Psyke & Logos, 1988, *9*, 307-336

PARALLEL DISTRIBUTED PROCESSING[1]

Kim Plunkett

What is PDP?

Parallel Distributed Processing (PDP) is a recent development in cognitive science modelling research that assumes that individual components of human information processing are highly interactive and that knowledge of events, concepts and language is represented diffusely in the cognitive system. This distributed feature distinguishes PDP from other »connectionist« models which make use of localist representations, e.g. a single node in a network standing for an entire concept. The PDP approach developed out of the explicit concern of some cognitive scientists that cognitive models should not be restricted to describing aspects of human behaviour that are idealised abstractions of competence. Rather, a unified model of cognition, straddling the traditional competence/performance distinction, has emerged as a worthwhile and realistic research endeavour. Cognitive scientists, working from a PDP perspective, have begun to construct models of human cognition that show promise that their ambitious goals might be achieved. To date, successes range from simulations at the low end of cognition that model speech perception or interpret the different configurations of a necker cube to higher order cognitive skills like sentence processing or language learning. All these models emphasise the context sensitivity of cognitive processes and they all share the assumption that complex behaviours can emerge from the interaction of relatively simple constituents and their environment.

The characteristics of these systems include; a tendency to account for the details of human behaviour within a single framework that does not require *ad hoc* assumptions to account for apparent exceptional behaviours; a robustness in performance to distortions in the input stimuli or knowledge base itself; a capacity to learn or organise representations of the environment to which they are exposed - prototypical representations emerge as a natural outcome of the learning process; flexibility in responding in an appropriate manner to situations never experienced before. All these characteristics can be attributed to the parallel, distributed architecture of the systems used to implement the models. Typically, PDP models differ from traditional symbolic accounts of human behaviour in their rejection of the need for rule-based processes. Instead, PDP models offer micro-structural accounts. Rule-based accounts are considered by connectionists as convenient

approximations to a system that is considerably more complex. For example, it is argued that categorial rule-systems cannot capture the fine-grained structure of human behaviour in a natural way.

A PDP system's potential for learning is rooted in its high sensitivity to variations in patterns of stimulation and to the structure of the environment to which it is exposed. By manipulating the pattern of connections between its component parts, a network is able to exploit new patterns of stimulation from the environment to create new input/output functions and hence demonstrate new behaviour. All this is achieved with a minimum amount of pre-wiring, i.e. the network organises itself often constructing subtle internal representations of the environment that it is processing. For example, the construction of prototypical representations of environments to which it has never been exposed is a powerful property of such a system. In contrast, many current cognitivist models of learning are highly nativist in their approach, typically relegating learning to the process of choosing between a pre-defined set of symbolic parameters that are innately given.

Graceful degradation, i.e. robustness of behaviour in the face of inadequate stimulation or internal damage, is achieved as a result of the distributed representation of knowledge. Many nodes in a network contribute to the representation of any given fact or proposition. One cannot point to the local representation of a concept as one can in a conventional semantic network. The global distribution of activity in the network has to be considered when evaluating its knowledge state. PDP networks are robust in that the overall pattern of activity in a network often remains stable in the face of perturbation. Similarly, the propensity of PDP networks to respond appropriately to new environments reflects the conservative, assimilative nature of their global representations. New patterns of stimulation are judged against old experiences. Networks modify their behaviour by accommodating to the parallel influence of new sources of information that are simultaneously impinging on the system. It is helpful to view a network as a multi-dimensional state space or energy landscape. The precise contours of the landscape vary with the environment in which the network finds itself. Adaptation in the network can be likened to the process of gradient descent. Final behaviour is determined by the parameters corresponding to the lowest level or valley in the energy landscape. Since the landscape varies from one environment to another, output behaviour will accommodate accordingly.

Historical Roots

PDP models demonstrate many characteristics that are desirable in simulations of human cognition. Though individual properties can be found in other approaches, it is rare to find a single system embracing so many important features. Proponents of the PDP approach regard the parsimony of

their models as heralding a new era in the study of cognition. However, Parallel Distributed Processing builds upon a long tradition of scientific endeavour which dates back at least as far as the British Empiricist David Hume. Many of the ideas embodied in the PDP approach can be found in writings of William James.

> *The amount of activity at any given point in the brain-cortex is the sum of the tendencies of all other points to discharge into it, such tendencies being proportional (1) to the number of times the excitement of each other point may have accompanied that of the point in question; (2) to the intensity of such excitements; and (3) to the absence of any rival point functionally disconnected with the first point, into which the discharges might be diverted. (James, 1892, p. 265).*

Theoretical developments that laid the foundations for many of today's models were already underway in the forties and fifties (Hebb, 1949; McCulloch & Pitts, 1943; Rosenblatt, 1959). Yet PDP has emerged as a new perspective only within the last six or seven years (Hinton & Anderson, 1981; Rumelhart & McClelland, 1986a). To understand the relationship between PDP and earlier, related approaches, it is necessary to review some of the core concepts of PDP. This brief review will also serve as an introduction to the more sophisticated simulations described below.

The heart of a parallel distributed processing system is a neural network. A neural network consists of a collection of units connected to each other by a set of pathways. Figure one illustrates an example of a simple network designed to solve the logical OR problem (see below).

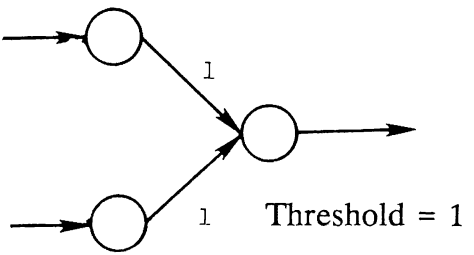

Figure One

The circles represent the units or constitutive nodes of the network and the solid lines indicate the pattern of connectivity between the units. Units take on a variety of activation values that can depend on some function of the net input to a unit from other units, the previous state of the unit itself and input from the external environment. The activation value of a unit determines its

output. A variety of functions of the activation value are typically used to determine the output from a unit: A squashing function constrains the output within certain limits: A threshold function determines which of a limited set of outputs will be produced: A linear function simply passes on the activation value itself. Networks may be homogeneous in that all units use the same output function or they may be heterogeneous in the output functions used. Note that it is unusual to find homogeneous nets of linear units. The range of tasks that such networks can perform is limited and they tend to be unstable (activation values have a potential to explode to enormously high values in linear systems). On the other hand, the input function that maps the net input to a unit onto its activation value is typically linear.

Units communicate with each other by passing their output values to the other units in the system with which they are connected. Connections are almost always unidirectional. The input/output functions of the units, together with their pattern of connectivity, define the architecture of the network. Units can excite or inhibit each other. Each connection may have a positive or negative real value that determines the degree and direction of influence of the source unit on the target unit. Target units may receive inputs from a large number of source units. The strength of the connection between two units is called the *weight* of the connection. The product of the output value of the source unit and the weight value between the source unit and the target unit determines the contribution of the source unit to the net input to the target unit. If the product is negative then the source unit inhibits the target unit. If the product is positive then the source unit excites the target unit. Some systems treat excitatory input and inhibitory input independently from one another (Grossberg, 1980). However, all the models described in this paper treat the two types of input homogeneously. A zero weight between two units is functionally equivalent to a lack of connection between those units in most systems.

The overall behaviour of the network is determined by the set of weights that define the pattern of connectivity of the system, and by the input of the system. The set of weight values embodies the knowledge of the system with respect to a given set of environmental stimuli. In figure one, the numbers on the solid lines represent the weights of the connections between the various pairs of units. The system consists of two inputs and a single output unit. The input units are linear and simply pass on the values received from the environment to the output unit. The output unit is a linear threshold unit i.e. if the activation value reaches or exceeds a given value, in this case »1«, then it outputs »1«, otherwise it outputs »0«. Logical OR can be represented by the following truth table:

LOGICAL »OR«		LOGICAL »AND«	
input	output	input	output
0 0	0	0 0	0
0 1	1	0 1	0
1 0	1	1 0	0
1 1	1	1 1	1

A similar network to that depicted in figure one can represent the boolean function Logical AND. The only change that needs to be made is that the threshold of the output unit must be changed to »2«.

