Backpropagation Can't Do Parity Generalisation

Chris Thornton

Cognitive and Computing Sciences

University of Sussex

Brighton

BN1 9QN

UK

Email: Chris.Thornton@cogs.susx.ac.uk

WWW: http://www.cogs.susx.ac.uk

Tel: (44)1273 678856

November 25, 1996

Abstract

It is accepted that the early connectionist learning methods such as the perceptron algorithm cannot solve parity learning problems. But since the early 1980s, there have been many demonstrations purporting to show that the backpropagation method *can* do so. However these demonstrations are misleading. Backpropagation in fact reliably *fails* to solve parity problems when they are posed as genuine, supervised learning problems, i.e., as problems involving generalisation. Thus backpropagation is subject to some of the same limitations as the perceptron method.

Keyworks: Learning, Theoretical limitations

1 Backpropagation performance on parity generalisation

The parity problem is one of the best established of all benchmarks for neural-network learning methods. In a parity problem we have a number of boolean input variables and one boolean output variable. The input/output rule states that the output is true iff an *odd* number of input values are true. If there are just two input variables the problem is known as 'Exclusive-OR' (XOR) since it is effectively the rule that either of the inputs can be true, but not both. The full mapping for the 3-bit parity problem can be written as a training set (using 1=true, 0=false) as follows.

x_1	x_2	x_3		y_1
1	1	1	$\Longrightarrow$	1
1	1	0	$\Longrightarrow$	0
1	0	1	$\Longrightarrow$	0
1	0	0	$\Longrightarrow$	1
0	1	1	$\Longrightarrow$	0
0	1	0	$\Longrightarrow$	1
0	0	1	$\Longrightarrow$	1
0	0	0	$\Longrightarrow$	0

The parity rule turns out to be surprisingly hard to learn. Learning procedures such as the perceptron learning algorithm are known to be *incapable* of acquiring parity mappings [1]. But even state-of-the-art symbolic methods such as C4.5 [2] and backpropagation [3] generalise poorly from incomplete parity mappings. With 4-bit parity, 16 minimally incomplete training sets (i.e., training sets which contain all but one of the possible cases) can be constructed. C4.5 actually generalises incorrectly in *all 16 cases*; i.e., it always 'gets the answer wrong'.

Backpropagation performs no better. In an extensive empirical analysis, backpropagation was tested for its ability to generalise to one, randomly selected unseen case in the 4-bit parity mapping. In this analysis a standard, two-layer, (strictly) feed-forward network was used with the number of hidden units being varied between 3 and 80. Data were collected for 20 successful runs (i.e., achievement of negligible error on the training data) with each architecture. The learning rate was 0.2 and the momentum value was 0.9.

The results are summarised in Figure 1. This shows the post-training mean error for seens and unseens averaged over the 20 successful training runs which were performed in each architecture. The basic error value used here is simply the average difference between the target output and actual output produced. The graph shows negligible mean error for seen cases due to the fact that data were only recorded for successful runs. More interestingly, it shows that the mean error on the unseen case is very poor for all architectures used, i.e., no generalisation is achieved. (The reason why the generalisation error is so much *worse* than chance is explained below.)

2 Performance on related problems

Generalisation failures on parity mappings are sometimes dismissed on the grounds that the parity problem is an artificial construct upon which learning methods cannot be expected to perform properly. To show that this is not the case we need to demonstrate that backpropagation fails on other problems as well as parity. The key property of the parity mapping is that, in its full form, it is statistically neutral. That is to say, the conditional probability of

