Barsalou wrote sufficient rude remarks on his copy of the manuscript that we responded by giving increased attention to non-traditional symbolic models in the publication version. We express our deep thanks to these four individuals for their contributions. In addition, much of the material in this book was presented informally to the Cognitive Science Group at Georgia State University, which included James L. Pate, Richard Thompson Putney, Paul Allopenna, David Washburn, Robert Mankoff, and Quinton Gooden. Also, a study group at the University of Cincinnati, consisting of Christopher Gauker, Kelly Hite, Melinda Hogan, William E. Morris, and Robert Richardson, read the manuscript during the fall of 1989. We appreciate the comments made by the members of these groups, which helped to improve the manuscript. We have benefited more generally from discussion of issues explored in this book with a number of individuals, including Richard Billington, Dorrit Billman, and Suge-Yuki Kuroda. Henri Madigan and Britten Poulson provided valuable assistance in collecting bibliographical materials that were used in preparing the text. We also wish to acknowledge a Georgia State University Research Grant which provided computer resources for the simulations presented in this book. Finally, during fall 1989, one of us (Abrahamsen) was a visiting scholar at Skidmore College, and is grateful for the college's support and stimulating discussions with a number of its faculty.

1

Networks versus Symbol Systems: Two Approaches to Modeling Cognition

A Revolution in the Making?

The rise of cognitivism in psychology, which, by the 1970s, had successfully established itself as a successor to behaviorism, has been characterized as a Kuhnian revolution (Baars, 1986). Using Kuhn's (1962/1970) term, the emerging cognitivism offered its own *paradigm*, that is, its way of construing psychological phenomena and its research strategies, both of which clearly distinguished it from behaviorism (for overviews, see Neisser, 1967; Lindsay and Norman, 1972). This change was part of a broader cognitive revolution that not only transformed a number of disciplines such as cognitive and developmental psychology, artificial intelligence, linguistics, and parts of anthropology, philosophy, and neuroscience; it also led to an active cross-disciplinary research cluster known as *cognitive science*. As the cognitive paradigm developed, the idea that cognition involved the manipulation of symbols became increasingly central. These symbols could refer to external phenomena and so have a semantics. They were enduring entities which could be stored in and retrieved from memory and transformed according to rules. The rules that specified how symbols could be composed (syntax) and how they could be transformed were taken to govern cognitive performance. Given the centrality of symbols in this approach, we shall refer to it as the *symbolic paradigm*.

In the 1980s, however, an alternative framework for understanding cognition has emerged in cognitive science, and a case can be made that it is a new Kuhnian (Schneider, 1987). (We shall be using the term *cognition* very broadly to cover a range of mental processing, including not just activities involving reasoning and memory, but also language, perception, and motor control.) This new class of models are variously

known as *connectionist, parallel distributed processing* (*PDP*), or *neural network* models. The "bible" of the connectionist enterprise, Rumelhart and McClelland's two volumes entitled *Parallel Distributed Processing* (1986), sold out its first printing prior to publication and sold 30,000 copies in its first year. Clearly connectionism has become the focus of a great deal of attention.

Connectionism can be distinguished from the traditional symbolic paradigm by the fact that it does not construe cognition as involving symbol manipulation. It offers a radically different conception of the basic processing system of the mind–brain. This conception is inspired by our knowledge of the nervous system. The basic idea is that there is a network of elementary *units* or nodes, each of which has some degree of *activation*. These units are *connected* to each other so that active units excite or inhibit other units. The network is a *dynamical system* which, once supplied with initial input, spreads excitations and inhibitions among its units. In some types of network, this process does not stop until a *stable state* is achieved. To understand a connectionist system as performing a cognitive task, it is necessary to supply an interpretation. This is typically done by viewing the initial activations supplied to the system as specifying a problem, and the stable configuration produced at the end of processing as the system's solution to the problem.

Both connectionist and symbolic systems can be viewed as computational systems. But they advance quite different conceptions of what computation involves. In the symbolic approach, computation involves the transformation of symbols according to rules. This is the way we teach computation in arithmetic: we teach rules for performing operations specified by particular symbols (e.g., $+, \div$) on other symbols which refer to numbers. When we treat a traditional computer as a symbolic device, we view it as performing symbolic manipulations specified by rules which typically are written in a special data-structure called the *program*. The connectionist view of computation is quite different. It focuses on causal processes by which units excite and inhibit each other and does not provide either for stored symbols or rules that govern their manipulations.

While connectionism has achieved widespread attention only in the 1980s, it is not a newcomer. Network models, which were predecessors of contemporary connectionist models, were developed and widely discussed during the early years of the cognitive revolution in the 1960s. The establishment of the symbolic paradigm as virtually synonymous with cognitive science only occurred at the end of the 1960s, when the symbolic approach promised great success in accounting for cognition

and the predecessors of connectionism seemed inadequate to the task. A brief recounting of this early history of network models will provide an introduction to the connectionist approach and to the difficulties which it is thought to encounter. The issues that figured in this early controversy still loom large in contemporary discussions of connectionism and will be discussed extensively in subsequent chapters. (For additional detail see Cowan and Sharp (1988) from which we have largely drawn our historical account, and Anderson and Rosenfeld (1988) which gathers together many of the seminal papers and offers illuminating commentary.)