Networks like these have been investigated since the forties (McCulloch and Pitts, 1943) and are typical, though simplified, versions of the neural nets that originally sparked off interest in the area. Larger nets with greater numbers of input and output units can represent more complicated input/output algorithms. However, their mode of operation is essentially identical to the network described above; input patterns are mapped onto a set of output units via weighted connections. The activation values of the output units, together with their threshold values, determine the resultant pattern of output activity. It is also possible to make networks like these learn, i.e. given an input and a desired output it is possible to manipulate the value of the weights automatically so that the required input/output function is achieved. The learning algorithm originally used by Rosenblatt (1962) is called the Perceptron Learning Rule. According to this rule, the weights between two units are modified if the desired output differs from the actual output. The desired output is determined by a »teacher« signal. The activation value is determined by the propagation of the input signal through the network. The actual output on each output unit is then compared to the target specified for that unit in the teacher signal. If there is a mismatch between these two values, then all the weighted connections feeding into the given output unit are adjusted according to the following rule:

$$\Delta w_i = (t_p - o_p)\, i_{pi} = \delta_p\, i_{pi}$$

where Δw_i specifies the change in weight on the connection from input unit to output unit p, t_p specifies the teacher signal for output unit p, o_p specifies the actual signal on output unit p, and i_{pi} is the output from input unit i. The change in threshold on the output unit is given by the following rule:

$$\Delta\theta = -(t_p - o_p) = -\delta_p$$

This procedure is guaranteed to find a set of weights that produces the correct input/output mappings, *provided such a set of weights exists.* The Perceptron Learning procedure can be applied to a surprisingly wide range of pro-

blems and is still used in many influential models. However, as Minsky and Papert (1969) pointed out, perceptrons are still quite limited in the class of input/output mappings they can learn.

In particular, perceptrons are unable to solve the Exclusive OR problem. The truth table for Exclusive OR is given below:

EXCLUSIVE »OR«

input	output
0 0	0
0 1	1
1 0	1
1 1	0

Exclusive OR demands a non-linear partitioning of the problem space (figure two).

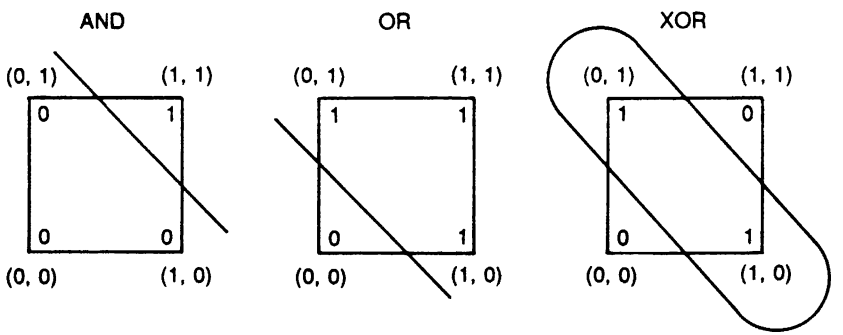

Figure Two
Geometric representations of the three problems

It is possible for networks to *learn* a non-linear partitioning only when there is an intermediate level of inhibitory units between the input and output units. The Perceptron Learning rule provides only for the adjustment of weights directly connecting the input and output units. Therefore, perceptrons cannot perform non-linear classifications. Since it is known that the class of problems to which Exclusive OR belongs is common in computer science, the demise of neural net models based on the Perceptron Learning procedure followed swiftly on the publication of Minsky and Papert's book. Like spreading excitation in semantic nets, neural network research in general went out of fashion. For many years, the dominant symbolic approach to computational psychology reigned supreme.

Though neural net research receded into the background in the late sixties and seventies, the effects of Minsky and Papert's critique were as much so-

ciological as scientific (Papert, 1988). Perceptrons represent only a small class of possible network architectures and learning procedures and it was to this small class of networks alone that Minsky and Papert's comments were addressed. However, Minsky and Papert's critique was mistakenly interpreted as applying to neural nets in general and funding for most neural net projects dried up. Nevertheless, during the dark years, a small group of researchers continued to investigate the properties of different neural nets and extend their domain of application. J.A. Anderson (1977), Grossburg (1976) and Kohonen (1977) are notable contributors amongst this group. Many of the principles and insights embodied in PDP research today are due to the sustained efforts of this small group of researchers.

As some cognitive scientists became increasingly dissatisfied with the achievements of the traditional symbolic approach to computational psychology, it was recognised that neural nets possessed precisely the kind of properties which traditional symbolic models seem to lack. Furthermore, cognitive scientists began to recognise that various mathematical tools could help extend the generality of the Perceptron Learning procedure to a greater variety of problems, including Exclusive OR. For example, Rumelhart, Hinton and Williams (1986) describe a learning algorithm called Back Propagation (also known as the Generalised Delta-rule), which specifies a procedure for manipulating the weights in a multi-layered network. Back propagation can be used to solve the Exclusive OR problem. Figure three depicts an example of a network that implements the Exclusive OR truth table above.

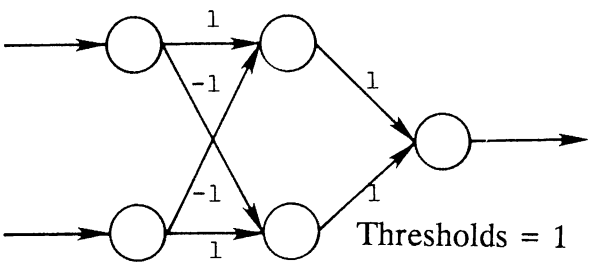

Figure Three
A network for solving Exclusive OR

Notice that the network in figure three contains a layer of units between the two input units and the single output unit. Any units in a network that are not exposed to the external environment but only to other units in the network are called *hidden units*. Back propagation[2] provides a method for adjusting the threshold values of these hidden units as well as the weighted connections between hidden units and *visible units* (visible units receive input directly from the environment or send output directly to the environment).

Recent work with networks using the back propagation algorithm has shown that hidden units often organise themselves in a way that reflects the structure of the problem at hand. For example, Hinton (1986) describes a back propagation network designed to learn kinship relationships. Hidden units in this network organise themselves to represent the salient dimensions in kinship relationships, such as gender and generation. Sometimes, the representations discovered by hidden units parallel human interpretations of a problem. At other times, the hidden units discover partitions of problem space not obvious to humans. For example, a back propagation network discovered a novel solution to the Exclusive OR problem. Hidden units have also been used to filter out the redundancies in an input signal, compressing the information for later retrieval (Ackley, Hinton & Sejnowski, 1985). The ability of hidden units to extract and represent regularities of the environment to which they are exposed has triggered a controversial discussion of the nature of representation in neural nets. Hidden units have no referential function, and yet they seem to share some properties with the symbolic entities embodied in traditional rule-based accounts of cognition.

Another important refinement in neural network architectures has been the development of the Boltzmann machine. The Boltzmann machine belongs to a class of *constraints satisfaction* networks that are capable of finding solutions to problems which require the simultaneous fulfilment of a selected set of criteria. For example, the process of language comprehension may be reviewed as a constraint satisfaction problem in which the various component parts of a sentence must be integrated to obtain a coherent interpretation (McClelland & Kawamoto, 1986). These criteria may be mutually supportive or they may be in competition with each other. In the former case, interactive constraint satisfaction networks quickly converge on a best solution. However, if the criteria are in competition with each other, a network may possess a variety of final stable states. Some of these states (often called local minima) may underestimate the potential of the network to solve the problem at hand. Boltzmann learning is a technique for avoiding these local minima (see figure four) and finding the best possible solution to the problem in a given network.

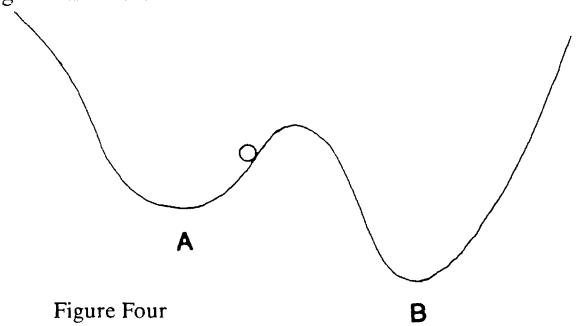

Figure Four
Local minimum (A) in an energy landscape

Boltzmann learning is a stochastic process inspired by statistical mechanics. Just as misaligned crystals or metal can be realigned by a gradual process of heating and cooling called annealing, so can a network be made to explore the energy landscape of a given problem space until it finds a best possible solution. This process is called simulated annealing. As we shall see in the next section, constraint satisfaction networks are particularly useful for simulating psychological phenomena that involve the disambiguation of situations that demand interpretation under the control of contextual determinants. In contrast to feed-forward, back propagation networks, constraint satisfaction networks are typically more highly interconnected. Such networks need not have an obvious layered-structure and many units may be visible to the environment.

