CHAPTER 13

Information Processing

DAVID KLAHR and BRIAN MACWHINNEY
1. The human information-processing system architecture is isomorphic to a production-system architecture. This premise derives from observations about similarities in terms of both structural organization and behavioral properties. Structurally, production systems provide a plausible characterization of the relations between long-term memory and working memory, and about the interaction between procedural and declarative knowledge. Behaviorally, strong analogies can be seen between humans and production systems with respect to their abilities to mix goal-driven and event-driven processes, and with their tendency to process information in parallel at the recognition level and serially at higher cognitive levels.

2. Change is a fundamental aspect of intelligence; we cannot say that we fully understand cognition until we have a model that accounts for its development. The first 20 years of information-processing psychology devoted scant attention to the problems of how to represent change processes, other than to place them on an agenda for future work. Indeed, almost all of the information-processing approaches to developmental issues followed the two-step strategy outlined in the Simon quotation that opened this chapter: First construct the performance model, and then follow it with a change model that operates on the performance model. In recent years, as researchers have begun to work seriously on the change process, they have begun to formulate models that inextricably link performance and change. Self-modifying production systems are one such example of this linkage.

3. All information-processing-system architectures, whether human or artificial, must obey certain constraints in order to facilitate change. It is these constraints that give rise to the seemingly complex particulars of individual production-system architectures. Thus, an understanding of production-system models of change is a step toward understanding the nature of human development and learning.

**CONNECTIONIST SYSTEMS**

In this section we examine work conducted from the connectionist perspective. Because both production system modelers and connectionists are pursuing common goals, there are many points where their pathways converge. Both approaches rely heavily on computational modeling. Both approaches understand the importance of matching theory to data. Both perspectives have come to understand the importance of emergent properties in understanding transition mechanisms. Since the final understanding of transition mechanisms may well require insights from both perspectives, it makes little sense to advance strong claims for superiority of one approach over the other. Rather, we need to understand why researchers are currently exploring different paths, invoking different incantations, and wielding different computational weapons. To do this, we need to better understand the differences in the goals and constraints assumed by the two approaches.

We start with a brief description of the basic features of connectionist models. Then we address a few important aspects of connectionism that distinguish it from production system approaches. One basic distinction comes from the fact, noted earlier, that production systems take the symbol as their basic building block, while connectionist systems take a "sub-symbolic" perspective. Although we have reserved most of the "compare and contrast" discussion in this chapter for the final section, it is important to treat this distinction at the outset of our presentation of connectionist models. Following that discussion we turn to a review of actual work conducted in the connectionist framework.

**Basic Principles of Neural Networks**

Connectionist models are implemented in terms of artificial neural networks. Neural networks that are able to learn from input are known as "adaptive neural networks." In practice, all current neural network frameworks are based on adaptive neural networks. The architecture of an adaptive neural network can be specified in terms of eight design features:

1. **Units.** The basic components of the network are a number of simple elements called variously neurons, units, cells, or nodes. In Figure 13.3, the units are labeled with letters such as "x_i."

2. **Connections.** Neurons or pools of neurons are connected by a set of pathways which are variously called connections, links, pathways, or arcs. In most models, these connections are unidirectional, going from a "sending" unit to a "receiving" unit. This unidirectional assumption corresponds to the fact that neural connections also operate in only one direction. The only information conveyed across connections is activation information. No signals or codes are passed. In Figure 13.3, the connection between units x_i and y_j is marked with a thick line.
3. Patterns of connectivity. Neurons are typically grouped into pools or layers. Connections can operate within or between layers. In some models, there are no within-layer connections; in others all units in a given layer are interconnected. Units or layers can be further divided into three classes:

(a) **Input units** which represent signals from earlier networks. These are marked as x units in Figure 13.3.

(b) **Output units** which represent the choices or decisions made by the network. These are marked as y units in Figure 13.3.

(c) **Hidden units** which represent additional units juxtaposed between input and output for the purposes of computing more complex, nonlinear relations. These are marked as z units in Figure 13.3.

