Commentary on **Andy Clark and Chris Thornton (1997). Trading spaces: Computation, representation, and the limits of uninformed learning. BBS 20:57–90.**

Abstract of the original article: Some regularities enjoy only an attenuated existence in a body of training data. These are regularities whose statistical visibility depends on some systematic recoding of the data. The space of possible recodings is, however, infinitely large – it is the space of applicable Turing machines. As a result, mappings that pivot on such attenuated regularities cannot, in general, be found by brute-force search. The class of problems that present such mappings we call the class of "type-2 problems." Type-1 problems, by contrast, present tractable problems of search insofar as the relevant regularities can be found by sampling the input data as originally coded. Type-2 problems, we suggest, present neither rare nor pathological cases. They are rife in biologically realistic settings and in domains ranging from simple animat (simulated animal or autonomous robot) behaviors to language acquisition. Not only are such problems rife – they are standardly solved! This presents a puzzle. How, given the statistical intractability of these type-2 cases, does nature turn the trick? One answer, which we do not pursue, is to suppose that evolution gifts us with exactly the right set of recoding biases so as to reduce specific type-2 problems to (tractable) type-1 mappings. Such a heavy-duty nativism is no doubt sometimes plausible. But we believe there are other, more general mechanisms also at work. Such mechanisms provide general (not task-specific) strategies for managing problems of type-2 complexity. Several such mechanisms are investigated. At the heart of each is a fundamental ploy – namely, the maximal exploitation of states of representation already achieved by prior, simpler (type-1) learning so as to reduce the amount of subsequent computational search. Such exploitation both characterizes and helps make unitary sense of a diverse range of mechanisms. These include simple incremental learning (Elman 1993), modular connectionism (Jacobs et al. 1991), and the developmental hypothesis of "representational redescription" (Karmiloff-Smith 1979; 1992). In addition, the most distinctive features of human cognition – language and culture – may themselves be viewed as adaptations enabling this representation/computation trade-off to be pursued on an even grander scale.

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.
rid@ecs.soton.ac.uk WWW:http://isis.ecs.soton.ac.uk/

Abstract: Clark & 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 (1996t) (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 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 Damper (1996), 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, p. 87) 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 generalization.

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 counter arguments are insufficient.

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 it is 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, p. 87): "Damper implies that parity cannot be a generalization 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 (1996) in his commentary: 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; 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 generalising 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, p. 87) 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 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.

C&T then consider the standard two-spirals problem which, they say, is "parity-like" and "has never been treated as anything other than a generalization 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 an "extended parity problem" (my sect. 3). 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 is not a generalisation problem, what are we to make of the claim of Berkeley (1996) in his commentary (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 and 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?"

Authors' Response

Reading the generalizer's mind

Chris Thornton[a] and Andy Clark[b]

[a]*Cognitive and Computing Sciences, University of Sussex, Falmer, Brighton, BN1 9QH, England;* **chris.thornton@cogs.susx.uk.ac**
www.cogs.susx.ac.uk/

[b]*Philosophy/Neuroscience/Psychology Program, Washington University in St Louis, St Louis, MO 63130;* **andy@twinearth.wustl.edu**

Abstract: In his new commentary, **Damper** re-emphasises his claim that parity is not a generalisation problem. But when proper account is taken of the arguments he puts forward, we find that the proposed conclusion is not the only one that can be drawn.

In adding the word "still" to the title of his ongoing commentary, **Damper** re-emphasises his claim that "parity is not a generalisation problem." His view is that in our Response (Clark & Thornton 1997r) we failed to accept or even properly address his argument. However, as we hinted in the first paragraph of section R6, the interpretation of his claim is not a straightforward matter. "Parity" refers to the truth function whose rule is that the output is true if an odd number of the inputs are true. Parity functions may be of any order but we are particularly familiar with the 2-place variant, also known as "exclusive-or" (XOR).