More recently, network architectures have been introduced that attempt to integrate *time* as a dynamic dimension in the representations embodied within a network. These networks involve the use of *recurrent* connections (units that feed back on themselves). Recurrent connections permit a network to maintain an image of its previous states. Since these recurrent connections can themselves be manipulated by a variety of learning algorithms, such networks can develop a capacity to predict or control future events. For example, Jordan (1986) describes a system which attempts to deal with the classical problem of serial order in behaviour (Lashley, 1951) through the parallel, distributed implementation of a planning structure. Similarly, Elman (1988) shows how a recurrent network can be used to capture some syntactic properties of sentences without the explicit specification of grammatical rules. The problem of serial order in behaviour was impressively resolved by symbolic accounts of cognition (Miller, Galanter & Pribram, 1960). Connectionist accounts of this problem are needed if PDP is to be regarded as a serious alternative to the classical symbolic approach. In the final section of this article, we will return to a comparative evaluation of connectionist and symbolic contributions to our understanding of cognition. In the next section, we turn to a presentation of some concrete examples of connectionist simulations of psychological phenomena.

Connectionist simulations

In this section, I shall briefly review three connectionist simulations. Each simulation uses a different network architecture and addresses a different type of psychological phenomenon. The first model describes a simulation of the learning of past tense forms of verbs by young children. The heart of this model is a simple perceptron learning system. The second model shows how a constraint satisfaction network can be used to simulate the different interpretations of a knecker cube. Finally, a recurrent network that is able to recover lexical classes from word-order is described.

Learning the past tense of verbs

A founding assumption of research in child language is that children, like adults, use language productively. That is, after the initial phases of learning, language usage cannot simply result from mimicking what is heard in the input, but rather children acquire the ability to generate syntactic and morphological combinations that they could never have heard before (Ervin, 1964; Berko, 1958). From most current perspectives, linguistic knowledge is framed in terms of general principles, i.e. »rules«, which govern the productive and (sometimes) interestingly innovative usage of language. The goal for the acquisitionist has been to work through the various domains of language, outlining how and when children come to master the systems of rules governing the production and comprehension of novel forms and utterances. However, current perspectives also acknowledge that certain pockets of linguistic knowledge do not appear to be rule-governed in the same sense. For example, about 150 or so of the most frequently used verbs in English fall outside the domain of the regular past tense rule, »add-ed« to the stem. Irregular, or strong verbs (Pinker and Prince, 1988), do not form their past tenses by applying a suffixation process, but rather are memorised as separate lexical entries. These verbs can be grouped into three general categories according to the relationship they exhibit to their present tense form:[3] (a) *identity mapping* (or no marking - doing nothing to the stem, e.g., hit $- - >$ hit); (b) *vowel change* (changing the vowel, e.g., come $- - >$ came); (c) *arbitrary* (there is no obvious structural relationship between the present and past tense form, e.g., go $- - >$ went).

Across acquisition, the fact that two different types of verbs coexist in the lexicon, one group using a general rule and others not, sometimes presents problems for the language learner. In both naturalistic and experimental contexts, children frequently exploit the regularities of the past tense system, applying a general rule to the irregular »exceptions« to the rule (e.g., Bybee and Slobin, 1982; Kuczaj, 1977). Children will produce errors such as »goed« or »sitted« in which the regular »add-ed« ending is applied to verb stems which, in the adult grammar, are not subject to this procedure. In addition, several researchers have noted that the time course in the acquisition of these rules (and their exceptions) suggests that children will get worse in their production of forms before they get better. That is, children make mistakes on some types of past tense forms (e.g., comed) after they have been using the forms correctly (e.g., came). After some extended period of overapplication of the general rule to irregular verbs, children reorganise their lexical categorisations and settle into the correct set or irregular stems and corresponding past tense forms, using both regular and irregular verbs in the past tense. This characterisation of the acquisition of morphological regularities (and irregularities) has been described as a »U-shaped« developmental course in which children pass through two stages before attaining adult com-

petence in handling the past tense in English (Pinker & Prince, 1988).

Because these erroneous forms are present only infrequently in the input to children, their timely avoidance of »exceptions« has been taken as the most convincing piece of behavioural evidence that language learning involves the process of recognising and organising linguistic knowledge into a coherent system of general rules. Why else would children produce erroneous forms such as the overgeneralisations of the »add-ed« rule? Acquisiton, then, involves the construction of a system of generalised statements about the structure of the lexicon, and the accompanying lists of exceptions to those general rules. Traditionally, the notion of a »rule« has provided linguists and psychologists with an elegant means to neatly package what children and adults »know« about the intricacies and complexities of language while at the same time accounting for productive language use. As noted recently by Pinker and Prince (1988),

> »{*language*}*researchers who could agree on little else have all agreed on one thing: that linguistic knowledge is couched in the form of rules and principles*« (pg. 74).

Indeed, invoking rules serves to elevate language learning above the level of rote imitation and allows linguists »to factor a complex phenomena into simpler components that feed representations into one another« (Ibid. pg. 84).

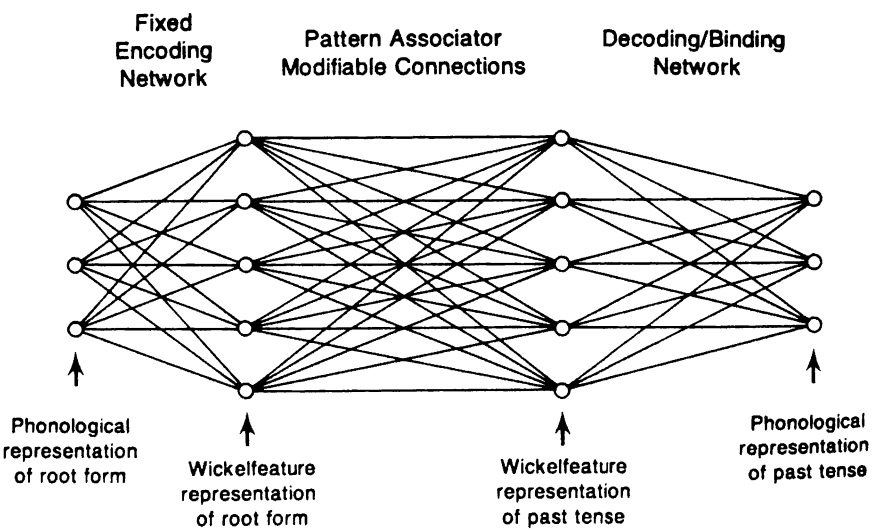

Figure Five

Recently, work within the PDP perspective has promoted reevaluations of the constructs and processes guiding language acquisition. In an attempt to illustrate the applicability of parallel systems to traditionally rule-based domains, Rumelhart & McClelland (1986b) set out to capture several of the »facts« of the acquisition of the English past tense; in particular, that children overgeneralise the »add-ed« suffix to irregular verbs and that development of this system proceeds along a »U-shaped« course. The goal of this work was to suggest how an account of language processing and acquisition might be able to avoid the reliance on symbol-manipulating, rule-governed processes, while still capturing these phenomena of acquisition. Rumelhart and McClelland's past tense simulation contains three major components. (See figure five).

First, an encoding devide takes the present tense stems of English verbs, symbolised as binary characters, and converts each stem into a distributed representation of context sensitive phonological features called Wickelfeatures. Output from the encoding device consists of a vector of activation across the set of output units (460 in all).[4] Secondly, a single-layered, pattern association network maps the set of Wickelfeatures, which it takes as input, onto a set of output units. The activity on these output units constitute the Wickelfeature representation of the past tense form of the verb that was originally presented to the simulator in its present tense form. The task of the pattern association network is to learn to map correctly input to output vectors through adjusting the set of weights which connect the input and output units. The adjustment of the weights is achieved by using the error signal obtained from comparing the actual output of the network with the desired output stipulated by a teacher signal. The weights connecting the input and output units of the network are then adjusted using the Perceptron Learning rule. This second component of the simulator is entirely responsible for the learning that is required to map present tense stems of verbs onto their corresponding past tense forms. This mechanism, then, can be seen to be the foundation for the overgeneralisations reported by Rumelhart & McClelland. The third component of the simulator takes as its input the vector representing the activity of the output units of the pattern associator. Its function is to generate the set of Wickelphones that best fit the output vector description. In principle, the decoder provides a Wickelphone description of the past tense form of the verb that was originally provided in the Wickelphone representation of the present tense stem to the encoder. Several researchers as well as Rumelhart & McClelland themselves have acknowledged several difficulties with this type of decoding process (Pinker & Prince, 1988). The usefulness of Wickelfeatures as a technique for encoding linguistic information in networks of this types is not crucial for the issues discussed in this paper, and the reader is referred to the original source for details.

The results of the Rumelhart & McClelland simulation are important for several reasons. First, within some reasonable limits, the learning curves and overgeneralisations created by the simulation resemble many of the errors and stages of development that children are reported to make in the acquisition of past tense verb forms.

