Trading Spaces: Computation, Representation and the Limits of Uninformed Learning*

Andy Clark†
Philosophy/Neuroscience/Psychology Program
Washington University in St Louis
Campus Box 1073
St Louis
MO-63130
USA
Email: andy@twinearth.wustl.edu

Chris Thornton
Cognitive and Computing Sciences
University of Sussex
Brighton
BN1 9QH
U.K.
Email: Chris.Thornton@cogs.sussex.ac.uk

July 19, 1994

Abstract

It is widely appreciated (e.g. Marr, 1982) that the difficulty of a particular computation varies according to how the input data are presented. What is less well understood is the effect of this computation/representation trade-off within familiar learning paradigms. We argue that existing learning algorithms are often poorly equipped to solve problems involving a certain type of important and widespread statistical regularity, which we call 'type-2 regularity'. The solution in these cases is to trade achieved representation against computational search. We investigate several ways in which such a trade-off may be pursued including simple incremental learning, modular connectionism, and the developmental hypothesis of 'representational redescription'. 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.

*Research on this paper was partly supported by a Senior Research Leave fellowship granted by the Joint Council (SERC/MRC/ESRC) Cognitive Science Human Computer Interaction Initiative to one of the authors (Clark). Thanks to the Initiative for that support.

†The order of names is arbitrary.

Long abstract

Some regularities enjoy only an attenuated existence in a body of training data. These are regularities whose statistical visibility depends on some systematic re-coding of the data. The space of possible re-codings is, however, infinitely large - it is the space of applicable Turing machines. As a result, mappings which pivot on such attenuated regularities cannot, in general, be found by brute force search. The class of problems which 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 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 re-coding 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 viz. the maximal exploitation of states of representation already achieved by prior (type-1) learning so as to reduce the amount of subsequent computational search. Such exploitation both characterises and helps make unitary sense of a diverse range of mechanisms. These include simple incremental learning (Elman, 1993), modular connectionism (Jacobs, Jordan and Barto, 1991), and the developmental hypothesis of 'representational redescription' (Karmiloff-Smith, 1979, Karmiloff-Smith, 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.

Keywords

Learning, connectionism, statistics, representation, search

Introduction. The Limits of Uninformed Learning.

> In any multilayered PDP System, part of the job of intermediate layers is to convert input into a suitable set of intermediate representations to simplify the problem enough to make it solvable. One reason PDP modelling is popular is because nets are supposed to learn intermediate representations. They do this by becoming attuned to regularities in the input. What if the regularities they need to be attuned to are not in the input? Or rather, what if so little of a regularity is present in the data that for all intents and purposes it would be totally serendipitous to strike upon it? It seems to me that such a demonstration would constitute a form of the poverty of stimulus argument. (Kirsh, 1992, p.317)

Kirsh's worry about regularities which enjoy only a marginal existence 'in the input' is, we suggest, an extremely serious one. In this paper we offer a statistical framework which gives precise sense to the superficially vague notion of such marginal regularities. We show that problems involving such marginal regularities are much more pervasive than many working connectionists optimistically imagine. And we begin the task of developing a unified framework in which to understand the space of possible solutions to such problems; a space centered around the key notions of incremental learning and representational trajectories (Elman, 1993; Clark, 1993, ch.7).

Having emphasised the foundational role which an understanding of these notions must play in cognitive science, we go on to argue that a wide variety of superficially distinct ploys and mechanisms can be fruitfully understood in these terms. Such ploys and mechanisms range from simple evolved filters and feature-detectors all the way to complex cases involving the use and re-use of acquired knowledge. The goal, in every case, is to systematically re-configure a body of input data so that computationally primitive learning routines can find some target mapping, i.e. to trade representation against computation. Uninformed learning — learning which attempts to induce the solutions to problems involving these 'marginal regularities' solely on the basis of the gross statistics of the input corpus — is, we show, pretty much doomed to failure. But the variety of ways in which a learning device can circumvent such problems is surprisingly large and includes some quite unexpected cases.

The strategy of the paper is as follows. We begin (section 1) by distinguishing two kinds of statistical regularity. This distinction (between what we term 'type-1' and 'type-2' mappings) gives precise sense to Kirsh's notion of robust versus 'marginal' exemplification of regularities in specific bodies of data. We go on (section 2) to look at two case studies. These concern a simple animat behavior called 'conditional approach' and the grammar acquisition task studied by Elman (1993). The final substantive section (section 3) develops a broader perspective on the issues and canvasses a variety of partial solutions to the problems posed by type-2 mappings. These solutions build on and extend Elman's (1993) perspective on incremental learning, and relate it to other strategies for maximising the usefulness of achieved states of representation.

1 'Marginal' Regularities and the Complexity of Learning.

Kirsh's question concerned regularities whose presence 'in the data' was so weak as to make discovery 'totally serendipitous'. But how should we understand this notion of regularities which are in some way present in the data and yet threaten to remain invisible to any uninformed learning device? One way to give concrete sense to such a notion is to distinguish between two ways in which a regularity can be statistically present in a training set. In the first (basic) way the regularity may be discovered by sampling over the input data. In the second (derived) way, the regularity may emerge only as a result of some systematic re-coding of the input features. In the latter case, it is unlikely that any uninformed learning device (one which does not receive some extra prompt or push to enable it to choose the riqht re-coding out of an infinity of possible re-codings) will discern the regularity.

This account of the idea of 'marginal regularities' can be made statistically precise. Let us treat the process of learning implemented by some arbitrary uninformed learning mechanism as the attempt to acquire a target input/output mapping. To have any chance of success the learner requires some source of feedback regarding the mapping to be acquired. In the much studied supervised learning scenario, this feedback takes the form of a set of training examples taken from the target mapping. The learner's aim is to arrive at the point at which it is able to map any input taken from the mapping onto its associated output. In more general terms, the learner's aim is to be able to give a high probability to the correct output for an arbitrary input taken from the mapping.