Now, of course, a function is a function is a function. It is not, in itself, a "problem." And incontrovertibly, therefore, it cannot be a "generalisation problem." However the parity function (like any other function) is easily used as the *basis* for a generalisation problem. The procedure is straightforward: we take the complete mapping for a given parity function (e.g., all 16 input/output associations for the 4-bit parity function) and we present a *subset* of these cases to the generalisation mechanism, for example a supervised learner. We then test the generalisation performance of the mechanism by examining its response on the unseen cases, or the "validation" set as Damper calls it. This is the standard method for presenting a generalisation problem to a supervised learning mechanism. And we note that in both his commentaries, Damper describes the way he used the method to test the generalisation abilities of a backpropagation network. His account in his first commentary is particularly clear: he describes how he provided his backpropagation network with just "the first three lines of the [XOR] truth table" so as to see whether it could "generalise on the 2-variable parity (XOR) problem." Clearly, then, Damper is familiar with the procedure by which a parity mapping is used as the basis for the presentation of a generalisation problem. His intention thus cannot be to argue that parity cannot be used as the basis of a generalisation problem. What, then, are we to make of his declaration that "parity is not a generalisation problem."

A clue to his intentions comes in his second commentary. Here he suggests that generalisation "refers to the fitting of a smooth function to an input-output mapping." For Damper, this implies that "parity is not a generalisation problem because binary-to-binary mappings are inherently discontinuous." The argument here is that since parity involves a particular operation – namely the "fitting of a smooth function to an input-output mappings" – the setting up of a generalisation problem in which this operation cannot, in principle, be applied, inevitably results in failure. In experimental terms, the procedure should be deemed a pointless and vacuous exercise and "parity generalisation" classified as an oxymoron.

Several aspects of Damper's first and second commentaries suggest that this is indeed his intended argument and that we should therefore treat "parity [still] isn't a generalisation problem" as a claim about the impossibility of *solving* parity-generalisation problems. And yet, nagging doubts [still] remain. Can Damper's view that generalisation involves the fitting of a "smooth function" to an input-output mapping be taken seriously? A significant proportion of contemporary generalisation models are symbolic in nature and do not trade in *any* sort of numerical representation. Is it Damper's intention that these should be ruled out of court? His attitude to the performance of backpropagation on parity-generalisation also presents problems. We expect Damper to take the failure of *any* generalisation method on parity generalisation to corroborate his view that parity-generalisation cannot be regarded as a genuine problem. And yet in his conclusion, Damper suggests that "C&T are wrong to read too much into [backpropagation's] failures" on the parity-generalisation task.

And what are we to make of his treatment of the generalisation method described by Berkeley? Damper notes that Berkeley's method provides a "solution" to a parity generalisation task and we naturally expect Damper's position to be an emphatic rejection of Berkeley's claim that the method performs anything approximating "genuine generalisation." And yet, paradoxically, Damper's view is that Berkeley's method *satisfactorily* accomplishes the task it is set even though – as he puts it – the method may appear to "mind-read by discovering just that solution which happens to be in the experimenter's mind!" This sounds suspiciously

like a muted round of applause. And in fact it turns out that Damper's view is that "the lesson of all this is that learning is not homogeneous – different algorithms learn different things."

Reading these words we must struggle with the contrast between the Damper who thinks parity generalisation is not a genuine learning problem and the Damper who considers that certain methods satisfactorily solve parity generalisation problems. But to get hung up on these apparent contradictions would, we believe, be a mistake. Rather, we should try to determine Damper's intended meaning by carefully reading between the lines of his commentary.

In appearing to present inconsistent views with respect to generalisation's technicalities, Damper may be cunningly shaking out the thorny problem which lies at the subject's core, namely Hume's problem, or the "problem of induction." This is the observation that since inductive generalisations do not have (by definition) a logical derivation, they can never be regarded as entirely certain. Any inductively-acquired knowledge (e.g., scientific knowledge) is thus necessarily uncertain.

