1

Networks versus Symbol Systems: Two Approaches to Modeling Cognition

1.1 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 research strategies and its way of construing psychological phenomena, 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* (see Bechtel, Abrahamsen, and Graham, 1998). Its domain of inquiry centrally included reasoning, memory, and language but also extended to perception and motor control. 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 will refer to it as the *symbolic paradigm*.

In the 1980s, however, an alternative framework for understanding cognition emerged in cognitive science, and a case can be made that it is a new Kuhnian paradigm (Schneider, 1987). 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. The years since have seen a steady stream of additional research as well as a number of textbooks (J. A. Anderson, 1995; Ballard, 1997; Elman et al., 1996; McLeod, Plunkett, and Rolls, 1998; O'Reilly and Munakata, 2000; Quinlan, 1991) and new journals (e.g., *Connection Science*, *Neural Computation,* and *Neural Networks*). Clearly connectionism has continued to attract 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, one 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.[1] 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 resulting stable configuration 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., + and ÷) 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. (For further discussion of the notion of *computation*, and whether it extends to the type of processing exhibited by connectionist networks, see B. C. Smith, 1996; van Gelder, 1995; and chapter 8, below.)

While connectionism has achieved widespread attention only since the 1980s, it is not a newcomer. The predecessors of contemporary connectionist models were developed in the mid-twentieth century and were still being 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 (at least for researchers in artificial intelligence and computational modeling in psychology) 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) and Anderson, Pellionisz, and Rosenfeld (1990), which gather together many of the seminal papers and offer illuminating commentary.

1.2 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, it was to model

cognitive phenomena in systems that exhibited some of the same basic properties as networks 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 McCulloch–Pitts neurons was elaborated in a variety of research endeavors in succeeding decades. John von Neumann (1956) showed how networks of such units 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 (for example, 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 of 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 section 2.2.4).

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. The first paper by McCulloch and Pitts 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 when quite different in appearance. 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. In his *elementary perceptron*, a single layer of McCulloch–Pitts units (shown as triangles in figure 1.1) received input from sensory units. Each McCulloch–Pitts unit was influenced in its own way by the input activations, as determined by a modifiable connection with each input that could range from strongly inhibitory to strongly excitatory. Whether the resulting activation was sufficient for the McCulloch–Pitts unit to fire depended upon its threshold (t). In this example,

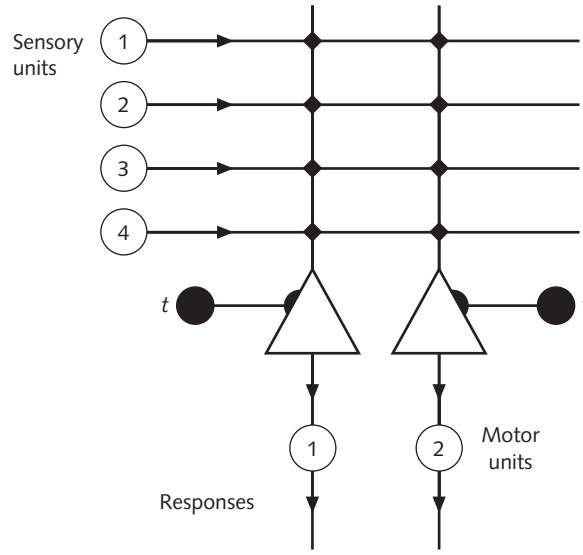

Figure 1.1 An elementary perceptron, as investigated by Rosenblatt (1958). Inputs are supplied on the four sensory units on the left and outputs are produced on the two motor units at the bottom. The network's computational units are the two McCulloch–Pitts neurons (large triangles), each of which has an inhibitory connection to a threshold unit (small dark circles). Each intersection between horizontal and vertical lines represents the synapse of one sensory unit on one of the McCulloch–Pitts neurons. This way of diagramming a network arranges the synapses such that, if their modifiable weights were shown, they would be in tabular format. Reprinted with permission from J. D. Cowan and D. H. Sharp (1988) Neural nets and artificial intelligence, *Daedalus*, 117, p. 90.

the output was sent to a motor unit (not an essential part of the architecture). Rosenblatt also explored networks with multiple layers of McCulloch–Pitts units, including some in which later layers might send excitations or inhibitions back to earlier layers.