If the learner is to have any chance of achieving this goal, the feedback it receives must contain information which justifies the assigning of particular probabilities to particular outputs. Learning is essentially the process of discovering and exploiting such justifications. To understand the nature of the process we need to analyze the ways in which supervisory feedback can provide justifications for assignments of particular probabilities to particular outputs. The problem, in general, is thus

From a source of feedback, i.e., a set of input/output examples

Produce an implementation of an appropriate, conditional probability distribution over outputs; i.e., produce an implementation that will identify the value of $P(y|x)$, the probability that y is the correct output for input x, for any x and y taken from the target input/output mapping.

Given this specification, any analysis of the acquisition task must show how the probability assignments produced are justified by the input/output examples. Ignoring the trivial case in which $x \rightarrow y$ is a member of the example set (which trivially justifies the conclusion that $P(y|x) = 1$), there remain three substantial forms of justification. $P(y|x) = p$ might be justified if

(1) $P(y) = p$,

(2) $P(y|x') = p$, where x' is some selection of values from input-vector x, or

(3) $P(y|g(x)) = p$, where g is some arbitrary function.

This holds for any value of p. Thus we know that any acquisition mechanism must exploit some combination of these three forms of justification. In the absence of any special background knowledge, the complexity of exploiting a particular probability (as a justification) is related to the size of its distribution. This prompts us to split the justification forms into two basic categories: the 'direct' forms $P(y)$ and $P(y|x)$ and the 'indirect' form $P(y|g(x))$.

$P(y)$ and $P(y|x)$ are direct in the sense that that their distributions can be obtained by direct sampling of the inputs. $P(y|g(x))$ is indirect since it can only be obtained following identification of the 'recoding' function g. The significance of this distinction relates to complexity. Both of the direct forms have finite distributions. The indirect form on the other hand has an infinite and highly 'explosive' distribution since it is grounded in the space of computable functions. Problems which involve exploiting either of the two direct forms thus have lower theoretical complexity than problems which involve exploiting the indirect form.

The added complexity in the indirect case consists in the need to discover a recoding of the training data, i.e., to discover the function g on which the justification depends. Such a function must depend on non-absolute values of its argument vector since otherwise, it would follow that there would be some $P(y|x')$ such that

$$P(y|g(x')) = P(y|x')$$

and the supposed indirect justification would thus be reduced to a direct justification. From this we can infer that problems which require the exploitation of indirect forms of justification involve finding functions which test (or measure) *relational properties* of the input values.[1] In what follows we will call problems which are only solvable through exploitation of indirect justifications 'type-2' and all others 'type-1'.

To illustrate this notion of indirect justification, consider the following training set. This is based on two input variables (x1 and x2) and one output variable (y1). There are six training examples in all. An arrow separates the input part of the example from the output part.

```
x1  x2      y1

1   2  -->  1
2   2  -->  0
3   2  -->  1
3   1  -->  0
2   1  -->  1
1   1  -->  0
```

A variety of direct justifications are to be found in these training data. For example, we have the unconditional probability $P(y1 = 1) = 0.5$, and the conditional probability $P(y1 = 1|x1 = 2) = 0.67$. These probabilities, and in fact all the probabilities directly observed in these data, turn out to be close to their chance values. Indirect justifications are to be found via some recoding function g. In the case at hand imagine that the function effectively substitutes the input variables in each training pair with a single variable whose value is just the difference between the original variables. This gives us a set of derived pairs as shown below (the value of x4 here is the difference between the values of x1 and x2).

```
Original pairs               Derived pairs (x4 = |x1-x2|)

x1  x2      y1               x4      y1
```

[1] This is a satisfying rediscovery of the old AI rule which states that 'relational learning is hard', cf. (Dietterich, London, Clarkson and Dromey, 1982).

```
1   2  -->  1              1  -->  1
2   2  -->  0              0  -->  0
3   2  -->  1              1  -->  1
3   1  -->  0              2  -->  0
2   1  -->  1              1  -->  1
1   1  -->  0              0  -->  0
```

Note how the recoding has produced data in which we observe a number of extreme probabilities relating to the output variable y1, namely P(y1=0|x4=0)=1, P(y1=1|x4=1)=1 and P(p1=0|x4=2)=1. The recoding thus provides us with indirect justification for predicting y1=0 with a probability of 1, if the difference between the input variables is 1. It also provides us with indirect justification for predicting y1=1 with a probability of 1, if the difference between the input variables is either 2 or 0. In short, we have indirect justification for the output rule 'y1=1 if x4=1; otherwise y1=0'. Kirsh's 'marginal regularities' we conclude, are precisely those whose justification is in our sense indirect. They thus involve (1) deriving a recoding of the training examples and (2) deriving probability statistics within the recoded data.

The number of indirect justifications is the number of direct justifications (derivable from the relevant recoded data) plus the number of possible recodings of the data. The number of possible recodings is simply the number of distinct Turing machines we can apply to those data. There are infinitely many of these. Thus the space of indirect justifications is infinitely large. To hit on the right one by brute-force search would indeed be 'serendipitous.'

Thus consider the much studied case of learning parity mappings (see e.g., Rumelhart Hinton and Williams (1986), Hinton and Sejnowski (1986).) These are indeed cases of type-2 (relational) input/output mappings. The input/output rule for a parity mapping is simply that the output should be 1 (or true) just in case the input vector contains an odd number of 1s (or, in general, an odd number of odd values). The complete mapping for the third-order, binary-valued parity problem (i.e., 3-bit parity) is as follows.

```
x1  x2  x3        x4

1   1   1   -->   1
1   1   0   -->   0
1   0   1   -->   0
1   0   0   -->   1
0   1   1   -->   0
0   1   0   -->   1
0   0   1   -->   1
0   0   0   -->   0
```

Every single conditional probability for this mapping (for values of the output variable x4) is at its chance level of 0.5. Since the probabilities for parity mappings are *always* like this they cannot be solved by exploiting direct justifications. Parity problems are thus always pure type-2.