An interesting corollary is that, since generalisation *products* are always uncertain, we should arguably treat all generalisation *methods* as being of equal status. And, indeed, this thesis has recently been given a mathematical foundation in the form of the No-Free-Lunch theorem of Wolpert (1996b; 1996a) and the Conservation Law of Schaffer (1994).[1]

Damper's assertion that "backpropagation has no special status" seems to confirm our suspicion that his underlying aim is not so much to demonstrate that parity isn't a generalisation problem but rather to demonstrate that the performance of particular learning methods on particular problems does not tell us very much. But if this *is* his intention then all parties can breathe a sigh of relief. There is nothing about our position which would cause us to do anything but wholeheartedly concur.

Recall that our target article used a probability argument to show that inductive generalisations may be justified either though type-1 (statistical) effects or through type-2 (relational) effects. We demonstrated that learning which depends exclusively on the exploitation of type-1 effects cannot deal with relational problems because such problems do not present exploitable, type-1 effects. (In both his commentaries, Damper takes time to illustrate what this means in the context of the parity mapping.) Following the presentation of the type-1/type-2 distinction we then introduced a case study involving the backpropagation method. This was intended merely to provide an illustrative example. Our suggestion was that the in-principle intractability that relational problems present to methods relying on type-1 effects "may help to explain why backpropagation . . . often fails to solve low-order parity problems when presented as generalisation problems." In other words, we were speculating that backpropagation may be bad at parity generalisation as a result of depending too heavily on the exploitation of type-1 effects. We could easily have made the same remark about any other method adopting the same strategy.

We believe that wires may have become crossed over this reference to backpropagation partly because of the unconventional way in which the type-1/type-2 distinction was formulated. But we hope that the ensuing commentary has clarified the fact that the distinction introduced is uncontentious and that it is in fact one which has been expressed in a wide variety of ways over a large number of years. As it turns out, it can even be formulated in terms of Damper's own "smooth function" concept.

To formulate the distinction in these terms we first need to visualise the generalisation process and associated input/output mapping in pictorial terms. We view the input/output mapping in terms of an input space whose datapoints correspond to individual inputs. The label attached to each datapoint is then the output associated with the input; a method then solves a generalisation problem by successfully using a sample of labelled datapoints to predict the labels of inputs not included in the sample.

In a "smooth" input/output mapping – the type that Damper believes presents a genuine generalisation problem – the labeling of datapoints varies smoothly across the space. Datapoints with the same label cluster together and there is a gradual transition between different labels as we move across the space. Generalisation methods must associate groups of inputs with specific outputs and in the "smooth mapping" context, this is easily accomplished. Because of the way inputs with the same output tend to cluster together, the process of separating them can be accomplished straightforwardly by introducing simple bounding constructs (planes, spheres, etc.) into the space. If the input/output mapping is not smooth and datapoints with the same label do not cluster together, then separation of groups of inputs requires the introduction of more complex bounding constructs.

These observations might lead us to introduce a "new" fundamental distinction between smooth and non-smooth input/output mappings and to point out that only smooth input/output mappings allow for learning/generalisation processes based on the introduction of simple bounding constructs. Particular learning methods could then be divided up according to whether they utilise simple or complex bounding constructs. Key members of the "simple" camp would turn out to be the Perceptron method (Minsky & Papert 1988) which introduces a single, planar boundary, ID3 (Quinlan 1983) which adds an arbitrary number of axis-aligned, extreme boundaries, backpropagation (Rumelhart et al. 1986), which manipulates a fixed number of linear boundaries, LVQ (Kohonen et al. 1990), which manipulates a fixed number of spherical boundaries and the k-nearest-neighbours method (Duda & Hart 1973), which utilises the implicit planar boundaries between datapoints. Key members of the "complex" camp would turn out to be methods such as AQ15 (Michalski et al. 1986), `cigol` (Muggleton & Buntine 1988), and `foil` (Quinlan 1990), which utilise background knowledge of one form or another for the purposes of forming complex separations among classes of inputs.