Forerunners of Connectionism: Pandemonium and Perceptrons

The initial impetus for developing network models of cognitive performance was the recognition that the brain is a network. Obviously, given the complexity of the brain and the limited knowledge available then or now of actual brain functioning, the goal was not to model brain activity in complete detail. Rather, the goal was to model cognitive phenomena in systems that exhibited some of the same basic properties as the network of neurons in the brain. The foundation was laid by Warren McCulloch and Walter Pitts in a paper published in 1943. They proposed a simple model of neuron-like computational units and then demonstrated how these units could perform logical computations. Their "formal neurons" were binary units (i.e., they could either be on or off). Each unit would receive excitatory and inhibitory inputs from certain other units. If a unit received just one inhibitory input, it was forced into the *off* position. If there were no inhibitory inputs, the unit would turn *on* if the sum of the excitatory inputs exceeded its threshold. McCulloch and Pitts showed how configurations of these units could perform the logical operations of AND, OR, and NOT. McCulloch and Pitts further demonstrated that any process that could be performed with a finite number of these logical operations could be performed by a network of such units, and that, if provided with indefinitely large memory capacity, such networks would have the same power as a Universal Turing machine.

The idea captured by Pitts–McCulloch "neurons" was elaborated in a variety of research endeavors in succeeding decades. John von Neumann (1956) showed how such networks could be made more reliable by significantly increasing the number of inputs to each particular unit and determining each unit's activation from the statistical pattern of activations over its input units (e.g., by having a unit turn on if more

than half of its inputs were active). In von Neumann's networks each individual unit could be unreliable without sacrificing the reliability of the overall system. Building such redundancy into a network seems to require vastly increasing the number or units, but Winograd and Cowan (1963) developed a procedure whereby a given unit would contribute to the activation decision of several units as well as being affected by several units. This constitutes an early version of what is now referred to as "distributed representation" (see chaper 2).

In addition to formal characterizations of the behavior of these networks, research was also directed to the potential applications of these networks for performing cognitive functions. McCulloch and Pitts' first paper was devoted to determining the logical power of networks, but a subsequent paper (Pitts and McCulloch, 1947) explored how a network could perform pattern-recognition tasks. They were intrigued by the ability of animals and humans to recognize different versions of the same entity even though they might appear quite different. They construed this task as requiring multiple transformations of the input image until a canonical representation was produced, and they proposed two networks that could perform some of the required transformations. Each network received as input a pattern of activation on some of its units. The first network was designed to identify invariant properties of a pattern (properties possessed by a pattern no matter how it was presented), while the second transformed a variant into a standard representation. Because their inspiration came from knowledge of the brain, they presented evidence that the first type of network captured properties of the auditory and visual cortex, while the second captured properties of the superior colliculus in controlling eye movements.

Frank Rosenblatt was one of the major researchers to pursue the problem of pattern recognition in networks. Like Pitts and McCulloch, he worked principally with binary units in layered networks, that is, networks in which one set of units receives inputs from outside and sends excitations and inhibitions to another set of units, which may then send inputs to yet a third group. He also explored networks in which later layers of units might send excitations or inhibitions back to earlier layers. Rosenblatt referred to such systems as *perceptrons* (see figure 1.1). He supplemented McCulloch and Pitts' networks by making the strengths (commonly referred to as the *weights*) of the connections between units continuous rather than binary, and by introducing procedures for changing these weights, enabling the networks to be trained to change their responses. For networks with two layers and connections running only from units in the first layer to

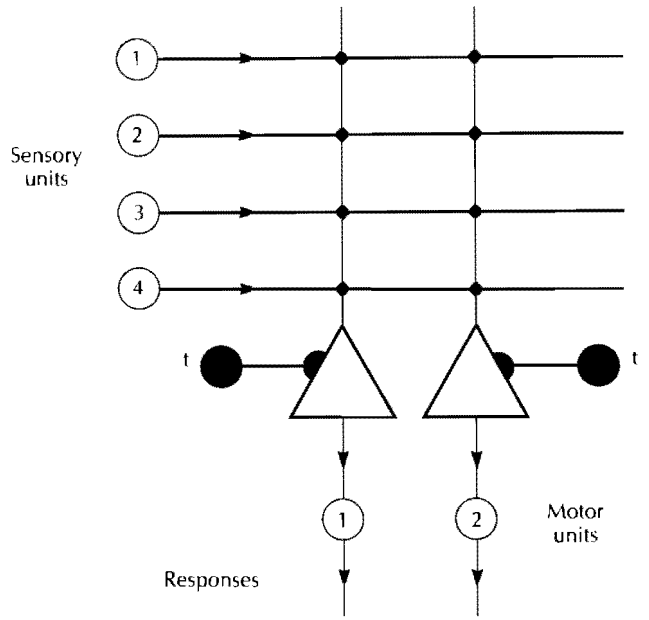

Figure 1.1 An elementary perceptron, as investigated by Rosenblatt (1958). Inputs are supplied on the four sensory units on the left and outputs are produced on the two motor units at the bottom. The horizontal and vertical lines represent connections; the diamonds at their intersections represent synapses whose weights can be modified if incorrect outputs are generated. From J. D. Cowan and D. H. Sharp (1988) Neural nets and artificial intelligence, *Daedalus*, 117, p. 90, Reprinted with permission.

those in the second, Rosenblatt's procedure was to have the network generate, using existing weights, an output for a given input pattern. The weights on connections feeding into any unit that gave what was judged to be an *incorrect* response were changed (those feeding into units giving the correct response were left unaltered). If the unit was off when it should have been on, an increase was made to all weights on connections that had carried any activation to it (i.e., came from units that had been active). Conversely, if the unit was on when it should have been off, these weights were reduced. Rosenblatt demonstrated the important Perceptron Convergence Theorem with respect to this training procedure. The theorem holds that if a set of weights existed that would produce the correct responses to a set of patterns, then through a finite number of repetitions of this training procedure the network would in fact learn to respond correctly (Rosenblatt, 1961; see also Block, 1962).

