HEURISTIC PATTERN REDUCTION

T.D. Gedeon and **T.G. Bowden**
School of Computer Science and Engineering,
The University of New South Wales,
P.O. Box 1, Kensington, 2033, AUSTRALIA

ABSTRACT

It has been estimated that some 70% of applications of neural networks use some variant of the multi-layer feed-forward network trained using back-propagation. The use of such networks has a number of problems, including the speed of training, and the avoidance of local minima. Here we report on a series of experiments to test the hypothesis that a reduction in the complexity of a training set can improve learning. We have found that a simple heuristic method of reduction of the size of a training set can produce a trained network with improved performance on the validation test set.

ASSUMPTIONS

In this paper we will assume a multi-layer feed-forward network trained using back-propagation, and will use the general expression “neural network” to mean such a network. All connections are from units in one level to units in the next level, with no lateral, backward or multi-layer connections. Each unit is connected to each unit in the preceding layer by a simple weighted link. The network is trained using a training set of input patterns with desired outputs, using the back-propagation of error measures. The network is tested using a validation set of patterns which are never seen by the network during training and thus can provide a good measure of the generalisation capabilities of the network. The separation of the total set of patterns into training and test sets is generally at random to avoid introducing experimenter bias.

By back-propagation we mean the general concept of developing the error gradient with respect to the weights, and not restricted to the original gradient descent method. In the examples we use here, we have used the basic sigmoidal logistic activation function $y = (1 + e^{-1})^{-1}$, though this is not germane to the substance of our results.

INTRODUCTION

A number of contributions in the domain of pruning of neural networks have shown that a reduction in network size to

COMP1111 More Computing Sorted on student ID 18 May 92 10:34:12 Page 1

Regno	Crse/Prog	S	ES	Tutgroup	lab2	tutass	lab4	h1	h2	lab7	p1	f1	mid	lab10	final
					3	5	3	20	20	3	20	20	45	3	100
.															
.															
0275000	3400	1	F	T10-yh	2.5	3	3	18	4.5	3	14	18.5	24	2.5	68
0275105	3420	1	F	T9-ko	3	4	2.5	17	17	3	5	14	10	2.4	56
0275139	3420	1	F	T4-ko	0	5	2.5	18	17	3	6	10	28	2.4	57
0275164	3400	1	F	T2-no	.	3	1.5	8.5	.	1.5	.	.	10.2	2.4	44
0275279	3420	1	F	T2-no	3	3	.	19	18	2	5.5	4	20	2.4	60
0275282	3400	1	F	T4-ko	2.5	3	3	19	.	3	.	10	16	2.4	51
0275298	3400	1	F	T9-ko	3	5	2.5	17	18	3	8.5	18	21	2.4	61
0275315	3420	1	F	T10-yh	2	3	0.5	14	.	1	.	.	7	2	26
0275567	3400	1	F	T10-yh	.	3.5	2.5	19.5	.	2.5	.	.	11.5	.	36
.															
.															

Table 1: Raw data

some minimal size can improve the generalisation capabilities of the neural network. The seminal work on pruning by inspection was by Sietsma and Dow (1991); more recently Gedeon and Harris (1991a) introduce an automatable method called distinctiveness analysis for network size reduction, and include a survey of other work in this area.

Kruschke (1989) has shown that a reduction in the dimensionality of the space spanned by the hidden unit weight vectors (without reducing the number of hidden units) also improves the generalisation capabilities of a neural network.

The use of validation sets to stop training before generalisation degrades is now well established (eg, Morgan and Boulard, 1990).

The commonality between the above methods is of course the introduction of some constraints which limit the resources available to the network and force it to generalise rather than learn the specific patterns presented. This sets the scene for our hypothesis that a reduction in the number of training patterns may be usable as a resource limiting constraint in a similar fashion to improve generalisation.

A suggestion that more is not better when it comes to the number of training patterns can be found in Chauvin (1990), citing the Runge effect where an overfitting error can increase with the size of the sampling data set.

Finally, a cautionary note, in that there are dangers in reducing the size of the training set. In the realm of reducing network size, Gedeon and Harris (1991b) have discussed the relationship between the minimal size of a neural network and the (lack of) robustness of the solution. Thus, while the use of a validation set to determine when to stop training is in general a useful idea, it must be noted that it is essential to use a validation set when we reduce the number of training patterns. This is because the time it would take to overtrain the network (number of epochs of training) is also reduced in parallel with any speeding up of the learning done.

APPLICATION DOMAIN

The experiments were performed on a sample of 153 patterns taken from the class results of an undergraduate Computer Science subject COMP1111 at the University of New South Wales.