4. **Weights.** Each connection has numerical weight that is designed to represent the degree to which it can convey activation from the sending unit to the receiving unit. Learning is achieved by changing the weights on connections. For example, the weight on the connection between x₁ and y₁ is given as .54 in Figure 13.3.

5. **Net inputs.** The total amount of input from a sending neuron to a receiving neuron is determined by multiplying the weights on each connection to the receiving unit times the activation of the sending neuron. This net input to the receiving unit is the sum of all such inputs from sending neurons. In Figure 13.3, the net input to y₁ is .76, if we assume that the activation of x₁ and x₂ are both at 1 and the x₁y₁ weight is .54 and the x₂y₁ weight is .22.

6. **Activation functions.** Each unit has a level of activation. These activation levels can vary continuously between 0 and 1. In order to determine a new activation level, activation functions are applied to the net input. Functions that "squash" high values can be used to make sure that all new activations stay in the range of 0 to 1.

7. **Thresholds and biases.** Although activations can take on any value between 0 and 1, often thresholds and bias functions are used to force units to be either fully on or fully off.

8. **A learning rule.** The basic goal of training is to bring the neural net into a state where it can take a given input and produce the correct output. To do this, a learning rule is used to change the weights on the connections. *Supervised* learning rules need to rely on the presence of a target output as the model for this changing of weights. *Unsupervised* learning rules do not rely on targets and correction, but use the structure of the input as their guide to learning.

All connectionist networks share this common language of units, connections, weights, and learning rules. However, architectures differ markedly both in their detailed patterns of connectivity and in the specific rules used for activation and learning. For excellent, readable introductions to the theory and practice of neural network modeling, the reader may wish to consult Bechtel and Abrahamsen (1991) or Fausett (1994). For a mathematically more advanced treatment, see Hertz, Krogh, and Palmer (1991).

To illustrate how connectionist networks can be used to study cognitive development, let us take as an example the...
model of German gender learning developed by MacWhinney, Leinbach, Taraban, and McDonald (1989). This model was designed to explain how German children learn how to select one of the six different forms of the German definite article. In English we have a single word “the” to express definiteness. In German, the same idea can be expressed by der, die, das, des, den, or den. Which of the six forms of the article should be used to modify a given noun in German depends on three additional features of the noun: its gender (masculine, feminine, or neuter), its number (singular or plural), and its role within the sentence (subject, possessor, direct object, prepositional object, or indirect object). To make matters worse, assignment of nouns to gender categories is often quite nonintuitive. For example, the word for fork is feminine, the word for spoon is masculine, and the word for knife is neuter. Acquiring this system of arbitrary gender assignments is particularly difficult for adult second language learners. Mark Twain expressed his own consternation at this aspect of German in a treatise entitled “The awful German language” (Twain, 1935) in which he accuses the language of unfairness in assigning pretty young girls to the neuter gender, while allowing the sun to be feminine and the moon masculine. Along a similar vein, Maratsos and Chalkley (1980) argued that, since neither semantic nor phonological cues can predict which article accompanies a given noun in German, children could not learn the language by relying on simple surface cues.

Although these relations are indeed complex, MacWhinney et al. show that it is possible to construct a connectionist network that learns the German system from the available cues. This model, like most current connectionist models, involves a level of input units, a level of hidden units, and a level of output units (Figure 13.4). Each of these levels or layers contains a number of discrete units or nodes. For example, in the MacWhinney et al. model, the 35 units within the input level represent features of the noun that is to be modified by the article. Each of the two hidden unit levels includes multiple units that represent combinations of these input-level features. The six output units represent the six articles in the German language that correspond to the word the in English.

As noted, a central feature of such connectionist models is the very large number of connections among processing units. As shown in Figure 13.4, each input-level unit is connected to first-level hidden units; each first-level hidden unit is connected to second-level hidden units; and each second-level hidden unit is connected to each of the six output units. None of these hundreds of individual node-to-node connections are illustrated in Figure 13.4, since graphing each individual connection would lead to a blurred pattern of connecting lines. Instead a single line is used to stand in place of a fully interconnected pattern between levels. Learning is achieved by repetitive cycling through three steps. First, the system is presented with an input pattern that turns on some, but not all of the input units. In this case, the pattern is a set of sound features for the noun being used. Second, the activations of these units send activations through the hidden units and on to the output units. Third, the state of the output units is compared to the correct target and, if it does not match the target, the weights in the network are adjusted so that connections that suggested the correct answer are strengthened and connections that suggested the wrong answer are weakened.