Rosenblatt emphasized how the perceptron differed from a symbolic processing system. Like von Neumann, he focused on statistical patterns over multiple units (e.g., the proportion of units activated by an input), and viewed noise and variation as essential. He contended that by building a system on statistical rather than logical (Boolean) principles, he had achieved a new type of information processing system:

> It seems clear that the class C' perceptron introduces a new kind of information processing automaton: For the first time, we have a machine which is capable of having original ideas. As an analogue of the biological brain, the perceptron, more precisely, the theory of statistical separability, seems to come closer to meeting the requirements of a functional explanation of the nervous system than any system previously proposed.... As a concept, it would seem that the perceptron has established, beyond doubt, the feasibility and principle of non-human systems which may embody human cognitive functions at a level far beyond that which can be achieved through present day automatons. The future of information processing devices which operate on statistical, rather than logical principles seems to be clearly indicated. (Rosenblatt, 1958, p. 449, quoted in Rumelhart and Zipser, 1986, in *PDP:5*, pp. 56–7)

Oliver Selfridge (1959) was another of the early investigators of the pattern recognition capabilities of network models. Unlike Rosenblatt, he assigned a particular interpretation to each of the units in his network. One of the pattern-recognition tasks he explored was recognition of letters, a task that is made difficult by the fact that different people write their letters differently. He called his model *pandemonium*, capturing the fact that his model was composed of *cognitive demons* that performed computations in parallel without attention to one another, and each "shouted out" its judgement of what letter had been presented (see figure 1.2). These cognitive demons each specialized in gathering evidence for one particular letter; the greater the evidence the louder they shouted. The decision demon then made the identification of the letter on the basis of which unit shouted the loudest. The evidence gathered by each cognitive demon was supplied by a lower layer of feature demons. Each feature demon responded if its feature (e.g., a horizontal bar) was present in the image. The feature demon was connected to just those cognitive demons whose letters contained its feature. Thus, a cognitive demon would respond most loudly if all of its features were present in the image, and less loudly if some but not all of its features were present. One of the virtues of this type of network is that it would still make a correct or plausible judgement about a letter even if some of its features were missing or atypical (see Selfridge, 1959; Selfridge and Neisser, 1960).

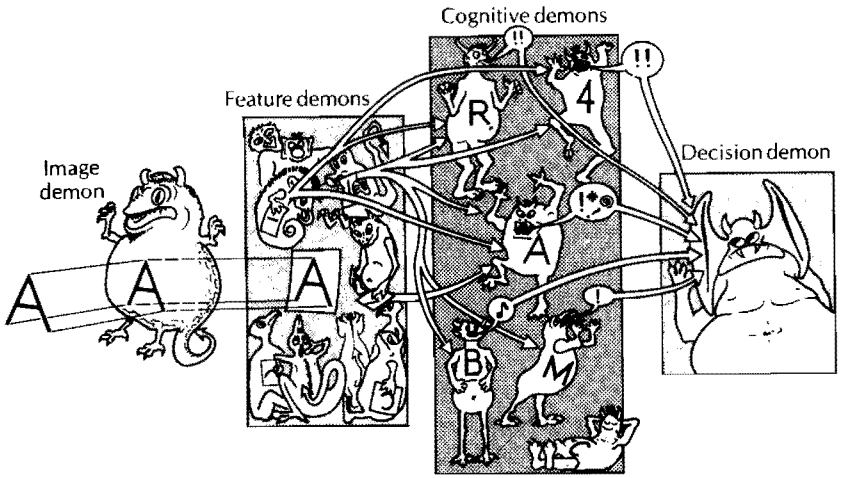

Figure 1.2 Selfridge's pandemonium model. The *demons* at each level beyond the image demon (which merely records the incoming image) extract information from the demons at the preceding level. Thus, a given feature demon responds positively when it detects evidence of its feature in the image, and a cognitive demon responds to the degree that the appropriate feature demons for its letter are active. Finally, the decision demon selects the letter whose cognitive demon is most active. From P. Lindsay and D. A. Norman (1972) *Human Information Processing*, San Francisco: Freeman, p. 116. Reprinted with permission.

Early researchers recognized that, in addition to modeling pattern recognition, networks might be useful as models of how memories were established. In particular, researchers were attracted to the problem of how networks might store associations between different patterns. An extremely influential proposal was developed by Donald Hebb (1949), who suggested that when two neurons in the brain were jointly active, the strength of the connection might be increased. This idea was further developed by Wilfrid Taylor (1956), who explored networks of analog units that took activations within a continuous range (e.g., -1 to $+1$). In the network he proposed, a single set of motor units was connected to two different sets of sensory units (which we shall call the base units and the learning units). The network was set up such that each pattern on the base units was associated with a pattern on the motor units. A different set of patterns was defined for the learning units. No associations to the motor units were specified, but each learning unit pattern was assigned an association with one base unit pattern. When the network was run, the associated sensory patterns were activated at the same time. The eventual outcome was that the learning

units acquired the ability to generate the same motor patterns as the base units with which they were associated.