Yet parity problems, as is well-known, can be solved by, e.g., backpropagation learning. Moreover such solutions are typically said to involve the algorithm deriving what can be thought of as an internal re-coding scheme. However, we should not overestimate the generality of such solution methods. All of them introduce restrictive assumptions about the nature of the type-2 regularity to be discovered. Backpropagation for example effectively assumes that the required re-coding can be expressed in terms of the user-fixed architecture of semi-linear transfer functions, and that it can be discovered by the gradient descent method embodied in the learning algorithm. If the assumption is invalid, the learning necessarily fails.

This may help to explain why backpropagation often fails to solve low-order parity problems when they are presented as generalization problems, i.e., when some cases are held back for testing purposes. The graph shown in Figure 1 was produced from an empirical survey that involved running standard backpropagation (Rumelhart, Hinton and Williams, 1986) on 4-bit parity generalization problems (with four, randomly selected cases used as unseens) using a wide range of internal architectures. All the curves in the upper half of the graph are error profiles[2] for the testing set of four cases. All the curves in the lower half of the graph are error profiles for the training set. There are 32 pairs of curves in all although many of them are bunched together in two clumps at the far left of the graph.

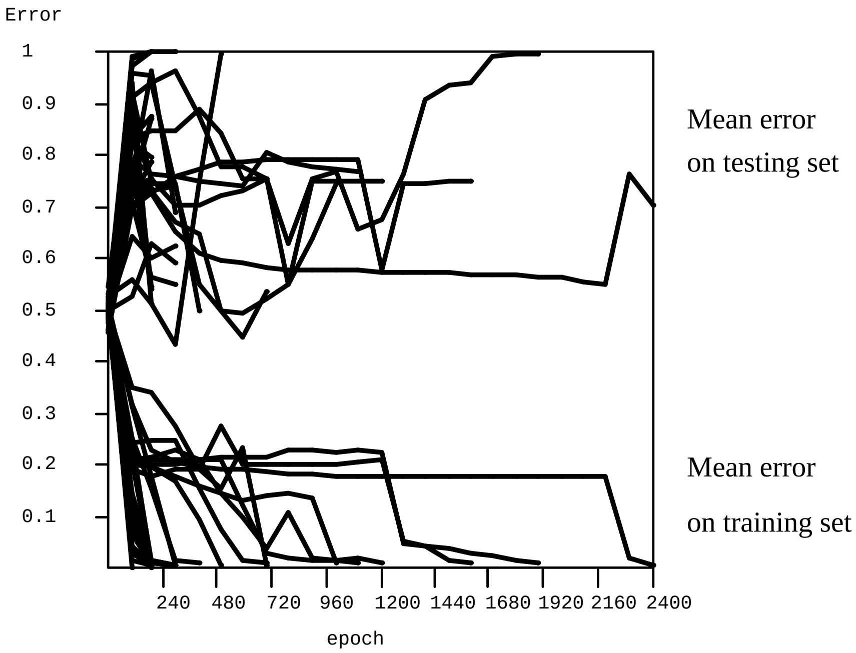

Figure 1:

Rather obviously, generalization over the testing cases was never observed to improve much beyond the chance level in any of the runs recorded. But the point to note is that the training-set error profiles typically go to zero rather rapidly (usually within 1000 epochs). This tells us that the generalization failure occurs in the context of perfectly ‘successful’ learning, i.e., perfect acquisition of the training cases. This is a particularly concrete sort of generalization failure since it cannot be overcome by increasing the amount of training or by changing parameters. Once a supervised algorithm has learned the training cases perfectly, generalization grinds to a halt. As far as the algorithm ‘knows’, it is *already* producing perfect performance.

Parity cases, we conclude do not really warrant any optimism concerning the chances of backpropagation

[2] The error measure is the average difference between actual and target activations. For these experiments we used standard learning parameters; i.e., a learning rate of 0.2 and a momentum of 0.9.

in a multilayer net hitting on the right re-codings to solve type-2 cases. Instead as we move towards larger scale, more realistic cases, we find a robust pattern of failure. In the next section we consider two such cases. The first concerns a simple robotics-style problem called ‘conditional-approach’. The second concerns learning about grammatical structure.

2 Two case studies

In order to investigate the difficulty of type-2 learning problems in an experimental setting, we conducted a comparative survey focussed on a superficially simple animat behavior called 'conditional approach'. The production of this behavior in an animat requires a proximity sensing system of some sort and motor abilities enabling forward and rotational movements. The behavior involves moving in on any relatively small object in the sensory field but standing clear of (i.e., *not* moving in on) any large object.

The behavior was investigated using computer simulations. The simulations used a 2-dimensional, rectangular world and a single animat. This had two free-wheeling castors situated fore and aft and two drive wheels situated along the central, latitudinal axis (see Figure 2). The animat was equipped with a range-finding system. This sensed the proximity of the nearest object — subject to 10% noise — along seven rays, evenly spaced within a 100 degree, forwards facing arc (see further details below).

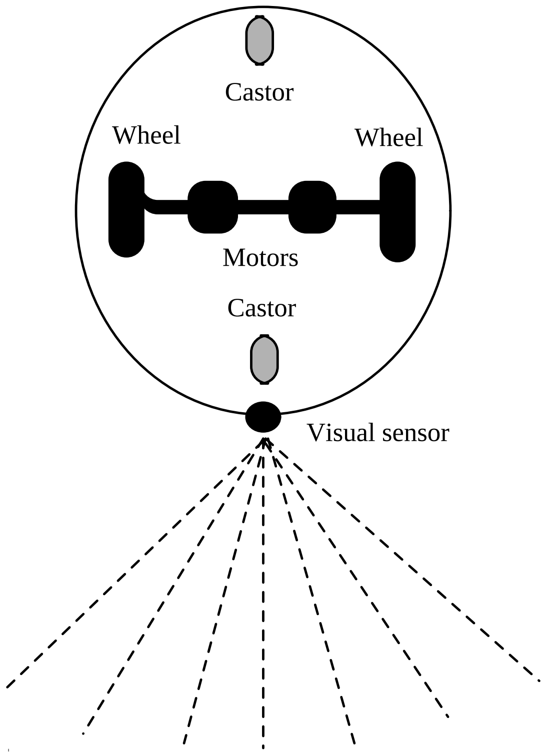

Figure 2:

The plan view shown in Figure 3 illustrates the basic simulation setup. The animat, situated in the lower part of the space, is represented as a small box with an arrow pointing in its direction of motion. The seven dashed lines are its probe rays. The boundaries of the space — here shown as unbroken lines — are actually transparent to the animat. Thus, in the situation shown, the animat senses only the circular blob directly ahead of it. That is to say, within its seven proximity inputs, the two associated with the rays intersecting the blob will be relatively high but the other five will be zeros indicating 'no object sensed'.