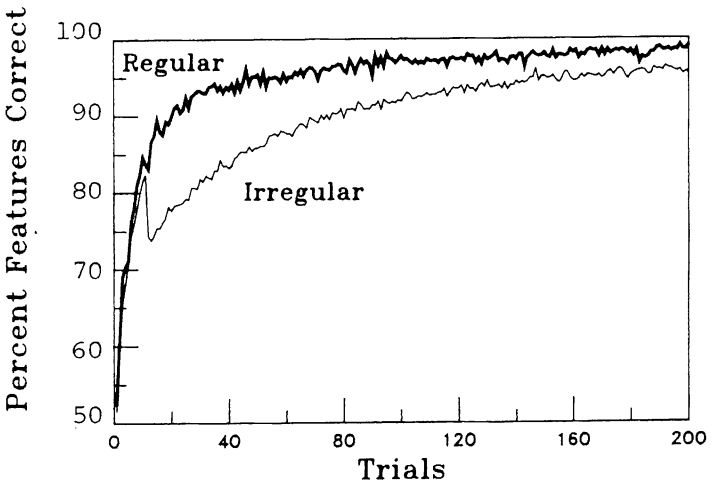

Figure Six
The percentage of correct features for regular and irregular high-frequency verbs as a function of trials

Figure six shows the »U-shaped« dip for irregular verbs during the early stages of learning. This regression represents the stage of learning in which irregular verbs are treated as though they are regulars. Even more impressively, Rumelhart & McClelland's simulation provides distinct learning curves for the different *classes* of irregulars which closely map the types and relative timing of errors made by young children. For example Kuczaj (1977) reports that past tense errors of the form »ated« occur later in development than errors which simply »add-ed« to the present tense stem (»eated«). Rumelhart & McClelland's simulator is also delayed in producing these former types of error.

The excitement engendered in the cognitive science community by Rumelhart & McClelland's simulation was partly to do with the accuracy with which it seemed to be able to mimic young children's behaviour. More importantly, however, the simulation showed how one might construct a *developmental* model that could acquire linguistic knowledge without assuming much nativist baggage. In particular, the simulation does not rely on rules in any obvious way. Furthermore, the simulation embodied an account of the process of morphological reorganisation which is assumed to be crucial to

the achievement of mature linguistic skills (Bowerman, 1982). The model achieves this bt playing off the learning properties of the pattern association network against the structural relationships implicit in the information it must process, i.e. the regularities of English verb morphology.

The past tense simulation has been severely criticised. In addition to their criticism of the Wickelfeature representation used by Rumelhart & McClelland, Pinker & Prince (1988) question the input assumptions of the simulation and point out that the U-shaped developmental curve followed by the simulator is an artifact of the discontinuity in vocabulary size and structure to which the model is exposed. For example, the performance dip for irregular verbs (figure six) occurs just after the proportion of regular to irregular transformations in the training set has been greatly increased. However, Plunkett & Marchman (in press) show that »U-shaped« learning can be achieved with *continuous* input to a back propagation network, *provided certain realistic assumptions are made about the relative frequencies of the different classes of verbs*. Furthermore, the Plunkett & Marchman simulation provides information as to the conditions under which such a network acts *as if* it is learning a set of rules and the conditions when behaviour is less categorical. Despite the qualms of classical symbol theorists, the PDP approach still holds out the promise of an alternative developmental account for acquisition and the potential for a new approach to language processing.

Interpreting the Necker cube

Constructivist accounts of human perception often cite the shifting interpretations of the necker cube as testifying to the top-down nature of cognitive processes. The necker cube can be interpreted in different ways because we are able to project distinct orientation models onto one and the same stimulus set. On this view, the interpretation of a necker cube is contingent upon the construction of an internal representation. More recently (Feldman, 1981 and Rumelhart, Smolensky, McClelland & Hinton, 1986), connectionists have shown how the orientation of a necker cube can be computed through the interaction of mental models and bottom-up visual information in a constraint satisfaction network. Figure 7 provides a summary of some of the constraints involved in perceiving the necker cube.

The bottom part of the figure illustrates the necker cube itself whilst the top part of the figure illustrates two interconnected networks, each representing alternative interpretations of the necker cube. Each unit in the network represents a hypothesis about the correct interpretation of a vertex of a necker cube. For example, the unit in the lower left-hand part of the network represents the hypothesis that the lower left-hand vertex of the drawing is a front lower left (FLL) vertex. The upper right-hand unit of the network represents

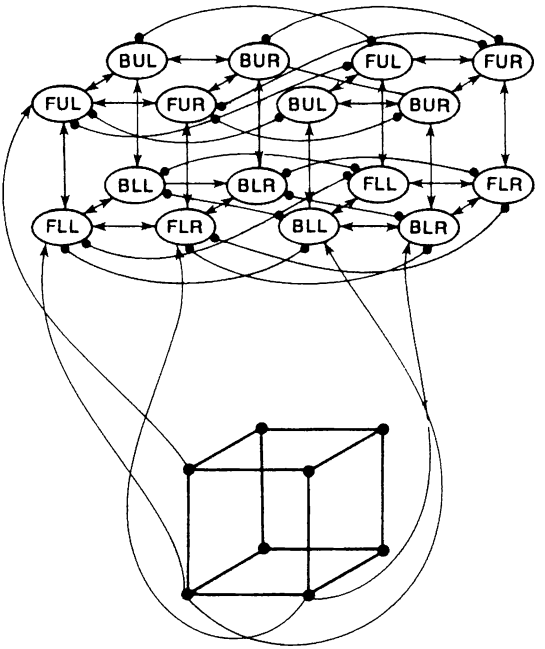

Figure Seven

the hypothesis that the upper right-hand vertex of the necker cube represents a front upper right (FUR) vertex. Notice that both these interpretations are inconsistent. Normally, only one of those vertices can occupy concurrently the frontal plane of the cube at a time. Figure 7 also illustrates the different types of constraints operating in the network. Since each vertex can have only one interpretation, alternative interpretations of the same vertex are connected by inhibitory weights (BUL - - -> FUL). Similarly, since the same interpretation cannot be given to more than one vertex, units representing the same interpretation are mutually inhibiting (BLL - - -> BLL). Thirdly, units that represent locally consistent interpretations are mutually exciting (FUL - - -> FLL). Without these excitatory and inhibitory constraints, it is extremely unlikely that the network will find a solution corresponding to the correct interpretation of the necker cube (all the units in one sub-network turned on and all the units in the other sub-network turned off). In fact, without any constraints whatsoever, there are in principle 2^{16} possible configurations of the network. However, once constraints are introduced to the system, the likehood of many of these states occurring is considerably reduced. Hopfield (1982) has shown that the behaviour of constraint satisfaction networks can be characterised as a process of hill-climbing[5] on a *goodness contour.* Since many of the units in a constraint satisfaction

network, like the one depicted in Figure 7 above, are in competition with each other, it is possible to evaluate any given configuration of the network as having a particular *goodness of fit.* Goodness of fit depends essentially on the extent to which each unit satisfies the constraints imposed upon it by other units, the likehood of an individual unit turning on itself (bias), and finally direct evidence from the input that suggests a given unit should turn on. We can summarise the goodness of fit in a constraint satisfaction network for all units in the network in the following equation:

$$Goodness = \Sigma_{ij}\, w_{ij}\, a_i\, a_j + \Sigma_i\, input_i\, a_i + \Sigma_i\, bias_i\, a_i$$

where w_{ij} represents the weight connecting unit j to unit i, and »a« represents the activation of a given unit.

Given this equation and an algorithm that stipulates an updating procedure for the units in a network, one can describe an energy landscape or goodness contour that corresponds to the various configurations of the network. In this way, *local* computational operations, in which each unit adjusts its activation up or down on the basis of its net input, serve to allow the network to converge towards states that maximise a *global* measure of goodness or degree of constraint satisfaction. For example, assume that a network like that depicted in figure 7, with the stipulated set of excitatory and inhibitory connections, is provided with a set of randomly assigned biases on each of the 16 units. A bias determines the probability that a given unit will turn on or off. If the network is allowed to run for a succession of discrete times steps, the activation values of each of the units will adjust themselves in such a way as to *relax* into a stable global state or maximum goodness of fit for the network.

