CHAPTER 4

PDP Models and General Issues in Cognitive Science

D. E. RUMELHART and J. L. MCCLELLAND

We are naturally optimistic about parallel distributed processing as a valuable framework for creating cognitive models. This does not mean, however, that there are no tough problems to be solved. Indeed, we have spent much of our effort convincing ourselves that PDP models could form a reasonable basis for modeling cognitive processes in general. In this chapter we shall address some of the objections that we and others have raised to the work and sketch our answers to these objections. However, we should like to say at the outset that we do not believe that any such general considerations as those discussed here will, in the end, bear much weight. The real proof is in the pudding. If PDP models are a valuable way to proceed, their usefulness will be proved in the added insights they bring to the particular substantive areas in which they are applied. The models we describe in later chapters are largely intended to constitute the beginnings of such a proof.

Many of the questions and issues raised below are addressed by material described in detail in other chapters in the book. For this reason, much of our present discussion is in the form of pointers to the relevant discussions. In this sense, this chapter serves not only as a discussion of our approach but as an overview of the issues and topics that are addressed in the chapters that follow.

SOME OBJECTIONS TO THE PDP APPROACH

PDP Models Are Too Weak

The one-layer perceptron. The most commonly heard objection to PDP models is a variant of the claim that PDP models cannot perform any interesting computations. One variant goes like this: "These PDP models sound a lot like perceptrons to me. Didn't Minsky and Papert show that perceptron-like models couldn't do anything interesting?" This comment represents a misunderstanding of what Minsky and Papert (1969) have actually shown. A brief sketch of the context in which Minsky and Papert wrote will help clarify the situation. (See Chapter 5 for a somewhat fuller account of this history.)

In the late 1950s and early 1960s there was a great deal of effort in the development of self-organizing networks and similar PDP-like computational devices. The best known of these was the *perceptron* developed by Frank Rosenblatt (see, for example, Rosenblatt, 1962). Rosenblatt was very enthusiastic about the perceptron and hopeful that it could serve as the basis both of artificial intelligence and the modeling of the brain. Minsky and Papert, who favored a *serial symbol processing* approach to artificial intelligence, undertook a very careful mathematical analysis of the perceptron in their 1969 book entitled, simply, *Perceptrons.*

The perceptron Minsky and Papert analyzed most closely is illustrated in Figure 1. Such machines consist of what is generally called a *retina*, an array of binary inputs sometimes taken to be arranged in a two-dimensional spatial layout; a set of *predicates*, a set of binary threshold units with fixed connections to a subset of units in the retina such that each predicate computes some local function over the subset of units to which it is connected; and one or more decision units, with modifiable connections to the predicates. This machine has only one layer of modifiable connections; for this reason we will call it a *one-layer* perceptron.

Minsky and Papert set out to show which functions can and cannot be computed by this class of machines. They demonstrated, in particular, that such perceptrons are unable to calculate such mathematical functions as parity (whether an odd or even number of points are on in the retina) or the topological function of connectedness (whether all points that are on are connected to all other points that are on either directly or via other points that are also on) without making use of absurdly large numbers of predicates. The analysis is extremely elegant and demonstrates the importance of a mathematical approach to analyzing computational systems.

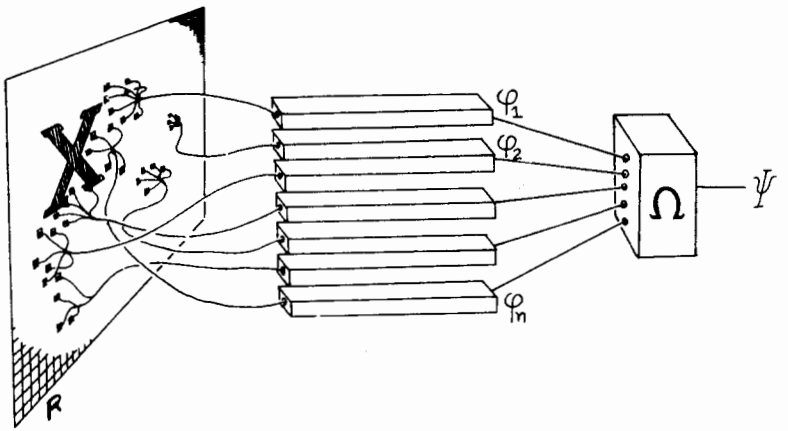

FIGURE 1. The one-layer perceptron analyzed by Minsky and Papert. (From *Perceptrons* by M. L. Minsky and S. Papert, 1969, Cambridge, MA: MIT Press. Copyright 1969 by MIT Press. Reprinted by permission.)

Minsky and Papert's analysis of the limitations of the one-layer perceptron, coupled with some of the early successes of the symbolic processing approach in artificial intelligence, was enough to suggest to a large number of workers in the field that there was no future in perceptron-like computational devices for artificial intelligence and cognitive psychology. The problem is that although Minsky and Papert were perfectly correct in their analysis, the results apply only to these simple one-layer perceptrons and not to the larger class of perceptron-like models. In particular (as Minsky and Papert actually conceded), it can be shown that a multilayered perceptron system, including several layers of predicates between the retina and the decision stage, can compute functions such as parity, using reasonable numbers of units each computing a very local predicate. (See Chapters 5 and 8 for examples of multilayer networks that compute parity). Similarly, it is not difficult to develop networks capable of solving the connectedness or inside/outside problem. Hinton and Sejnowski have analyzed a version of such a network (see Chapter 7).

Essentially, then, although Minsky and Papert were exactly correct in their analysis of the *one-layer perceptron,* the theorems don't apply to systems which are even a little more complex. In particular, it doesn't apply to multilayer systems nor to systems that allow feedback loops.

Minsky and Papert argued that there would not be much value to multilayer perceptrons. First, they argued that these systems are sufficiently unrestricted as to be vacuous. They pointed out, for example, that a universal computer could be built out of linear threshold units. Therefore, restricting consideration of machines made out of linear threshold units is no restriction at all on what can be computed.

We don't, of course, believe that the class of models sketched in Chapter 2 is a small or restrictive class. (Nor, for that matter, are the languages of symbol processing systems especially restrictive.) The real issue, we believe, is that different algorithms are appropriate to different architectural designs. We are investigating an architecture in which cooperative computation and parallelism is natural. Serial symbolic systems such as those favored by Minsky and Papert have a natural domain of algorithms that differs from those in PDP models. Not everything can be done in one step without feedback or layering (both of which suggest a kind of "seriality"). We have been led to consider models that have both of these features. The real point is that we seek algorithms that are *as parallel as possible.* We believe that such algorithms are going to be closer in form to the algorithms which could be employed by the hardware of the brain and that the kind of parallelism we employ allows the exploitation of multiple information sources and cooperative computation in a natural way.

A further argument advanced by Minsky and Papert against perceptron-like models with hidden units is that there was no indication how such multilayer networks were to be trained. One of the appealing features of the one-layer perceptron is the existence of a powerful learning procedure, the perceptron convergence procedure of Rosenblatt. In Minsky and Papert's day, there was no such powerful learning procedure for the more complex multilayer systems. This is no longer true. Chapters 5, 6, 7, and 8 all provide schemes for learning in systems with hidden units. Indeed, Chapter 8 provides a direct generalization of the perceptron learning procedure which can be applied to arbitrary networks with multiple layers and feedback among layers. This procedure can, in principle, learn arbitrary functions including, of course, parity and connectedness.

The problem of stimulus equivalence. A second problem with early PDP models—and one that is not necessarily completely overcome by multilayer systems—is the problem of invariance or *stimulus equivalence.* An *A* is an *A* is an *A*, no matter where on the retina it appears or how large it is or how it is oriented; and people can, in general, recognize patterns rather well despite various transformations. It has always seemed elegant and natural to imagine that an *A*, no matter where it is presented, is normalized and then processed for recognition using stored knowledge of the appearance of the letter (Marr, 1982; Neisser, 1967).

In conventional computer programs this seems to be a rather straightforward matter requiring, first, normalization of the input, and,

second, analysis of the normalized input. But in early PDP models it was never clear just how normalization could be made to work. Indeed, one of the main criticisms of perceptrons—one that is often leveled at more recent PDP models, too—is that they appear to provide no mechanism of attention, no way of focusing the machine on the analysis of a part of a larger whole and then switching to another part or back to the consideration of the whole.

While it is certainly true that certain PDP models lack explicit attentional mechanisms, it is far from true that PDP mechanisms are in principle incapable of exhibiting attentional phenomena. Likewise, while it is true that certain PDP models do not come to grips with the stimulus equivalence problem, it far from true that they are incapable of doing this in principle. To prove these points, we will describe a method for solving the stimulus equivalence problem that was described by Hinton (1981b). The idea is sketched in Figure 2. Essentially, it involves two sets of feature detectors. One (at the bottom of the figure) consists of *retinocentric* feature detectors and the other (above the retinocentric units) consists of canonical feature detectors. Higher order units that recognize canonical patterns (in this example, letters) sit above the canonical feature detectors and can have mutually excitatory connections to these feature detectors, just as in the interactive activation model of word recognition. What Hinton described was a method for mapping retinocentric feature patterns into canonical patterns. In general, for patterns in three-space, there are six degrees of freedom, but for present purposes we will consider only figures that are rotated around a fixed point in the plane. Here normalization simply amounts to a one-dimensional rotational transformation.