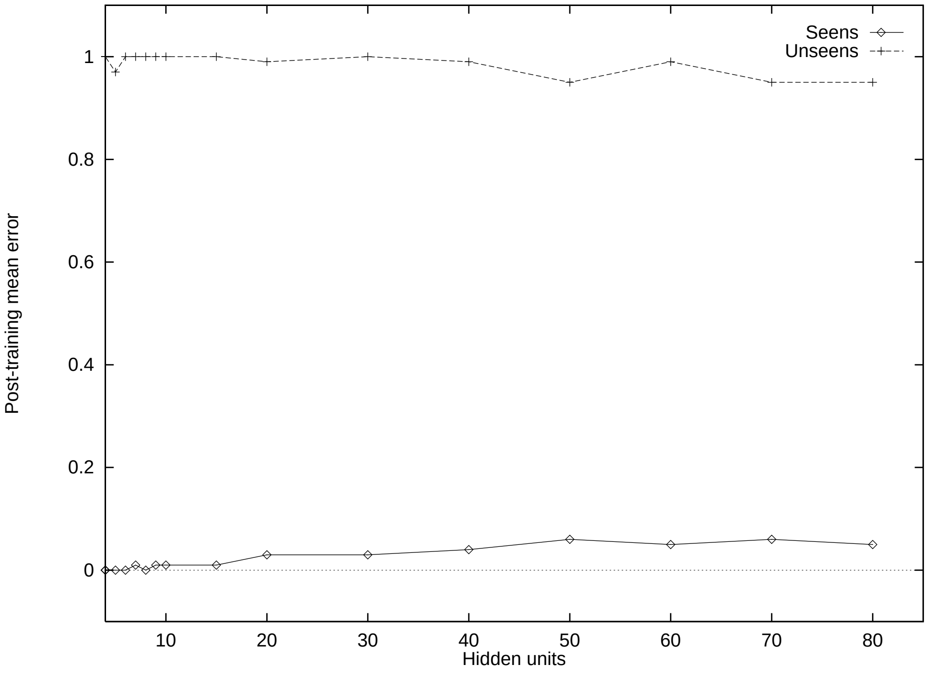

Figure 1: Post-training mean-error curves for parity generalisation.

seeing a particular output value given the presence of a particular input value always has the chance value of 0.5. It is this property which makes the parity problem so hard to learn. No associations exist between particular inputs and particular outputs. A natural hypothesis is that backpropagation will perform badly on any problem which exhibits statistical neutrality. As it turns out, empirical experiments tend to confirm this.

A fertile source of statistically neutral mappings is the 'modulus-addition' learning problem, i.e., a mapping in which the output value in each pair is always the modulus sum of the input values. It turns out that any such problem is guaranteed to be statistically neutral provided that the number of possible values for any given input variable is equal to, or a multiple of the number of possible output values.

If the training set for some learning problem is statistically neutral then all the conditional output probabilities must be at their chance levels. This tells us that in the training set we will see each value of an input variable X appearing with each output value an equal number of times (i.e., that the number of possible values of X must be equal to, or a multiple of the number of possible outputs). We could therefore map the N values of our input variable X onto integers in the range $0 ... N-1$ and the M output values onto integers in the range $0 ... M-1$. Moreover, since the only constraint is that each input value must associate with each output value an equal number of times, we could do this in such a way as to ensure that the output value is always the modulus to base M of the value of X.

If we do this to each input variable in turn, using a fixed mapping of the output values, we end up with a purely numeric version of the training set. Then, 'incrementing' the integer value of any variable (i.e., switching attention to a case in the training set showing the next highest value of the variable) always has the effect of 'incrementing' the output value. The training set therefore instantiates a modulus-addition rule. The general conclusion is that any problem which can be translated into a modulus-addition problem (such that the cardinality of the set of output values is a factor of the cardinalities of all the input-value sets) is guaranteed to be statistically neutral.

3 Dealing with neutrality in non-parity problems

Consider the 'likelihood problem' whose target mapping is shown below. This is a straightforward learning problem with a relatively obvious input/output rule. However, we can translate it into a modulus-addition problem with the following substitutions: person/0, computer/1, consumes/0, dislikes/1, heat/0, electricity/1, moisture/2, silicon/3, yes/0, no/1. Under this translation, the requirement that the input-set cardinalities are equal to, or a multiple of the output-set cardinality is met so we know that the problem is *necessarily* statistically neutral. It is in fact easy to confirm that every single conditional output-probability has the chance value, which is 0.5 here because there are just two output values.

x_1	x_2	x_3		y_1
person	consumes	heat	$\Longrightarrow$	yes
person	consumes	electricity	$\Longrightarrow$	no
person	consumes	moisture	$\Longrightarrow$	yes
person	consumes	silicon	$\Longrightarrow$	no
person	dislikes	heat	$\Longrightarrow$	no
person	dislikes	electricity	$\Longrightarrow$	yes
person	dislikes	moisture	$\Longrightarrow$	no
person	dislikes	silicon	$\Longrightarrow$	yes
computer	consumes	heat	$\Longrightarrow$	no
computer	consumes	electricity	$\Longrightarrow$	yes
computer	consumes	moisture	$\Longrightarrow$	no
computer	consumes	silicon	$\Longrightarrow$	yes
computer	dislikes	heat	$\Longrightarrow$	yes
computer	dislikes	electricity	$\Longrightarrow$	no
computer	dislikes	moisture	$\Longrightarrow$	yes
computer	dislikes	silicon	$\Longrightarrow$	no

