simplest model (e.g., the one with the least "free" parameters). In fact, smaller networks are well known to provide superior generalization (e.g., Baum & Haussler 1989). In this respect, the arguments of C&T would have been more convincing if six- or more bit parity were used, so that the mapping could be carried out with fewer free parameters (i.e., weights) than training patterns. Since avoiding local minima in minimal six- (or more) bit parity networks is extremely difficult and since it is unlikely that real brains use minimal networks we shall pass over this point.

One natural way to achieve model simplification is by constraining the search space, and one natural constraint might be the imposition of symmetry, that is, start learning assuming maximal symmetry and only relax that assumption as each level of symmetry is found to fail to exist. This will automatically reduce the effective number of free parameters. For example, imposing a symmetry on the weights is sufficient to give good generalization for the four-bit parity problem. Here we constrain the weight solutions to lie on the hyperplanes in weight space corresponding to weights that are symmetric with respect to the input units. This might be implemented in a learning network by constraining the weight changes to be the same for each input unit. This reduces the problem to 13 degrees of freedom and requires only 16.3 million random attempts to find 20 solutions. The symmetry guarantees that all these solutions will generalize correctly. Such "weight sharing" is known to improve generalization more generally (e.g., Nowlan & Hinton 1992).

Another natural constraint we may impose is to assume that local information is more important than distant information until such an assumption is proven incorrect. We may view this to be at work in Elman's grammar acquisition network as discussed by C&T. Elman (1993) implemented these constraints with incremental learning schemes. This is in fact another poor example, since the network not only fails to generalize but also has insufficient processing power to learn even the raw training data (Elman 1993, p. 76). A more powerful recurrent network, or a network with appropriate input buffers or time delay lines, should not have this problem, but there is no reason to suppose that this would improve generalization as well. In time-buffered networks we can constrain solutions to make maximal use of local information by having a smaller learning rates for weights corresponding to longer range dependencies. This approach has also, for example, been shown to improve generalization in past tense acquisition models for which the inflection is usually, but not always, determined by the final phoneme of the stem and in models of reading aloud for which long range dependencies are relatively rare (Bullinaria 1994). Similar constraints may be implemented by weight decay and are also known to improve generalization (e.g., Krogh & Hertz 1992).

Simple constraints on the weight space may not be sufficient to improve generalization for all type-2 problems, but the examples given above indicate that it does have a range of applicability. One might argue that such constraints are just a convenient way to implement the representational recodings of Clark & Thornton, but if that is the case we would seem to have a continuous spectrum of constraints and their type-1/type-2 distinction becomes rather fuzzy.

What is the type-1/type-2 distinction?

Nick Chater

Department of Psychology, University of Warwick, Coventry, CV4 7AL, United Kingdom. **nick@psy.ox.ac.uk**

Abstract: Clark & Thornton's type-1/-2 distinction is not well-defined. The classes of type-1 and type-2 problems are too broad: many nocomputable functions are type-1 and type-2 learnable. They are also too narrow: trivial functions, such as identity, are neither type-1 nor type-2 learnable. Moreover, the scope of type-1 and type-2 problems appears to be equivalent. Overall, this distinction does not appear useful for machine learning or cognitive science.

1. Why probabilities? Clark & Thornton (C&T) frame the learning problem as deriving a conditional probability distribution $P(Y|X)$, where X and Y are sets of possible inputs and outputs, from a set of input-output pairs, (x, y). This is puzzling, because the learning systems that C&T consider (e.g., feedforward neural networks) produce a *single* output, given each input, rather than a conditional probability distribution over all possible outputs.[1] Moreover, C&T state that if a pattern (x, y) has been encountered, then $P(y|x) = 1$ (sect. 2, para. 4), which indicates that they assume that the conditional probability distribution is degenerate – that is, for each input there is a single output. So they appear not be concerned with learning arbitrary conditional probability distributions, but rather with learning *functions* from input to output.

2. All conditional probability distributions are Type 1 learnable. C&T say a distribution to be learned "$P(y|x) = p$ might be [type-1] justified if . . . $P(y|x') = p$, where x' is some selection of values from input-vector x. . ." (sect. 2, para. 4). Suppose x' is the selection of *all* values of x – that is, $x' = x$. Then it trivially follows that $P(y|x) = p$ if and only if $P(y|x') = p$. That is, all conditional probability distributions, including as a special case all functions (including the uncomputable functions), are type-1 learnable.