Another researcher who pursued this type of associative memory network was David Marr (1969), who proposed that the cerebellum is such a network which can be trained by the cerebrum to control voluntary movements. The cerebellum consists of five different kinds of cell or unit, with the modifiable connections lying between the granule cells and Purkinje cells. The other cell types serve to set the firing thresholds on these two cell types. The development of connections between the granule cells and Purkinje cells, he proposed, underlay the learning of sequences of voluntary movements in activities like playing the piano. Marr subsequently proposed similar models for the operation of the hippocampus (Marr, 1971) and the neocortex (Marr, 1970).

The early history of network models we have summarized in this section indicates that there was an active research program devoted to exploring the *cognitive* significance of such networks. It is important to emphasize that while some of this research was explicitly directed at modeling the brain, for Rosenblatt and some other researchers the goal was to understand cognitive performance more generally. The relative prominence of research devoted to network models diminished in the late 1960s and early 1970s, as the alternative approach of symbolic modeling became dominant. (Semantic networks, hybrid models that place symbols in network structures, also arose and thrived in the 1970s; as discussed in chapter 4.) In the next two sections we shall examine what made the symbolic approach so attractive to cognitive researchers, and how network research (in the original tradition pioneered by Rosenblatt) declined until rejuvenated in the 1980s. Finally, we shall sketch the relation between the network and symbolic models of the 1980s.

The Allure of Symbol Manipulation

The symbol manipulation view of cognition has several roots. One of these lies in philosophy, in the study of logic. A logical system consists of procedures for manipulating symbols. In propositional logic the symbols are taken to represent propositions or sentences and connectives such as AND and OR. Generally, there is a clear goal in such manipulation. For example, in *deductive logic* we seek a set of rules that will enable us to generate only true propositions as long as we start with true propositions. A system of such rules is spoken of as *truth preserving*. The simple inference rule *modus ponens* is an example of a

truth-preserving rule. From one proposition of the form "If p, then q" and another of the form "p," we can infer a proposition of the form "q" (where p and q are placeholders for specific propositions).

We have actually adopted two perspectives in the previous paragraph, and it is the relation between them that makes logic, and systems designed to implement logic, so powerful. From one perspective, we treat the symbols for propositions as representational devices. For example, we conceive of a proposition as depicting a state of affairs that might or might not hold in the world. From this perspective, we speak of a proposition as either *true* (if the proposition corresponds to the way the world is) or *false* (if it does not correspond). This perspective is generally known in logic as a *model theoretic* perspective. We think of a model as a set of entities and identify those propositions as *true* whose ascriptions correspond to the properties that the entities in the model actually possess. Within this framework we can evaluate whether a pattern of inference is such that for any model in which the premises are true, the conclusion will also be true. The second perspective, known as the *proof theoretic* perspective, focuses not on the relations between the propositions and the objects they represent, but simply on the relations among the propositions themselves, construed as formal entities. When we specify inference rules in a logical system, we focus only on the syntax of the symbols and disregard what they refer to. What gives logic its power is, in part, the possibility of integrating these two perspectives, of designing proof procedures that are complete, that is, that will enable us to derive any proposition that will be true in all models in which the premises are true.

The relation between proof theory and model theory gives rise to a very powerful idea. If intelligence depended only upon logical reasoning, for which the goal was truth preservation, then it would be possible to set up formal proof procedures, which will achieve intelligent performance. However, intelligence does not depend solely on being able to make truth-preserving inferences. Sometimes we need to make judgements as to what is likely to be true. This is the domain of inductive logic. The goal of inductive logic is to establish formal rules, analogous to the proof theoretic procedures of deductive logic, that lead from propositions that are true to those that are likely to be true. If such rules can be identified, then we may still be able to set up formal inference procedures that produce intelligent performance.

The crucial assumption in both deductive and inductive logic is that in order to process a symbol, we only need to consider its formal properties. We can disregard its representational function, that is, whether it is true or not, and if true, what state of affairs it describes. Thus, with a

formal system, it is often possible to reinterpret the symbols that are used (i.e., assign them a new representational role) without affecting how the symbol processing system itself operates.

The idea that intelligent cognitive processes are essentially processes of logical reasoning has a long history, captured in the long-held view that the rules of logic constitute rules of thought. It is found in authors such as Hobbes, who treated reasoning as itself comparable to mathematical computation and suggested that thinking was simply a process of formal computation:

> When a man *reasoneth*, he does nothing else but conceive a sum total, from *addition* of parcels; or conceive a remainder, from *subtraction* of one sum from another; which, if it be done by words, is conceiving of the consequence of the names of all the parts, to the name of the whole; or from the names of the whole and one part, to the name of the other part.... These operations are not incident to numbers only, but to all manner of things that can be added together, and taken from one out of another. For as arithmeticians teach to add and subtract in *numbers*; so the geometricians teach the same in *lines*, *figures*, solid and superficial, *angles*, *proportions*, *times*, degrees of *swiftness*, *force*, *power*, and the like; the logicians teach the same in *consequences of words*; adding together two *names* to make an *affirmation*, and two *affirmations* to make a *syllogism*; and *many syllogisms* to make a *demonstration*; and from the *sum* or *conclusion* of a *syllogism*, they subtract one *proposition* to find the other. (Hobbes [1651], 1962, p. 41)

The idea of thinking as logical manipulation of symbols was further developed in the works of rationalists such as Descartes and Leibniz and empiricists such Locke and Hume, all of whom conceived of the symbols as ideas, and formulated rules for properly putting together or taking apart ideas.