The neutrality of the likelihood problem implies that we should expect generalisation performance on this problem by backpropagation (and C4.5) to be just as poor as it was in the case of parity. And in fact this is exactly what we *do* find. Backpropagation's generalisation performance on the likelihood problem using one, randomly selected unseen case is summarised in the Figure 2. Again, the generalisation performance is very poor in all architectures tested. As expected, C4.5 generalises incorrectly on all 16, minimally incomplete training sets for this problem.

It is interesting to note that the generalisation performance on this new problem hovers around the chance level. This is of course where we would expect

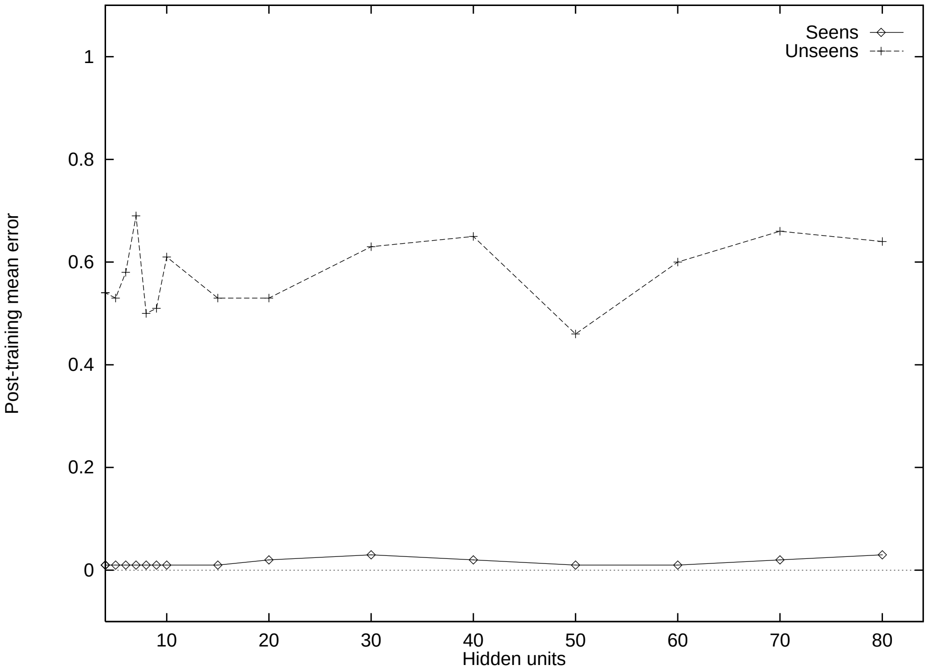

Figure 2: Post-training mean-error curves for likelihood generalisation.

it to be on the hypothesis that backpropagation cannot deal with statistically netural mappings. In the case of the true parity problem, it will be recalled that the generalisation was considerably *worse* than chance. The explanation for this appears to be that in deleting a single case from a parity mapping, a strong but misleading association is created between input cases one Hamming unit away from the deleted case and the complement of the output for those cases (i.e., the 'wrong' output). Backpropagation detects and exploits this phoney association and is thus led to always produce the complement of the correct generalisation.

4 Concluding comment

The paper has shown that backpropagation reliable fails to solve parity learning problems when they are posed as genuine supervised learning problems. The algorithm is thus subject to at least one of the limitations that Minsky and Papert attributed to the perceptron method in the late 1960s. The firm confidence which researchers sometimes place in backpropagation may therefore be less than fully justified.

References

[1] Minsky, M. and Papert, S. (1988). *Perceptrons: An Introduction to Computational Geometry* (expanded edn). Cambridge, Mass.: MIT Press.

[2] Quinlan, J. (1993). *C4.5: Programs for Machine Learning.* San Mateo, California: Morgan Kaufmann.

[3] Rumelhart, D., Hinton, G. and Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, *323* (pp. 533-6).