estimator*, constraints must be placed on the structure of the estimator and hence introduce bias, or a very large number of examples of the relationship must be used to construct the estimator.

Non-parametric statistics is concerned with model free estimation. When employed for classification both parametric and non-parametric statistics seek to construct decision boundaries between the various classes using a collection of training samples. Non-parametric methods differ from parametric methods in that there is no particular structure assigned to the decision boundaries, *a priori*.

The obvious advantage of parametric techniques is efficiency. By setting the structure of the decision boundary before estimation begins then fewer data points (or training examples) are required. This is because there are (hopefully) a small number of parameters in the parametric model that require estimation. Non-parametric or model free estimation potentially requires the estimation of an infinite number of parameters and hence needs a much larger number of training examples. However, the efficiency of parametric methods comes at a cost. If the actual form of the decision boundary departs substantially from the assumed form, then parametric methods can result only, in the "best" approximation for the decision boundary from within the adopted class of decision boundaries. Non-parametric methods place no restriction on the class in which the decision boundary used in estimation must reside.

Informally, *consistency* is the asymptotic convergence of the estimator to the object of estimation. In this context asymptotic refers to the sample size or the number of patterns in the training set approaching infinity. Most non-parametric algorithms are consistent for any regression function $E[\mathbf{y}|\mathbf{x}]$[2]. Indeed it has been shown (White, 1989, and Gallant and White, 1988) that feed forward networks are consistent, under appropriate conditions relating to the architecture of the network. Consistent in the sense that the weights in the network will, in the limit of training set size approaching infinity, converge to the optimal weights $\mathbf{w}^*$. Although this is an encouraging property, non-parametric methods can be very slow to converge, and this has indeed been observed in the training times for neural networks.

Non-parametric estimators are guaranteed to perform optimally in the limit. In the context of neural networks, they are only guaranteed to outperform other parametric estimators when the size of the training set approaches infinity. For a finite sample, non-parametric estimators can be very sensitive to the actual realisations of $(\mathbf{x},\mathbf{y})$ contained in the sample. This sensitivity results in an estimator that is high in what is known as *variance*. The only way to control this variance is to introduce some *a priori* structure into the estimator, that is to use parametric methods. This approach also has its pitfalls. In complex classification problems, it is difficult to know the structure to impose on the estimator. As mentioned

above, this can result in estimators that converge to an incorrect solution. This creates models that are high in what is known as *bias*. The performance function used in back-propagation can be readily decomposed into a bias and a variance term (Geman et al, 1992).

It is this dilemma between bias and variance that can explain the limitations of non-parametric learning. Low bias and low variance requires large numbers of training examples. In situations where it is not possible to obtain sufficiently large numbers of training examples it is necessary to allow some bias into the neural network training procedure.

A number of methods have been used to introduce bias including:

- Pruning - by the removal of hidden units, the class of functions the network can produce is restricted (eg Sietsma and Dow, 1991, Gedeon and Harris, 1991).
- Dynamic node addition - network training starts with few units and thus high bias (eg Ash, 1989, White, 1989, Harris & Gedeon, 1991).
- Extra terms in performance function - act as smoothing terms decreasing variance. The cost is an increase in bias as details of the object of estimation $E[\mathbf{y}|\mathbf{x}]$ are blurred and lost.
- Cross validation - used to halt training, the network is restricted from building some decision boundaries.
- Outlier removal - reduce the initial variance in the training set and thus improve the variance/bias trade-off (Geman et at, 1992).

In the next section we will examine a number of outlier detection methods before introducing our own.

Outlier detection methods:

In terms of the statistical framework we use, there are several ways that a noisy training set can occur. Either, the input pattern **x** does not obey the environmental probability law ν, or the target pattern **y** does not obey the conditional probability law γ. Both these cases result in an atypical or irregular mapping between input and output. In its quest to identify γ, network training cannot help but be adversely affected by the presence of these errors.