Note that adding the stipulation that x' cannot include all of x does not help. This can be circumvented by adding irrelevant "dummy" values to each input vector (e.g., a string of 0s) – the learning problem is now just as hard as before. Then the selection x' does not take all elements of the input; it ignores the dummy values. But as before $P(y|x) = p$ if and only if $P(y|x') = p$.

3. The problem of novel outputs. From the above, it seems that C&T's definition does not capture their intuitive notion successfully. From the examples they give, it seems that they intend that $P(y|x')$ is not an arbitrary probability distribution, but rather that it is estimated from frequencies in the input data by $F(y, x')/F(x')$, where $F(x')$ is the number of occurrences of patterns which match x on the selection x' of values, and $F(y, x')$ is the number of such patterns associated with output y.

But this definition is also inadequate in general, because it means that any novel output y_{novel} must be assigned probability 0, because $F(y_{novel}, x') = 0$, precisely because y_{novel} has not occurred in the training set. This means that the class of type-1 problems is very restrictive. It does not include the *identity* function (in which each input is mapped to a different and hence novel output).

C&T face a dilemma. If they follow their stated definition, then all conditional probability distributions are type-1 learnable. If they follow the frequency-based analysis they use in the text, then no conditional probability distribution which assigns a nonzero probability to any unseen output is Type 1 learnable, which seems drastically restrictive.

Furthermore, the frequency-based approach also faces the problem that probabilities can never be estimated exactly from a finite amount of data, and therefore that the $F(y, x')/F(x')$ will not in general equal $P(y|x') = p$. The best that such an estimate can be is probably approximately correct, in some sense (e.g., Valiant 1984).

4. What does type-2 learning mean? C&T say a distribution to be learned "$P(y|x) = p$ might be [Type 2] justified if . . . $P[y|g(\in X) = z] = p$, where g is some arbitrary function, $\in X$ is any seen input, and z is the value of function g applied to x." (sect. 2, para. 4).

This formulation is difficult to interpret, because it uses notation in an unconventional way. But from the later discussion, the appropriate interpretation appears to be this: the function g maps some subset S of previously seen inputs onto a common output, z. We estimate the conditional probability (presumably that which C&T call "$P[y|g(\in X) = z] = p$") by the number of members of S which produce output y, divided by the total number of members of S.

As with type-1 problems, this means that the conditional proba-

bility of all novel outputs must be zero for a problem to be type-2 learnable, for the same reason: the frequency count for novel outputs is necessarily 0. So the identity function is not type-2 learnable either.

But worse, *all* the *non*novel outputs can be justifiably predicted with probability 1. Suppose that a previous input, x_{prev} was paired with the output y_{prev}. Then define g such that $g(x) = z$ (where x is the novel input), and $g(x_{prev}) = z$; $g(x_{rest}) = z + 1$, for all other previously seen inputs x_{rest}. g is a "recoding" of the inputs that classifies the novel input x with a single past input x_{prev}. The subset, S, defined above, has one member, which produced output y_{prev}, so that the estimated conditional probability is $1/1 = 1$. Hence, the arbitrary output y_{prev} is justifiably predicted with probability 1. An analogous argument extends not just to a single novel x, but to all possible novel x. In short, any function whatever which generalizes from the seen instances to the unseen instances is type-2 learnable, even the noncomputable ones (so long as there are no novel outputs).

Note that type-2 problems appear to have the same (rather bizarre) scope as type-1 problems. They are both too broad and too narrow in the same way.

NOTE

1. The output of neural networks can be viewed as a probability distribution over possible outputs if, for example, outputs are binary and intermediate values are interpreted as probabilities (e.g., Richard & Lippman 1991). A different approach assumes that outputs are distorted (for example by Gaussian noise). This is useful in understanding learning in Bayesian terms (Mackay 1992). Moreover, some networks implicitly produce conditional probability distributions by generating a distribution of outputs over time (e.g., the Boltzmann machine; Hinton & Sejnowski 1986). None of these approaches seems relevant to C&T's discussion.