MacWhinney et al. tested this system’s ability to master the German article system by repeatedly presenting 102 common German nouns to the system. Frequency of presentation of each noun was proportional to the frequency with which the nouns are used in German. The job of the network was to choose which article to use with each noun in each particular context. After it did this, the correct answer was presented, and the simulation adjusted connection strengths so as to optimize its accuracy in the future.

After training was finished, the network was able to choose the correct article for 98% of the nouns in the original set. The ability to learn the input set is not a demonstration of true learning, since the network may have simply memorized each presented form by rote. However, when the simulation was presented with a previously encountered noun in a novel context, it chose the correct article on 92% of trials, despite the noun’s often taking a different article in the new context than it had in the previously encountered ones. This type of cross-paradigm generalization is clear evidence that the network went far beyond rote memorization during the training phase. In addition, the simulation was able to generalize its internalized knowledge to entirely novel nouns. The 48 most frequent nouns in German that had not been included in the original input set were presented in a variety of sentence contexts. On this completely novel set, the simulation chose the correct article from the six possibilities on 61% of trials, versus 17% expected by chance. Thus, the system’s learning mechanism, together with its representation of the noun’s phonological and semantic properties and the context, produced a good guess about what article would accompany a given noun, even when the noun was entirely unfamiliar.
The network's learning paralleled children's learning in a number of ways. Like real German-speaking children, the network tended to overuse the articles that accompany feminine nouns. The reason for this is that the feminine forms of the article have a high frequency because they are used both for feminines and for plurals of all genders. The simulation also showed the same type of overgeneralization patterns that are often interpreted as reflecting rule use when they occur in children's language. For example, although the noun Kleid (which means clothing) is neuter, the simulation used the initial "kl" sound of the noun to conclude that it was masculine. Because of this, it invariably chose the article that would accompany the noun if it were masculine. Further, the same article-noun combinations that are the most difficult for children proved to be the most difficult for the simulation to learn and to generalize to on the basis of previously learned examples.

How was the simulation able to produce such generalization and rule-like behavior without any specific rules? The basic mechanism involved adjusting connection strengths between input, hidden, and output units to reflect the frequency with which combinations of features of nouns were associated with each article. Although no single feature can predict which article would be used, various complex combinations of phonological, semantic, and contextual cues allow quite accurate prediction of which articles should be chosen. This ability to extract complex, interacting patterns of cues is a particular characteristic of the type of connectionist algorithm, known as back-propagation, that was used in the MacWhinney et al. simulations. What makes the connectionist account for problems of this type particularly appealing is the fact that an equally powerful set of production system rules for German article selection would be quite complex (Mugdan, 1977) and learning of this complex set of rules would be a challenge in itself.

Connectionist Constraints on Computational Models

As we pointed out earlier, theoretical claims regarding production-system models do not extend to the underlying architecture of the computer on which they run. However, production systems have the potential to embody the same computational power as the von Neumann serial computer. Such models only become plausible as theories of human cognition when additional constraints are added, such as the size of working memory, the total amount of activation, and so on. However, some connectionists have not been satisfied with this analysis of the relation between von Neumann machines and human cognition. Instead, they have argued that the very nature of the underlying neural system yields emergent properties that are quite different from those implicit in production-system architectures. In particular, adaptive neural network models (Grossberg, 1987; Hopfield, 1982; Kohonen, 1982) deliberately limit this descriptive power of their models by imposing two stringent limitations on their computational models: a prohibition against symbol passing and an insistence on self-organization rather than hand wiring. We will describe each of these constraints below.