During the back error propagation step of the back-propagation algorithm, each weight is changed by an amount which is a function of the discrepancy between actual network output and desired output. When presented to the network, the erroneous patterns in the training set produce a high disparity between desired and actual output. This produces large weight changes as the network tries to minimise the error on these patterns. As most networks are trained until the mean square error over the training set is below some threshold, these erroneous patterns will prolong training. This growth in training time greatly increases the chance of the network overfitting the training set. So these inaccurate patterns seem to have two possible effects on training:

- they force the network to slow its learning of the majority of patterns in order to learn those few erroneous patterns, and
- they cause training time to escalate, thereby increasing the effect of overfitting.

A simple way of lessening the effect of these incorrect patterns is to remove them altogether. This approach, though rudimentary in theory, is complicated in practice. Joines and White (1992) identify a number of approaches, all of which have some problems:

<u>Absolute Criterion Method</u>

This method attempts to minimise the absolute value of the error as opposed to normal back-propagation which attempts to minimise the mean square error. The outliers in the training set will have larger errors relative to the rest of the training set. This method does not propagate these large errors by design, consequently the changes in weight are smaller. This method may allow backpropagation to find simple models. The obvious problem with the Absolute Criterion Model is of stability or oscillation. Being parabolic, very small weight changes are needed to minimise the error function for normal back propagation, when the error is near zero.

This method propagates the same error throughout training, resulting in relatively large weight changes, which prevent the network from reaching a stable minimum. It is possible to revert to normal back-

propagation error function in the region of small error, to alleviate the problem of oscillation, since the normal backpropagation function is very smooth around the origin.

Least Median Squares (LMS)

This method seeks to minimise the median of the residual errors. The procedure involves calculating the mean square error for each pattern in the training set then back-propagating the median of those errors.

This method is similar to batch learning in normal back-propagation and shows the same slowness in convergence.

Least Trimmed Squares (LTS)

This method is designed to speed up convergence of normal back-propagation training. The basic premise is to minimise the mean square error over only a percentage of the training set. At every 5th epoch all the patterns in the training set are sorted in ascending order based on their mean square error. The patterns connected with the lowest mean square errors are used to train the network for the next 5 epochs. The process is then repeated and a new subset of the full training set is selected as training patterns. This method may result in better generalisation because the outliers in the training set will never be used to train the network, because they produce large mean square errors. As a result, the weights will never adjusted to fit these outliers.

Joines and White (1992) tested the above method on a clean data set with noise specifically introduced into the training set (but not the test set). The result was excellent, with the mean square error over the training set increasing and decreasing over the test set - the reverse of the usual case. There are some major problems. Firstly in the real world our test set would also contain noise. Secondly the details of the method requires knowing (or assuming) *a priori* how many outliers are present in the training set.

Bimodal distribution removal:

This section introduces our method for outlier detection called *Bimodal Distribution Removal* (BDR), which addresses all the weaknesses of the other methods indicated in the previous section.