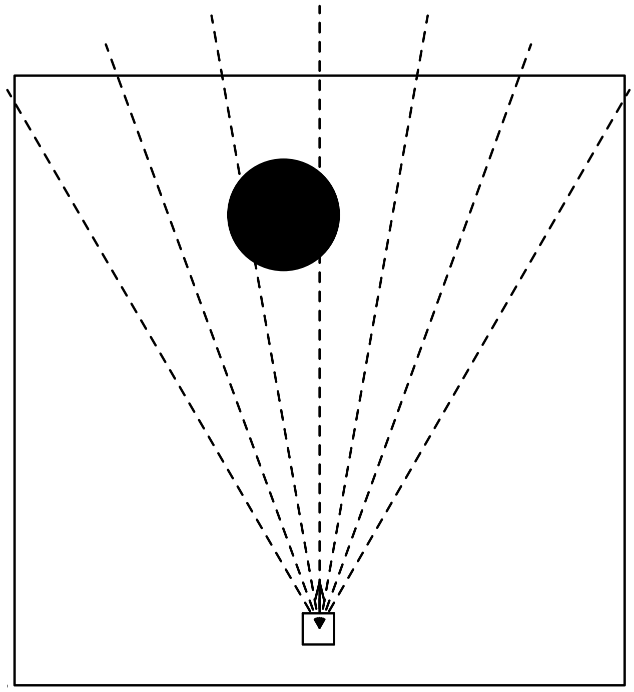

Figure 3:

The aim of the empirical investigation was to see how well supervised learning algorithms performed when used to train an animat to perform conditional-approach. To obtain training sets for the learning we hand-crafted an animat to produce perfect conditional-approach behaviour and then sampled its reactions during simulation runs. This involved interrupting our simulation program in the middle of each time cycle and recording the sensory input received by the animat at that point, and the amount of drive being sent to the two wheels. The input/output pairs produced gave us the required training set.

The conditional-approach behavior entails producing three, basic behavioral responses to four scenarios. With no object appearing in the sensory field the animat must swivel right ten degrees. With an object appearing at long-range, or a *small* object appearing at close-range the animat must execute a forwards move towards that object. (This might or might not involve a change in direction.) With a large object appearing at close-range the animat should remain stationary.

The inputs from the sensory system were represented (for purposes of training) in the form of real numbers in the range 0.0-1.0. The inputs formed a normalized measure of proximity and embodied 10% noise. The amount of drive applied to the two wheels in each simulation step was represented in the form of two real numbers, also in the range 0.0-1.0. Thus, a full right turn with no forwards motion would appear in the training set as the pair <1.0,0.0> (given the assumption that the first number sets the drive on the left wheel and the second number the drive on the right wheel).

The use of standard-format training sets enabled us to test the performance of any supervised learning algorithm on the conditional-approach problem. In practice we tested the performance of a wide range of algorithms including ID3 (Quinlan, 1986) and C4.5 (Quinlan, 1993), feed-forward network learning algorithms of the backpropagation family including 'vanilla' backpropagation (Rumelhart, Hinton and Williams, 1986), a second-order method based on conjugate-gradient descent (Becker and Cun, 1988) and a second-order method based on Newton's method called 'quickprop' (Fahlman and Lebiere, 1990). We also tested a constructive network learning method called 'cascade-correlation' (Fahlman and Lebiere, 1990) and a classifier/genetic-algorithm combination based on Goldberg's 'simple classifier system' (Goldberg, 1989).

All the network algorithms tested operate by modifying the connection weights in a fixed, non-recurrent network of artificial neurons (using the standard logistic activation function). The efficiency of network learning is determined by feeding in novel inputs to the network and seeing what outputs are generated after the activation has propagated across all the relevant connections. When applying network learning algorithms the user must decide the internal architecture of the network[3] and, in some cases, the learning and momentum rate. When testing the various network learning algorithms we experimented with a range of two-layered, feed-forward architectures (with complete inter-layer connectivity) but found that the best performance was obtained using nine hidden units; i.e. we settled on a 7-9-2 feed-forward architecture. All the results reported relate to this case.

The results were surprisingly robust. C4.5 and nearest-neighbors performed better on the learning task than the connectionist algorithms or the classifier system, but none of the algorithms provided satisfactory performance on this problem. In general, following training the animat would tend to either approach all objects (large or small) or no objects. It would only very occasionally produce the desired discrimination between large and small objects.

We measured the success of the training in several ways. First of all we measured conventional error rates (i.e. proportion of incorrect responses on unseens). However, these figures give a misleading impression of success. The majority of responses in the conditional-approach behavior do not entail making the crucial discrimination between large and small objects. They merely involve continuing rotatory behavior or moving further towards a small and/or distant object. A better performance measure is provided by sampling the frequencies with which the animat actually arrives at large and small objects. The former frequency we call the 'nip frequency', the latter the 'meal frequency'. These frequencies tend to show the extent to which the animat's behavior embodies the necessary size discrimination.

Our main results are summarized in Table 1.

	Error rate	Meal freq.	Nip freq.
NN	0.161	0.117	0.191
quickprop	0.221	0.201	0.321
c4	0.233	0.479	0.371
CS	0.344	0.251	0.275

Table 1:

The lowest error rate on the testing cases was 0.161 (16.1%) and this was produced by the nearest-neighbours algorithm (NN). This figure seems low but actually reveals relatively poor performance

[3] The configuration of input and output units is fixed by the learning problem. When testing standard backpropagation we found that a learning rate of 0.2 and a momentum of 0.9 gave best results and these were the settings used in all the cases reported. When testing iterative learning algorithms (i.e., the network learning algorithms) we ran the algorithms for a minimum of 100,000 epochs of training (i.e., 100,000 complete sweeps through the entire training set).