Rosenblatt differed from McCulloch and Pitts in making the strengths (commonly referred to as the *weights*) of the connections continuous rather than binary and in introducing procedures for changing these weights so that perceptrons could learn. For elementary perceptrons, 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 not. If the unit was off when it should have been on, the weight on the connection from each active input unit was increased. Conversely, if the unit was on when it should have been off, the weight from each active input unit was reduced. Rosenblatt offered a proof of his 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. 156–7)

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 its reliance upon *cognitive demons* that performed computations in parallel without attention to one another, each of them "shouting out" its judgment of what letter had been presented (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 judgment about a letter even if some of its features were missing or atypical (see Selfridge, 1959; Selfridge and Neisser, 1960).

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 will 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

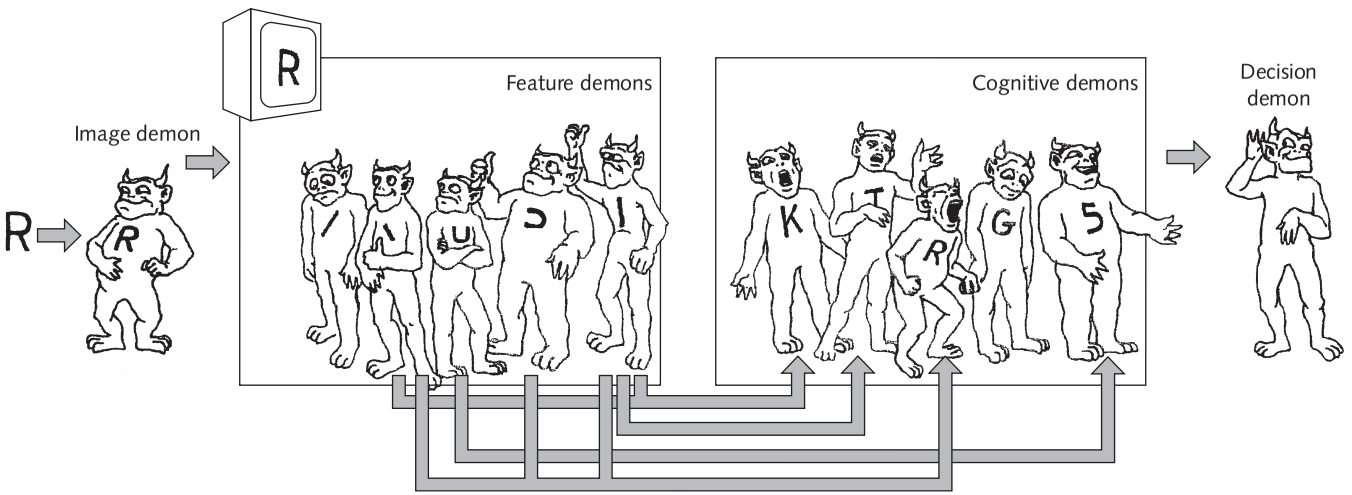

Figure 1.2 Selfridge's (1959) pandemonium model. The "demons" at each level beyond the image demon (which merely records the incoming image) extract information from the preceding level. Thus, a given feature demon responds positively when it detects evidence of its feature in the image, and a cognitive 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. Figure drawn by Jesse Prinz.

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 models for the operation of the hippocampus (Marr, 1971) and 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. In section 1.3 we will examine what made the symbolic approach so attractive to cognitive researchers, and in section 1.4 we will see that interest in networks declined until revived by connectionism in the 1980s. Finally, in section 1.5 we will get an overview of connectionism's continued development in the 1990s via alliances with other new approaches to cognition and end by raising the prospect of a *rapprochement* with the symbolic approach.

1.3 The Allure of Symbol Manipulation

1.3.1 From logic to artificial intelligence

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 (i.e., sentences) and connectives (e.g., *and*, *or*, *if–then*). 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, e.g., "If I think, then I exist").

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 properties 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 entities 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 by 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 judgments as to what is probably (but not necessarily) 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 making inferences involving a symbolic expression, we consider only its form. We can disregard the expression's representational function, that is, whether it is true or not, and if true, what state of affairs it describes. For example, the form of the expression (*p and* (*q or r*)) is that of a particular connective (*and*) with two arguments; one is a proposition (*p*) and the other is composed from another connective (*or*) with two propositional arguments (*q*, *r*). Based just on the form of the expression, without knowing anything about *p* or the other propositions, we infer *p*. If (*p and* (*q or r*)) is in fact true this is a sound inference, but if it is false then *p* may or may not be true and inferring it risks error. Thus, it is important to take care that the initial expressions (premises) are true before undertaking inference in a formal system. One advantage gained is the efficiency of attending only to form; another is that the symbols may be reinterpreted (i.e., assigned new representational roles) without affecting the validity of the inferences made using them.