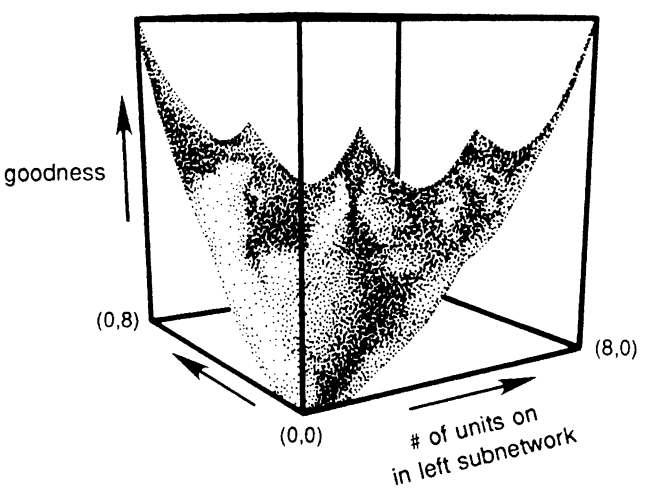

Figure Eight

Figure 8 depicts a goodness contour for the possible configurations of the necker cube network. We may interpret this picture as a visual conceptualisation of the possible states of the network. The low point (0,0) corner, corresponds to the start state in which no units are turned on. The peaks on the left and right correspond to the standard interpretations of the necker cube. These goodness peaks are the states to which the network will most often be attracted in the relaxation process. The choice between interpretations is determined by the position on the goodness contour at which the network starts and the particular sequence of updates that is chosen for the units in the network. Thus, it is possible to *push* the network to a particular interpretation of the necker cube by giving a large bias to one or more of the units. For example, notice that the goodness contour in figure 8 contains a number of smaller peaks. These peaks represent *impossible* interpretations of the necker cube such as that depicted in figure 9 in which two surfaces are interpreted as being foremost.

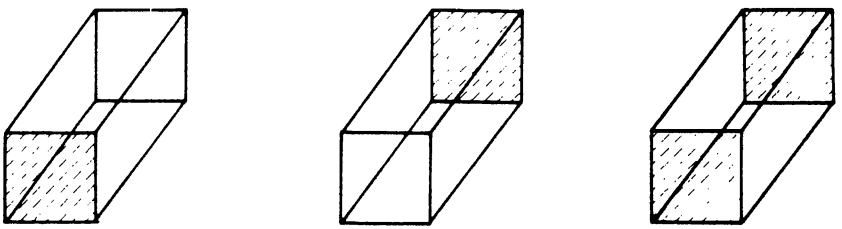

Figure Nine
Three interpretations of the necker cube

Since these peaks represent local maxima on the *goodness contour,* the hill-climbing process halts and the network remains in this stable, though *impossible,* state.

Dynamic accounts of the interpretative process show us how a single ambiguous source of information can be resolved into a definite single solution. The goodness or energy landscape provides a global picture of the characteristics of any given constraint satisfaction network. The process of hill-climbing describes the process of change and conflict resolution within the network. Note how the energy landscape is molded by reference to stable states of the network. We may conceptualise these stable states, visualised as peaks in the landscape, as *default* configurations which the network will move towards in the absence of any conflicting information. Thus, constraint satisfaction networks can be seen to implement representational entities often referred to in the literature as *frames* (Minsky, 1975) or *scripts* (Schank & Abelson, 1977). The mutual constraints within a network and the environment (external input) in which the network finds itself interact to mold a dy-

namic, context sensitive energy landscape. Global maxima in this landscape correspond to prototypical resolutions of the constraint satisfaction problem. Local maxima correspond to intermediate solutions, reflecting competition between constraints. However, local maxima need not represent *impossible* solutions as in the case of the necker cube. They may equally well represent unorthodox but acceptable configurations of a given frame or script. In this fashion, the continuous nature of the energy landscape provides a foundation for the non-categorial forms of behaviour typical of these networks. Furthermore, stable configurations are achieved on the basis of *local* computations. No symbolic executive supervises the relaxation process.

Discovering lexical classes from word-order

The determination of word-order in an utterance is known to reflect a variety of constraints such as syntactic structure, selectional restrictions, subcategorisation and discourse considerations. Traditionally, psycholinguistic accounts of language production and comprehension have invoked symbolic processing systems to express the abstract structural relationships between words in an utterance. For example, in the sixties, psycholinguistics was dominated by the view that language processing involved some psychological implementation of transformational grammar (Fodor, Bever & Garrett, 1974). Although this approach turned out to be incorrect, the procedurally oriented theories (e.g. Miller & Johnson-Laird, 1976) which took over, are still symbolically based. Indeed, it has been argued (Fodor & Pylyshyn, 1988) that a symbolic level of representation is a necessary foundation for psycholinguistic processing. On this view, neural nets are capable of capturing only the most trivial structural relationships found between the words in an utterance. Neural nets fail in precisely the same way that Markov grammars failed to provide an account of linguistic structure earlier this century. As a first attempt to answer this challenge, Elman (1988) used a recurrent network to simulate word-order prediction. As we shall see, his network was able to assign lexical items to their correct grammatical category and to predict appropriate category orderings in the output from the network.

Earlier in this paper, we saw that recurrent networks can maintain an image of their previous states and so develop the capacity to predict future events. The architecture of the recurrent network used by Elman is shown in figure 10. Notice that figure 10, like the network for solving Exclusive OR, contains a layer of hidden units. Thus, the network possesses the capacity to extract regularities from the input patterns and construct internal representations thereof. In addition to the layer of hidden units, Elman's network contains a set of context units. The hidden units and the context units are equal in number. However, the context units can only communicate with the hidden layer. The weights cconnecting the hidden units to the context units are fixed

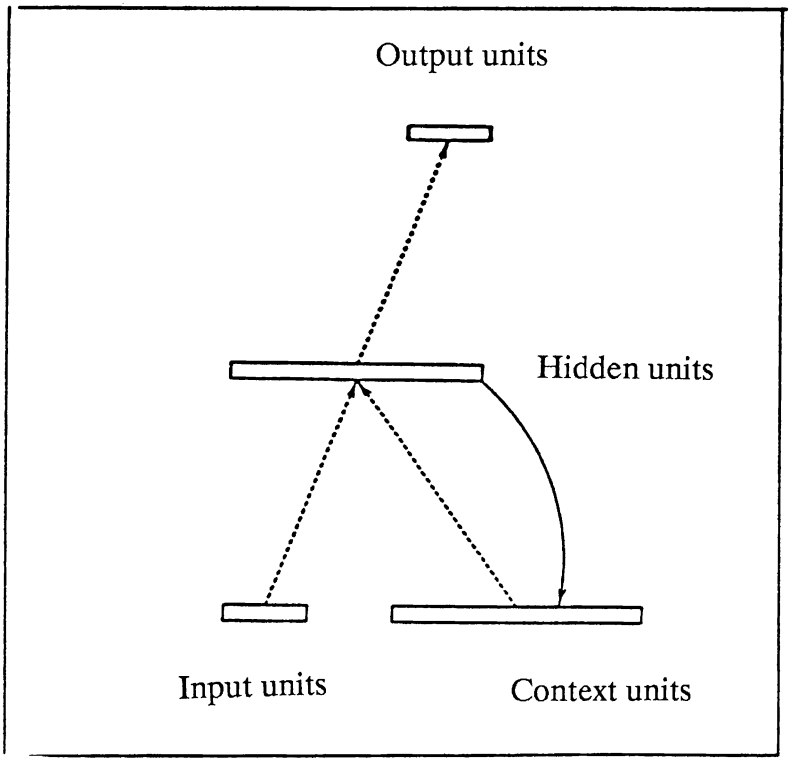

Figure Ten

and constitute a one-to-one mapping. In effect, on every time-step the context units establish a copy of the previous states of the hidden units. The context units are connected to the hidden units in a one-to-many mapping. The dotted line connecting the context units to the hidden units indicates that the weights are adjustable. The context units display an image of the previous states of the hidden units to the hidden units themselves. In this way, Elman's recurrent network goes beyond the finite state Markov grammar. The recurrent connections, through the context units, provide the system with an indefinitely embedded representation of previous states of the network. The context units act as a contextual memory for the network.

