Parity: The Problem that Won't Go Away

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/users/christ/index.html
Tel: (44)1273 678856

December 20, 1995

Abstract

It is well-known that certain learning methods (e.g., the perceptron learning algorithm) cannot acquire complete, parity mappings. But it is often overlooked that state-of-the-art learning methods such as C4.5 and backpropagation cannot generalise from *incomplete* parity mappings. The failure of such methods to generalise on parity mappings is sometimes dismissed as uninteresting on the grounds that it is 'impossible' to generalise over such mappings, or on the grounds that parity problems are mathematical constructs having little to do with real-world learning. However, this paper argues that such a dismissal is unwarranted. It shows that parity mappings are hard to learn because they are statistically neutral and that statistical neutrality is a property which we should expect to encounter frequently in real-world contexts. It also shows that the generalization failure on parity mappings occurs even when large, minimally incomplete mappings are used for training purposes, i.e., when claims about the impossibility of generalization are particularly suspect.

1 Introduction

The parity rule is easily stated (e.g., 'the output value is true if and only if an odd number of input values are true') but it is surprisingly hard to learn by conventional methods. The reason is related to the fact that parity mappings are statistically neutral. The probability (i.e., frequency) of seeing some particular input value mapped onto some particular output in a parity mapping always turns out to be the chance value of 0.5. This means that it is impossible to build successful rules which focus on particular input values: any successful rule must attend to *all* the input values in order to get the answer right in all cases.

For the machine learning researcher, the significance of the parity problem has traditionally been its effectiveness as a small but challenging benchmark problem. However, I will argue that this view underestimates the fundamental nature and wider implications of the problem. I claim that the parity problem is at the heart of a major class of hard learning problems and that our understanding of the way in which it can be solved will thus be a key component in our understanding of learning.

The paper divides up into three main sections. The next section (section two) analyses the statistical basis of the parity problem and clarifies its relationship with the wider class of statistically neutral problems. Section three presents a task analysis of learning which leads to a basic distinction between hard and easy learning problems. Section four shows how the class of hard learning problems are statistically neutral in principle but not in practice and demonstrates how incidental, statistical effects can be exploited by standard learning methods. Section five is a discussion and summary.

2 Statistical properties of the parity problem

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 should be true just in case an *odd* number of input values are true.[1] If there are just two input variables the problem is known as 'Exclusive-OR' (or **XOR**) since it is effectively the rule that either of the inputs can be true, but not both.

It is well known that parity problems are statistically *neutral* [1]. This means that all conditional output probabilities exhibited by a parity mapping have 'chance' values, i.e., that no input/output associations exist. Consider the 3-bit (i.e., 3-input) parity problem, which 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

In Table 1 we see the unconditional and conditional probabilities for all input-variable instantiations. Note that all the probabilities are exactly the chance value for a boolean value, namely 0.5. This is a *necessary* consequence of the nature of the input/output rule. The frequencies with which we see each of the two possible outputs when we put an input variable into a fixed state must always be equal since there will be just as many cases of the unfixed variables which produce parity as non-parity. Thus the output probabilities conditional on any particular input variable instantiation will always be at the chance level. The statistical neutrality of the parity problem means that it cannot be solved by statistical methods: any process of searching for dependencies between specific input and output variables is of no benefit because such associations simply do not exist. Thus, the performance of any learning method which exploits such processes is necessarily compromised on a parity problem. (This is of course what makes the parity problem a challenging benchmark.) But interestingly, it turns out that parity is not the *only* type of problem for which statistical neutrality is guaranteed. Any problem that can be converted into a modulus-addition problem is guaranteed to be statistically neutral provided that the number of possible values for any given input variable is equal to, or an exact multiple of the number of possible output values. To show this we argue backwards from observations about the statistically neutral training set.

[1] Arguably, since the rule tests for an odd number rather than an even number, the problem should be called the *dis*parity problem.

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

We can confirm the neutrality of this training set empirically by tabulating the relevant conditional probabilities. (Table 2 shows the complete set of probabilities which have a first or zeroth-order condition.)