A fixed mapping from retinocentric units to canonical units would involve connecting each retinocentric feature detector to the corresponding canonical feature detector. Thus, to correct for a 90° clockwise rotation in the plane, we would want each retinal unit to project to the canonical unit corresponding to it at an offset of 90°.

How to implement variable mappings? Hinton proposed the use of a set of mapping units which act to switch on what amount to *dynamically programmable connections* from the retinocentric units to the canonical units. In the figure, three different mapping units are shown on the right: one that produces no rotation at all, one that produces a 90° clockwise rotation, and one that produces a 90° counterclockwise rotation. When one of these mapping units is active, it provides one of two inputs to a subset of the programmable connections. Thus, when the 90° clockwise mapping unit is active, it provides one of two inputs to the connection from each retinocentric unit to the central unit that corresponds to it under the 90° clockwise rotation. These connections are multiplicative—they pass the product of their two inputs on to the

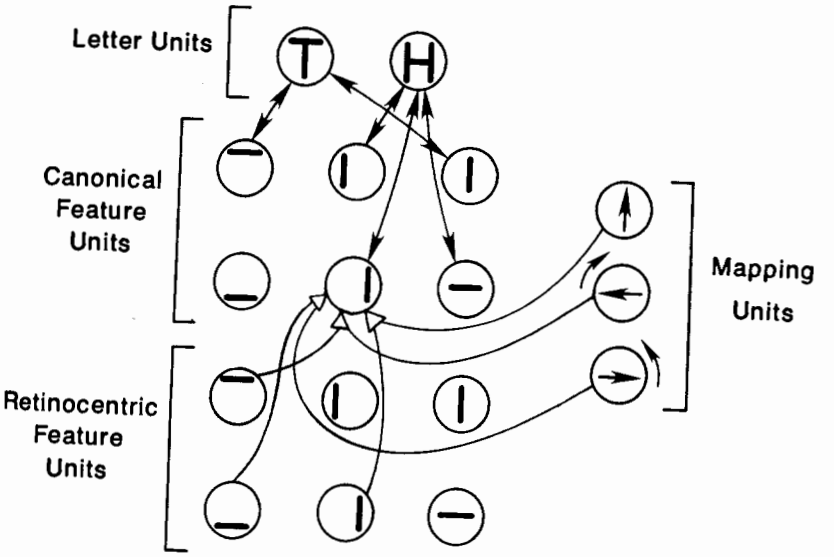

FIGURE 2. Hinton's (1981b) scheme for mapping patterns in one coordinate system into patterns in another coordinate system. At the top are two letter-detector units, with mutual excitatory connections to the six canonical feature units (the position and orientation of the line segment each of these detectors represents is indicated by the line segment in the "body" of each unit). At the bottom are six retinocentric feature units, and at the right are units corresponding to each of three different mappings from retinocentric to canonical features. (The arrows on the units indicate which direction in the retinocentric frame corresponds to upright in the canonical frame, and the arrow outside the unit indicates the nature of the transformation imposed on the retinocentric pattern). Each canonical unit receives three pairs of inputs, with each pair arriving at a multiplicative connection. These inputs are illustrated for one canonical unit only.

receiving unit. In this case, if a particular retinocentric feature is on and the 90° clockwise mapping unit is on, then the canonical feature corresponding to the active retinal feature will receive an excitatory input. If just one of the two inputs to the connection is on, no activation will flow to the central unit. In this way, when a mapping unit is active, it effectively programs the multiplicative connections needed to implement the corresponding mapping by activating one of the two inputs to each of the programmable connections.

Using this mechanism, it is possible to map from retinal to central coordinates if the mapping is known in advance. Object recognition can now proceed as follows: A mapping is chosen (perhaps on the basis of processing the preceding stimulus), and this is used to map a retinal input onto the canonical units. In a system involving variable

translational mappings, in addition to the rotational mappings shown here, it would be possible to focus the attention of the system successively on each of several different patterns merely by changing the mapping. Thus it would not be difficult to implement a complete system for sequential processing of a series of patterns using Hinton's scheme (a number of papers have proposed mechanisms for performing a set of operations in sequence, including Grossberg, 1978, and Rumelhart & Norman, 1982; the latter paper is discussed in Chapter 1).

So far, we have described what amounts to a PDP implementation of a conventional pattern recognition system. First, map the pattern into the canonical frame of reference, then recognize it. Such is the procedure advocated, for example, by Neisser (1967) and Marr (1982). The demonstration shows that PDP mechanisms are in fact capable of normalization and of focusing attention successively on one pattern after another.

But the demonstration may also seem to give away too much. For it seems to suggest that the PDP network is simply a method for implementing standard sequential algorithms of pattern recognition. We seem to be left with the question, what has the PDP implementation added to our understanding of the problem?

It turns out that it has added something very important. It allows us to begin to see how we could solve the problem of recognizing an input pattern even in the case where we do not know in advance either what the pattern is or which mapping is correct. In a conventional sequential algorithm, we might proceed by serial search, trying a sequence of mappings and looking to see which mapping resulted in the best recognition performance. With Hinton's mapping units, however, we can actually perform this search in parallel. To see how this parallel search would work, it is first necessary to see how another set of multiplicative connections can be used to choose the correct mapping for a pattern given both the retinal input and the correct central pattern of activation.

In this situation, this simultaneous activation of a central feature and a retinal feature constitutes evidence that the mapping that connects them is the correct mapping. We can use this fact to choose the mapping by allowing central and retinal units that correspond under a particular mapping to project to a common multiplicative connection on the appropriate mapping unit. Spurious conjunctions will of course occur, but the correct mapping units will generally receive more conjunctions of canonical and retinal features than any other (unless there is an ambiguity due to a symmetry in the figure). If the mapping units compete so that the one receiving the most excitation is allowed to win, the network can settle on the correct mapping.

We are now ready to see how it may be possible to simultaneously settle on a mapping and a central representation using both sets of multiplicative connections. We simply need to arrange things so that when the retinal input is shown, each possible mapping we wish to consider is partially active. Each retinal feature then provides partial activation of the canonical feature corresponding to it under each of the mappings. The correct mapping allows the correct canonical pattern to be partially activated, albeit partially obscured by noise generated by the other partially activated mappings. Interactive activation between this central pattern and higher level detectors for the pattern then reinforces the elements of the pattern relative to the noise. This process by itself can be sufficient for correct recognition. Further cleanup of the central pattern can be achieved, though, by allowing the pattern emerging on the central units to work together with the input pattern to support the correct mapping over the other partially active mappings via the multiplicative connections onto the mapping units. This then results in further suppression of the noise. As this process continues, it eventually locks in the correct interpretation of the pattern, the correct canonical feature representation, *and* the correct mapping, all from the retinal input alone. Prior activation of the correct mapping facilitates the process of settling in, as do prior cues to the identity of the figure (see Rock, 1973, and Palmer, 1980, for evidence that these clues do facilitate performance), but are not, in general, essential unless the input is in fact ambiguous without them.

Hinton's mapping scheme allows us to make two points. First, that parallel distributed processing is in fact compatible with normalization and focusing of attention; and second, that a PDP implementation of a normalization mechanism can actually produce a computational advantage by allowing what would otherwise be a painful, slow, serial search to be carried out in a single settling of a parallel network. In general, Hinton's mapping system illustrates that PDP mechanisms are not restricted to fixed computations but are quite clearly capable of modulation and control by signals arising from other parts of an integrated processing system; and that they can, when necessary, be used to implement a serial process, in which each of several patterns is considered, one at a time.

The introduction of multiplicative or contingent connections (Feldman & Ballard, 1982) is a way of greatly increasing the power of PDP networks of fixed numbers of units (Marr, 1982; Poggio & Torre, 1978; see Chapter 10). It means, essentially, that each unit can perform computations as complex as those that could be performed by an entire one-layer perceptron, including both the predicates and the decision unit. However, it must also be noted that multiplicative connections are not strictly necessary to perform the required conjunctive computational operations. Nonlinear, quasi-multiplicative interactions can be implemented in a variety of ways. In the worst case, whole units could

be dedicated to each multiplicative operation (as in the predicate layer of the perceptron).[1]

While Hinton's mapping mechanism indicates how attention might be implemented in PDP systems and imports some of the power of parallel distributed processing into the problem of simultaneously solving the mapping problem and the recognition problem, it does leave something to be desired. This is the fact that it allows only a single input pattern to be processed at one time since each pattern must be mapped separately onto the canonical feature units. Serial attention is sometimes required, but when we must resort to it, we lose the possibility of exploiting simultaneous, mutual constraints among several patterns. What has been processed before can still influence processing, but the ensemble of to-be-processed patterns cannot exert simultaneous, mutual influence on each other.

There is no doubt that sequentiality is forced upon us in some tasks—precisely those tasks in which the thought processes are extended over several seconds or minutes in time—and in such cases PDP mechanisms should be taken to provide potential accounts of the internal structure of a process evolving in time during the temporally extended structure of the thought process (see Chapter 14). But, in keeping with our general goals, we have sought to discover ways to maximally exploit simultaneous mutual constraints—that is, to maximize parallelism.