 Thou Shalt Not Pass Symbols

The brain is not constructed like a standard digital computer. The crucial difference between the two machines lies in the structure of memory storage and access (Kanerva, 1993). In the random-access memory of a standard digital computer (von Neumann, 1956), there are a series of hard locations, each of which can store a single "word" of data. The size of the memory depends on the length of the word of data. Because the computer is built out of highly reliable electrical components, the integrity of each memory location can be guaranteed. Neural hardware is made out of noisy, unstable components and no such guarantees can be issued. To compensate for the lower reliability of individual components, the brain relies on massive parallelism and distributed memory encodings. In the type of neural memory that appears to be implemented in the cerebellum (Albus, 1981; Marr, 1969), the address space is huge and sparse. Because the system cannot rely on locating individual hard addresses at the site of individual neurons (Kanerva, 1993), it must perform retrieval by locating addresses in the general vicinity of the stored memory. These addresses are called "soft" memory addresses, since they refer not to a single location, but to a general position in address space. The address space has a huge number of dimensions; but, because it is so sparsely populated, retrieval of memories does not require the exact determination of hard addresses.

An alternative method for passing symbols between neurons would view individual neurons as separate processing units capable of sending and receiving signals. But we know that the signals sent and received by neurons are entirely limited in shape. Neurons do not send Morse code down axons, symbols do not run across synapses, and brain waves do not pass phrase structures. In general, the brain provides no obvious support for the symbol passing architecture that provides the power underlying the von Neumann machine.
Instead, computation in the brain appears to rely ultimately on the formation of redundant connections between individual neurons.

The ways in which the brain has adapted to these limitations are not yet fully understood. The cerebellar addressing system is probably only one of several neural memory systems that use soft addresses and other storage techniques. We know that the hippocampus is also involved in aspects of memory storage (Schmajuk & DiCarlo, 1992) and it appears that its role may involve techniques involving data compression. There are also various rehearsal pathways designed to implement the learning of verbal material (Gathercole & Baddeley, 1993; Gupta & MacWhinney, 1994, 1996). Our emerging understanding of the various memory systems of the brain points to a complex interaction between cortex, thalamus, hippocampus, cerebellum, and other brain structures that work both on line and during sleep to facilitate storage, learning, and retrieval of memories. All of this work is done in ways that circumvent the limitations on symbol passing imposed by the biological structure of neurons.

**Thou Shalt Not Hand-Wire**

By itself, the requirement that computation be performed locally without symbol passing or homunculi is not enough to fully constrain the descriptive power of our models. One could still hand-wire a neural network to perform a specific function or to model a particular behavior. In neural networks, hand-wiring can be accomplished by creating a little program or homunculus that gets inside the network and sets weights on individual links between nodes. For example, we could hand-wire an animal category by linking nodes labeled “cat,” “dog,” and “tiger” to a hand-coded node labeled “animal.” By detailed weight setting and the use of gating and polling neurons, virtually any function can be wired into a neural network (Hertz et al., 1991). An early example of a fully hand-wired neural network was Lamb’s (1966) stratificational grammar. More recently, we have seen hand-wired neural networks in areas such as interactive activation models of reading (McClelland & Rumelhart, 1981), speech errors (Dell, 1986; MacWhinney & Anderson, 1986; Steemberg, 1985), ambiguity resolution (Cottrell, 1985), and lexical activation (Marslen-Wilson, 1987). Although these networks fit within the general framework of connectionist models, the fact that they are constructed through hand-wiring makes them less interesting as developmental models.

Certain “hybrid” models move the process of hand-wiring away from the network level onto an alternative symbolic level. This “implementational” approach to hand-wiring spares the modeler the tedium of hand-wiring by running the wiring procedure off symbolic templates. For example, Touretzky (1990) has shown that there are techniques for bottling the full power of a LISP-based production-system architecture into a neural net. These demonstrations are important because they show how difficult it is to control excessive modeling power.