(for reasons explained above). The same goes for the other error rates shown. The columns headed 'Meal freq.' and 'Nip freq.' show the 'meal' and 'nip' frequencies respectively for the various simulated animats. Note that the trained animats do quite poorly, with the quickprop, NN and CS animats achieving nip-frequencies in excess of the meal-frequencies.

The reason conditional approach, despite its surface simplicity is so hard to learn is that it is a classic type-2 problem. What makes it type-2 is that the input/output rule for this behavior is (relative to the input coding) inherently *relational*. The robot must learn to produce behaviors which depend on the ratio between apparent closeness and apparent width. Successful performance can be straightforwardly achieved by a hand re-coding in which this ratio is calculated and made explicit.

Moving up the scale of task-complexity, we next consider Elman's recent and important work on grammar acquisition (Elman, 1993). Elman studied a grammar acquisition problem in which a simple recurrent net was required to learn a grammar involving features such as verb-subject number agreement and long distance (cross-clausal) dependencies . He discovered that ordinary backpropagation learning was unable to prompt a net to acquire knowledge of the grammar. But success could be achieved in either of two ways. First, it was possible successfully to train a net if the training data was divided into graded batches beginning with simple sentences and progressing to more complex (multi-clausal) ones. Second, success could be achieved by providing the network with a limited initial window of recurrency (re-setting the context units to 0.5 after every 3rd/4th word) which was allowed to increase as training progressed. In the latter case there was no need to batch the training data as the restricted initial memory span in effect filtered out the mappings involving cross-clausal dependencies and allowed in only the simpler constructions: the data was thus 'automatically sorted'. It is clear that the two techniques are functionally equivalent and that the reason that they work is, as Elman comments, that

> The effect of early learning ... is to constrain the solution space to a much smaller region. The solution space is initially very large, and contains many false solutions (in network parlance, local error minima). The chances of stumbling on the correct solution are small. However, by selectively focussing on the simpler set of facts, the network appears to learn the basic distinctions — noun/verb/relative pronoun, singular/plural etc. — which form the necessary basis for learning the more difficult set of facts which arise with complex sentences. (Elman, 1993, p.84)

By 'false solutions' Elman means the extraction of the wrong regularities, i.e. finding spurious type-1 regularities which will fail to determine successful performance on unseen cases. Both of Elman's solution techniques force the net to learn certain basic mappings first (e.g. verb/subject number agreement). Once this knowledge is in place, the more complex mapping-tasks (e.g. agreement across an embedded clause) alter in statistical character. Instead of searching the explosive space of possible relations between input variables, the net has been alerted (by the simpler cases) to a specific relation (agreement) which characterizes the domain.

Elman-style incremental learning works because the early learning alters the shape of the subsequent search space. In a sense, once the early learning is in place, the device is no longer uninformed. Instead, it benefits from a substantial bias towards solutions which involve re-coding inputs in terms of e.g. verb, subject, number (singular or plural) etc.. And, relative to such a re-coding, the otherwise invisible higher-level grammatical regularities pop out. In this way the incrementally trained net avoids what Elman calls the Catch-22 situation in which:

> the ... ... crucial primitive notions (such as lexical category, subject/verb agreement etc.) are obscured by the complex grammatical structures ... [and] the network is also unable to

> learn about the complex grammatical structures because it lacks the primitive representations necessary to encode them. (Elman, 1993, p.94)

Learning these 'primitive representations' is learning a specific re-coding scheme; one which later simplifies the task of accounting for more complex grammatical regularities such as long-distance dependencies. Relative to the new encodings such elusive regularities are transformed into directly observable frequencies in the (now re-coded) data set. The need for such re-coding, in the grammar case, was long ago demonstrated. Here, we merely recapitulate Miller and Chomsky's (1963) observation (also cited by Elman (1993) p.86) that regularities such as long distance dependency cannot (on pain of unimaginably large search) be learnt by reliance on co-occurrence statistics defined over individual words i.e. defined over the original input features. By contrast, once the input is viewed through the lens of a re-coding scheme involving features such as subject and number (singular/plural) even a 17-word displaced agreement relation will show up as a mere second order direct frequency, i.e. one involving the absolute values of 2 variables. What Elman in effect demonstrates is that this re-coding scheme can itself be learnt as a function of directly sampled frequencies provided the early training data is either carefully selected (as in the 'graded batches' technique) or effectively filtered (as in the memory-restriction case). In these ways a problem whose full expression poses an intractable type-2 learning problem can be reduced to a developmental sequence of tractable type-1 mappings. Moreover, this can be achieved without going so far as to build in the required re-coding bias at the very outset. The full nativist solution favoured by Chomsky is, in such cases, not compulsory.

Kirsh's observation that target domains involving 'marginal regularities' represent a version of the poverty of the stimulus argument is thus correct. But — perhaps surprisingly — such domains (now fixed as those built around type-2. indirect frequency effects) sometimes yield to a temporal sequence of type-1 learning episodes. In the next section we consider the potential scope and power of this basic strategy and some related ploys and techniques. The upshot is, we believe, a much more unified and theoretically well-grounded idea of the role of a variety of developmental, computational and even cultural and historical constraints and processes. The combined effect of such constraints and processes is to enable us to achieve a kind of cognitive hyper-acuity: to regularly and robustly solve types of problem whose statistical profiles are prima facie cause for despair.

3 Putting Representations to Work.

The real trouble with type-2 learning problems is, we saw, that they cannot in general be solved by any kind of uninformed search. The trouble with informed search, of course, is identifying the informant. In many cases, positing better informed search simply begs the question. Just where did those feature detectors, or those biases towards trying such and such a re-coding first, come from? Unless we are comfortable with a very heavy-duty nativism and an amazing diversity of task-specific on-board learning devices[4], we will hope in addition to uncover at least a few more general strategies or ploys. Such strategies (tricks, ploys, heuristics) cannot be problem specific, since this would be to fall back on the full nativist solution. Instead, they will constitute general techniques aimed at maximising the ways in which achieved representations can be traded against expensive search. They will thus maximise the chances of a learner successfully penetrating some random type-2 domain. Elman has already alerted us to one such trick - the provision of an extended developmental period whose early stages are characterised by weaker computational resources able to act 'like a protective veil, shielding the infant from stimuli which ... require prior learning to be interpreted' (Elman, 1993, p.95). What other strategies might reduce the search space for type-2 cases?