One mechanism which appears to make some progress in this direction is the connection information distribution mechanism described in Chapter 16. That mechanism uses multiplicative connections like those used in Hinton's model to send connection information out from a central knowledge store so that it can be used in local processing networks, each allocated to the contents of a different display location. The mechanism permits multiple copies of the same knowledge to be used at the same time, thereby effectively allowing tokens or local copies of patterns to be constructed from centrally stored knowledge of types in a parallel distributed processing system. These tokens then can interact with each other, allowing several patterns, all processed using the same centrally stored information, to exert simultaneous, mutual constraints on each other. Since these ideas, and their relation to attention, are discussed at length in Chapter 16, we will not elaborate on them further here.

[1] The linear threshold unit provides a quasi-multiplicative combination rule, and Sejnowski (1981) has described in detail how close approximation of the quantitative properties of multiplication of signals can be achieved by units with properties very much like those observed in real neurons.

Recursion. There are many other specific points that have been raised with respect to existing PDP models. Perhaps the most common one has to do with recursion. The ability to perform recursive function calls is a major feature of certain computational frameworks, such as augmented transition network (ATN) parsers (Woods, 1973; Woods & Kaplan, 1971), and is a property of such frameworks that gives them the capability of processing recursively defined structures such as sentences, in which embedding may produce dependencies between elements of a surface string that are indefinitely far removed from each other (Chomsky, 1957). It has often been suggested that PDP mechanisms lack the capacity to perform recursive computations and so are simply incapable of providing mechanisms for processing sentences and other recursively defined structures.

As before, these suggestions are simply wrong. As we have already seen, one can make an arbitrary computational machine out of linear threshold units, including, for example, a machine that can carry out all the operations necessary for implementing a Turing machine; the one limitation is that real biological systems cannot be Turing machines because they have finite hardware. In Chapter 14, however, we point out that with external memory aids (such as paper and pencil and a notational system) such limitations can be overcome as well.

We have not dwelt on PDP implementations of Turing machines and recursive processing engines because we do not agree with those who would argue that such capabilities are of the essence of human computation. As anyone who has ever attempted to process sentences like "The man the boy the girl hit kissed moved" can attest, our ability to process even moderate degrees of center-embedded structure is grossly impaired relative to that of an ATN parser. And yet, the human ability to use semantic and pragmatic contextual information to facilitate comprehension far exceeds that of any existing sentence processing machine we know of.

What is needed, then, is not a mechanism for flawless and effortless processing of center-embedded constructions. Compilers of computer languages generally provide such facilities, and they are powerful tools, but they have not demonstrated themselves sufficient for processing natural language. What is needed instead is a parser built from the kind of mechanism which facilitates the simultaneous consideration of large numbers of mutual and interdependent constraints. The challenge is to show how those processes that others have chosen to explain in terms of recursive mechanisms can be better explained by the kinds of processes natural for PDP networks.

This challenge is one that has not yet been fully met. However, some initial steps toward a PDP model of language processing are described in Chapter 19. The model whose implementation is

described in that chapter illustrates how a variety of different constraints may be combined by PDP models to aid in the assignment of underlying roles to the constituents of sentences. The chapter also provides a discussion of three different ways in which the model could be extended to process embedded clauses in a way that is roughly consistent with human capabilities and limitations in this regard.

We do not claim to have solved these problems. Our existing models have limitations and much remains to be done. Our explorations have just begun. The question is not, is the job done—no computational framework can claim much on this score. The question instead is, can more progress be made through further exploration of the PDP perspective on the microstructure of cognition? The discovery of multilayer learning rules, the use of multiplicative connections to implement transformations of input patterns, the distribution of connection information, and the host of other developments described throughout this book, indicate to us that the answer to the question is "yes."

PDP Models Are Not Cognitive

We have observed that the cooperative character of parallel distributed processing often allows us to account for behavior which has previously been attributed to the application of specific rules of grammar or rules of thought. This has sometimes led us to argue that lawful behavior is not necessarily *rule-driven* behavior. Here, we must distinguish between *rules* and *regularities*. The bouncing ball and the orbiting planet exhibit regularities in their behavior, but neither is applying rules. We have demonstrated the power of this approach in our earlier work on word perception (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982) and on the learning of English morphology (Chapter 18). In these cases we have been able to show how the apparent application of rules could readily *emerge* from interactions among simple processing units rather than from application of any higher level rules.

Some have viewed our argument against explicit rules as an argument against the cognitive approach to psychology. We do not agree. We believe that we are studying the *mechanisms* of cognition. The application of a rule (e.g., the firing of a production) is neither more nor less cognitive than the activation of our units. The real character of cognitive science is the attempt to explain mental phenomena through an understanding of the mechanisms which underlie those phenomena.

A related claim that some people have made is that our models appear to share much in common with behaviorist accounts of behavior. While they do involve simple mechanisms of learning, there is a crucial difference between our models and the radical behaviorism of Skinner and his followers. In our models, we are explicitly concerned with the problem of internal representation and mental processing, whereas the radical behaviorist explicitly denies the scientific utility and even the validity of the consideration of these constructs. The training of hidden units is, as is argued in Chapters 5 to 8, the construction of internal representations. The models described throughout the book all concern internal mechanisms for activating and acquiring the ability to activate appropriate internal representations. In this sense, our models must be seen as completely antithetical to the radical behaviorist program and strongly committed to the study of representation and process.

PDP Models Are the Wrong Level of Analysis

It is sometimes said that although PDP models are perfectly correct, they are at the wrong level of analysis and therefore not relevant to psychological data.[2] For example, Broadbent (1985) has argued that psychological evidence is irrelevant to our argument about distributed memory because the distribution assumption is only meaningful at what Marr (1982) has called the *implementational* (physiological) level and that the proper psychological level of description is the *computational* level.

The issues of levels of analysis and of theorizing is difficult and requires a good deal of careful thought. It is, we believe, largely an issue of scientific judgement as to what features of a lower level of analysis are relevant to a higher one. We are quite sure that it is not a matter for prescription. We begin our response to this objection with a review of Marr's analysis and his three levels of description. We then suggest that indeed our models are stated at the same level (in Marr's sense) as most traditional models from cognitive science. We then describe other senses of levels, including one in which higher level accounts can be said to be convenient approximations to lower level accounts. This sense comes closest to capturing our view of the

[2] The following discussion is based on a paper (Rumelhart & McClelland, 1985) written in response to a critique by Donald Broadbent (1985) on our work on distributed memory (cf. Chapter 17 and McClelland & Rumelhart, 1985).

relation between our PDP models and other traditional information processing models.

Marr's Notion of Levels

David Marr (1982) has provided an influential analysis of the issue of levels in cognitive science. Although we are not sure that we agree entirely with Marr's analysis, it is thoughtful and can serve as a starting point. Whereas Broadbent acknowledges only two levels of theory, the computational and the implementational, Marr actually proposes three, the *computational*, the *algorithmic*, and the *implementational* levels. Table 1 gives a description of Marr's three levels. We believe that PDP models are generally stated at the algorithmic level and are primarily aimed at specifying the representation of information and the processes or procedures involved in cognition. Furthermore, we agree with Marr's assertions that "each of these levels of description will have their place" and that they are "logically and causally related." Thus, no particular level of description is independent of the others. There is an implicit computational theory in PDP models as well as an appeal to certain implementational (physiological) considerations. We believe this to be appropriate. It is clear that different algorithms are more naturally implemented on different types of hardware and, therefore, information about the implementation can inform our hypotheses at the algorithmic level.

TABLE 1

THE THREE LEVELS AT WHICH ANY MACHINE CARRYING OUT INFORMATION PROCESSING TASKS MUST BE UNDERSTOOD

Computational Theory	Representation and Algorithm	Hardware Implementation
What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?	How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?	How can the representation and algorithm be realized physically?

Note. From *Vision* by D. Marr, 1982, San Francisco: W. H. Freeman. Copyright 1982 by W. H. Freeman. Reprinted by permission.

Computational models, according to Marr, are focused on a formal analysis of the problem the system is solving—not the methods by which it is solved. Thus, in linguistics, Marr suggests that Chomsky's (1965) view of a *competence* model for syntax maps most closely onto a *computational* level theory, whereas a psycholinguistic theory is more of a *performance* theory concerned with how grammatical structure might actually be computed. Such a theory is concerned with the algorithmic level of description. It is the algorithmic level at which we are concerned with such issues as efficiency, degradation of performance under noise or other adverse conditions, whether a particular problem is easy or difficult, which problems are solved quickly and which take a long time to solve, how information is represented, etc. These are all questions to which psychological inquiry is directed and to which psychological data is relevant. Indeed, it would appear that this is the level to which psychological data speaks most strongly. At the computational level, it does not matter whether the theory is stated as a program for a Turing machine, as a set of axioms, or as a set of rewrite rules. It does not matter how long the computation takes or how performance of the computation is affected by "performance" factors such as memory load, problem complexity, etc. It doesn't matter how the information is represented, as long as the representation is rich enough, in principle, to support computation of the required function. The question is simply *what function* is being computed, not *how* is it being computed.

Marr recommends that a good strategy in the development of theory is to begin with a careful analysis of the goal of a particular computation and a formal analysis of the problem that the system is trying to solve. He believes that this top-down approach will suggest plausible algorithms more effectively than a more bottom-up approach. Thus, the computational level is given some priority. However, Marr certainly does not propose that a theory at the computational level of description is an adequate psychological theory.