Parity is not 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-isis.ecs.soton.ac.uk

Abstract: Uninformed learning mechanisms will not discover "type-2" regularities in their inputs, except fortuitously. Clark & Thornton argue that error back-propagation only learns the classical parity problem – which is "always pure type-2" – because of restrictive assumptions implicit in the learning algorithm and network employed. Empirical analysis showing that back-propagation fails to generalise on the parity problem is cited to support their position. The reason for failure, however, is that generalisation is simply not a relevant issue. Nothing can be gleaned about back-propagation in particular, or learning in general, from this failure.

1. Introduction. Clark & Thornton (C&T) argue that many learning problems involve "type-2" mappings, characterised by "attenuated existence in . . . training data." Thus, their discovery by an uninformed learning device (such as back-propagation) presents intractable problems of search. Once serendipitously found, however, the type-2 mappings can be exploited in further, modular learning. It is hard to argue against the principle of such recoding playing an important part in learning: indeed, it is almost a truism in cognitive science. Rather, I wish to show that C&T's detailed supporting arguments based on the empirical inability of back-propagation to generalise on the classical parity problem are mistaken.

2. Parity, generalisation, and mind reading. As with many others, my interest in neural computing blossomed when I obtained McClelland and Rumelhart's *Explorations in parallel distributed processing* in 1988. As its advertised purpose was to encourage experimentation, I tried to get the **bp** program to generalise on the 2-variable parity (XOR) problem. Given a network with 2 input nodes, 2 hidden nodes, and a single output, together with the first three lines of the truth table:

x_1	x_2		y
0	0	⇒	0
0	1	⇒	1
1	0	⇒	1

could **bp** generalise to find the missing line:

$$1 \quad 1 \quad \Rightarrow \quad 0\,?$$

I soon realised that was a silly thing to expect. How could *any* machine learning procedure generalise on this problem? The y-output for the unseen mappings is entirely arbitrary and hence unpredictable from the three seen mappings, although the unconditional probability $P(y = 1) = 0.67$ on C&T's argument favours a 1 output corresponding to a learned OR solution. Expecting the back-propagation network to generalise consistently to the XOR function – rather than the simpler OR function – *solely* on the basis that this is what in the experimenter's mind, amounts to expecting the network to be a mind-reader. Parity is not a generalisation problem! Yet this seems to be a cornerstone of C&T's thesis. To quote: "parity cases . . . do not really warrant any optimism concerning the chances of backpropagation . . . hitting on the right recodings." Thus, the inability to generalise on the parity problem is taken to have implications for cognition when, clearly, it does not.

This inability is apparently well-known to Elman who writes (1993, p. 85): "If the fourth pattern (for example) is withheld until late in training, the network will typically fail to learn XOR. Instead, it will learn logical OR since this is compatible with the first three patterns." Presumably, by "compatible" he means the unconditional probabilities favour the OR solution. As pointed out above, however, the polarity of the output for the unseen input combination is arbitrary. To this extent, both XOR and OR are equally "compatible" with the first three patterns.

I ran several simulations on the first three patterns using **bp**. In 20 out of 20 repetitions, the linearly-separable OR solution was always found. The hidden-layer representation was always that of one unit having weights and biases corresponding to an OR separating "hyperplane" like that labelled *line 1* on Figure 1 while the other "did nothing," that is, its weights and biases corresponded to a line which did not separate the input patterns at all. While there is a finite chance that this "do nothing" line just

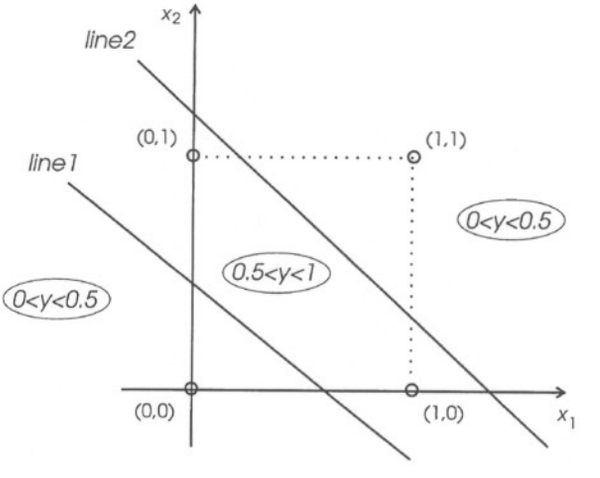

Figure 1 (Damper). Possible separating "hyperplanes" for the 2-input parity problem, corresponding to the decision boundaries formed by the two hidden units. In this simple case, the hyperplanes are lines.