Recall that the key to success, when faced with a type-2 case, is to use achieved representations to reduce the complexity of subsequent search. This is the operational core of incremental learning in which the bare input data is effectively re-coded through the lens of the early knowledge. Such a re-coding, in e.g. the cases studied by Elman, is, however, task-specific. That is to say, the achieved representations (the results of the early learning) are only available for use along a single, fixed processing channel. Once the system has exploited the early knowledge to achieve success at the adult grammar, the enabling resource (the 'building-block' knowledge) is in effect used up. (There are ways around this, but they all require either extensions of the basic connectionist model (e.g. wholesale copying of the early net) and/or are restricted to the rare cases in which the dimensionality of the inputs is identical for both the original task and any later ones - for a full discussion see Clark and Karmiloff-Smith (1993), Karmiloff-Smith and Clark (1993).)

One useful trick would thus be to somehow 'free-up' any acquired representational resources so as to allow such resources to participate in a multitude of different kinds of future problem-solving. Representational resources originally developed to solve a problem P in a domain D would, in such a case, be exploitable in an open-ended number of future learning episodes. Whereas, in the Elman example the representational trajectory is a one-off (one sequence of learning culminating in a successful network), we are now imagining cases in which one set of early 'building block' knowledge can be used as often as required and can thus participate in multiple representational trajectories (temporal sequences of learning). Achieved representational resources, on such a model, do double duty as general purpose feature detectors which can be used to re-code subsequent inputs in an effort to unmask lurking type-2 regularities.

Complete and unbounded mobility and re-useability of existing knowledge is probably impractical. But partial mobility is a realistic and realizable goal. One way of attaining it is to pursue a more determinedly modular connectionist approach. Thus Jacobs, Jordan and Barto (1991) describe a system which comprises a variety of architecturally distinct sub-nets. These sub-nets compete to be allowed to learn to represent a given input pattern. Whichever net, early on in the training, gives the output closest to the target, is allowed to learn that pattern. In the trained-up system a gating network selects which sub-net should come into play to yield the output for a given input. In a task such as multiple-speaker vowel recognition (Jacobs, Jordan, Nowlan and Hinton, 1991) such a modular system can avoid the intractable task of finding a single position in weight space capable of solving the problem for all

[4] Gallistel (1994) is an eloquent defence of just such a profligate nativism.

types of voices and instead tackle a set of more tractable ones, viz. one sub-net learns to identify vowels in children's voices, another in men's and another in women's. (See also Churchland and Sejnowski, 1992p.130.) Such modularization is one possible key to the flexible and multiple re-use of the valuable products of early learning. The goal, as noted above, is to ensure that a detector for some property P is not inextricably embedded into the solution to a single more complex problem, since P may be just the property or sensitivity which would render some other subsequently encountered, problem tractable. Assigning specific tasks to specific modules allows for the future re-use of a trained-up module in some other overall task (see Jacobs, Jordan and Barto, 1991).

An even more general version of this idea (concerning the benefits of flexible and multiple re-useability for achieved representations) is captured by Karmiloff-Smith's (1979, Karmiloff-Smith, 1992) 'Representational Redescription Hypothesis'. Karmiloff-Smith's claim is that a special and distinguishing feature of higher cognition is that it involves an endogenous drive to (a) seek increasingly general, flexible and abstract re-codings of achieved knowledge and (b) make those re-codings available for use outside the original problem domain. Such re-coding is, moreover, to be conservative in that the previous codings are never lost and can themselves be invoked as required.

Despite a frustrating lack of concrete mechanisms (but see Clark and Karmiloff-Smith (1993), Clark (1993) for some suggestions) the idea is attractive. For endogenous pressure to re-code is precisely self-generated pressure to explore continuously the space of incremental problem solutions without commitment to the solution of any specific problem. Each such re-coding may just happen to reduce a problem that was previously type-2 (and hence effectively outside the scope of individual learning) to a tractable type-1 incarnation. The learner will thus be engaged in a kind of continuous search for new problems insofar as each re-coding changes the shape of the space defined by the inputs and hence opens up new cognitive horizons. An individual, endogenously specified tendency to engage in representational redescription would thus amount to a natural injunction to persistently pull as much as possible into the space negotiable by our on-line weak type-1 learning methods. With direct task-driven exploration of type-2 spaces out of the question, evolution bestows on the individual a generic drive to code and re-code and re-re-code. Once again, we are trading spaces — using achieved representation to reduce the complexity of computation.

Such a tendency would also help offset a serious constraint on standard connectionist learning. This is what Elman (1993) calls the constraint of 'continuity of search'. The worry is that gradient descent search techniques impose a limitation viz. that the hypotheses to be considered (here, hypotheses are identified with locations in weight space) at time t1, cannot be 'wildly different' from those already under consideration at the previous processing cycle (time t). This is because of the nature of gradient descent learning itself; it explores a space by multiple restricted local weight updates. Hence 'learning occurs through smooth and small changes in hypotheses' (Elman (1993) p.91). But while this is true so long as we restrict our attention to the search performed by any single network, it is not true if we consider the use of multiple searches exploiting a variety of networks. Within a larger, more modular space, we can indeed explore 'wildly different' hypotheses in rapid succession. This would be the case if e.g. new inputs were at first gated to one sub-network and then, if that does not look promising (large error signal), gated to a wholly different sub-net and so on. Such sub-nets (as in the Jacobs, Jordan and Barto work) could encode very different states of achieved knowledge and hence provide a multitude of different 'lenses' to apply to the data. In such a manner, distant points in hypothesis space could indeed be successively explored. Networks of networks, comprising multiple, re-useable representational resources, may thus provide for more wide-ranging search and hence the maximal use of achieved representation.