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 arithmeti-

> cians 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 1s and 0s, implemented by *on* and *off* settings of switches). The basic operations of the computer allow recall of the symbols from memory and execution of 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 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 pointed 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 regarded this internal semantics as a major advance over formal symbol systems such as those of logic, and argued 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)

Newell and Simon thus disagreed with those cognitive scientists who, in emphasizing the continuity between computers and formal logic, retained the assumption that syntax should be autonomous from semantics. They saw computers as providing an

advantageous dovetailing of syntax and semantics that was not available within abstract formal logic. A similar difference in perspective arose with respect to what work the computer is regarded as carrying out. From a continuity perspective, computers are powerful devices for implementing logical operations: programs can be written to serve the same function as inference rules in a logical system. From the 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 grandmaster'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, until recently this analogy provided a locus of unity among cognitive scientists.

1.3.2 From linguistics to information processing

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*: to write a grammar was to specify an automaton that could generate sentences (which could comprise an infinite set if at least one recursive rule was included). One way to evaluate such a grammar was to ask whether it could generate 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 phrase structure trees by applying a succession of rewrite rules

(rules which expand one symbol into a string of subordinate symbols, each of which can itself be expanded, and so forth). Indefinitely long constituents can be embedded within such a 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 cognition.

1.3.3 Using artificial intelligence to simulate human information processing

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 are meaningful), 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, gained increased salience with the reemergence of network models in the 1980s. We turn now to a brief history of networks as an alternative to the symbolic tradition.

1.4 The Decline and Re-emergence of Network Models

1.4.1 Problems with perceptrons

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 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. Thereafter 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). But their work attracted only limited attention and funding. What is less clear is whether Minsky and Papert's book precipitated the decline, 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 an elementary perceptron (one in which input units are connected to a single layer of McCulloch–Pitts units). The centerpiece of their critique was their demonstration that there are functions,

such as those determining whether a figure is connected or whether the number of elements is odd or even, which cannot be evaluated by such a network. An example is the logical connective *exclusive or* (usually abbreviated as "XOR"). The expression *p XOR q* is defined as true if *p* is true and *q* is not, or *q* is true and *p* is not. In order for a perceptron to compute XOR, it is necessary to include an additional layer of McCulloch–Pitts units (now known as *hidden units*) between the input units and the original layer of McCulloch–Pitts units (now known as *output units*). 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 will discuss in section 3.2.2, an adaptation 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 judgment 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 fundamental problem: 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 recursive operations for generating trees and manipulating them. 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 will see in chapters 5 and 6, many advocates of the symbolic tradition continue to fault modern connectionism on precisely this ground.

1.4.2 Re-emergence: The new connectionism

In the early 1980s the type of network research pioneered by Rosenblatt began once again to attract attention and to gain adherents within what had now become known as cognitive science. Geoffrey Hinton and James A. Anderson's (1981) *Parallel Models of Associative Memory* was a harbinger, based on a 1979 conference that brought together UCSD's core group of cognitive scientists (especially David Rumelhart and Donald Norman) with some key researchers who had never abandoned networks (e.g., Anderson, Hinton, Teuvo Kohonen, and David Willshaw) and others who were newly attracted to them (e.g., Terrence Sejnowski from computational neuroscience and Jerome Feldman from artificial intelligence). Papers that employed networks to model various cognitive performances began to appear in cognitive journals. 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 Van Lehn arguing that networks were inadequate devices for achieving cognitive performance. Debate at that session and others during the conference occasionally became acrimonious as these "new connectionists"[2] began to press their alternative and challenged the supremacy of the symbolic approach. Connectionist research increased dramatically across the 1980s and became part of the established order in the 1990s, as departments hired young connectionists and many senior researchers added connectionist modeling techniques to their repertoire as tools to be employed for at least some purposes.

An intriguing question is why connectionism should have re-emerged so strongly when it did. Probably there was a confluence of factors. First, powerful new approaches to network modeling were developed around the early 1980s, including new architectures, new techniques for training multi-layered networks, and advances in the mathematical description of the behavior of nonlinear systems. Many of these innovations could be applied directly to the task of modeling cognitive processes. Second, the credibility and persuasiveness of some of the key innovators helped their message to get a hearing within cognitive science. 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 commented:

> 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, a related factor that was probably not essential but helped jump-start the new developments was that certain people were in the right place at the right time (e.g., Hinton and Anderson were visitors at UCSD, a leading center of symbolic cognitive science that became a leading center of network modeling, especially parallel distributed processing). Fourth, cognitive science had remained, either intentionally or unintentionally, somewhat 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 began to see advantages in the neural-like architecture of connectionist models. Fifth, this attraction to networks 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. The desire for parsimony, which earlier had characterized behaviorism, re-emerged. Sixth, 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 in learning from experience, inadequate generalization, domain specificity, and inefficiencies due to serial search. Human cognition, which the rule systems were supposed to be modeling, seemed to be relatively free of such limitations.