As psychologists, we are committed to an elucidation of the algorithmic level. We have no quarrel with Marr's top-down approach as a strategy leading to the discovery of cognitive algorithms, though we have proceeded in a different way. We emphasize the view that the various levels of description are interrelated. Clearly, the algorithms must, at least roughly, compute the function specified at the computational level. Equally clearly, the algorithms must be computable in amounts of time commensurate with human performance, using the kind and amount of hardware that humans may reasonably be assumed to possess. For example, any algorithm that would require more specific events to be stored separately than there are synapses in the brain should be given a lower plausibility rating than those that require much less storage. Similarly, in the time domain, those algorithms that

would require more than one serial step every millisecond or so would seem poor candidates for implementation in the brain (Feldman & Ballard, 1982).

In short, the claim that our models address a fundamentally different level of description than other psychological models is based on a failure to acknowledge the primary level of description to which much psychological theorizing is directed. At this level, our models should be considered as *competitors* of other models as a means of explaining psychological data.

Other notions of levels. Yet we do believe that in some sense PDP models are at a different level than other cognitive models such as prototype theories or schema theory. The reason is that there is more between the computational and the implementational levels than is dreamt of, even in Marr's scheme. Many of our colleagues have challenged our approach with a rather different conception of levels borrowed from the notion of levels of programming languages. It might be argued that a model such as, say, schema theory or the ACT* model of John R. Anderson (1983) is a statement in a "higher level" language analogous, let us say, to the Pascal or LISP programming languages and that our distributed model is a statement in a "lower level" theory that is, let us say, analogous to the assembly code into which higher level programs can be compiled. Both Pascal and assembler, of course, are considerably above the hardware level, though the latter may in some sense be closer to the hardware and more machine dependent than the former.

From this point of view one might ask why we are mucking around trying to specify our algorithms at the level of assembly code when we could state them more succinctly in a high-level language. We believe that most people who raise the levels issue with regard to our models have a relationship something like this in mind. People who adopt this notion have no objection to our models. They only believe that psychological models are more simply and easily stated in an equivalent higher level language—so why bother?

We believe that the programming language analogy is very misleading, unless it is analyzed more carefully. The relationship between a Pascal program and its assembly code counterpart is very special indeed. It is necessary for the Pascal and assembly language to map *exactly* onto one another only when the program was *written* in Pascal and the assembly code was compiled from the Pascal version. Had the original assembly code was compiled from the Pascal version. Had the original "programming" taken place in assembler, there is no guarantee that such a relationship would exist. Indeed, Pascal code will, in general, compile into only a small fraction of the possible assembly code programs that could be written. Since there is every reason to suppose

that most of the programming that might be taking place in the brain is taking place at a "lower level" rather than a "higher level," it seems unlikely that some particular higher level description will be identical to some particular lower level description. We may be able to capture the actual code approximately in a higher level language—and it may often be useful to do so—but this does not mean that the higher level language is an adequate characterization.

There is still another notion of levels which illustrates our view. This is the notion of levels implicit in the distinction between Newtonian mechanics on the one hand and quantum theory on the other.[3] It might be argued that conventional symbol processing models are macroscopic accounts, analogous to Newtonian mechanics, whereas our models offer more microscopic accounts, analogous to quantum theory. Note, that over much of their range, these two theories make precisely the same predictions about behavior of objects in the world. Moreover, the Newtonian theory is often much simpler to compute with since it involves discussions of entire objects and ignores much of their internal structure. However, in some situations Newtonian theory breaks down. In these situations we must rely on the microstructural account of quantum theory. Through a thorough understanding of the relationship between the Newtonian mechanics and quantum theory we can understand that the macroscopic level of description may be *only an approximation* to the more microscopic theory. Moreover, in physics, we understand just when the macrotheory will fail and the microtheory must be invoked. We understand the macrotheory as a useful formal tool by virtue of its relationship to the microtheory. In this sense the objects of the macrotheory can be viewed as *emerging* from interactions of the particles described at the microlevel.

The basic perspective of this book is that many of the constructs of macrolevel descriptions such as schemata, prototypes, rules, productions, etc. can be viewed as emerging out of interactions of the microstructure of distributed models. These points are most explicitly considered in Chapters 6, 14, 17, and 18. We view macrotheories as approximations to the underlying microstructure which the distributed model presented in our paper attempts to capture. As approximations they are often useful, but in some situations it will turn out that an examination of the microstructure may bring much deeper insight. Note for example, that in a conventional model of language acquisition, one has to make very delicate decisions about the exact circumstances under which a new rule will be added to the rule system. In our PDP models no such decision need be made. Since the analog to a rule is

[3] This analogy was suggested to us by Paul Smolensky.

not necessarily discrete but simply something that may emerge from interactions among an ensemble of processing units, there is no problem with having the functional equivalent of a "partial" rule. The same observation applies to schemata (Chapter 14), prototypes and logogens (Chapter 18), and other cognitive constructs too numerous to mention. Thus, although we imagine that rule-based models of language acquisition—the logogen model, schema theory, prototype theory, and other macrolevel theories—may all be more or less valid approximate macrostructural descriptions, we believe that the actual algorithms involved cannot be represented precisely in any of those macrotheories.

It may also be, however, that some phenomena are too complex to be easily represented as PDP models. If these phenomena took place at a time frame over which a macrostructural model was an adequate approximation, there is no reason that the macrostructural model ought not be applied. Thus, we believe that the concepts of symbols and symbol processing can be very useful. Such models may sometimes offer the simplest accounts. It is, however, important to keep in mind that these models are approximations and should not be pushed too far. We suspect that when they are, some account similar to our PDP account will again be required. Indeed, a large part of our own motivation for exploring the PDP approach came from the failure of schema theory to provide an adequate account of knowledge application even to the task of understanding very simple stories.

Lest it may seem that we have given too much away, however, it should be noted that as we develop clearer understandings of the microlevel models, we may wish to formulate rather different macrolevel models. As pointed out in Chapter 3, PDP mechanisms provide a powerful alternative set of macrolevel primitives.[4]

Imagine a computational system that has as a primitive, "Relax into a state that represents an optimal global interpretation of the current input." This would be, of course, an extremely powerful place to begin building up a theory of higher level computations. Related primitives would be such things as "Retrieve the representation in memory best matching the current input, blending into it plausible reconstructions of details missing from the original memory trace," and "Construct a dynamic configuration of knowledge structures that captures the present situation, with variables instantiated properly." These sorts of primitives would be unthinkable in most conventional approaches to higher level cognition, but they are the kinds of emergent properties that PDP mechanisms give us, and it seems very likely that the availability of

[4] We thank Walter Schneider for stressing in his comments on an earlier draft of this chapter the importance of the differences between the computational primitives offered by PDP and those offered by other formalisms for modeling cognitive processes.

such primitives will change the shape of higher level theory considerably.

PDP mechanisms may also place some constraints on what we might realistically ask for in the way of computational primitives because of the costs of implementing certain kinds of computations in parallel hardware in a single relaxation search. The parallel matching of variablized productions is one case in point. Theories such as ACT* (J. R. Anderson, 1983) assume that this can be done without worrying about the implementation and, therefore, provide no principled accounts of the kinds of crosstalk exhibited in human behavior when processing multiple patterns simultaneously. However, it appears to be a quite general property of PDP mechanisms that they will exhibit crosstalk when processing multiple patterns in parallel (Hinton & Lang, 1985; Mozer, 1984; see Chapters 12 and 16).

High-level languages often preserve some of the character of the lower level mechanisms that implement them, and the resource and time requirements of algorithms drastically depends on the nature of the underlying hardware. Higher level languages that preserve the character of PDP mechanisms and exploit the algorithms that are effective descriptions of parallel networks are not here yet, but we expect such things to be coming along in the future. This will be a welcome development, in our view, since certain aspects of cognitive theory have been too strongly influenced by the discrete, sequential algorithms available for expression in most current high-level languages.

As we look closely, both at the hardware in which cognitive algorithms are implemented and at the fine structure of the behavior that these algorithms are designed to capture, we begin to see why it may be appropriate to formulate models which come closer to describing the microstructure of cognition. The fact that our microstructural models can account for many of the facts about the representation of general and specific information, for example, as discussed in Chapter 18, makes us ask why we should view constructs like logogens, prototypes, and schemata as anything other than convenient approximate descriptions of the underlying structure of memory and thought.

Reductionism and Emergent Properties

A slightly different, though related, argument is that the PDP enterprise is an exercise in reductionism—an exercise in which all of psychology is reduced to neurophysiology and ultimately to physics. It is argued that coherent phenomena which emerge at any level (psychology or physics or sociology) require their own language of description

and explanation and that we are denying the essence of what is cognitive by reducing it to units and connections rather than adopting a more psychologically relevant language in our explanations.

We do not classify our enterprise as reductionist, but rather as interactional. We understand that new and useful concepts emerge at different levels of organization. We are simply trying to *understand* the essence of cognition as a property emerging from the *interactions* of connected units in networks.