It may even be useful (though clearly highly speculative) to consider public language and culture as large scale implementations of the same kind of strategy. For language and culture, we may suspect, provide exactly the kind of augmentation to individual cognition which would enable uninformed learning devices to trade achieved representation against computation on a cosmic scale. Public language may be seen as a ploy which enables us to preserve the fruits of one generation's or one individual's explorations at the type-1/type-2 periphery and thus quickly to bring others to the same point in representational space. Otherwise put, we can now have learning trajectories which criss-cross individuals and outstrip human lifetimes. In addition, we can (by grace of such cultural institutions as schooling) easily re-create, time and again, the kind of learning trajectory which leads to the solution of key complex problems. In these ways, the occasional fruits of good fortune (the discovery of a powerful input re-coding (a concept) or a potent sequence of training items) can be preserved and used as the representational base-line of the next generation's mature explorations. Language and culture thus enable us to trade achieved representation in any member of the species, past or present, against computation for all posterity. Given the largely fortuitous nature of the search for new representations, this is an advantage whose significance cannot be exaggerated.

It is interesting to compare this vision (of language and culture as a means of preserving representational achievements and extending representational trajectories) with that of Dennett (1993). Dennett usefully identifies a weak type of learning which he calls ABC learning and defines as the 'foundational animal capacity to be gradually trained by an environment'. He depicts standard connectionist learning as falling into this class and asks what leads us to outstrip the achievements of such ABC-learners. His answer is: the augmentation of such learning by the symbol structures of public language. (See also Dennett (1991) p.190, 220, 298-302.)

We think Dennett is almost right. He is right to depict language as a key factor in our abilities to frequently and repeatedly appear to exceed the bounds of ABC (or, as we would have it, type-1) learning. Yet in a very real sense there is, we believe, no other type of learning to be had. What looks like type-2 learning is in fact the occasional re-formulation of a type-2 problem in terms which reduce it to type-1. Language, we suggest, simply enables us to preserve and build on such reductions, insofar as the key to each reduction is an achieved re-representation of a body of data. But language is not, we think, the root of such re-representations. Instead, such re-representations must be discovered essentially by chance (perhaps aided by an endogenous, though undirected, drive to continuously seek re-codings of existing knowledge) in either individual or species learning. Language is a preserver both of representational discoveries and of useful learning trajectories. Language-users will thus indeed steer a profoundly deeper course into the type-2 problem space than anyone else, but for reasons which are, we suspect, a little more pedestrian than Dennett imagines.

Finally, we should note that cultural transmission opens up a new avenue of quasi-evolutionary selection (see e.g. Campbell (1974)). For it allows the production of artifacts which are increasingly well-adapted to human needs. One intriguing possibility is that public language, as a kind of cultural artifact, has evolved to fit the profile of the human learner. Such a hypothesis effectively inverts the nativist image in which we are adapted to the space of humanly possible languages. Indeed, the conjecture is that those languages all represent careful adaptations to us. Thus, for example, English may exhibit a morphological structure selected so as to 'pop out' when English sentences are heard by a typical learner viz. a child with a limited short-term memory and window of attention. If this were so, then it would be as if the learner had 'feature detectors' already in place, geared to re-coding the gross inputs in a peculiarly morphology-revealing way. Yet in fact, it would be the language whose morphological form had evolved so as to be revealed by processing biases which the system already exhibited.

Just such a conjecture has been explored by E. Newport in the guise of her 'less is more' hypothesis. The basic hypothesis is similar to Elman's 'starting small' idea - though it is targeted on learning about morphology. The idea is that young children's inability to process and recall complex stimuli actually allows basic morphological structures to 'pop out' of the gross data. Learners able to process and recall more complex stimuli would have to extract these morphological building blocks by computational means. For our purposes, however, it is the next step in Newport's argument which matters. For she goes on to ask whether the nice 'fit' between the basic morphological structure and children's limited early capacities to perceive complex stimuli is just a lucky coincidence? The answer, she hypothesizes, (Newport, 1990, p.25) is 'No'. The fit is no accident. But neither is it the case that the child's early capacities were selected so as to facilitate language learning. Instead (and here is the inversion we mooted earlier), the structure of the language was selected so as to exploit those early (and independent) limitations. The upshot is a situation in which it looks as if the child has on-board resources tailored to simplifying the language acquisition task. But in fact, it is (in a sense) the language that has the prior knowledge of the child, and not vice versa.

A similar manoeuvre may, we conjecture, be needed to insulate Elman's 'starting small' story[5] from a similar criticism. The worry is that the 'protective veil' of early limitations is of value only insofar as it allows through just the data needed to learn the basic representational 'lenses' which will later make tractable the search for the more complex grammatical regularities. But why should the early limitations window the gross data in just that way? One possible response, echoing Newport, is that the problem itself has evolved so as to fit the computational profile of the early learner. Had our early limitations been different, language itself would have evolved a different structure so as to exploit them. Many superficially type-2 problems, we conclude, may be tractable simply because the problem itself has evolved so as to exploit the inherent biases of human learning mechanisms. The game is rigged and the search is, in a very peculiar sense, informed.

[5]Elman (1993) discusses Newport's work. But strangely, he does not address the cultural-evolutionary conjecture which, we believe, is crucial to any complete defence of his model.

4 Conclusions. A Cautious Optimism.

From a determinedly statistical point of view, things looked bleak. Uninformed learned, it was shown, had little chance of penetrating type-2 problem spaces. And such problem spaces looked set to permeate biological cognition right down to its roots in simple animat bahaviors. Since such problems are, repeatedly, solved by real learners, the question was 'How?'. What ploys, stratagems and tricks enable weak learning devices to discover regularities whose traces in the gross input data are (in a sense we made precise) merely marginal?