The task which Elman gives the network is fairly straightforward. A word is presented to the input units. In response, the network must predict which word, taken from a previously constructed list, will be presented next to the input units. The network shows that it can predict the next word in the list by generating the word on the output units. During the training phase, a teacher signal provides feedback to the network. Errors are propagated backwards through the network using the Generalised Delta rule. The previously

constructed list consists of sequences of sentences generated by a simple phrase structure grammar. The list contains 10,000 two- or three-word sentences. Each word is represented by a unique random binary string. Thus, if the first sentence presented to the network is »woman smash plate«, the first two responses of the network should be »smash« and »plate«. Elman trained the network for five complete passes through the data set. He discovered that the absolute level of predictive performance was not very good, indicating that the network had failed to memorise the sequence of words. However, after the training phase, the connections in the network were frozen and the individual words used in the grammar presented to the input units,

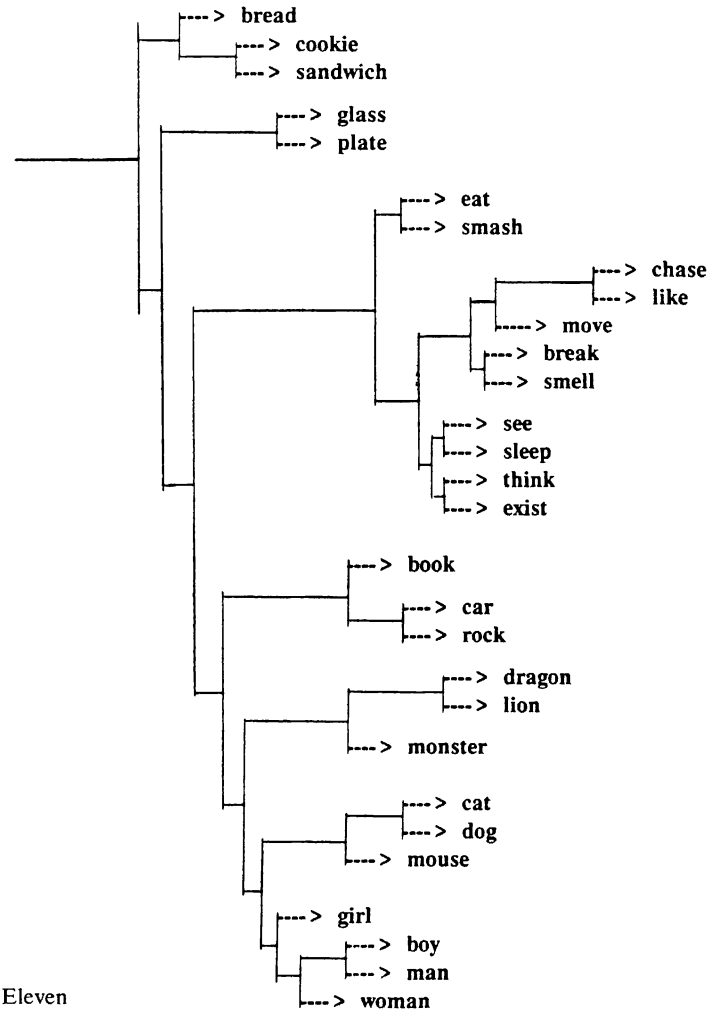

Figure Eleven

one at a time. Instead of recording the output from the network, Elman recorded the activity across the hidden units (represented as a vector) for each unique word in each of its sentential contexts. The hidden unit activations produced by a given word (in all its contexts) were averaged to yield a single 50-bit vector for each of the 35 unique words in the input stream. These internal representations were then subjected to a hierarchical clustering analysis.

Figure 11 shows the resulting tree; this tree reflects the similarity structure of the internal representations of these lexical items as perceived by the network. Lexical items which have similar properties are grouped together lower in the tree, and clusters of similar words which resemble other classes are connected higher in the tree. It is clear that the similarity structure depicted in the tree diagram reflects our human intuitions about grammatical similarity between the words which the network knows about. Thus, verbs are grouped together on one branch of the tree whilst nouns have been grouped together on another branch. Although the network has been told nothing about the semantics of the nouns presented, the cluster analysis reveals that the network has discovered appropriate semantic classifications of the words. For example, inanimate objects are distinguished from animate objects and humans are distinguished from small animals. Elman notes, »that this hierachy is implicit in the similarity structure of the representations, and not an explicit function of the architecture. The network does not have available any of the symbolic apparatus of semantic networks or tree structures.« (pg. 20). He also goes on to point out that the apparent semantic knowledge of the network is an illusory. The network has no knowledge of the meaning of words since each word is represented by a random binary string. It is simply the case that the network is able to classify the different words on the basis of their very similar behaviour with regard to serial order. However, one can imagine real language learners making use of the cues provided by word-order to make intelligent guesses about the meanings of novel words. This simulation suggests how distributional information in the input might be exploited by the learner without couching this knowledge in terms of explicit rules.

Elman admits that the structural relationships implicitly represented in the network in this simulation do not reflect a full-blown grammar of English! However, in a later set of simulations and using a similar network architecture, Elman addresses the problem of pronominal reference. For example, consider the following sentences:

a) If Leo_i wants, he_i will attend the meeting.
b) If he_i wants, Leo_i will attend the meeting.
c) Leo_i will attend the meeting if he_i wants.
d) He_i will attend the meeting if Leo_j wants.

Subscripts indicate acceptable coreference. Within Government and Binding Theory (Chomsky, 1982) coreference between a pronoun and a noun in a sentence is permitted if and only if there is a structural relationship called *c-command* holding between two linguistic symbols. *C-command* is defined entirely in terms of symbolic predicates. However, Elman is able to show that a network can decipher the reference of a pronoun in a sentence without ever being told about the structural relationship *c-command.* Indeed, it has been argued (Kuno, 1987) that *c-command* cannot account for much of the data on pronominal reference. Elman suggests that this failure may well be due to the symbolic, categorical nature of the mechanism embodied in the *c-command* rule. PDP networks may provide a better framework for dealing with the fluid dynamics of pronominal reference.

PDP and Symbol Processing

Imagine a neural net with a pyramidal structure; a large number of input units; a smaller number of hidden units; a single output unit. Using a back propagation algorithm, the network is trained to respond appropriately with the truth values of propositions, encoded as binary input, presented to the input units. Thus, the network when presented with the proposition »dogs have fur« responds with true (i.e. »1«) whereas when presented with the proposition »dogs have fins« it responds with false (i.e. »0«). This network would seem to be able to evaluate the truth values of propositions. It might even be able to evaluate some propositions which it has never been exposed to before (Ramsey & Stich, 1988). Clearly, the network has acquired some knowledge about the world. Contrast this network with a more traditional semantic network, where the elements of a proposition are represented as monadic nodes and propositions are represented by the activation of the correct set of constituent nodes. Typicallly, the nodes in a semantic network are assumed to have an intentional, symbolic relationship to the objects for which they stand in the real world. Similarly, the structure of the representation of a proposition in such a net is assumed to directly reflect the structure of the state of affairs in the world which the proposition describes. So in the case of the proposition »dogs have fur«, the discrete nodes »dog« and »fur« connected by a property »isa« link maps isomorphically onto a state of affairs in the world. In the back propagation network, propositions are not represented in the same discrete fashion. Rather, any given proposition is represented by the global pattern of activity throughout the network. Knowledge in the network is »represented« by the complete matrix of weights that defines the architecture of the network. In general, it is not possible to point at a discrete location in the network that can be said to represent the proposition. Similarly, the addition of new propositions to the two types of network involve distinct processes. In the case of a semantic network, it is pos-

sible to add some extra nodes and links without disturbing the pre-existing system. However, in the back propagation network, learning a proposition will involve, in general, the adjustment of the complete set of weights defining the network architecture. For reasons like these, PDP nets are often said to provide *sub-symbolic* representations of given knowledge domains, in contrast to the *symbolic* representations of say semantic networks (Smolensky, 1988a).

It is common place reasoning in folk-psychological theories (Stich, 1983) to assume that a person's *beliefs* often play an actively causal role in their behaviour. Beliefs are construed as an individual's holding a *propositional attitude*. Thus, in order for an individual to believe that »dogs have fur« they must be in possession of the propositional attitude »that p« where »p« is the proposition »dogs have fur«. Propositional attitudes have causal powers in that they play a causal role in the activation of other propositional attitudes (beliefs) and behaviour. The relationship between propositional attitudes is determined by some function which maps the relation between propositions. Some philosophers (e.g. Stalnaker, 1987) suppose that propositional attitudes are monadic entities and that the causal power of any given propositional attitude is represented solely by its associative connectivity to other propositional attitudes. However, within the classical symbol processing tradition (Fodor, 1987), propositional attitudes are assumed to have formal, internal structure. The constituents of propositional attitudes are themselves considered symbolic entities in much the same way as the nodes in a semantic net. From this perspective, propositional attitudes have causal power with respect to each other by virtue of their internal structure and a set of formal rules for acting on that structure. The transition between belief states is guided by the activity of a formal, syntactic machine. The language of thought is thus built out of a set of intentional, semantic symbols, and a grammar for manipulating those symbols.

Fundamental to the classical view of cognition is the existence of discrete symbolic entities that can be manipulated by a set of rules. Distributed representations of propositions lack internal structure and are not sufficiently modular to permit rule-governed transformations. Fodor & Pylyshyn (1988) argue that this difference in architecture counts decisively against PDP approach. For example, consider the two sentences:

1) John loves Mary.
2) Mary loves John.