	$P(y_1 =$no$\vert\ )$	$P(y_1 =$yes$\vert\ )$
	0.5	0.5
$x_1 =$person	0.5	0.5
$x_1 =$computer	0.5	0.5
$x_2 =$consumes	0.5	0.5
$x_2 =$dislikes	0.5	0.5
$x_3 =$electricity	0.5	0.5
$x_3 =$heat	0.5	0.5
$x_3 =$silicon	0.5	0.5
$x_3 =$moisture	0.5	0.5

Table 2: Conditional output probabilities in consumer mapping.

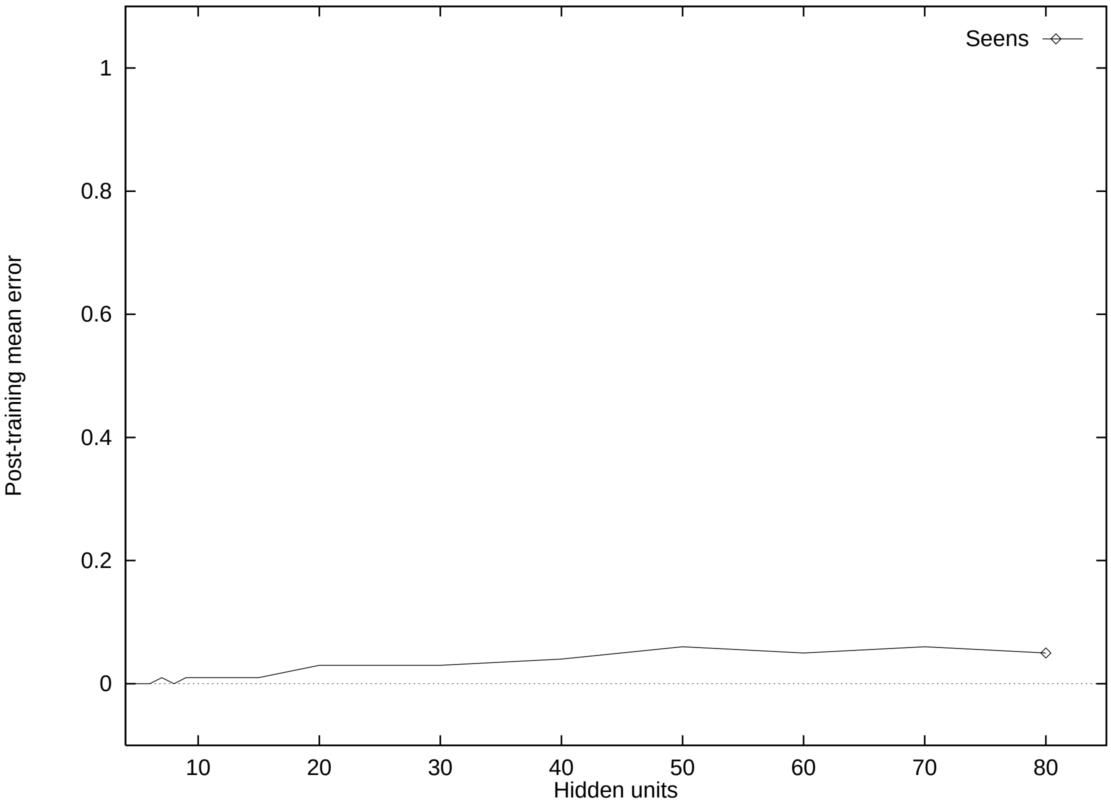
2.2 Performance of learning algorithms on neutral ma ings

The performance of learning methods that rely on the exploitation of statistical effects is necessarily compromised on statistically neutral problems. Learning methods that rely *solely* on the exploitation of statistical effects produce worst-case performance on such problems. Algorithms in the CART family [2] are a case in point. ID3 [3,4] for example, constructs a decision tree by recursively partitioning the training set until every pair in a given partition maps onto the same output value. At each stage of the process, a new partitioning is constructed by dividing up the cases in an existing partition according to which value they have on the variable whose values are most strongly correlated (within the partition) with specific output values. This has the effect of maximizing the output-value uniformity of new partitions and thus minimising (subject to horizon effects) the total number of hyper-rectangular partitions required in order to achieve full uniformity. The algorithm is thus guided only by statistical effects in the training data.