Cognitive scientists who were motivated by several of these factors became connectionists, and quite a battle ensued with advocates of the classic symbolic approach beginning in the mid-1980s. At the same time, though, developments within both

symbolic and network approaches often had the effect of softening the boundary between them. Some symbolic modelers, focusing on the fifth and sixth factors listed above, sought unified frameworks for cognitive modeling that shared some attributes with network models. ACT-R (John R. Anderson, 1993; Anderson and Lebière, 1998) uses a localist network architecture for its long-term memory and a production system architecture for operating on what is retrieved. The *Soar* architecture (Laird, Newell, and Rosenbloom, 1987) makes a production system do both jobs. However, as described in Newell's (1990) master work, *Unified Theories of Cognition*, it seems to approximate the spirit of connectionist models in its simplicity (e.g., fine-grained rules compete in parallel with no conflict resolution attempted).

On the connectionist side, some designers made *hybrid models* by implementing specific rule-based accounts in connectionist architectures so as to gain advantages of both approaches (e.g., Touretzky and Hinton, 1988; see section 6.2.1, below). Connectionists also found more general inspiration in certain approaches that emerged from the symbolic tradition shortly before connectionism itself emerged, and never fully resided in either the symbolic or connectionist camp; examples include schema theory and story grammars (Rumelhart, 1975), probabilistic feature models (Smith and Medin, 1981), symbol-based semantic networks with spreading activation (J. R. Anderson, 1983), prototype theory (Rosch, 1975), and scripts (Schank and Abelson, 1977). Some of these can be given a connectionist implementation, arguably superior to the original theory. 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*). Also, a major effort to implement scripts in networks is the focus of chapter 7. Work that combined aspects of the symbolic and connectionist approaches helped lay the groundwork for the more pluralistic, if not always less contentious, cognitive science that opened the twenty-first century.

1.5 New Alliances and Unfinished Business

The big story of recent years, however, is not the softening of the boundary between symbolic and connectionist approaches. It is the new alliances that specialized subgroups of connectionists have formed with other emerging frameworks for understanding cognitive and sensorimotor abilities. In this second edition we examine three such alliances.

- *Dynamical approaches* to cognition give long-overdue priority to the dimension of time, and the mathematical and visual tools of dynamical systems theory illuminate how certain types of connectionist networks achieve their success.
- *Embodied cognition* is the idea that mind cannot be understood only by modeling internal activity; it is crucial to extend inquiry outwards to the mind's interactive couplings with the body and environment. Creating network controllers for robots provides a way of pursuing this idea, and using simulated evolution as the method makes them especially relevant to a new research field called "artificial life."
- *Cognitive neuroscience* is a field that has thrived recently due to the availability of new ways to measure and form images of the activity of the brain during cognitive activity. Network modelers increasingly are moving their focus down into

> the brain, tailoring the architecture and tasks performed by networks to knowledge about particular brain areas that has been gained not only from neuroimaging but also from such traditional methods as lesion studies, ERP, and single-cell recording in animals.

These new alliances will produce some of the most exciting work of the first decade of the twenty-first century. Whatever their success, though, they will leave some unfinished business. For reasons that we still do not understand, systems with enough parallel, distributed, dynamical, embodied and neurally grounded activity to do just about anything – perhaps even achieving Turing equivalence – repeatedly find themselves in the same grooves. That is, they behave in ways that can be closely approximated by symbolic models, and for many purposes it is the symbolic models that are most convenient to use. This is especially clear in the case of language: network models of the brain's activities in processing language, however good they get, will not displace linguistics. The real challenge for connectionists will not be to defeat symbolic theorists, but rather to come to terms with the ongoing relevance of the symbolic level of analysis. That is, the ultimate new alliance may be as simple, and as difficult, as forming a new relationship with the long-time opponent.

In most circles this idea currently has little priority and few adherents. If the future of connectionism lies in yet another alliance – one with the symbolic approach it has been opposing vigorously for years – a glimpse of that future is available now in Optimality Theory (OT; see Prince and Smolensky, 1993). This new linguistic framework originated in an alliance between two people: Paul Smolensky, who was a major contributor to connectionism in the 1980s, and Alan Prince, who was a major opponent during that same period. They found common ground in the discovery that various phonological phenomena can be described using a universal set of soft constraints to select the optimal output among a large number of candidates. A given language has its own rigid rank ordering of these constraints, which settles the numerous conflicts between them.