In appropriate circumstances, we have no difficulty in identifying the two »Johns« as being one and the same person. This feature of cognitive inference can be easily accommodated within the symbolic tradition. The two occurrences of »John« are distinct *tokens* of the same mental symbol *type*. We appreciate the coreference of the tokens because they activate the same type

in our mental representations. Now consider the same two sentences represented in a back propagation network. As we saw above, propositions are represented in a distributed fashion throughout the network. In general, it is not possible to point at a local area of the network which represents the single proposition or part of a proposition. Sentences 1) and 2) will, therefore, be represented by distinct, global patterns of activity througout the network. How then, Fodor & Pylyshyn ask, are we to capture the obvious intuition that one and the same person, John, is involved in each proposition when we cannot identify John's token representation and hence his semantic type? Fodor & Pylyshyn argue that parallel distributed representations cannot provide an adequate account of semantic *compositionality* which is essential for understanding mental life.

In a similar vein, Fodor & Pylyshyn argue that PDP models og human cognition are unable to capture the universal *systematicity* inherent in cognitive inference. For example, given the logical expression »A&B«, one can deduce the corollaries »A« and »B«. To achieve this result, all you need to know is that there exist constituent symbols »A« and »B« and a simple transformatical rule for manipulating the premise. Furthermore, the same symbolic system can be easily extended to include the manipulation of such expressions as »A&B&C&D«. Fodor & Pylyshyn argue that the deductive powers encapsulated by this simple manipulating system are not merely matters of empirical fact but are *necessary* properties of cognitive systems. Similarly, they argue that any cognitive system that can perceive or learn the relation *a***R***b*, is equally likely to be able to learn the inverse relation *b***R***a*, not because of contingent fact but because of the nature of cognitive systems. Fodor & Pylyshyn point out that insofar as PDP models are limited to representing empirical statistical regularities they fail to account for the ubiquity and necessity of many of the characteristics of cognitive systems. In essence, their claim is that the poverty of the stimulus precludes an explanation of mind based purely on empirical foundations.

As Fodor & Pylyshyn openly acknowledge, the thrust of their critique of PDP closely parallels the cognitivist's critique of behaviourism in the fifties (Chomsky, 1959). Indeed, PDP has been heralded as born-again behaviourism. There may well be some truth in this characterisation but that need not be such a damning indictement. In contrast to the impenetrable, modular architectures of the classical symbolic account, PDP systems, like behaviourist accounts, are firmly grounded in the environments in which they are designed to operate. For example, the specification of stimulus and response representations are essential steps in simulations that use back propagation. By entrenching itself in the environment, a PDP network is able to offer an account of learning which goes beyond radical behaviourism in that it attempts to provide an account of the internal organisation of the processes which permit the wide range of stimulus/response functions observed in human behaviour.

Fodor & Pylyshyn's critique is substantial (as are those found in other articles, published in the same special issue of *Cognition)* and needs to be answered. However, it is likely that the issues will be resolved through empirical research rather than *a priori* philosophical argumentation. The issues here are complex. For instance, one's view as to what might count as a scientific theory of the mind is likely to influence one's position in the ongoing debate. Dennett (1988) characterises the classical symbolic tradition as seeking pure, universal, crystalline theories of the mind. He points out that evolution may not have been so kind to cognitive scientists. The mind might turn out to be more like a gadget,

> *»an object that one should not expect to be governed by »deep« mathematical laws but nevertheless a designed object, analysable in functional terms: ends and means, costs and benefits, elegant solutions on one hand and on the other, shortcuts, jury rigs, and cheap ad hoc fixes«* (Dennett, 1988, p. 286).

No self-respecting cognitive scientist today believes that the newborn mind is a tabula rasa. Neither do PDP researchers believe that a single PDP network architecture is adequate to the task of supporting all the diverse forms of human cognition. Indeed, as we have seen above, connectionists are deeply involved in investigating the properties of different kinds of networks and discovering the diverse potential across networks for performing certain cognitive tasks. Nevertheless, the general characterisation of neural nets as being statistical inference machines seems correct. Thus, to the extent that explanations of the human cognitive system requires the postulation of universal, platonic ideals, so will PDP models of human cognition, unaided by symbolic mechanisms, fail in their attempt.

However, there are other ways of construing notions of *compositionality* and *systematicity* than those found in the classical symbolic approach. For example, Smolensky (1988b), in a reply to Fodor and Pylyshyn, argues that an increased understanding of the mathematics of neural networks is likely to bring about a new conceptualisation of *compositionality* within a distributed framework. For example, we saw in Elman's (1988) simulation a technique called cluster analysis which provides a similarity metric for comparing distinct states of the same network. Factorial and regression analyses are also contributing to this endeavour. Although distributed representations cannot solve the type/token problem by appealing to discrete categorial entities, similarity metrics may capture better our intuitions about »John« in sentences 1) and 2) above. For though John may be one and the same person in the real world, whether he is doing the loving or being loved, it is a fact of our emotional lives that loving is not the same as being loved. Mental types must be malleable to the role they play in a representation. Distributed patterns of activity are able to deal with this malleability better than discrete, symbolic categorisations. Once we give up the requirement that mental representa-

tions be miniature models of our effective world, and once we acknowledge that contextualisation is an unavoidable given of all mental processing, so does the concept of an impenetrable, discrete mental symbol lose much of its appeal. Similarly, the *systematicity* in human reasoning is open to question. Accounts of logical inconsistence in human reasoning abound in the literature (Wason & Johnson-Laird, 1972). Human reasoning, with all its gaps and inconsistencies, may well turn out to be the product of a statistical inference machine. PDP models of cognition provide cognitive scientists with an alternative framework in searching for answers to what are, at base, empirical questions.

Concluding Remarks: Levels of Explanation

PDP goes under other names. *Connectionism* and *neural networks* are often used interchangeably with each other and with the term PDP. However the terms are not precisely synonymous. Connectionism is the most inclusive term, intended to cover a wide range of network architectures and representational systems that use parallel processing. PDP is used to characterise that style of connectionism which emphasises the importance of distributed representations. However, localist approaches abound (Feldman, 1982). The term neural network is perhaps the most difficult to pin down. It is clearly biologically oriented, but the term is often used in contexts which are intended as cognitive descriptions. This raises the whole issue as to which level of explanation connectionist systems are directed.

Connectionist models clearly have a neurological appeal. Parallel computations by units connected in a web of weighted lines provides a crude but effective abstraction of the central nervous system. Indeed, PDP'ers have themselves anointed their approach »brain-style processing«. Neural net simulations are indeed providing neurophysiologists with insights into the dynamics of massively parallel systems. But are connectionists' systems constrained to providing models at the neurological level of explanation? The testimony of research reported in this article indicates that PDP models have a role to play at higher, functional levels of description and explanation. This is a controversial claim. As we have seen, many researchers working within the classical symbol manipulation approach to cognitive science argue that functional accounts of the cognitive level must be couched in terms of discrete, categorical symbol processing systems. Furthermore, they argue that current PDP models do not behave in the necessary symbolic fashion. According to this argument then, PDP models will not be able to provide explanations and descriptions of the cognitive level. It is conceded, however, that PDP models, appropriately hard-wired, may be able to implement the foundations of a cognitive system in much the same way that the hardware of a computer provides the necessary environment for symbolic program-

mes (Pinker & Prince, 1988). Indeed, it is widely acknowledged that something like a PDP system must provide the neurological foundations for the apparent symbolic mind. On this view a symbol processing machine sits on top of a PDP implementation of the neurological system. It makes sense to talk about a two-level system because the symbolic machine operates according to its own autonomous set of principles.

The PDP research community resists this relegation of their domain of explanation to the implementational level. One of the primary motivations for building PDP models of cognitive processes was that symbolic approaches seem to lack certain characteristics that are needed at precisely the cognitive level of functioning. A compromise solution in which PDP mechanisms and symbol manipulating devices work side-by-side in an harmonious cognitive system, is currently coming into fashion. However, there is a danger in this approach. An advantage of PDP systems is that they can learn. On the other hand, symbolic systems are notoriously impenetrable and modular. They typically embody universal principles which cannot be extracted unaided from the stimuli to which they are exposed. To the extent that we accept a tandem architecture of PDP and symbolic systems we accept an a-developmental approach to human cognition. To be sure, the new-born mind comes equipped with a sophisticated set of constraints for processing its environment. However, we must be careful to ensure that the competences endowed on the new-born mind by a symbol processing device do not exceed the facts of human development. Given that we are only just beginning to scratch the surface of the potential learning capacitites of PDP systems, it would seem premature to compromise in favour of a more heterogeneous account. Parsimony requires a far more thorough investigation of the limits of the PDP approach to cognition.