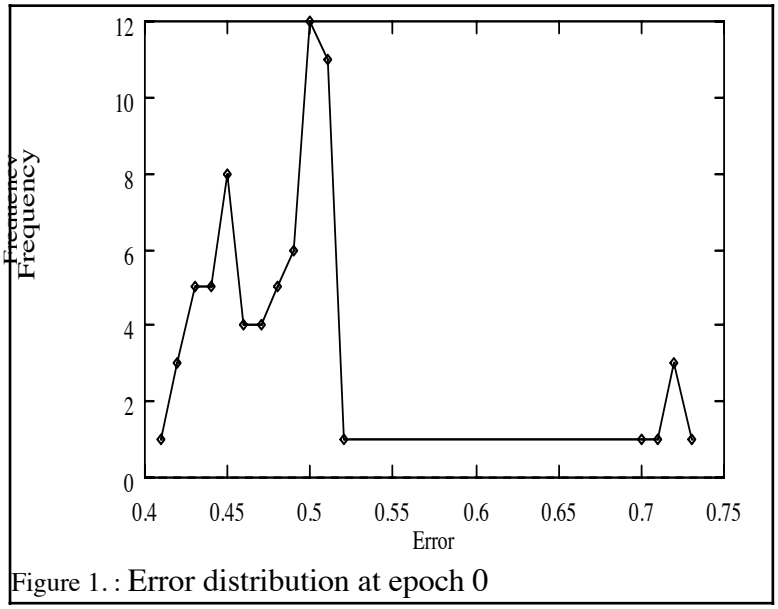

Figure 1. : Error distribution at epoch 0

In order to investigate the behaviour of outliers in the training set during training, frequency distributions of the errors for all patterns in the training set were produced every 50 epochs of training. shows that very early in training (0-100 epochs) the distribution of the pattern errors is approximately normal with a large variance. Very quickly however, the network dramatically reduces the errors for a majority of the training set. However, as evident from Figure 2 there remains patterns with relatively high error. This creates an almost bimodal error distribution, with the low error peak containing patterns the network has learnt well, and the high error peak containing the outliers. Figure 2 also reveals that the distribution is only approximately bimodal. There exist patterns with errors which are between the two peaks. These patterns are being slowly learnt by the network and shall hereafter be referred to as *slow coaches*.

From the two peaks in the error distribution it is clear that the network can identify outliers itself. The network is learning $E[\mathbf{y}|\mathbf{x}]$, the expected value of **y** given **x**. The patterns that appear in the high error peak are outliers in the sense that they are *not* what the network 'expects' **y** to be given **x**.

It would be difficult and time consuming to identify a bimodal error distribution during training.

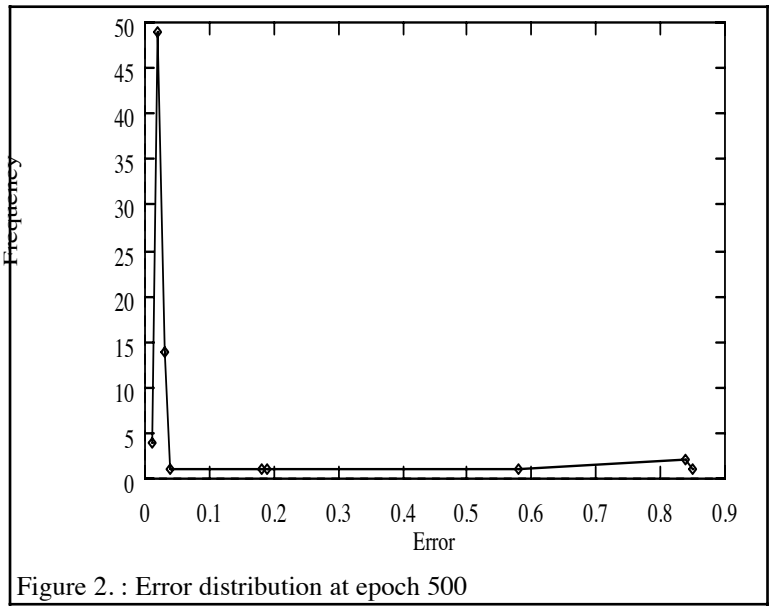

Figure 2. : Error distribution at epoch 500

Fortunately, a measure of the variance will achieve the same effect. As mentioned above, the variance υ_{ts}, of the error distribution is quite large very early in training. As the network begins to learn the majority of the patterns, υ_{ts} drops sharply (see Figure 3). This is due to the lower error peak dominating the distribution. Low variance ($\upsilon_{ts} \approx 0.1$ and below) indicates that the two error peaks have formed, so the removal of outliers can begin. All this usually occurs very quickly, normally within 200-500 epochs.The next step is to decide which patterns to remove. Patterns should not be removed too quickly, as those patterns with midrange errors could eventually be learnt by the network. In choosing which patterns to remove the first step is to calculate the mean of the errors for all the patterns in the training set $\overline{\delta}_{ts}$. Due to the dominance of the low error peak (Figure 2) $\overline{\delta}_{ts}$ will be very low, but greater than nearly all errors in the low error peak (due to the presence of the high error peak). Those patterns in the low error peak are not outlier candidates. In order to isolate potential outliers, all those patterns with error greater than $\overline{\delta}_{ts}$ are taken from the training set.