We certainly believe in emergent phenomena in the sense of phenomena which could never be understood or predicted by a study of the lower level elements in isolation. These phenomena are functions of the particular kinds of groupings of the elementary units. In general, a new vocabulary is useful to talk about aggregate phenomena rather than the characteristics of isolated elements. This is the case in many fields. For example, we could not know about diamonds through the study of isolated atoms; we can't understand the nature of social systems through the study of isolated individuals; and we can't understand the behavior of networks of neurons from the study of isolated neurons. Features such as the hardness of the diamond is understandable through the interaction of the carbon atoms and the way they line up. The whole is different than the *sum* of the parts. There are nonlinear interactions among the parts. This does not, however, suggest that the nature of the lower level elements is irrelevant to the higher level of organization—on the contrary, the higher level is, we believe, to be understood primarily through the study of the interactions among lower level units. The ways in which units interact is not predictable from the lower level elements as isolated entities. It is, however, predictable *if* part of our study involves the interactions among these lower level units. We *can* understand why diamonds are hard, not as an isolated fact, but because we understand how the atoms of carbon can line up to form a perfect lattice. This is a feature of the aggregate, not of the individual atom, but the features of the atom are necessary for understanding the aggregate behavior. Until we understand that, we are left with the unsatisfactory statement that diamonds are hard, period. A useful fact, but not an explanation. Similarly, at the social level, social organizations cannot be understood without understanding the individuals which make up the organization. Knowing about the individuals tells us little about the structure of the organization, but we can't *understand* the structure of the higher level organizations without knowing a good deal about individuals and how they function. This is the sense of emergence we are comfortable with. We believe that it is entirely consistent with the PDP view of cognition.

There is a second, more practical reason for rejecting radical reductionism as a research strategy. This has nothing to do with emergence; it has to do with the fact that we can't know everything and find out everything at once. The approach we have been arguing for suggests that to understand something thoroughly at some level requires knowledge at that level, plus knowledge of the lower levels. Obviously, this is impractical. In practice, even though there might be effects of lower levels on higher levels, one cannot always know them. Thus, attempting to formulate a description at this higher level as a first order of approximation is an important research strategy. We are forced into it if we are to learn anything at all. It is possible to learn a good deal about psychology without any reference whatsoever to any lower levels. This practical strategy is not, however, an excuse for ignoring what *is* known about the lower levels in the formulation of our higher level theories. Thus, the economist is wrong to ignore what we might know about individuals when formulating his theories. The chemist would be wrong to ignore what is known about the structure of the carbon atom in explaining the hardness of diamonds. We argued above that the view that the computational level is correct derives from experience with a very special kind of device in which the higher level was *designed* to give the right answers—exactly. In describing natural intelligence that can't, we suspect, be right—exactly. It can be a first order of approximation. As we learn more about a topic and as we look at it in more and more detail we are going to be forced to consider more and more how it might emerge (in the above sense) from the *interactions* among its constituents. Interaction is the key word here. Emergent properties occur whenever we have nonlinear interactions. In these cases the principles of interaction themselves must be formulated and the real theory at the higher level is, like chemistry, a theory of interactions of elements from a theory one level lower.

Not Enough Is Known From Neuroscience to Seriously Constrain Cognitive Theories

Many cognitive scientists believe that there will eventually be an understanding of the relationships between cognitive phenomena and brain functioning. Many of these same people feel, however, that the brain is such an exceptionally powerful computational device that it is capable of performing just about any computation. They suppose that facts now known from neuroscience place little or no restriction on what theories are possible at a cognitive level. In the meantime, they suppose, a top-down analysis of possible mechanisms of cognition can lead to an understanding of cognition that will stand independently of whatever might be discovered about brain functioning. Moreover, they

believe that neuroscientists can be guided in their bottom-up search for an understanding of how the brain functions.

We agree with many of these sentiments. We believe that an understanding of the relationships between cognitive phenomena and brain functions will slowly evolve. We also believe that cognitive theories can provide a useful source of information for the neuroscientist. We do not, however, believe that current knowledge from neuroscience provides no guidance to those interested in the functioning of the mind. We have not, by and large, focused on the kinds of constraints which arise from detailed analysis of particular circuitry and organs of the brain. Rather we have found that information concerning *brain-style* processing has itself been very provocative in our model building efforts. Thus, we have, by and large, not focused on *neural modeling* (i.e., the modeling of neurons), but rather we have focused on *neurally inspired* modeling of cognitive processes. Our models have not depended strongly on the details of brain structure or on issues that are very controversial in neuroscience. Rather, we have discovered that if we take some of the most obvious characteristics of brain-style processing seriously we are led to postulate models which differ in a number of important ways from those postulated without regard for the hardware on which these algorithms are to be implemented. We have found that top-down considerations revolving about a need to postulate parallel, cooperative computational models (cf. Rumelhart, 1977) have meshed nicely with a number of more bottom-up considerations of brain style processing.

There are many brain characteristics which ought to be attended to in the formulation of our models (see Chapters 20 and 21). There are a few which we have taken most seriously and which have most affected our thinking. We discuss these briefly below.

Neurons are slow. One of the most important characteristics of brain-style processing stems from the speed of its components. Neurons are much slower than conventional computational components. Whereas basic operations in our modern serial computers are measured in the nanoseconds, neurons operate at times measured in the milliseconds—perhaps 10s of milliseconds. Thus, the basic hardware of the brain is some 10^6 times slower than that of serial computers. Imagine slowing down our conventional AI programs by a factor of 10^6. More remarkable is the fact that we are able to do very sophisticated processing in a few hundred milliseconds. Clearly, perceptual processing, most memory retrieval, much of language processing, much intuitive reasoning, and many other processes occur in this time frame. That means that these tasks must be done in no more than 100 or so serial steps. This is what Feldman (1985) calls the *100-step program constraint.* Moreover, note that individual neurons probably don't compute very complicated functions. It seems unlikely that a single neuron computes a function much more complex than a single instruction in a digital computer. Imagine, again, writing an interesting program in even 1000 operations of this limited complexity of a serial computer. Evidently, the brain succeeds through *massive parallelism.* Thus, we conclude, the mechanisms of mind are most likely best understood as resulting from the cooperative activity of very many relatively simple processing units operating in parallel.

There is a very large number of neurons. Another self-evident, but important, aspect of brain-style processing is the very large number of processing units involved. Conventional estimates hold that there are on the order of 10^{10} to 10^{11} neurons in the brain. Moreover, each neuron is an *active* processing unit. This suggests parallelism on a very large scale indeed. An understanding of parallel computation involving a few hundred reasonably complex processors provides the wrong model. It may well be that it is the massive scale of the parallelism of the brain that gives it its amazing power.

Although the human brain is large, the number of neurons is not unlimited. It happens that our models sometimes push the limits of plausibility because of the large number of units they require. This is a real constraint, one that we and others have begun to take into account in evaluating our models (see Chapter 12 for a discussion of this issue).

Neurons receive inputs from a large number of other neurons. Another important feature of brain processing is the large fan-in and fan-out to and from each unit. Estimates vary, but single cortical neurons can have from 1,000 to 100,000 synapses on their dendrites and, likewise, can make from 1,000 to 100,000 synapses on the dendrites of other neurons. Generally, one or a small number of action potentials received are not enough to generate an action potential (see, for example, Chapter 20). This suggests that human computation does not involve the kind of logic circuits out of which we make our digital computers, but that it involves a kind of statistical process in which the single units do not make decisions, but in which decisions are the product of the cooperative action of many somewhat independent processing units. Reliability derives from the stability of the statistical behavior of large numbers of units. Again, this degree of connectivity should be contrasted with the number of immediate neighbors of processors in current parallel computers. Usually these numbers are measured in the tens (or less) rather than in the thousands. Moreover, this large degree of connectivity suggests that no neuron is very many synapses away from any other neuron. If, for argument's sake, we assume that every

cortical neuron is connected to 1,000 other neurons and that the system forms a lattice, all of the neurons in the brain would be within, at most, four synapses from one another. Thus, large fan-in and fan-out leads to shallow networks. It should finally be noted that even though the fan-in and fan-out is large, it is not unlimited. As described in Chapter 12, the limitations can cause problems for extending some simple ideas of memory storage and retrieval.

Learning involves modifying connections. Another key feature of our models which derives from our understanding of learning mechanisms in the brain is that the *knowledge is in the connections* rather than in the units themselves. Moreover, learning is generally assumed to involve modifying connection strengths. There are real computational advantages to such a simple learning procedure. Its simplicity and homogeneity allow us to develop powerful learning procedures which work simply and incrementally. (See Chapters 5, 6, 7, 8; Chapters 11, 17, 18, 24, and 25 consider the implications of this view.)

Neurons communicate by sending activation or inhibition through connections. Communication among neurons involves simple excitatory and inhibitory messages. Only a few bits can be communicated per second. Thus, unlike other parallel message passing systems such as Hewitt's (1975) ACTOR system which allows arbitrary symbolic messages to be passed among its units, we require simple, signed numbers of limited precision. This means that the currency of our systems is not symbols, but excitation and inhibition. To the degree that symbols are required, they must emerge from this subsymbolic level of processing (Hofstadter, 1979).

Connections in the brain seem to have a clear geometric and topological structure. There are a number of facts about the pattern of connections in the brain which, we believe, are probably important, but which have not yet had a large impact on our models. First, most connections are rather short. Some are long (these tend to be excitatory), but not most. There are rather strong geometric and topological constraints. There is a rough mapping in that input parameters (such as spatial location in vision or frequency in audition) are mapped onto spatial extent in the brain. In general it seems that nearby regions in one part of the brain map onto nearby regions in another part of the brain. Moreover, there is a general symmetry of connections. If there are connections from one region of the brain to another, there are usually connections in the reverse direction. Some of these features have been implemented in our models, though, interestingly, most often for computational reasons rather than for biological verisimilitude. For

example, rough symmetry was a feature of our earlier work on word perception (cf. McClelland & Rumelhart, 1981), and it is a feature of the work described in Chapters 6, 7, 14, 15, and 16. The error propagation learning rule of Chapter 8 requires a back path for an error signal to be propagated back through. In general, reciprocally interacting systems are very important for the kind of processing we see as characteristic of PDP models. This is the defining feature of *interactive activation* models. We have also employed the view that connections between systems are excitatory and those within a region are inhibitory. This is employed to advantage in Chapters 5 and 15.