With the development of automata theory and physical computers in the mid-twentieth century, there was a burgeoning of more subtle and varied views of symbols and symbol manipulation. From one perspective (well characterized in Haugeland, 1981), the digital computer is simply a device for implementing formal logical systems. Symbols are stored in memory registers (these symbols may simply be sequences of 1's and 0's, implemented by *on* and *off* settings of switches). The basic operations of the computer allow recalling the symbols from memory and executing changes in the symbols according to rules. In the earliest computers, the rules for transforming symbols had to be specially wired into the machine, but one of the major breakthroughs in early computer science was the development of the stored program. The stored program is simply a sequence of symbols that directly determines what operations the computer will perform on other symbols. The relation between the stored program and those other symbols is much like the relation between the formally written rule *modus ponens* and the symbol strings to which it can be applied. Like the formal rules of logic, the rules in the computer program do not consider the semantics of the symbols being manipulated, but only their form. This perspective has been given a variety of renderings by such symbolic theorists as Dennett (1978), Fodor (1980), and Pylyshyn (1984).

An alternative way to construe the semantics of computational systems was offered by Newell and Simon (1981). For them, a computer is a *physical symbol system* consisting of symbols (physical patterns), expressions (symbol structures obtained by placing symbol tokens in a physical relation such as adjacency), and processes that operate on expressions. They point out that there is a semantics (designation and interpretation) within the system itself; specifically, expressions in stored list-processing programs designate locations in computer memory, and these expressions can be interpreted by accessing those locations. They regard this internal semantics as a major advance over formal symbol systems such as those of logic, and argue that intelligence cannot be attained without it:

> *The Physical Symbol System Hypothesis.* A physical symbol system has the necessary and sufficient means for general intelligent action.
>
> By "necessary" we mean that any system that exhibits general intelligence will prove upon analysis to be a physical symbol system. By "sufficient" we mean that any physical symbol system of sufficient size can be organized further to exhibit general intelligence. (Newell and Simon, 1981, p. 41)

Hence, with respect to the question of the autonomy of syntax from semantics, some cognitive scientists have emphasized the continuity between computers and formal logical systems, whereas others (such as Newell and Simon) have viewed computers as enabling advances beyond formal systems. A similar difference in perspective arises with respect to what work the computer is regarded as carrying out. From a continuity perspective, computers are powerful devices for implementing logical operations; one can write programs that will serve the same function as inference rules in a logical system. From an alternative perspective (Simon, 1967), it took work in artificial intelligence to show us that *heuristics* (procedures that *might* obtain the desired result, often by means of an intelligent shortcut such as pruning

unpromising search paths) are often more useful than *algorithms* (procedures that are guaranteed to succeed in a finite number of steps but may be inefficient in a large system).

Hence, work in artificial intelligence is rooted in formal logic, but has achieved distinctive perspectives by pursuing the idea that computers are devices for symbol manipulation more generally. AI programs have replaced formal logic as the closest external approximation to human cognition; programs exist, for example, not only for proving logical theorems or performing logical inference, but also for playing chess at a master's level and diagnosing diseases. The (partial) success of these programs has suggested to many researchers that human cognitive performance also consists in symbol manipulation; indeed, this analogy provided, until recently, a locus of unity among cognitive scientists.

Yet another root of the symbolic approach is found in Noam Chomsky's program in linguistics. In his review of B. F. Skinner's *Verbal Behavior*, Chomsky (1959) argued that a behavioristic account was inadequate to account for the ability of humans to learn and use languages. Part of his argument focused on the "creativity" of language; Chomsky contended that any natural language has an infinite number of syntactically well-formed sentences, and that its speakers can understand and produce sentences that they had not previously encountered (Chomsky, 1957, 1968). This ability did not seem explicable in terms of learned associations between environmental stimuli and linguistic responses, even if these were augmented by such processes as generalization and analogy. In Chomsky's view, Skinner had not succeeded in adapting the constructs of behaviorism to the precise requirements of a linguistic account, and a quite different approach was needed.

In particular, Chomsky developed the notion of *generative grammar* as an approach to linguistic theory: to write a grammar was to specify an automaton that could generate infinite sets of sentences (this was easily assured by including at least one recursive rule). To evaluate such a grammar, the linguist must determine whether it generates all of the well-formed sentences of the target language, and only those sentences. Chomsky described and evaluated several different classes of generative grammars with respect to natural languages. Of particular importance, he argued that finite state grammars (those most consistent with a behaviorist account), were too weak even when they included recursive rules. They could generate an infinite set of sentences, but not the *correct* set. Specifically, they were unable to handle dependencies across indefinitely long strings (e.g., the dependency between *if* and *then* in sentences of the form "if A, then B" where A is indefinitely long).

To handle such dependencies, at least a phrase structure grammar (and preferably a transformational grammar) was required. These grammars produce constituent structures by applying a succession of rewrite rules (rules which expand one symbol into a string of subordinate symbols); indefinitely long constituents can be embedded within a phrase structure tree without affecting the surrounding dependencies. Transformational rules (rules that modify one phrase structure tree to obtain a related, or transformed, tree) provide additional power, but the most important and enduring part of Chomsky's argument is the rejection of finite state grammars.