The raw data consisted of the results from a number of laboratory exercises, assignments and a mid-term quiz all of which compose 40% of a student's mark for the subject. The exam mark which comprises the remaining 60% has been omitted, and the final aggregate mark is provided.
The goal of the exercise is to predict the final mark based on the partial marks. The educational imperative for such prediction is to be able to provide for students a reliable prediction of their final mark based on their current performance. This will of course be expected to invalidate the prediction in that students with low predicted final marks will take extra steps to improve their preformance. A sample of the raw data is shown in Table 1.

The raw data has been pre-processed to produce values in the range from 0 to 1, with adaptive banding being used rather than a simple mapping onto the range. Thus, the column for lab10 in the raw data in Table 1, becomes the fifth column from the right in Table 2 with the values from 2.5 and above being distinguished from the most common value of 2.4, and so on. The final mark is banded into the ranges for a Distinction, Credit, Pass or Fail and represented by a 1 in the appropriate output category.

```
   .
   .
p0275000 0.35 1.00 1.00 0.25 0.70 0.40 1.00 0.80 0.60 1.00 0.70 1.00 0.53 1.00 0 1 0 0
p0275105 0.35 1.00 1.00 1.00 1.00 0.70 0.70 0.40 1.00 1.00 0.30 0.70 0.22 0.50 0 0 1 0
p0275139 0.35 1.00 1.00 1.00 0.00 1.00 0.70 0.80 1.00 1.00 0.30 0.70 0.62 0.50 0 0 1 0
p0275164 0.35 1.00 1.00 0.50 0.00 0.40 0.00 0.20 0.00 0.00 0.00 0.00 0.23 0.50 0 0 0 1
p0275279 0.35 1.00 1.00 0.50 1.00 0.40 0.00 0.80 1.00 0.40 0.30 0.30 0.44 0.50 0 0 1 0
p0275282 0.35 1.00 1.00 1.00 0.70 0.40 1.00 0.80 0.00 1.00 0.00 0.70 0.36 0.50 0 0 1 0
p0275298 0.35 1.00 1.00 1.00 1.00 1.00 0.70 0.40 1.00 1.00 0.30 1.00 0.47 0.50 0 0 1 0
p0275315 0.35 1.00 1.00 0.25 0.40 0.40 0.00 0.40 0.00 0.00 0.00 0.00 0.16 0.00 0 0 0 1
p0275567 0.35 1.00 1.00 0.25 0.00 0.70 0.70 0.80 0.00 0.70 0.00 0.00 0.26 0.00 0 0 0 1
   .
   .
Table 2: Processed data
```

The actual pre-processing used does not affect our results, and are provided in Table 2 in the interests of completeness.

RESULTS

The network architecture used has 14 input units corresponding to the features of the processed student mark data, 5 units in the hidden layer, and 4 output units. This network size produces acceptable predictions of the final mark. As mentioned earlier, what we mean by acceptable in this case is being reported elsewhere.

The original set of 153 patterns is divided at random into 53 patterns to form a validation test set which will never be seen by the network during training. The remaining 100 patterns form the starting training set, which is later reduced in size. For each run, the same set of 53 patterns is used to validate the network trained by both the full size training set and the reduced size training sets. In each run, the same starting weights are used when training with the different sized training sets. This is to minimise the effects of the initial random functionality of the network unit weights.

The network was inititally trained for 1,000 epochs on the full 100 pattern training set. The individual training patterns were mapped onto a one-dimensional adjacency measure using their contribution to the total sum of squares. This adjacency was used as the base to make arbitrary assumtions as to sizes of clusters of training patterns. Thus, assuming an homogenous distribution of clusters of pairs of patterns, the training set can be reduced to half its size.

Clearly, this rough heuristic is unlikely to hold in general, nevertheless the results were encouraging as shown below.

We have performed 15 runs of each configuration, with different initial weights. A sample of the results are displayed in Table 3, with the prediction accuracy being represented by the total sum of squares (tss) value, the lower tss the better the prediction.

	-sec6-	-sec7-	-sec8-	-sec9-	-sec10-
-cont6-	37.980	28.719	34.488	35.549	40.820
-cont5-	30.417	46.022	34.858	36.695	43.113
-cont4-	44.050	49.013	49.568	32.536	36.981
-cont3-	42.258	52.686	43.832	46.727	42.592
-cont2-	37.216	48.361	36.547	35.132	43.892
-cont1-	36.713	41.121	55.530	40.834	29.800

Table 3: Prediction accuracy in terms of tss for an example set of 5 runs

Subsequent lines in Table 3 are the tss values for a series of smaller training set sizes. All of the training sets are different reductions of the original 100 pattern training set, and have been trained for 1,000 epochs. The ad-hoc nature of the reduction used accounts for the inconsistently varying results for smaller pattern set sizes. The success of the method is shown by the low error rates observed at the bottom of the table. Table 4 summarises the results for the experiments.