But in working through this argument we would, of course, simply be rehashing the type-1/type-2 distinction introduced in our paper. Input/output mappings are "smooth" just in case datapoints with the same label cluster together. This occurs if absolute input values (i.e., datapoint coordinates) are significant for the prediction of output. If input values are not significant, then there is no reason to expect datapoints with the same label to occupy the same part of the input space; there is no clustering and no smoothness. Damper suggests that expecting a method to generalise in this context – when absolute values are insignificant for the prediction of output (as they are in the parity mapping) – amounts to "expecting the [method] to

https://doi.org/10.1017/S0140525X98221200 Published online by Cambridge University Press

be a mind-reader." But although absolute values may not be significant, the relationship(s) among them may be. Generalisation then does not require mind-reading but merely an accurate identification of the relationship underlying the mapping. This brings us more or less back to the original point around which our paper was based. Mappings in which the underlying input/output rule is relational or type-2 cannot be generalised by methods which utilise simpler bounding constructs and thus implicitly assume a "smooth" (type-1) mapping.

As we suggested in our initial response, Damper is quite right to observe that absolute input values (datapoint coordinates) cannot be used as a basis for predicting outputs in parity mappings. But rather than demonstrating that parity cannot be treated as a generalisation problem, it actually demonstrates that parity forms the basis for a particular type of generalisation problem, namely, a *relational* problem in which the successful prediction of outputs involves the discovery of the relational rule underlying the mapping. Thus Damper's correction of his title should not involve the insertion of the word "still" but rather the insertion of "type-1," thus producing the correct conclusion "Parity is not a *type-1* generalisation problem."

NOTE

1. These methods make use of the fact that when we average the performance of a generalisation method over all possible scenarios, we inevitably find that each particular generalisation is correct just as often as it is incorrect. The effect is that all generalisation methods have an average performance which is identical to that achieved by random guessing.

References

Brady, M. L., Raghavan, R. & Slawney, J. (1989) Back-propagation fails to separate where perceptrons succeed. *IEEE Transactions on Circuits and Systems* 36:665–74. [RID]

Duda, R. & Hart, P. (1973) *Pattern classification and scene analysis.* Wiley. [rCT]

Kohonen, T., Barna, G. & Chrisley, R. (1990) Statistical pattern recognition with neural networks: Benchmarking studies. In: *Neurocomputing 2*, ed. J. A. Anderson, A. Pellionisz & E. Rosenfeld. MIT Press. [rCT]

Michalski, R., Mozetic, I., Hong, J. & Lavrac, N. (1986) The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proceedings of the Fifth National Conference on Artificial Intelligence*, 1041–45. Morgan Kaufmann. [rCT]

Minsky, M. & Papert, S. (1988) *Perceptrons: An introduction to computational geometry* (expanded edition). MIT Press. [rCT]

Muggleton, S. & Buntine, W. (1988) Machine invention of first order predicates by inverting resolution. In: *Proceedings of the Fifth International Conference on Machine Learning*, ed. J. Laird. Morgan Kaufmann. [rCT]

Quinlan, J. (1983) Learning efficient classification procedures and their application to chess and games. In: *Machine learning: An artifical intelligence*, ed. R. Michalski, J. Carbonell & T. Mitchell. Tioga. [rCT]

(1990) Learning logical definitions from relations. *Machine Learning* 5:239–66. [rCT]

Rosenblatt, F. (1962) *Principles of neurodynamics.* Spartan. [RID]

Rumelhart, D., Hinton, G. & Williams, R. (1986) Learning representations by back-propagating errors. *Nature* 323:533–36. [RID, rCT]

Schaffer, C. (1994) Conservation law for generalization performance. *Proceedings of the International Conference on Machine Learning.* July. Rutgers University. [rCT]

Wolpert, D. 1996a) The existence of a priori distinctions between learning algorithms. *Neural Computation* 8(7). [rCT]

(1996b) The lack of a priori distinctions between learning algorithms. *Neural Computation* 8(7). [rCT]