Chomsky viewed generative grammar as a model of linguistic *competence*; that is, a model of the knowledge of their language that speakers actually possess in their minds. Although he pioneered the use of (abstract) automata for specifying grammars, he did not intend to model linguistic *performance* (the expression of competence in specific, real-time acts such as the production and comprehension of utterances), nor did he implement his grammars on physical computers. Hence, his version of cognitivism is somewhat more abstract than that of information-processing psychology. Nevertheless, many psychologists were influenced by Chomsky as they moved from behaviorism to information processing, because his grammars suggested ways to model human knowledge using linguistic-style rules (that is, formally specified operations on strings of symbols).

Although Chomsky focused on linguistic competence, he did make some general, controversial claims about linguistic performance. One of these claims, that a process of hypothesis testing is involved in language acquisition, bore implications that were fruitfully developed by Jerry Fodor (1975). Before we can test a hypothesis, such as that the word *dog* refers to dogs, we must be able to state it. Fodor reasoned that this requires a language-like medium, which he called the *language of thought*. Further, since there is no way for a child to learn this language, it must be innate. Thus, Fodor contended that procedures for formal symbol manipulation must be part of our native cognitive apparatus. Fodor's argument represents a minority position within psychology, but virtually all researchers in the majority tradition of information processing assume some weaker version of a symbolic approach to cogniton.

We have briefly reviewed two strands of the symbolic approach: a strand leading from formal logic to artificial intelligence, in which computers came to be viewed as symbol manipulation devices, and a strand leading from linguistics to psychology, in which human cognition came to be viewed likewise as consisting in symbol manipulation.

In cognitive science, these two strands are often brought together in a cooperative enterprise: the design of computer programs to serve as models or simulations of human cognition. This raises a number of interesting issues that we can only briefly mention here (a number of penetrating discussions are available, e.g., Haugeland, 1985). Does a successful computer simulation closely approximate mental symbol processing at some appropriate level of abstraction, so that both the human and the computer are properly construed as symbol processors? Or should true symbol manipulation be attributed to only one of the two types of system; and if so, to the human or the computer? On one view, the human is the true symbol manipulator (because, for example, the human's symbols have causal relations to external referents), and the computer is merely a large calculator or scratchpad that can facilitate the process of deriving predictions from models of human performance (similar to the meteorologist's use of computers to calculate equations that describe the fluid dynamics of the atmosphere, for example). A contrasting view holds that the computer is the true symbol manipulator, and that human cognition is carried out quite differently (in less brittle fashion, as might be modeled in a network, for example). These issues, which have been troublesome for some time, have gained increased salience with the re-emergence of network models in the 1980s. We turn now to a brief history of networks as an alternative to the symbolic tradition.

The Disappearance and Re-emergence of Network Models

By the 1960s substantial progress had been made with both network and symbolic approaches to machine intelligence. But this parity was soon lost. Seymour Papert has provided a whimsical account:

> Once upon a time two daughter sciences were born to the new science of cybernetics. One sister was natural, with features inherited from the study of the brain, from the way nature does things. The other was artificial, related from the beginning to the use of computers. Each of the sister sciences tried to build models of intelligence, but from very different materials. The natural sister built models (called neural networks) out of mathematically purified neurones. The artificial sister built her models out of computer programs.
>
> In their first bloom of youth the two were equally successful and equally pursued by suitors from other fields of knowledge. They got on very well together. Their relationship changed in the early sixties when a new monarch appeared, one with the largest coffers ever seen in the kingdom of the sciences: Lord DARPA, the Defense Department's Advanced Research Projects Agency. The artificial sister grew jealous and was determined to keep for herself the access to Lord DARPA's research funds. The natural sister would have to be slain.
>
> The bloody work was attempted by two staunch followers of the artificial sister, Marvin Minsky and Seymour Papert, cast in the role of the huntsman sent to slay Snow White and bring back her heart as proof of the deed. Their weapon was not the dagger but the mightier pen, from which came a book – *Perceptrons* ... (1988, p. 3)

Clearly the publication of *Perceptrons* in 1969 represented a watershed. Research on network models, such as perceptrons and pandemonium, no longer progressed apace with work on symbolic models. Some researchers did continue to pursue and develop network models and in fact established some important principles governing network systems (see J. A. Anderson, 1972; Kohonen, 1972; Grossberg, 1976). Their work, however, attracted only limited attention and funding. What is less clear is whether Minsky and Papert's book precipitated the demise, or whether it was only a symptom.

Minsky and Papert's objective in *Perceptrons* was to study both the potential and limitations of network models. They used the tool of mathematics to analyze what kinds of computation could or could not be performed with a two-layer perceptron. The centerpiece of their criticism was their demonstration that there are functions, such as determining whether a figure is connected or whether the number of active units is odd or even, which cannot be evaluated by such a network. An example is the logical operation of *exclusive or* (XOR). The statement A XOR B is defined as true if A is true and B is not, or B is true and A is not. In order for a network to compute XOR, it is necessary to include an additional layer of units (now referred to as *hidden* units) between the input units and output units (see chapter 3). While Minsky and Papert recognized that XOR could be computed by such a multi-layered network, they raised an additional problem: there were no training procedures for multi-layered networks that could be shown to converge on a solution. As we shall discuss in chapter 3, an adaption of Rosenblatt's training procedure for two-layer networks has now been developed for multi-layered networks. But Minsky and Papert raised further doubts about the usefulness of network models. Even if the problem were overcome, would it be possible to increase the size of networks to handle larger problems? In more technical terms, this is a question as to whether networks will *scale* well. Minsky and Papert offered the intuitive judgement that research on multi-layered networks would be "sterile."