Ideally, we want to match the constraint against symbol passing with the requirement that networks be self-organizing. We want to make sure that specific representations are not hand-wired and that the connections between units are developed on the basis of automatic learning procedures. Although we will always be forced to “label” our input nodes and output nodes, we want our labeling systems to be general across problems and not hand-crafted anew for each particular problem. Rather, we want to use general forms of representation that lead to robust and emergent learning without recourse to hand-wiring. It is the emergent, self-organizing properties of neural networks that make them particularly interesting to the developmental psychologist. Such models can display further interesting and important properties, such as stage transitions (Shultz, Schmidt, Buckingham, & Mareschal, 1995), category leakage (McClelland & Kawamoto, 1986), graceful degradation (Harley & MacAndrew, 1992; Hinton & Shallice, 1991; Marchman, 1992), and property emergence (MacWhinney et al., 1989).

**Alternative Network Architectures**

One of the principal goals of connectionist theory over the last thirty years has been the exploration of the properties of competing network architectures. In this section we will review the most important network architectures with an eye toward understanding the types of developmental processes for which each might be most relevant. There is a great deal of evidence to suggest that no single architecture is ideal for all purposes and that the human brain probably uses different patterns of neural connectivity to solve different cognitive problems.

**Perceptrons**

In the late 1950s, researchers (e.g., Rosenblatt, 1959; Block, 1962; Widrow & Hoff, 1960) explored the properties of a simple connectionist model called a perceptron. This model connected a series of input units to one or more output units using simple unidirectional connections. The weights in the network were trained using an algorithm
called the perceptron learning rule. The perceptron learning rule comes along with the rather attractive guarantee that, if a perceptron can be configured to solve a problem, the algorithm will succeed in finding the solution. The rub is that often turns out that perceptrons cannot solve even very simple problems. For example, Minsky and Papert (1969) showed that perceptrons can encode a relation such as "black and tall," but not a relation such as "black but not tall." The problem with perceptrons is not with the learning rule, but with the strength of the basic computational mechanism. Today, perceptrons are only of historical interest.

**Pattern Associators and Backpropagation**

The successors to the perceptron are the pattern associators, and there are dozens of pattern associator architectures. Typically, these devices are designed as models of retrieval in human memory. They rely for their power on the holographic quality of neural networks which are able to retrieve stored patterns through vector manipulations. For example, a pattern associator should be able to take the sound /bal/ and retrieve the spelling B-A-L-L or it can take the smell of a rose and retrieve the vision of the thorns of the rose. Networks of this type are often trained using the delta rule or the extended delta rule. These rules compare the network's output patterns against some target signal and make weight adjustments to bring the network into line with the target.

The backpropagation architecture (Werbos, 1974) achieves additional computational power by adding an additional level of units between the input and output layers. These additional units are called "hidden units" because they have no direct connection to either the input or the output. Networks using backpropagation with hidden units and the delta rule can solve many types of problems that are difficult for simpler machines such as the perceptron. In fact, most current work in computational modeling of developmental phenomena makes use of the backpropagation framework. This single, simply characterized algorithm has demonstrated an ability to learn a wide variety of subtle patterns in the data.

Despite the proven success of backpropagation, there are several crucial problems that arise when we try to use this single architecture as an account for all aspects of cognitive and linguistic development. Each of the problems encountered by backpropagation has served as a stimulus to the development of interesting alternative frameworks. One basic problem that arises immediately as we try to match the backpropagation algorithm up to the brain is the fact that backpropagation assumes that connections which fire in a feed-forward fashion can also be trained in a feedbackward direction. However, we know that real neurons fire in only one direction and that this type of backwards training is not neurologically plausible. However, as Fussett (1990) shows, one can devise backpropagation networks that can be trained in a unidirectional and local manner by adding additional arrays of controlling units.

The study of the actual mechanics of weight changing in neural networks is very much the province of the cellular neurophysiologist. In this area, there is increasing evidence emphasizing the extent to which the neuron can compute complex functions. Hebb (1949) suggested that learning occurs when two cells fire simultaneously and the output of the postsynaptic cell functions to strengthen the firing of the synapse connecting the two cells. Although work by Kandel and Hawkins (1992) with the sea slug supports aspects of the Hebbian model of learning, Alkon and colleagues (1993) have found computationally more complex learning in higher organisms such as rabbits and rats. This non-Hebbian learning takes place locally on small areas of the dendritic cell membrane. Alkon has implemented a network model called Dystal that faithfully mimics these aspects of membrane activity and also works well as a connectionist pattern associator.