The implication is that ID3 should produce worst-case performance on parity profJ30.24020non the *entire* mapping in the training data, and have thus not tested the algorithm's ability to generalise to unseen cases.[3] However, if we test backpropagation's ability to generalise to one unseen case in, say, the 4-bit parity mapping (i.e., we present 15 of the 16 cases as training data, and test generalization on the one remaining case), then the results are unambiguously poor.

In an exhaustive 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 2. 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



poor for all architectures used, i.e., no generalization is achieved. (The fact that the generalization here is significantly worse than chance is explained below.) For purposes of comparison we carried out an identical analysis of backpropagation's generalization performance on the consumer problem (holding back one case as an unseen) and obtained qualitatively similar results, see Figure 3.

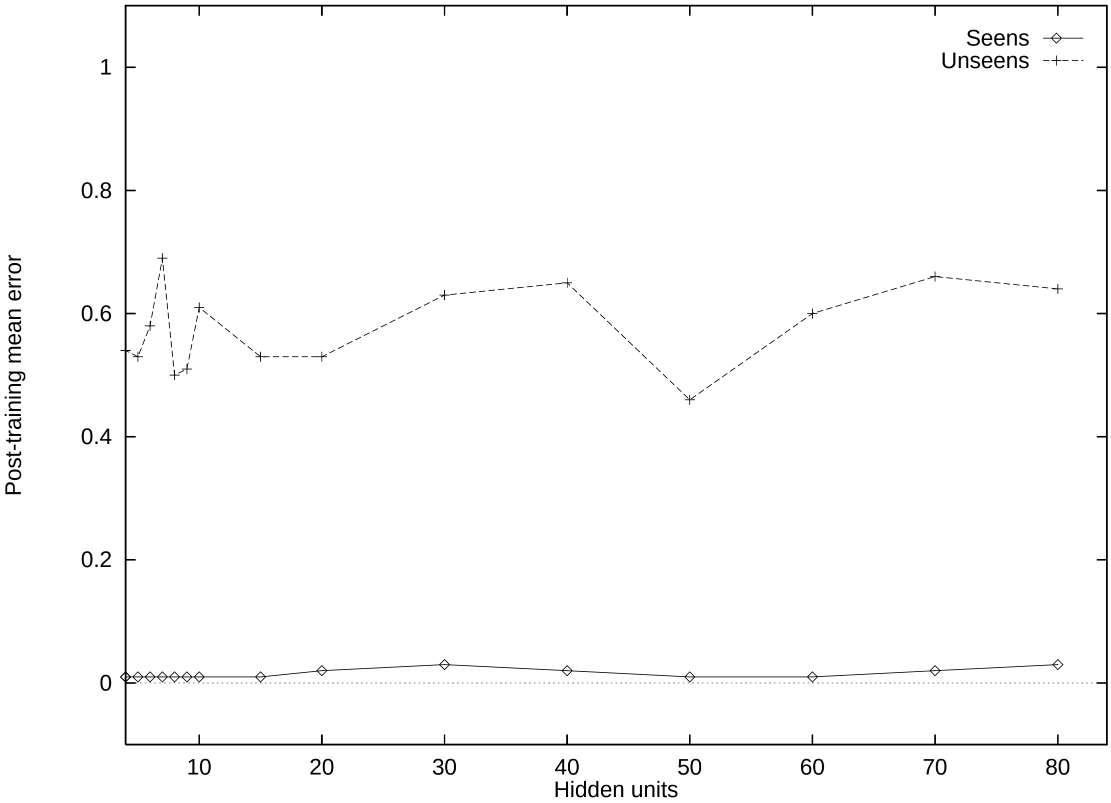


Figure 3: Post-training mean-error curves for consumer generalization.