This subset will contain the patterns from the high error peak (outliers) and the slow coaches between the two peaks. The dominance of the outliers in the subset will skew the distribution towards the outliers. The mean $\overline{\delta}_{ss}$ and standard deviation σ_{ss} of this skewed distribution is calculated. $\overline{\delta}_{ss}$ will be heavily influenced by the outliers and hence will be relatively high. From these two statistics is possible to decides which patterns to permanently remove from the training set. Those patterns with

$$\text{error} \geq \overline{\delta}_{ss} + \alpha\sigma_{ss} \qquad \text{where } 0 \leq \alpha \leq 1$$

are removed. BDR is intentionally conservative in its removal of patterns to give the network opportunity to learn the slow coaches. It is repeated every 50 epochs in order for the network to learn the features of each new training set.

Should BDR be continued indefinitely, eventually all the patterns would be removed from the training set. This is undesirable because as the training set becomes smaller the network is devoting 50 epochs of training to a reduced set of examples, thus potentially dramatically increasing the overfitting effect. Removal of the outliers from the training set causes the high error peak to shrink, resulting in a lower $\overline{\delta}_{ts}$ and a very much lower υ_{ts}. It is υ_{ts} that can be used as a halting condition for training. Once υ_{ts} is below a constant (typically 0.01) training is halted.

BDR attempts to address all the weaknesses of the outlier detection and removal methods discussed in the previous section in that:

- pattern removal does not start until the network itself has identified the outliers,
- the number of patterns removed is not hard wired, but instead is data driven,
- patterns are removed slowly, to give the network ample time to extract information from them, and
- a halting criterion naturally evolves preventing overfitting - results in significantly faster training time.

The remainder of this paper describes an experiment to gauge the effectiveness of BDR and some of the outlier detection methods presented in the previous section. The methods will be compared to normal back propagation, using a real world data set. The effectiveness of each method will be measured by the method's ability to control both bias and variance.

Experiment:

The data used in this experiment comprises student information, assessment and final mark for a

sample of students from a first year Computer Science subject at The University of New South Wales.