FOOTNOTES

1. Thanks to »Kognitionsforskningsgruppen« in Aarhus (Klaus Bærentsen, John Paulin Hansen, Tove Klausen, Steen Folke Larsen, Gerda Linnemann, Uffe Seilman and Steen Wackerhausen) and Lars Hem for detailed comments on the manuscript.
2. Back propagation is a least-squares algorithm. The error (delta) on each output unit is propagated back to each unit that feeds into the given output unit in proportion to the weight of the connection between the units. Thus, each hidden unit collects weighted error signals from all the units it feeds into. The net sum of the weighted error signals for each hidden unit determines the »delta« for that unit. This »delta« is then used as the »error« signal to determine weights adjustments on the next layer down. The process iterates backwards through all levels of the network. Back propagation is a rather clumsy and computationally demanding learning algorithm. More effective substitutes are already available and the future promises learning algorithms that are capable of dynamic re-configuring of the network.
3. Several different classifications of the irregular verbs in English exist. These classifications differ in detail but all draw major distinctions between arbitrary, identity mapping and vowel change transformations as I do here (Bybee & Slobin, 1982; Pinker & Prince, 1988).

4. Since the details of the encoding process are not of direct concern for the present article, the reader is referred to the original source for further information (see also Pinker & Prince, 1988, and Bever, in press, for reviews and criticisms).
5. Hill-climbing is simply the inverse of gradient-descent. The lowest level of a valley in an energy landscape can represent a stable configuration of the network. Similarly, a peak on a »goodness« contour can represent the same stable set of parameters.

BIBLIOGRAPHY

ACKLEY, D.H., HINTON, G.E. & SEJNOWSKI, T.J. (1984). A learning algorithm for Boltzmann machines. *Cognitive Science, 9,* 147-169.

ANDERSEN, J.A. (1977). Neural models with cognitive implications. In D. Laberge & S.J. Samuels (Eds.) *Basic processes in reading perception and comprehension.* Hillsdale, New Jersey: Erlbaum, 27-90.

BERKO, J. (1958). The child's learning of English morphology. *Word, 40,* 150-177.

BEVER, T.G. (in press). The Demons and the beast - modular and nodular kinds of knowledge. In C. Georgopoulus & R. Ishihara (Eds.), *Interdisciplinary approaches to language: Essays in honour of S.Y. Kuroda.* Kluwer: Dordrecht.

BOWERMAN, M. (1982). Reorganisational processes in lexical and syntactic development. In E. Wanner & L.R. Gleitman (Eds.) *Language acquisition: the state of the art.* Cambridge: Cambridge University Press, 319-346.

BYBEE, J.L. & SLOBIN, D.I. (1982). Rules and schemes in the development and use of the English past tense. *Language, 58,* 265-289.

CHOMSKY, N. (1959). Review of B.F. Skinner's verbal behaviour. *Language, 35,* 16-58.

CHOMSKY, N. (1982). *Some concepts and consequences of the theory of government and binding.* Cambridge: M.I.T. Press.

DENNETT, D.C. (1988). When philosophers encounter artificial intelligence. In *Dædalus. The American Academy of Art and Sciences.* Cambridge. 283-295.

ELMAN, J.L. (1988). *Finding structure in time.* Center for research in language. Technical Report 8801. University of California, San Diego.

ERVIN, S. (1964). Imitation and structural change in children's language. In E. Lenneberg (Ed.) *New directions in the study of language.* Cambridge: M.I.T. Press.

FELDMAN, J.A. (1981). A connectionist model of visual memory. In G.E. Hinton & J.A. Anderson (Eds.) *Parallel models of associate memory.* Hillsdale, New Jersey: Erlbaum. 48-81.

FELDMAN, J.A. (1982). Dynamic connections in neural networks. *Biological Cybernetics, 46,* 22-39.

FODOR, J.A. (1987). *Psychosemantics: The problem of meaning in the philosophy of mind.* Cambridge: M.I.T. Press.

FODOR, J.A. & PYLYSHYN, Z.W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28,* 3-72.

FODOR, J.A., BEVER, T.D. & GARRET, M.F. (1974). *The psychology of language.* New York: McGraw Hill.

GROSSBERG, S. (1976). Adaptive pattern classification and universal recoding: Part I. Parallel development and coding of neural feature detectors. *Biological Cybernetics, 23,* 121-134.

GROSSBERG, S. (1980). How does the brain build a cognitive code? *Psychological Review, 87,* 1-51.

HEBB, D.O. (1949). The *organisation of behaviour.* New York: Wiley.

HINTON, G.E. (1986). *Learning distributed representations of concepts.* Technical report. Pittsburg: Carnegie-Mellon University. Dept. of Computer Science.

HINTON, G.E. & ANDERSON, J.A. (Eds.) (1981). *Parallel models of associative memory.* Hillsdale, N.Y.: Erlbaum.

HOPFIELD, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA, 79,* 2554-2558.

JAMES, W. (1892). *Psychology (Briefer course):* Holt.

JORDAN, M.I. (1986). *Serial order: A parallel distributed processing approach.* Institute for cognitive science. Report 8604. University of California, San Diego.

KOHONEN, T. (1977). *Associative memory: A system theoretical approach.* New York: Springer.

KUNO, S. (1987). *Functional syntax: anaphora, discourse and empathy.* Chicago: University of Chicago Press.

KUCZAJ, S.A. (1977). The aquisition of regular and irregular past tense forms. *Journal of Verbal Learning and Verbal Behaviour, 16,* 589-600.

LASHLY, A.S. (1951). The problem of serial order in behaviour. In L.A. Jeffress (Ed.) *Cerebral mechanisms in behaviour.* New York: Wiley. 112-136.

MCCLELLAND, J.L. & KAWAMOTO, A.H. (1986). Mechanisms of sentence processing: Assigning roles to constituents. In D.E. Rumelbart & J.L. McClelland (Eds.) *Parallel distributed processing: Explorations in the microstructure of cognition, 2.* Cambridge: M.I.T. 272-326.

MCCULLOCH, W.S. & PITTS, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5,* 115-133.

MILLER, G.A. & JOHNSON-LAIRD, P.N. (1976). *Languages and perception.* Cambridge: Harvard University Press.

MILLER, G.A., GALANTER, E. & PRIBRAM, K.H. (1960). *Plans and the structure of behaviour.* New York: Holt, Rinehart & Winston.

MINSKY, M. (1975). A framework for representing knowledge. In P.H. Winston (Ed.). *The psychology of computer vision.* New York: McGraw Hill.

MINSKY, M. & PAPERT, S. (1969). *Perceptrons.* Cambridge: M.I.T. Press.

PAPERT, S. (1988). One AI or many? In *Dædalus. The American Academy of Art and Sciences.* Cambridge. 1-14.

PINKER, S. & PRINCE, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28,* 73-193.

PLUNKETT, K. & MARCHMAN, V. (in press). *Pattern association in a back propagation network: Implications for language acquisition.* Technical Report, Center for research in language, University of California, San Diego.

RAMSEY, W. & STICH, S. (1988). *Connectionism, eliminativism and the future of folk psychology.* Unpublished manuscript.

ROSENBLATT, F. (1959). Two theorems of statistical separability in the perceptron. In *Mechanisation of thought processes: Proceedings of a symposium held at the National Physical laboratory, November 1959. 1.* London: J.M. Stationary Office. 421-456.

ROSENBLATT, F. (1962). *Principles of neural dynamics.* New York: Spartan.

RUMELHART, D.E. & MCCLELLAND, J.L. (1986a). *Parallel distributed processing: Explorations in the microstructure of cognition. 1-2.* Cambridge: M.I.T.

RUMELHART, D.E. & MCCLELLAND, J.L. (1986b). On learning the past tense of English verbs. In D.E. Rumelhart & J.L. McClelland (Eds.) *Parallel distributed processing: Explorations in the microstructure of cognition, 2.* Cambridge: M.I.T. Press, 216-271.

RUMELHART, D.E., HINTON, G.E. & WILLIAMS, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.) *Parallel distributed processing: Explorations in the microstructure of cognition, 1.* Cambridge: M.I.T. Press, 318-362.

RUMELHART, D.E., SMOLENSKY, P., MCCLELLAND, J.L. & HINTON, G.E. (1986). Schemata and sequential thought processes in PDP models. In D.E. Rumelhart & J.L. McClelland, *Parallel distributed processing: Explorations in the microstructure of cognition, 2.* Cambridge: M.I.T. press. 7-57.

SCHANK, R.C. & ABELSON, R.P. (1977). *Scripts, plans, goals and understanding.* Hillsdale: New Jersey: Erlbaum.

SMOLENSKY, P. (1988a). On the proper treatment of conenctionism. *The Behavioral and Brain Sciences, 11.*

SMOLENSKY, O. (1988b). *The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn.* Unpublished manuscript.

STALNAKER, R.C. (1987). *Inquiry.* Cambridge: M.I.T. Press.

STICH, S. (1983). *From folk psychology to cognitive science.* Cambridge: M.I.T. Press.

WASON, P.C. & JOHNSON-LAIRD, P.N. (1972). *Psychology of reasoning: Structure and content.* London: Batsford.