2.4 Why algorithms fail to generalise over neutral mappings

The dynamics of the backpropagation process are complex. Explaining its generalization failure on neutral mappings is thus not straightforward. The simplest hypothesis may be that, despite its manifest success in the acquisition of small, complete parity mappings [8] backpropagation relies primarily on the exploitation of statistical effects and is thus unable to deal properly with neutral mappings. There are several arguments in favour of this idea.

First, the backpropagation learning algorithm is a generalization of the least-mean-squares algorithm [9] (and perceptron learning algorithm [10]) which is effectively an iterative method for deriving statistical input/output correlations. Thus the backpropagation learning method is rooted in a method for exploiting statistical effects. Second, the generalization performance observed in the 4-bit parity tests tended to be much *worse* than chance. This result is explained if the algorithm is primarily relying on statistical effects since the effect that is created when we delete one case from a parity mapping is a correlation 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). Thus if the algorithm exploits input/output correlations then it will tend to always generalise *incorrectly* from the minimally incomplete parity mapping. The fact that it does do so tends to confirm the hypothesis that backpropagation primarily exploits statistical correlations.

3 Relational problems are approximately neutral

It is sometimes argued that parity mappings are artificial, mathematical constructs and that we should therefore not be too concerned if we find that our learning methods fail to generalise over them. Parity mappings are hard to generalise because they are statistically neutral. But neutrality, or approximate neutrality is actually a common property of challenging learning problems. In fact it should be obvious that any learning problem with a complex, *relational* input/output rule (i.e., a rule which tests for a relationship among the inputs) will have an approximately nearly neutral training set.

If the input/output rule is relational then we do not expect to see any associations between specific input values and specific output values showing up in the training set. There is an association; but it involves a relationship among the inputs. Thus, we should expect that any relational learning problem will have a neutral target mapping. Unfortunately, the truth of the matter is less clear-cut. Relational rules, in fact, do not guarantee neutrality. The way in which a particular relational rule is encoded in a set of input/output examples ca90Td[(lucrea(y)]TJ9(v)102(exampl(Relatioexamplinc5c)plinc5c)pliuut5999.7(-'-

values (see Table 4).

	$P(y_1$

3.1 S arse codings am lify incidental effects

The fact that generalising incidental effects can be created through the encoding of the underlying, relational rule means that we can sometimes turn a 'hard' learning problem into an 'easier' problem simply by applying an encoding which maximizes the strength and range of generalising incidental effects. A simple approach involves using a sparse coding in which each input variable records9dtal

5 Acknowledgements

Many of the ideas in this paper were developed in collaboration with Jim Stone.

References

[1] Hinton, G. and Sejnowski, T. (1986). Learning and relearning in boltzmann machines. In D. Rumelhart, J. McClelland and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Vols I and II* (pp. 282-317). Cambridge, Mass.: MIT Press.

[2] Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees.* Wadsworth.

[3] Quinlan, J. (1986). Induction of decision trees. *Machine Learning, 1* (pp. 81-106).

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

[5] Wnek, J. and Michalski, R. (1994). Discovering representation space transformations for learning concept descriptions combining DNF and m-of-n rules. *Proceedings of ML-COLT'94.*

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

[7] Beale, R. and Jackson, T. (1990). *Neural Computing: An Introduction.* Adam Hilger.

[8] Rumelhart, D., Hinton, G. and Williams, R. (1986). Learning internal representations by error propagation. In D. Rumelhart, J. McClelland and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Vols I and II* (pp. 318-362). Cambridge, Mass.: MIT Press.

[9] Thornton, C. (1992). *Techniques in Computational Learning: An Introduction.* London: Chapman & Hall.

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

[11] Thrun, S., Bala, J., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dzeroski, S., Fisher, D., Fahlman, S., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R., Mitchell, T., Pachowicz, P., Reich, Y., Vafaie, H., Van de Welde, W., Wenzel, W., Wnek, J. and Zhang, J. (1991). The MONK's problems - a performance comparison of different learning algorithms. CMU-CS-91-197, School of Computer Science, Carnegie-Mellon University.