Parity Still Isn't a Generalisation Problem

R.I. Damper,

Cognitive Sciences Centre

and

Department of Electronics and Computer Science,

University of Southampton,

Southampton SO17 1BJ,

England.

Email: `rid@ecs.soton.ac.uk`

WWW: `http://isis.ecs.soton.ac.uk/`

(Continuing commentary on "Trading Spaces: Computation, Representation, and the Limits of Uninformed Learning", target article by Andy Clark and Chris Thornton: *BBS* **20**, pp.57–90.)

Abstract

Clark and Thornton take issue with my claim that parity is not a generalisation problem, and that nothing can be inferred about back-propagation in particular, or learning in general, from failures of parity-generalisation. They advance arguments to support their contention that generalisation is a relevant issue. In this continuing commentary, I examine generalisation more closely in order to refute these arguments. Different learning algorithms will have different patterns of failure: back-propagation has no special status in this respect. This is not to deny that a particular algorithm might fortuitously happen to produce the 'intended' function in an (oxymoronic) parity-generalisation task.

Clark and Thornton (C&T) distinguish between straightforward type-1 problems which are "statistical" and problems of type-2 which are "relational". The former are learnable by an 'uninformed' learning device, they say, while the latter require some sort of recoding to become learnable. C&T cite parity as an example of a type-2 problem, demonstrate the inability of back-propagation to generalize on this problem, and draw the conclusion that (sect. 5, para. 1): "Uninformed learning ... had little chance of penetrating the space of type-2 problems". In my commentary (*BBS* **20**, pp.69–70), I showed that the parity problem – at least where this involves binary-to-binary input-output mappings – cannot be considered a generalisation problem.

My argument was actually stated more eloquently by C&T in their Authors' Response (sect. R6, para. 1) than I had managed myself:

> "Damper ... worries that holding back even a single pattern on the classical (2 variable, XOR) parity problem simply makes the problem insoluble (the machine would need to read our minds to know the intended function) as the learning algorithm lacks sufficient data. He concludes that it must be wrong to link parity learning to issues about generalisation."

Having grasped my argument so well, however, they are strangely reluctant to accept it. Instead, they advance some counter arguments to support their original position. I seek to show here that these counters are unfounded.

Let us first be clear what generalisation is. Actually, it is not a very well-defined concept: basically, it refers to the fitting of a smooth function to the input-output mapping, avoiding over-fitting of the training data. So, it is not well-defined because one can ask: how smooth does smooth have to be? Although not well defined, however, we can still assert that "parity is not a generalisation problem" because binary-to-binary mappings are inherently discontinuous.

C&T go on to say (sect. R6, para. 2): "Damper implies that parity cannot be a generalisation problem because parity mappings exhibit neutral statistics". Leaving aside the matter that I certainly did more than "imply", the argument was not based on statistics. I showed through the results of simulations on the 2-variable XOR problem with the 11 ($\Rightarrow 0$) input held back that the learned function always reflected the most obvious input-output mapping (the OR function), rather than anything to do with probabilities or statistics. This point was also well made by Chater in his commentary (*BBS* **20**, pp.68–69): he writes that feedforward neural networks are not "concerned with learning arbitrary conditional probability distributions, but rather with learning *functions* from input to output". Put even more concretely, the back-propagation algorithm is concerned with searching heuristically an error surface in weight space for a minimum, and this is only loosely related to input-output statistics – or, indeed, to generalisation. This lack of a very direct relation explains why, in practical applications, the evolving generalisation ability of a network has to be tested during training with held-out data (so-called validation testing), rather than merely by monitoring the training-set error, if over-fitting is to be avoided.

C&T next opine (sect. R6, para. 3) that expecting generalisation on a 4-variable problem (holding out just one case in 16 to leave 15 cases as the basis for generalisation) "somehow does not seem quite so unreasonable". But they themselves showed that the problem would not generalise, and drew some strong inferences from this failure! The essential nature of the problem is not changed by adding more variables. The reason for the failure is precisely the same as the reason for failure in the 2-variable case: parity is not a generalisation problem.

They then consider the standard two-spirals problem which, they say, is "parity-like" and "has never been treated as anything other than a generalisation problem". The clue here is in the "-like" qualification. Because the inputs are co-ordinates in the plane and the output is a discrete label from one of two classes, this problem involves continuous-to-binary mappings and so is an instance of what I called (my sect. 3) an "extended parity problem". The extra information in the continuous input is crucial in making this a genuine generalisation problem (one where it makes sense to think of a smooth interpolation of the training data-points) where the 'true' (binary-to-binary) parity problem is not.

Now, if 'true' parity isn't a generalisation problem, what are we to make of the claim of Berkeley in his commentary (*BBS* **20**, pp.66–67) to have a solution for it? The key point here is that his learning procedure is quite unlike distributed back-propagation; it is localist in that the "value units" have restricted (nonmonotonic, Gaussian) receptive fields. I mentioned in my commentary (sect. 4) that constructive techniques with localist units are trivially capable of 'solving' the parity-generalisation (actually an oxymoron) problem, and gave the example of constructing an AND-OR network to illustrate the principle. There is, of course, nothing to stop a learning device *appearing* to mind-read by discovering just that solution which happens to be in the experimenters' mind!

The lesson of all this is that learning is not homogeneous – different algorithms learn different things. Indeed, quite subtle differences between learning procedures can produce quite profoundly different results. For instance, most people would imagine that it matters little whether one uses back-propagation (Rumelhart *et al.*, 1986) or the perceptron rule (Rosenblatt, 1962) to train a single-layer perceptron: the former is just an extension to the latter which allows hidden-unit weights to be estimated. Yet, as Brady *et al.* (1989) have shown, back-propagation actually fails on some linearly-separable problems where perceptron learning succeeds. This result deserves to be much better known than it is. So back-propagation learning has no special status, and C&T are wrong to read too much into its failures (especially on an insoluble problem!). Echoing Chater's question "Why probabilities?", we could as well ask of C&T "Why back-propagation?"

References

Brady, M.L., Raghavan, R. & Slawney, J. (1989) Back-propagation fails to separate where perceptrons succeed. *IEEE Transactions on Circuits and Systems* 36: 665–674.

Rosenblatt, F. (1962) *Principles of neurodynamics*. Spartan, New York.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986) Learning representations by back-propagating errors. *Nature* 323: 533–536.