The geometric structure of connections in the brain have not had much impact on our work. We generally have not concerned ourselves with *where* the units might physically be with respect to one another. However, if we imagine that there is a constraint toward the conservation of connection length (which there must be), it is easy to see that those units which interact most should be the closest together. If you add to this the view that the very high-dimensional space determined by the *number* of interconnections must be embedded into the two- or three-dimensional space (perhaps two and a half dimensions) of the cortex, we can see the importance of mapping the major dimensions physically in the geometry of the brain (see Ballard, in press, for a discussion of embedding high-dimensional spaces into two dimensions).

Information is continuously available. Another important feature of neural information processing is that the neurons seem to provide *continuously available output* (Norman & Bobrow, 1975). That is, there does not seem to be an appreciable decision phase during which a unit provides no output. Rather it seems that the state of a unit reflects its current input. To the degree that a unit represents a hypothesis and its activation level (instantaneous firing rate or probability of firing) represents the degree to which evidence favors that hypothesis, the activation level of the unit provides continuous information about the current evaluation of that hypothesis. This hypothesis was incorporated into the precursors of our own work on parallel distributed processing, especially the *cascade* model (McClelland, 1979) and the interactive model of reading (Rumelhart, 1977), and it is a feature of virtually all of the PDP models in this book.[5] Interestingly, this contrasts starkly with what used to be the standard approach, namely, *stage* models of information processing (Sternberg, 1969), and thereby offers a very

[5] Though some PDP models use discrete binary units (e.g., Hinton, 1981a; Hopfield, 1982), they generally use large numbers of these to represent any object, so that when a few of the units that form part of a pattern are on, the pattern can be said to be partially active.

different perspective on decision-making processes and the basic notion of stages.

Graceful degradation with damage and information overload. From the study of brain lesions and other forms of brain damage, it seems fairly clear there is not some single neuron whose functioning is essential for the operation of any particular cognitive process. While reasonably circumscribed *regions* of the brain may play fairly specific roles, particularly at lower levels of processing, it seems fairly clear that within regions, performance is characterized by a kind of *graceful degradation* in which the system's performance gradually deteriorates as more and more neural units are destroyed, but there is no single critical point where performance breaks down. This kind of graceful degradation is characteristic of such global degenerative syndromes as Alzheimer's disease (cf. Schwartz, Marin, & Saffran, 1979). Again, this is quite different from many serial symbol processing models in which the disruption of a single step in a huge program can catastrophically impact the overall performance of the system. Imagine the operation of a computer in which a particular instruction did not work. So long as that instruction was not used, there would be no effect on the system. However, when that instruction was employed in some process, that process simply would not work. In the brain it seems that the system is highly redundant and capable of operating with a loss in performance roughly similar in magnitude to the magnitude of the damage (see Chapter 12 for details). This is a natural performance characteristic of PDP models.

Distributed, not central, control. There is one final aspect of our models which is vaguely derived from our understanding of brain functioning. This is the notion that there is *no central executive* overseeing the general flow of processing. In conventional programming frameworks it is easy to imagine an executive system which calls subroutines to carry out its necessary tasks. In some information processing models this notion of an executive has been carried over. In these models, all processing is essentially *top-down* or *executive-driven*; if there is no executive, then no processing takes place at all.

Neuropsychological investigation of patients with brain damage indicates that there is no part of the cortex on whose operation all other parts depend. Rather it seems that all parts work together, influencing one another, and each region contributes to the overall performance of the task and to the integration into it of certain kinds of constraints or sources of information. To be sure, brainstem mechanisms control vital bodily functions and the overall state of the system, and certain parts of the cortex are critical for receiving information in particular

modalities. But higher level functions seem very much to be characterized by distributed, rather than central control.

This point has been made most clearly by the Russian neuropsychologist Luria (1966; 1973). Luria's investigations show that for every integrated behavioral function (e.g., visual perception, language comprehension or production, problem solving, reading), many different parts of the cortex play a role so that damage to any part influences performance but is not absolutely crucial to it. Even the frontal lobes, most frequently associated with executive functions, are not absolutely necessary in Luria's view, in that some residual function is generally observed even after massive frontal damage (and mild frontal damage may result in no detectable symptomatology at all). The frontal lobes have a characteristic role to play, facilitating strategy shifts and inhibiting impulsive responding, but the overall control of processing can be as severely impaired by damage to parietal lobe structures that appear to be responsible for maintaining organized representations that support coordinated and goal-directed activity.

Our view of the overall organization of processing is similar to Luria's. We have come to believe that the notion of subroutines with one system "calling" another is probably not a good way to view the operation of the brain. Rather, we believe that subsystems may *modulate* the behavior of other subsystems, that they may provide constraints to be factored into the relaxation computation. An elaboration of some aspects of these ideas may be found in Chapter 14.

Relaxation is the dominant mode of computation. Although there is no specific piece of neuroscience which compels the view that brain-style computation involves relaxation, all of the features we have just discussed have led us to believe that the primary mode of computation in the brain is best understood as a kind of *relaxation system* (cf. Chapters 6, 7, 14, 15, and 21) in which the computation proceeds by iteratively seeking to satisfy a large number of weak constraints. Thus, rather than playing the role of wires in an electric circuit, we see the connections as representing constraints on the co-occurrence of pairs of units. The system should be thought of more as *settling into a solution* than *calculating a solution*. Again, this is an important perspective change which comes out of an interaction of our understanding of how the brain must work and what kinds of processes seem to be required to account for desired behavior.

As can be seen, this list does not depend on specific discoveries from neuroscience. Rather, it depends on rather global considerations. Although none of these general properties of the brain tell us in any detail how the brain functions to support cognitive phenomena, together they lead to an understanding of how the brain works that

serves as a set of constraints on the development of models of cognitive processes. We find that these assumptions, together with those that derive from the constraints imposed by the tasks we are trying to account for, strongly influence the form of our models of cognitive processes.

PDP Models Lack Neural Realism

On the one hand, it is sometimes said—as indicated in the previous section—that there is little or no constraint to be gained through looking at the brain. On the other hand, it is often said that we don't look closely enough. There are many facts of neuroscience that are not factored directly into our models. Sometimes we have failed to capture the fine structure of neural processing in our models. Other times we have assumed mechanisms that are not known to exist in brains (see Chapter 20). One prominent example is the near-ubiquitous assumption that units can have both excitatory and inhibitory connections when it seems reasonably clear that most cortical units are either excitatory or inhibitory. If, as we argued above, it is important to understand the microstructure of cognition, why do we ignore such detailed characteristics of the actual physical processes underlying that microstructure?

To be sure, to the extent that our models are directly relevant to brains, they are at best coarse approximations of the details of neurophysiological processing. Indeed, many of our models are clearly intended to fall at a level between the macrostructure of cognition and the details of neurophysiology. Now, we do understand that some of our approximations may have ramifications for the cognitive phenomena which form our major area of interest; by missing certain details of neurophysiology, we may be missing out on certain aspects of brain function that would make the difference between an accurate account of cognitive-level phenomena and a poor approximation. Our defense is simply that we see the process of model building as one of successive approximations. We try to be responsive to information from both the behavioral and the neural sciences. We also believe that the key to scientific progress is making the right approximations and the right simplifications. In this way the structure can be seen most clearly. This point is considered further in Chapter 21.

We have been pleased with the structure apparent through the set of approximations and simplifications we have chosen to make. There are, however, a number of other facts from neuroscience that we have not included in most of our models, but that we imagine will be important when we learn how to include them. The most obvious of these is

the fact that we normally assume that units communicate via numbers. These are sometimes associated with mean firing rates. In fact, of course, neurons produce spikes and this spiking itself may have some computational significance (see Chapters 7 and 21 for discussions of the possible computational significance of neural spiking). Another example of possibly important facts of neuroscience which have not played a role in our models is the diffuse pattern of communication which occurs by means of the dispersal of chemicals into various regions of the brain through the blood stream or otherwise. We generally assume that communication is point-to-point from one unit to another. However, we understand that diffuse communication can occur through chemical means and such communication may play an important role in setting parameters and modulating the networks so that they can perform rather different tasks in different situations. We have employed the idea of diffuse distribution of chemicals in our account of amnesia (Chapter 25), but, in general, we have not otherwise integrated such assumptions into our models. Roughly, we imagine that we are studying networks in which there is a fixed setting of such parameters, but the situation may well be much more complex than that. (See Chapter 24 for some discussion of the role of norepinephrine and other neuromodulators.)

Most of our models are homogeneous with respect to the functioning of our units. Some of them may be designated as inhibitory and others as excitatory, but beyond that, they are rarely differentiated. We understand that there are perhaps hundreds of kinds of neurons (see Chapter 20). No doubt each of these kinds play a slightly different role in the information processing system. Our assumptions in this regard are obviously only approximate. Similarly, we understand that there are many different kinds of neurotransmitters and that there are different systems in which different of these neurotransmitters are dominant. Again, we have ignored this difference (except for excitatory and inhibitory connections) and presume that as more is understood about the information processing implications of such facts we will be able to determine how they fit into our class of models.