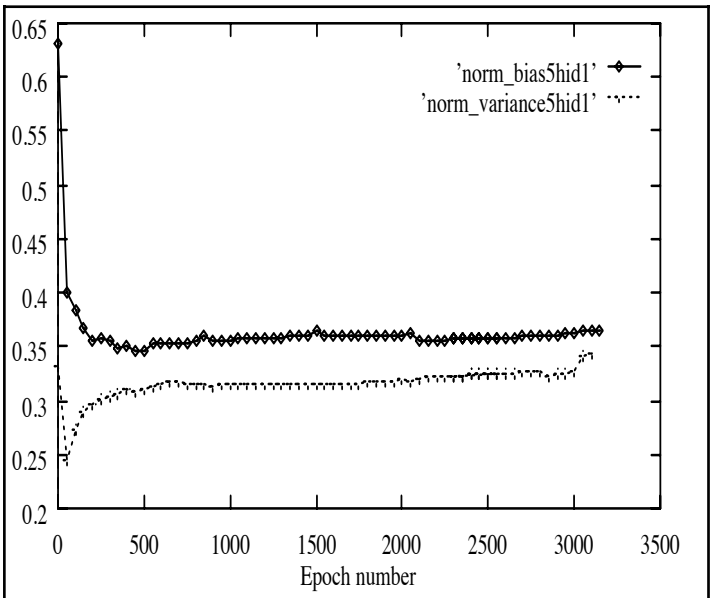

Figure 3. : Normal back-propagation

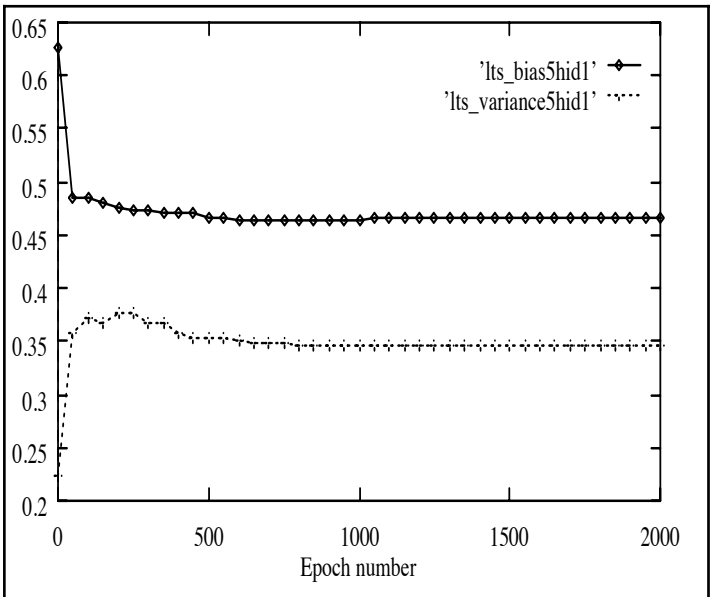

Figure 4. : Least Trimmed Squares (LTS)

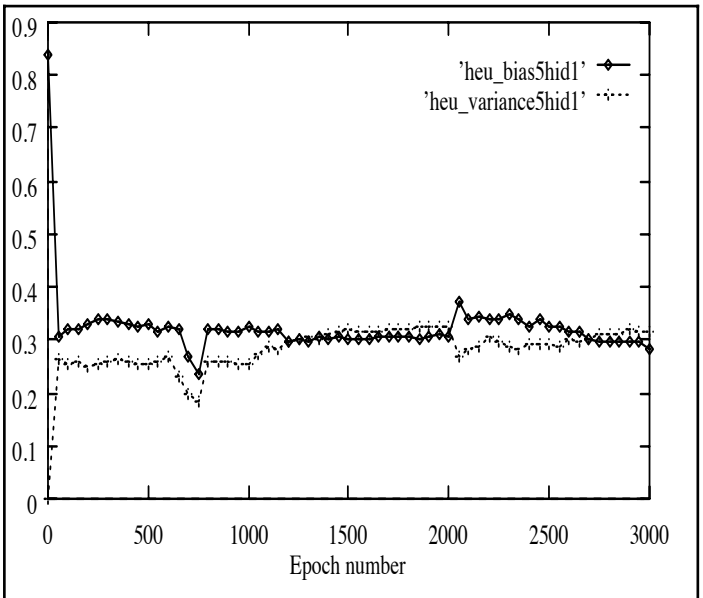

Figure 5. : Heuristic Reduction

The sample contains the marks for 150 students. The exam mark is excluded to introduce noise, the task of the network is to predict the final grade (HD, D, CR, P) based on the 40% of the mark which comes from assessment prior to the exam. For further details see Gedeon and Bowden (1992). A student's assessment may not reflect their grade at one extreme by copying assignments and so on to get a good mark during the year but not understand the material and hence do very badly in the final exam, or at the other extreme expend very little effort during the year and achieve low course assessment marks but study very hard for the final exam and do well. An analysis of the data indicates that these two cases are relatively rare, of the order of 10%, and can be classed as noise.

Normal backpropagation is usually classified as a high variance / low bias estimator. As such, normal backpropagation will control variance if the network does not begin overfitting the training set. Overfitting can happen if as in this experiment, the training set is noisy.

The network structure is 14-5-4, being input, hidden and output units. The experiment was repeated with a 14-10-4 network with no significant difference in the results.

Fifty patterns were set aside as a test set and were never used in training. The remaining 100 patterns were used to create 50 sets of 70 pattern training sets at random. Fifty networks for each of the Absolute Criterion, LMS, LTS, Heuristic Reduction (described in Gedeon and Bowden, 1992), and Bimodal Distribution Removal were trained, as well as normal back-propagation. The integrated bias and variance were then calculated. The results for the latter 4 cases are shown. The Absolute Criterion and LMS methods performed less well than the LTS method and are not shown.

Least Trimmed Squares provides the necessary control of variance at the expense of higher bias. This control of variance becomes significant if training is continued for a long period of time (the overfitting effect increases). The LTS method to control variance is hard wired and requires *a priori* knowledge of the amount of noise in the training set.

Heuristic Pattern Removal produces an almost contradictory result. The asymptotic nature of neural networks indicates that network performance becomes optimal as the size of the training set approaches infinity. Yet, measurements of bias and variance for training on a half size training set show the Heuristic

method performs as well as the Bimodal Distribution Removal method. Bias and variance are very sensitive to the complexity of the data and by how much the training set is reduced every 1,000 epoch. This can be seen by the slope of the variance plot in Figure 5 - the Heuristic method leads to the most uncontrolled increase in variance of all the methods. The problem remains in determining the 'correct' time to halt training.