Note that the values for the total sum of squares (tss) are the minima of the number of runs done and can be from different runs. Given that we have used a simplistic method to reduce the number of patterns we did not expect a consistent improvement, nor the degree of improvement possible. The reduction in tss value and the improvement in prediction when half of the pattern were removed is significant. Further, the result for 25 patterns is actually better than or equal to the majority of results found by the 5 networks displayed in Table 3 on the full 100 training patterns.

Number of Patterns	Tss value
100	28.719
80	27.379
67	30.477
50	24.777
33	32.903
25	34.688

Table 4: Minimum tss values for numbers of patterns

Finally, it must be noted that a statistical evaluation of the same data by an independent statistical consulting group reported that there was insufficient data present for reliable statistical predictions to be made. Nevertheless, one of the networks we have produced performs very well indeed. The difference between the best statistical result and that from the best neural network prediction we have produced is being reported elsewhere.

TESTING

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, Bimodal Distribution Removal (Slade and Gedeon, 1993), and Heuristic Reduction (Gedeon and Bowden, 1992) were trained, as well as normal back-propagation. The integrated bias and variance were then calculated. The results for the latter 2 cases are shown here.

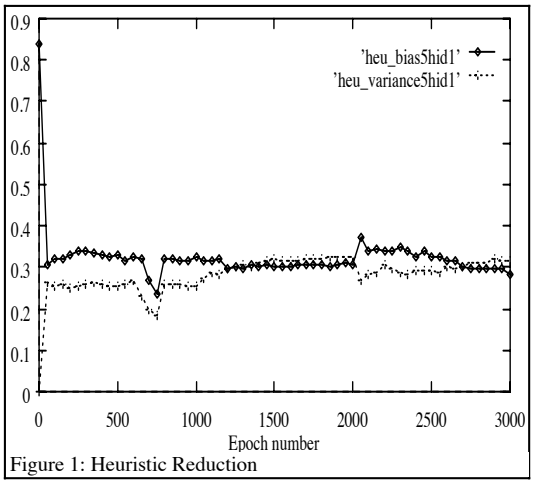

Figure 1: Heuristic Reduction

Heuristic Pattern Removal produces an interesting 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 1.

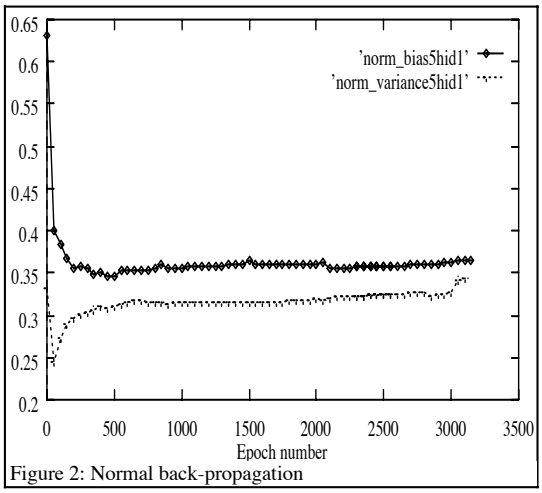

Figure 2: Normal back-propagation

CONCLUSION

We have shown that a simple heuristic method can be used to reduce the size of the training pattern set considerably, with an improvement of performance on the validation set. This improvement is most likely due to the simplification of the error surface in pattern space traversed by the network as it attempts to locate the minimum. That the minima found after simplification can be better than those found with the original pattern set indicates that none of the significant features of the original pattern set have been lost. A reduction in the number of training patterns also has possibilities in speeding up the training of feed-forward networks as the time taken to learn is related to the number of patterns used during training.

REFERENCES

Chauvin, H "Dynamic Behaviour of Constrained Back-Propagation Networks," *Proc. NIPS-2*, pp. 642-649, 1990.

Gedeon, TD & T.G. Bowden, TG, “Heuristic Pattern Reduction,” *Int. Joint Conf. on Neural Networks*, Beijing, vol. 2, pp. 449-453, 1992.

Gedeon, TD & Harris, D, “Network Reduction Techniques,” *Proc. Int. Conf. on Neural Networks Methodologies and*

Applications, AMSE, San Diego, vol. 2, pp. 25-34, 1991a.

Gedeon, TD & Harris, D, “Creating Robust Networks,” *Int. Joint Conf. on Neural Networks*, Singapore, vol. 3, pp. 2553-2557, 1991b.

Kruschke, JK, “Improving generalization in back-propagation networks with distributed bottlenecks,” *Int. Joint Conf. on Neural Networks*, vol. 1, pp. 443-447, 1989.

Morgan, N & Boulard, H "Generalisation and Parameter Estimation in Feedforward Nets: Some Experiments," *Proc. NIPS-2*, pp. 630-637, 1990.

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

Slade, P & Gedeon, TD “Bimodal Distribution Removal,” *Proc. IWANN Int. Conf. on Neural Networks*, Barcelona, 1993.