The inability of networks to solve particular problems was, for many investigators, only symptomatic of a more general problem. For them,

the fundamental problem was that the only kind of cognitive processes of which networks seemed capable were those involving associations. Within limits, a network could be trained to produce a desired output from a given input, but that merely meant that it had developed procedures for associating that input with the desired output. Associationism was exactly what many of the founders of modern cognitivism were crusading against. Chomsky contended, for example, that finite automata or simple associationistic mechanisms were inadequate to generate all the well-formed sentences of the language. One needed a more powerful automaton capable of performing recursive operations. The identification of network models with associationism thus undercut their credibility and supported the pursuit of symbolic programs as the major research strategy in cognitive science. As we shall see in chapter 7, many advocates of the symbolic tradition continue to fault modern connectionism on precisely this ground.

In the early 1980s the type of network research pioneered by Rosenblatt began once again to attract attention. Papers that employed networks to model various cognitive performances began to appear in cognitive journals. Geoffrey Hinton and James A. Anderson's (1981) *Parallel Models of Associative Memory* offered an accessible presentation of the re-emerging network research. At the 1984 meeting of the Cognitive Science Society, two symposia presented the network approach and debated its role in cognitive science. One, entitled "Connectionism versus Rules: The Nature of Theory on Cognitive Science," featured David Rumelhart and Geoffrey Hinton advocating network modeling (connectionism) and Zenon Pylyshyn and Kurt VanLehn arguing that networks were inadequate devices for achieving cognitive performance. Debate at that session and others during the conference occasionally became acrimonious as the connectionists began to press their alternative and challenged the supremacy of the symbolic approach.

Connectionist research has increased dramatically in the 1980s. While opposition continues, a growing number of cognitive scientists have either "converted" to connectionism or have added connectionist modeling techniques to their repertoire as tools they will employ for at least some purposes. An intriguing question is why connectionism should have re-emerged so strongly in the 1980s. While we do not offer a comprehensive answer to this question, there are a number of factors that seem relevant.

First, powerful new approaches to network modeling were developed, including new architectures, new techniques for training multilayered networks, and advances in the mathematical description of the behavior of nonlinear systems. Many of these innovations can be directly applied to the task of modeling cognitive processes. Second, the credibility of some of the researchers attracted to network research has played a role. For example, in chapters 2 and 3 we discuss an important mathematical insight into network behavior that was proposed by John Hopfield, a distinguished physicist. Anderson and Rosenfeld comment:

> John Hopfield is a distinguished physicist. When he talks, people listen. Theory in his hands becomes respectable. Neural networks became instantly legitimate, whereas before, most developments in networks had been in the province of somewhat suspect psychologists and neurobiologists, or by those removed from the hot centers of scientific activity. (1988, p. 457)

Third, cognitive science had remained, either intentionally or unintentionally, rather isolated from neuroscience through the 1970s. In large part this was because there was no clear framework to suggest how work in the neurosciences might bear on cognitive models. But by the 1980s cognitive scientists' interest in the neurosciences had increased, and network models were attractive because they provided a neural-like architecture for cognitive modeling. Fourth, the interest in neuroscience was one reflection of a more general interest in finding a fundamental explanation for the character of cognition. Rule systems, as they became more adequate, also became more complex, diverse, and *ad hoc*. The desire for parsimony, which earlier had characterized behaviorism, re-emerged. Fifth, a number of investigators began to confront the limitations of symbolic models. While initially the task of writing rule systems capable of accounting for human behavior seemed tractable, intense pursuit of the endeavor raised doubts. Rule systems were hampered by their "brittleness," inflexibility, difficulty, learning from experience, inadequate generalization, domain specificity, and inefficiencies due to serial search through large systems. Human cognition, which the rule systems were supposed to be modeling, seemed to be relatively free of such limitations.

These and other factors operated together to make networks models attractive to some cognitive scientists, beginning with a few pioneers in the early 1980s and reaching substantial proportions by the end of the decade. During the same period, however, other cognitive scientists were also concerned about the limitations of traditional symbolic models; no one who models performance wants a brittle system, for example. These investigators focused only on the fifth factor above, rather than all five factors, and adopted the conservative strategy of modifying the existing approach rather than initiating a new, relatively

untried approach. Hence, if the symbolic approach is a target of criticism on the part of network modelers, it is a moving target and therefore harder to hit.

Most of the modifications incorporated in the most recent symbolic models have narrowed the gap between symbolic and network models. (It could even be argued that the real revolution is the development of a variety of ways to overcome the limitations of earlier models, including but not limited to connectionist modeling.) First, a large number of rules at a fine grain of analysis (microrules) can capture more of the subtleties of behavior than a smaller number of rules at a larger grain of analysis. Second, rule selection, and perhaps rule application as well, can be made to operate in parallel. Third, the ability to satisfy soft constraints can be gained by adding a strength parameter to each rule and incorporating procedures that use those values in selecting rules. Fourth, resilience to damage can be gained by building redundancy into the rule system (e.g., making multiple copies of each rule). Fifth, increased attention can be given to learning algorithms (such as the genetic algorithm), knowlege compilation and "chunking" of rules into larger units, and ways of applying old knowledge to new problems (such as analogy).