One solution would be to leave it all up to biological evolution; to suppose that we are simply gifted with innate tendencies to re-code and process the gross data in just those ways needed to simplify specific kinds of learning. And no doubt this is sometimes the case. We believe, however, that other, more general mechanisms are also at work. The goal of our treatment was therefore two-fold. First, to give a precise, statistical account of the difference between 'marginal' and robust statistical regularities and hence to distinguish two kinds of learning task whose computational demands are very different. Second, to explore some of the ways (short of full-blooded nativism) in which the harder type of learning task may be successfully performed.

The statistical story is, we believe, robust. We have shown that a variety of existing learning algorithms tend to rely predominantly (and in some cases exclusively) on the extraction of a specific type of regularity from a body of input data. This type of regularity lies close to the surface of the training data, in the form of pronounced frequency effects and is thus fairly straightforwardly extracted by a variety of direct sampling methods. Some appearances to the contrary, the extraction of these (type-1) regularities is really all we currently know how to achieve — and no wonder, once the size of the search space for the other form is appreciated.

The extraction of the more opaque type-2 regularities is not, however, impossible. The crucial manoeuvre in such cases is somehow to trade achieved representation against computational search. Achieved representational states act as a kind of filter or feature detector allowing a system to re-code an input corpus in ways which alter the nature of the statistical problem it presents to the learning device. Thus are type-2 tigers reduced to type-1 kittens. It is exactly this strategy which characterises Elman's recent and important work on incremental learning. Several extensions to Elman's basic strategy were pursued. In particular, we noted the potential value of allowing achieved early representational states to participate in multiple episodes of future problem-solving, thus making maximal use of any re-coding leverage ever obtained. Modular connectionism, we suggested, may provide a partial implementation of such a maximising strategy. Annette Karmiloff-Smith's work on representational redescription was seen to constitute a general vision of endogenous drives in human learning consistent with the commitment to such maximisation. Most speculatively, language and culture were themselves depicted as evolved tools enabling a kind of species-wide implementation of the same strategy. Language and culture, thus viewed, enable us to construct problem-solving trajectories which exploit already achieved representational encodings and allow such exploitation to extend across individuals and generations. We thus trade achieved representation against individual computational search on a truly global scale. Finally, we noted that certain kinds of problem space (such as that of language acquisition) may have themselves evolved so as to exploit whatever biases happen to characterise the search strategies of real learners. To the extent that this is so, we may again see type-2 problems solved with unexpected ease.

It is the combination of these kinds of factor which, we believe, explains our otherwise baffling facility at uncovering deeply buried regularities. But despite the grab-bag of specific mechanisms the underlying trick is always the same; to maximise the role of achieved representation, and thus minimise the space of subsequent search. This now familiar routine is, as far as we can tell, obligatory. The computationally weak will inherit the earth. But only if they are representationally rich enough to afford it.

Acknowledgements.

Thanks to Bruce Katz, Noel Sharkey and Inman Harvey for useful comments. Thanks also to the six anonymous BBS referees whose patient comments and suggestions have greatly helped improve the text.

References

[1] Becker, S. and Cun, Y. (1988). Improving the convergence of back-propagation learning with second-order methods. CRG-TR-88-5, University of Toronto Connectionist Research Group.

[2] Campbell, D. (1974). Evolutionary epistemology. In P. Schillp (Ed.), *The Philosophy of Karl Popper* (pp. 413-463). Open, Court, Illinois.

[3] Chomsky, N. and Miller, G. (1963). Introduction to the formal analysis of natural language. In R.D. Luce, R.R. Bush and E. Galanter (Eds.), *Handbook of mathematical psychology.* Vol. II (pp. 269-322).

[4] Churchland, P. and Sejnowski, T. (1992). *The Computational Brain.* Cambridge, Ma.: M.I.T./Bradford Books.

[5] Clark, A. (1993). *Associative Engines: Connectionism, Concepts and Representational Change.* MIT, Bradford, Cambridge, MA.

[6] Clark, A. and Karmiloff-Smith, A. (1993). The cognizer's innards: a psychological and philosophical perspective on the development of thought. *Mind and Language.*

[7] Dennett, D. (1991). *Consciousness Explained.* Boston, Ma.: Little, Brown Co.

[8] Dennett, D. (1993). Learning and labelling. *Mind and Language*, *8*, No. 4 1003 (p. 1003540-548).

[9] Dietterich, T., London, B., Clarkson, K. and Dromey, G. (1982). Learning and inductive inference. In P. Cohen and E. Feigenbaum (Eds.), *The Handbook of Artificial Intelligence: Vol III.* Los Altos: Kaufmann.

[10] Elman, J. (1993). Learning and development in neural networks: the importance of starting small. *Cognition*, *48* (pp. 71-99).

[11] Fahlman, S. and Lebiere, C. (1990). *The Cascade-Correlation Learning Architecture.* CMU-CS-90-100, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213.

[12] Gallistel, R. (1994). Interview. *Journal of Cognitive Neuroscience*, *6*, No. 2 (pp. 174-179.).

[13] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

[14] 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.

[15] Jacobs, R., Jordan, M. and Barto, A. (1991). Task decomposition through competition in a modular connectionist architecture: the what and where visual tasks. *Cognitive Science*, *15* (pp. 219-250).

[16] Jacobs, R., Jordan, M., Nowlan, S. and Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation, 3* (pp. 79-87).

[17] Karmiloff-Smith, A. (1992). Nature, nurture and PDP: preposterous development postulates?. In A. Clark (Ed.), *Connection Science, special issue on Philosophical Issues in Connectionist Modelling, 4*, No. 3 and 4 (pp. 253-270).

[18] Karmiloff-Smith, A. (1979). *A Functional Approach to Child Language*. London: Cambridge University Press.

[19] Karmiloff-Smith, A. and Clark, A. (1993). What's special about the development of the human mind/brain?. *Mind and Language, 8*, No. 4 (pp. 569-581).

[20] Kirsh, (1992). TITLE. In Davis (Ed.), *Connectionism: Theory and Practice*. New York: O.U.P.

[21] Marr, D. (1982). *Vision*. New York: W.H. Freeman.

[22] Newport, E. (1990). Maturational constraints on language learning. *Cognitive Science, 14* (pp. 11-28).

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

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

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