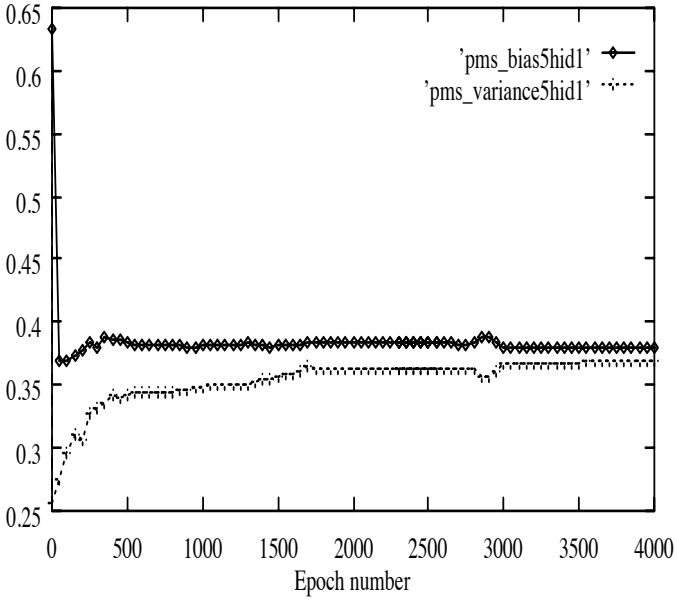

Figure 6. : Bimodal Distribution Removal

Bimodal Distribution Removal provides a similar control of variance as LTS. It is an improvement over LTS since both bias and variance are lower during training, and a data driven halting condition results.

The values for bias in the LTS and BDR methods for this data set in comparison to normal back-propagation indicates that even theugh these methods perform well, the noisy data points are being useful in this case. This means our implicit assumption that the probability law γ is approximately degenerate is barely valid. This points to the requirement for appropriate choice of a data set. An independant statistical analysis of the data was commissioned, and found there was little correlation between assessment data and final grade. Of course, the choice of data for the purpose of demonstrating a new method such 'difficult' data is ideal.

Conclusion:

The choice of training method depends on goal of the user. If a training set is known to be very noisy then an outlier detection method should be employed. However if the training examples are 'clean' and large in number, normal backpropagation will best approximate the regression $E[\mathbf{y}|\mathbf{x}]$. If the test set is known to be clean, then the LTS method can be used.

The BDR method has been shown to perform as well as existing methods including on 'real' noisy data, with none of the disadvantages. The method can improve generalisation by removing sources of noise, speed up training by reducing the number of patterns, and provides a natural stopping criterion to terminate training.

References:

Ash, T, "Dynamic node creation in back-propagation networks," TechRep, Univ. of Calif., 1989.

Gallant, AR and White, H, *A Unified Theory of Estimation and Inference for Nonlinear Dynamic Models*, Basil Blackwell, Oxford, 1988.

Gedeon, TD and T.G. Bowden, TG, "Heuristic Pattern Reduction," *IJCNN*, Beijing, 1992.

Gedeon, TD and Harris, D, "Network Reduction Techniques," *Proc. Int. Conf. on Neural Networks Methodologies and Applications*, San Diego, vol. 2, pp. 25-34, 1991.

Geman, S, Bienenstock, E and Doursat, R, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, pp. 1-58, 1992.

Harris, D and Gedeon, TD, "Adaptive insertion of units in feed-forward neural networks," *4th Int. Conf. on Neural Networks and their Applications*, Nîmes, 1991.

Joines, M and White, M, "Improving generalisation by using robust cost functions," *IJCNN*, vol. 3, pp. 911-918, Baltimore, 1992.

Sietsma, J and Dow, RF, "Creating Artificial Neural Networks That Generalize," *Neural Networks*, vol. 4, pp. 67-79, 1991.

White, H, "Learning in artificial neural networks: A statistical perspective," *Neural Computation*, vol 1., pp. 425-464, 1989.