As a very simple example (see Tesar, Grimshaw, and Prince, 1999, for the five-constraint version from which this is drawn), the constraint NoCoda is violated by any syllable ending in a consonant (the coda) and the constraint NoInsV is violated if a vowel is inserted in the process of forming syllables (the output) from a phoneme string (the input). If these were the only two constraints to consider (in fact there always are more), the input string /apot/ would be syllabified as *.a.pot.* in a language that ranks NoInsV higher (e.g., English), but as *.a.po.to.* or some other vowel-final form in a language that ranks NoCoda higher (e.g., Japanese). Working with talented collaborators, Smolensky and Prince developed Optimality Theory into such an elegant account that in just a few years it came to dominate work in phonology.

One unfinished task for optimality theorists is to achieve an equally compelling OT account of syntax. Another is to achieve a well-motivated interface between OT and the network-like level that is assumed to be its substrate (see Prince and Smolensky, 1997, for the recent status of this effort). As we will see in chapter 2, networks can be viewed as devices for constraint satisfaction and hence should provide a fairly natural implementation of OT. In Smolensky's harmonic grammar, for example, weighted connections can be used to optimally satisfy a set of linguistic constraints (in accord with Smolensky's more general Harmony Theory; see Smolensky, 1986, in *PDP:6*). The problem is that the networks of harmonic grammar engage in competition quantitatively – various input patterns and the weights of

various connections can yield many different outcomes – but a strict ranking of constraints always emerges at the higher level of description provided by OT. Why? Nobody knows. Until that problem is solved, the network level of description is of limited explanatory utility with respect to OT. But the solution, when and if it is found, may create a *rapprochement* between network models and symbolic accounts that triggers an era of dramatic progress in which alignments are found and used all the way from the neural level to the cognitive/linguistic level.

We mention this future possibility in order to now put it aside. Classic connectionism and its battle with the classic symbolic approach fill the next six chapters of this book, and the alliances that are currently most influential within connectionism are the focus of the last three chapters. Specifically, we introduce network architectures in chapter 2 and learning procedures in chapter 3. Then some specific network models are presented in the context of philosophical positions: some that are concordant with connectionism in chapter 4, followed by battles over rules in chapter 5, and battles over representations in chapter 6. A modular network implementation of a quasi-symbolic framework, scripts, is presented in some detail in chapter 7. We then move to alliances with the dynamical approach in chapter 8 (a prickly alliance, it will be seen), artificial life and embodied cognition in chapter 9, and cognitive neuroscience in chapter 10. It will become increasingly apparent in these later chapters that classic connectionism is just one way of "doing networks" and that an era of pluralism is already well under way.

Notes

1 If one were trying to model the ongoing life of the mind, as opposed to its response to a specific input, one might not want the network to really stabilize but only to achieve temporarily stable states, which might then be disrupted by new inputs or other internal processes.

2 The earliest connectionists were not neural network modelers of the mid-twentieth century like Rosenblatt, but associationists who viewed higher-order competencies as arising from connections among simpler elements. For Wernicke in the late nineteenth century the elements were neurally realized sensory and motor encodings; for Thorndike in the early twentieth century they were stimuli and responses. Each called his approach "connectionism."

Sources and Suggested Readings

Anderson, J. A. and Rosenfeld, E. (1998) *Talking Nets: An Oral History of Neural Networks*. Cambridge, MA: MIT Press.

Anderson, J. A. and Rosenfeld, E. (eds) (1988) *Neurocomputing: Foundations of Research*. Cambridge, MA: MIT Press.

Anderson, J. A., Pellionisz, A., and Rosenfeld, E. (1990) *Neurocomputing 2: Directions for Research*. Cambridge, MA: MIT Press.

Churchland, P. M. (1995) *The Engine of Reason, the Seat of the Soul*. Cambridge, MA: MIT Press.

Clark, A. (1989) *Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing*. Cambridge, MA: MIT Press.

Clark, A. (1993) *Associative Engines: Connectionism, Concepts, and Representational Change*. Cambridge, MA: MIT Press.

Cowan, J. D. and Sharp, D. H. (1988) Neural nets and artificial intelligence. *Daedalus*, 117, 85–121.

Cummins, R. and Cummins, D. D. (eds) (2000) *Minds, Brains, and Computers: The Foundations of Cognitive Science: An Anthology*. Oxford: Blackwell.

Franklin, S. (1995) *Artificial Minds*. Cambridge, MA: MIT Press.

Grossberg, S. (1982) *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*. Dordrecht: Reidel.

Rutgers Optimality Archive (ROA): electronic repository of papers on Optimality Theory at *http://ruccs.rutgers.edu/roa.html*.