The most comprehensive and successful nontraditional rule systems, such as J. R. Anderson's (1983) ACT* and Newell's (1988) SOAR, incorporate some of these design features (and Anderson makes explicit use of networks in addition to rules). Some differences with networks remain, but their importance and consequences are not as obvious as those involving traditional symbolic models. One of the remaining differences is that nontraditional symbolic models retain the use of ordered symbol strings whereas connectionist networks have no intrinsic ordering of their elements. In the most common architecture, the *production system*, these strings are rules of the form "If A, then B" where A is a Boolean combination of conditions, and B is a set of actions to be carried out when the conditions are met. Another difference is that sequenced operations and nonlocal control are inherent capabilities of symbolic models but not of networks. There presently is no adequate research base for determining what differences in empirical adequacy might result from these differences, but the differences are likely to be small enough that empirical adequacy will not be the primary determinant of the fate of symbolic versus connectionist models. Within either tradition, if a particular inadequacy is found, design innovations that find some way around the failure are likely to be forthcoming. Personal taste, general assumptions about cognition, the sociology of science, and a variety of other factors

can be expected to govern the individual choices that together will determine what approaches to cognitive modeling will gain dominance.

Given this state of affairs, in this book we shall draw our primary contrasts between traditional symbolic models and connectionist models. In this way we can convey, to some extent, why connectionists decided to abandon the traditional symbolic approach as a medium for modeling. In chapter 8 we shall present an argument that there are important tasks, other than modeling the cognitive mechanism, for which traditional symbolic theories are the theories of choice. In our view, connectionist and traditional symbolic inquiries should be carried out as distinctive enterprises, each of which can make contributions to the other; the availability of both approaches can strengthen cognitive science by providing multiple perspectives. The key to successful cooperation is that each approach be used for the tasks most suitable to it, rather than fighting for the same turf. For example, linguistic theories will always have a distinctive role to play, and presumably will remain symbolic. These theories efficiently describe the domain in which a connectionist (or other mechanistic model) must perform.

Within this framework, nontraditional symbolic theories do not have the same role to play as traditional ones: they are indeed fighting for the same turf as connectionism (that is, fine-grained modeling of the workings of the cognitive mechanism). However, the degree of polarization is not as great as it may seem, and the future could bring a pluralistic approach to mechanistic modeling within which connectionist themes and techniques are more distributed than is currently the case. Recent history provides some support for this scenario. Connectionist networks, in their incarnation as cognitive models, have origins in the symbolic tradition of the 1970s as well as in the neural network tradition. Schema theory and story grammars (Rumelhart, 1975), probabilistic feature models (Smith and Medin, 1981), prototype theory (Rosch, 1975), and scripts (Schank and Abelson, 1977) all emerged from the symbolic tradition but do not fully reside in either the symbolic or connectionist camp. All can be given a connectionist implementation, and these arguably are superior implementations. For example, schemata should be flexible and easy to modify, but this is much harder to achieve in a symbolic than in a connectionist implementation (Rumelhart, Smolensky, McClelland, and Hinton, 1986, in *PDP:14*). Furthermore, semantic networks with spreading activation (J. R. Anderson, 1983) are hybrid models that place symbols in network structures that dynamically change their activations; they can be regarded as a predecessor of connectionist models of cognition.

We shall point out where nontraditional and hybrid models are

relevant at various points in the discussion. There is such a variety of models, however, that we cannot provide a full treatment or make detailed comparisons within a book of this scope. Also, although we are favorably inclined to connectionist models, we decline to predict the outcome of the competition between connectionist and nontraditional symbolic models. The degree to which accommodation will be found, as in hybrid models or pluralism, simply is not known at this time. It is clear, however, that the cognitive science of the year 2000 will be a quite different cognitive science than would have emerged in the absence of the new connectionism.

2

Connectionist Architectures

Connectionist networks are intricate systems of simple units which dynamically adapt to their environments. Some have thousands of units, but even those with only a few units can behave with surprising complexity and subtlety. This is because processing is occurring in parallel and interactively, in marked contrast with the serial processing to which we are accustomed. To appreciate the character of these networks it is necessary to observe them in operation. Thus, in the first section of this chapter we shall describe a simple network that illustrates several features of connectionist processing. In the second section we shall examine in some detail the various design principles that are employed in developing networks. In the final section we shall discuss several appealing properties of networks that have rekindled interest in using them for cognitive modeling: their neural plausibility, satisfaction of "soft constraints," graceful degradation, content-addressable memory, and capacity to learn from experience. Connectionists maintain that the investment in a new architecture is amply rewarded by these gains.

The Flavor of Connectionist Processing: A Simulation of Memory Retrieval

We shall begin by describing a connectionist model which was designed by McClelland (1981) in order to illustrate how a network can function as a content-addressable memory system. Its architecture is atypical in some respects, but it conveys the flavor of connectionist processing in an intuitive manner. The information to be encoded concerns the members of two hypothetical gangs, the Jets and the Sharks, and some of their demographic characteristics (figure 2.1). Figure 2.2 shows how this information is represented in a network, focusing on just five of the