It is also true that we have assumed a number of mechanisms that are not known to exist in the brain (see Chapter 20). In general, we have postulated mechanisms which seemed to be required to achieve certain important functional goals, such as, for example, the development of internal representations in multilayer networks (see Chapter 8). It is possible that these hypothesized mechanisms do exist in the brain but have not yet been recognized. In that sense our work could be considered as a source of hypotheses for neuroscience. It is also possible that we are correct about the computations that are performed, but that they are performed by a different kind of neural mechanism

than our formulations seem at first glance to suggest. If this is the case, it merely suggests that the most obvious mapping of our models onto neural structures is incorrect.

A neuroscientist might be concerned about the ambiguity inherent in the fact that many of the mechanisms we have postulated could be implemented in different ways. From our point of view, though, this is not a serious problem. We think it useful to be clear about how our mechanisms *might* be implemented in the brain, and we would certainly be worried if we proposed a process that could not be implemented in the brain. But since our primary concern is with the computations themselves, rather than the detailed neural implementation of these computations, we are willing to be instructed by neuroscientists on which of the possible implementations are actually employed. This position does have its dangers. We have already argued in this chapter that the mechanism whereby a function is computed often has strong implications about *exactly what* function is being computed. Nevertheless, we have chosen a level of approximation which seems to us the most fruitful, given our goal of understanding the human information processing system.

We close this section by noting two different ways in which PDP models can be related to actual neurophysiological processes, apart from the possibility that they might actually be intended to model what is known about the behavior of real neural circuitry (see Chapters 23 and 24 for examples of models of this class). First, they might be intended as idealizations. In this approach, the emergent properties of systems of real neurons are studied by idealizing the properties of the individual neurons, in much the same way that the emergent properties of real gasses can be studied by idealizing the properties of the individual gas molecules. This approach is described at the end of Chapter 21. An alternative is that they might be intended to provide a higher level of description, but one that could be mapped on to a real neurophysiological implementation. Our interactive activation model of word recognition has some of this flavor, as do most of the models described in Chapters 14 through 19. Specifically with regard to the word recognition model, we do not claim that there are individual neurons that stand for visual feature, letter, and word units, or that they are connected together just as we proposed in that model. Rather, we really suppose that the physiological substrate provides a mechanism whereby various abstract informational states—such as, for example, the state in which the perceptual system is entertaining the hypothesis that the second letter in a word is either an *H* or an *A*—can give rise to other informational states that are contingent upon them.

Nativism vs. Empiricism

Historically, perceptron-like models have been associated with the idea of "random self-organizing" networks, the learning of arbitrary associations, very general, very simple learning rules, and similar ideas which show the emergence of structure from the *tabula rasa*. We often find, especially in discussion with colleagues from linguistics surrounding issues of language aquisition (see Chapters 18 and 19), that PDP models are judged to involve learning processes that are too general and, all in all, give too little weight to innate characteristics of language or other information processing structures. This feeling is brought out even more by demonstrations that some PDP learning mechanisms are capable of learning to respond to symmetry and of learning how to deal with such basic perceptual problems as perceptual constancy under translation and rotation (see Chapter 8). In fact, however, PDP models are, in and of themselves, quite agnostic about issues of nativism versus empiricism. Indeed, they seem to us to offer a very useful perspective on the issue of innate versus acquired knowledge.

For the purposes of discussion let us consider an organism that consists of a very large set of very simple but highly interconnected processing units. The units are assumed to be homogeneous in their properties except that some are specialized to serve as "input" units because they receive inputs from the environment and some are specialized to serve as "output" units because they drive the effectors of the system. The behavior of such a system is thus entirely determined by the pattern of inputs, the pattern of interconnections among the units, and the nature of and connections to the effectors. Note, that interconnections can have various strengths—positive, negative, and zero. If the strength of connection is positive, then activity in one unit tends to increase the activity of the second unit. If the strength of connection is negative, then the activity in the first unit tends to decrease the activity of the second unit. If the strength is zero, then activity of the first unit has no effect on the activity of the second.

In such a system the radical nativism hypothesis would consist of the view that all of the interconnections are genetically determined at birth and develop only through a biologically driven process of maturation. If such were the case, the system could have any particular behavior entirely wired in. The system could be designed in such a way as to respond differentially to human speech from other acoustic stimuli, to perform any sort of computation that had proven evolutionarily adaptive, to mimic any behavior it might observe, to have certain stimulus dimensions to which it was pretuned to respond, etc. In short, if all of the connections were genetically predetermined, the system could

perform *any* behavior that such a system of units, interconnections, and effectors might ever be capable of. The question of what behaviors it actually did carry out would presumably be determined by evolutionary processes. In this sense, this simple PDP model is clearly consistent with a rabidly nativist world view.

The radical empiricist hypothesis, on the other hand, suggests that there are no a priori limits on how the network of interconnections could be constituted. Any pattern of interconnections is possible. What determines the actual set of connections is the pattern of experiences the system gets. In this sense there is no prior limit on the nature of language; any language that could be processed by such a network could be learned by such an organism. The only limitations would be very general ones due to the nature of the learning rule in the system. With a sufficiently powerful learning rule, the organism could organize itself into whatever state proved maximally adaptive. Thus, there would be no limitation on the degree to which the behavior of the system could adapt to its environment. It could learn completely arbitrary associations. In short, if all connections in the system were modifiable by experience, the system could learn to perform any behavior at all that such a system of units, interconnections, and effectors might ever be capable of. The question of what behaviors it actually did carry out would presumably be determined by the learning process and the patterns of inputs the system actually experienced. In this sense, the simple PDP model is clearly consistent with a rabidly empiricist world view.

Obviously, it would be a straightforward matter to find a middle ground between the radical nativist view and the radical empiricist view as we have laid them out. Suppose, for sake of argument, that we have an organism whose initial state is wholly determined genetically. Suppose further that all of the connections were modifiable so that whatever the start state, any pattern of interconnections could emerge through interaction of the organism with its environment.[6] In such a system as this we have, it seems to us, the benefits of both nativism and empiricism. Like good nativists, we have given the organism a starting point that has been selected by its evolutionary history. We have not, however, strapped the organism with the rigid predeterminism that traditionally goes along with the nativist view. If there are

[6] Obviously both of these views are overstatements. Clearly the genes do not determine *every* connection at birth. Probably some sort of random processes are also involved. Equally clearly, not *every* pattern of interconnectivity is possible since the spatial layout of the neurons in the cortex, for example, surely limit the connectivity. Still, there is probably a good deal of genetic specification of neural connection, and there is a good deal of plasticity in the pattern of connectivities after birth.

certain patterns of behavior which, in evolutionary time, have proven to be useful (such as sucking, reaching, or whatever) we can build them in, but we leave the organism free to modify or completely reverse any of these behavioral predispositions.[7] At the same time, we have the best of the empiricist view—namely, we place no a priori limitations on how the organism may adapt to its environment. We do, however, throw out the weakest aspect of the empiricist dogma—namely, the idea of the *tabula rasa* (or totally random net) as a starting point. The organism could start at whatever initial state its evolutionary history prepared it for.

Perhaps, at this stage, all of this seems painfully obvious. It seems obvious to us too, and nevertheless, it gives us a new perspective on the nativism/empiricism issue. The issue is not what is *the* set of predetermined modules as some would suggest (cf. Fodor, 1983). On this view it seems quite reasonable, we submit, that to the degree that there are modules, they are co-determined by the start state of the system (the genetic predisposition) and by the environment. (We take a module to be roughly a set of units which are powerfully interconnected among themselves and relatively weakly connected to units outside of the set; of course, this concept admits all gradations of modularity, just as our view of schemata allows all degrees of schematization of knowledge.) There is, on this view, no such thing as "hardwiring." Neither is there any such thing as "software." There are only connections. All connections are in some sense hardwired (in as much as they are physical entities) and all are software (in as much as they can be changed.) Thus, it may very well be that there is a part of the network prewired to deal with this or that processing task. If that task is not relevant in the organism's environment, that part of the network can be used for something else. If that part of the network is damaged, another part can come to play the role "normally" carried out by the damaged portion. These very properties have been noted characteristics of the brain since Hughlings-Jackson's work in the late 19th century (e.g., Jackson, 1869/1958); Jackson pointed them out as difficulties for the strict localizationist views then popular among students of the brain. Note too that our scheme allows for the organism to be especially sensitive to certain relationships (such as the relationship between nausea and eating, for which there might be stronger or more direct prewired

[7] Here again, our organism oversimplifies a bit. It appears that some parts of the nervous system—particularly lower level, reflexive, or regulatory mechanisms—seem to be prewired and subject only to control by trainable modulatory connections to higher level, more adaptive mechanisms, rather than to be directly modifiable themselves; for discussion see Teitelbaum (1967) and Gallistel (1980).

connections) while at the same time allowing quite arbitrary associations to be learned.

Finally, it should be mentioned that all of the learning schemes that have been proposed for networks of the sort we have studied are incremental (cf. Chapters 7, 8, 11, 18, 19, and 25), and therefore as an organism moves from its primarily genetically predetermined start state to its primarily environmentally determined final state, it will pass through a sequence of more or less intermediate states. There will be a kind of trajectory through the space of possible networks. This trajectory will constitute the developmental sequence for the organism. To the degree that different individuals share the same genetics (start state) and to the degree that their environments are similar, they will pass through similar trajectories. It should also be said that since, in PDP systems, what is learned is a product of both the current state of the organism and the current pattern of inputs, the start state will have an important effect on what is learned and the shape of the network following any given set of experiences. However, the greater the amount of experience, the more independent the system should be from its start state and the more dependent it should be on the structure of its environment.

Of course, not all connections may be plastic—certainly, many subcortical mechanisms are considerably less plastic than cortical ones. Also, plasticity may not continue throughout life (see Chapter 24). It would, of course, be a simple matter to suppose that certain connections are not modifiable. This is an issue about which our framework provides no answer. The major point is that there is no inconsistency between prewired, innate knowledge, and mutability and adaptability.

We cannot resist making one more point about the nativism/empiricism issue. This is that our PDP account of innate knowledge seems to provide a rather plausible account of how we can come to have innate "knowledge." To the extent that stored knowledge is assumed to be in the form of explicit, inaccessible rules of the kind often postulated by linguists as the basis for linguistic competence (see Chapter 18), it is hard to see how it could "get into the head" of the newborn. It seems to us implausible that the newborn possesses elaborate symbol systems and the systems for interpreting them required to put these explicit, inaccessible rules to use in guiding behavior. On our account, we do not need to attribute such complex machinery. If the innate knowledge is simply the prewired connections, it is encoded from the start in just the right way to be of use by the processing mechanisms.

Why Are People Smarter Than Rats?

Some have argued that since we claim that human cognition can be explained in terms of PDP networks and that the behavior of lower animals such as rats can also be described in terms of such networks we have no principled way of explaining why rats are not as smart as people. Given all of the above, the question does seem a bit puzzling. We are not claiming, in any way, that people and rats and all other organisms start out with the same prewired hardware. People have much more cortex than rats do or even than other primates do; in particular they have very much more prefrontal and parietal cortex—more brain structure not dedicated to input/output—and presumably, this extra cortex is strategically placed in the brain to subserve just those functions that differentiate people from rats or even apes. A case in point is the part of the brain known as the *angular gyrus*. This part of the brain does not exist even in chimpanzees. It sits at the intersection between the language areas of the temporal lobe and the visual areas of the parietal lobe, and damage to this area produces serious deficits in language and in the mapping of words onto meanings. While it is possible that structures like the angular gyrus possess some special internal wiring that makes them fundamentally different, somehow, in the kinds of cognitive operations they perform, their cytoarchitecture is not markedly different from that of other parts of the brain (see Chapters 20 and 21). Thus it seems to us quite plausible that some of the differences between rats and people lie in the potentiality for forming connections that can subserve the vital functions of language and thought that humans exhibit and other animals do not.

But there must be another aspect to the difference between rats and people as well. This is that the human environment includes other people and the cultural devices that they have developed to organize their thinking processes. Some thoughts on how we imagine these cultural devices are exploited in higher forms of intelligent behavior are presented in Chapter 14.

Conscious Knowledge and Explicit Reasoning

There may be cognitive scientists who accept some or all of what we have said up to this point, but still feel that something is missing, namely, an account of how we guide behavior using explicit, conscious knowledge, how we reason from what we know to new conclusions based on that knowledge, and how we find a path through a problem

space through a series of sequential steps. Can parallel distributed processing have anything to say about these explicit, introspectively accessible, temporally extended acts of thinking? Some have suggested that the answer is no—that PDP models may be fine as accounts for perception, motor control, and other *low-level* phenomena, but that they are simply unable to account for the higher level mental processing of the kind involved in reasoning, problem solving, and other higher level aspects of thought.

We agree that many of the most natural applications of PDP models are in the domains of perception and memory (see, for example, Chapters 15, 16, and 17). However, we are convinced that these models are equally applicable to higher level cognitive processes and offer new insights into these phenomena as well. We must be clear, though, about the fact that we cannot and do not expect PDP models to handle complex, extended, sequential reasoning processes as a single settling of a parallel network. We think that PDP models describe the microstructure of the thought process, and the mechanisms whereby these processes come, through practice, to flow more quickly and run together into each other.

Partly because of the temporally extended nature of sequential thought processes—the fact that they involve many settlings of a network instead of just one—they are naturally more difficult to deal with, and our efforts in these areas are, as yet, somewhat tentative. Nevertheless, we have begun to develop models of language processing (Chapter 19), language acquisition (Chapter 18), sequential thought processes and consciousness (Chapter 14), and problem solving and thinking in general (Chapters 6, 8, and 14). We view this work as preliminary, and we firmly believe that other frameworks provide additional, important levels of description that can augment our accounts, but we are encouraged by the progress we have made in these areas and believe that the new perspectives that arise from these efforts are sufficiently provocative to be added to the pool of possible explanations of these higher level cognitive processes. Obviously, the extension of our explorations more deeply into these domains is high on our ongoing agenda. We see no principled reasons why these explorations cannot succeed, and every indication is that they will lead us somewhat further toward an understanding of the microstructure of cognition.

MANY MODELS OR JUST ONE?

Before concluding this chapter, some comment should be made about the status of the various models we and other members of the PDP research group offer throughout the book. As the title of the book suggests, we understand our work as an *exploration*. We have been impressed with the potential of PDP models for changing our perspectives on the human information processing system. We have tried to maintain the kinds of general principles outlined in this chapter, but we have felt free to vary the details from application to application. Sometimes the variations are due to the fact that certain features of the models need to be elaborated to deal with certain phenomena but can be suppressed for other phenomena. Other times, we have simply made a different choice to explore a different part of the space of PDP models. We do not see ourselves capable as yet to produce the supermodel which would connect all of our areas of exploration together. Rather, we feel that the PDP framework which we are developing forms a kind of *metatheory* from which specific models can be generated for specific applications. The success of the particular models reflects indirectly on the metatheory, but we feel that the proper approach is to study detailed models of detailed applications while at the same time keeping one eye on the bigger picture. Thus, we don't really have a single model. Rather, we have a family of related models. In the best of all worlds each of our specific models may turn out to be a rough approximation to some unifying, underlying model as specialized to the problem area in question. More likely, however, each represents an exploration into a more or less uncharted region of the space of PDP models. Each application has lead to useful insights—both into the phenomena under study and into the behavior of the specific versions of the models used to account for them.

CONCLUSION

Some of the issues we have considered in this chapter are quite specific to our particular enterprise, but in the main, they are more general. They concern such questions as the scope of cognitive theory, the relation between levels, the question of nature vs. nurture, and the relevance of neural mechanisms to an analysis of cognition.

The present chapter has provided an overview of our views on a number of these central questions. In so doing, it has also provided an overview of the work that is described in the rest of the book, along with some of the reasons for doing it. Indeed, in many ways the rest of the book *is* our response to the issues we have touched on here. The chapters in Part II seek ways to overcome the computational limitations of earlier network models, and the chapters in Part III provide some of the formal tools that are crucial in pursuing these kinds of goals. The

chapters in Part IV address themselves to cognitive constructs and attempt to redefine the cognitive structures of earlier theories in terms of emergent properties of PDP networks. The chapters in Part V consider the neural mechanisms themselves and their relation to the algorithmic level that is the focus of most of the work described in Parts II and IV.

ACKNOWLEDGMENTS

We would like to thank the many people who have raised the questions and the objections that we have attempted to discuss here. These people include John Anderson, Francis Crick, Steve Draper, Jerry Fodor, Jim Greeno, Allen Newell, Zenon Pylyshyn, Chris Riesbeck, Kurt van Lehn, and many others in San Diego, Pittsburgh, and elsewhere. In addition, we would like to thank Allan Collins, Keith Holyoak, and Walter Schneider for organizing seminars and symposia around these issues, and we would like to thank our many colleagues who have helped us formulate our answers to some of these questions, particularly Geoff Hinton, Paul Smolensky, and the other members of the PDP Research Group.

Preparation of this chapter was supported by ONR contracts N00014-82-C-0374, NR 667-483 and N00014-79-C-0323, NR 667-437, by a grant from the System Development Foundation, and by a Research Scientist Career Development Award MH00385 to the second author from the National Institute of Mental Health.

PART II

BASIC MECHANISMS

The chapters of Part II represent explorations into specific architectures and learning mechanisms for PDP models. These explorations proceed through mathematical analysis coupled with results from simulations. The major theme which runs through all of these explorations is a focus on the learning problem. How can PDP networks evolve to perform the kinds of tasks we require of them? Since one of the primary features of PDP models in general is their ability to self-modify, these studies form an important base for the application of these models to specific psychological and biological phenomena.

In Chapter 5, Rumelhart and Zipser begin with a summary of the history of early work on learning in parallel distributed processing systems. They then study an unsupervised learning procedure called *competitive learning*. This is a procedure whereby feature detectors capable of discriminating among the members of a set of stimulus input patterns evolve without a specific teacher guiding the learning. The basic idea is to let pools of potential feature detector units *compete* among themselves to respond to each stimulus pattern. The winner within each pool—the one whose connections make it respond most strongly to the pattern—then adjusts its connections slightly toward the pattern that it won. Several earlier investigators have considered variants of the competitive learning idea (e.g., Grossberg, 1976; von der Malsberg, 1973). Rumelhart and Zipser show that when a competitive network is trained through repeated presentations of members of a set of patterns, each unit in a pool comes to respond when patterns with a particular