**Networks That Deal with Time**

In the standard backpropagation framework, processing is idealized as occurring at a single moment in time. This idealization may make sense for processes that are extremely brief or for decisions in which many factors are being weighed without time constraints. However, for problems such as word recognition, sentence production, seriation, and speeded chess playing, temporal components are crucial components of the task. One network architecture that deals with this problem is a variation on back propagation developed by Jordan (1986) and Elman (1990). This variation takes the standard three-layer architecture of pools A, B, and C shown in Figure 13.5 and adds a fourth input pool D of context units which has recurrent connections to pool B. Because of the recurrent or bidirectional connections between B and D, this architecture is known as recurrent backpropagation.

A recurrent backpropagation network encodes changes over time by storing information regarding previous states in the pool of units labeled as D. Consider how the network deals with the processing of a sentence such as "Mommy loves Daddy." When the first word comes in, pool C is activated and this activation is passed on to pool B and then
pools A and D. The complete state of pool B at Time 1 is stored in pool D. The activation levels in pool D are preserved, while pools A, B, and C are set back to zero. At time 2 the networks hears the word "love" and a new pattern of activations is established on pool C. These activations are passed on to pools B, C, and D. However, because pool D has stored activations from the previous word, the new state is blended with the old state and pool C comes to represent aspects of both "Mommy" and "love."

Processing in a network of this type involves more than just storage of a superficial sequence of words or sounds. For example, in the simulations of sentence processing developed by Elman (1993), the output units are trained to predict the identity of the next word. In order to perform in this task, the network needs to implicitly extract part-of-speech information from syntactic co-occurrence patterns. Alternatively, the output units can be used to represent comprehension decisions, as in the model of MacWhinney (1996). In that model, part-of-speech information is assumed and the goal of the model is to select the agent and the patient using a variety of grammatical and pragmatic cues.

Another method for dealing with temporal ordering was developed by Grossberg (1978). In this system, linear ordering of elements such as the phonemes in a word is controlled by cluster units which sit above the component phoneme units and control their ordering as what Grossberg calls an "avalanche." The Elman and Grossberg systems are designed for markedly different problems. Grossberg's system works well for the learning of invariant serial orderings such as those found in lexical phonology and Elman's system is more appropriate for the learning of flexible, variant patterns of serial ordering, such as those found in syntax. It would not be surprising to find that other problems in serial ordering required still other network architectures.

Avoiding Catastrophes. A serious limitation of the backpropagation algorithm is its tendency toward developmental instability. A backpropagation network trained on one set of inputs can undergo a process of "catastrophic interference" (McCloskey & Cohen, 1989) when the input corpus is shifted to a markedly different structure. The problem of catastrophic interference shows up clearly when a network is trained with one language (L1) and then suddenly switched to dealing with input from a second language (L2). What happens is that learning of L2 wipes out knowledge of L1 (MacWhinney, 1996). Of course, no such catastrophic interference occurs in real life. When we learn a second language in real life, our knowledge of our first language remains firm.

Catastrophic interference occurs in backpropagation networks because new memories tend to overwrite old memories. One class of solutions tries to address this problem by making minor changes to backpropagation. This can be done by making weight changes only for novel aspects of the input (Kortge, 1990), hand-tuning the input corpus to avoid sudden changes (Hetherington & Seidenberg, 1989), localizing the receptive fields for units (Kruschke, 1992), or adding units with different learning rates (Hinton & Plaut, 1987). Although these solutions solve the problem of catastrophic interference, they often force us to make overly restrictive assumptions about the possible distributions of cues in the environment.

Localized Memories. A more general approach to the problem of catastrophic interference and other forms of crosstalk focuses on the role of neuronal topology in controlling neuronal recruitment and memory development. In topological models, units are more specifically devoted to specific memories, interactions between memories tend to be confined to local areas, and major shifts in the character of the input do not overwrite these localized memories.

Kanerva's Sparse Distributed Memory (SDM) is one such topological approach. The SDM model allows for one-shot storage of new memories without crosstalk. However, memories must be stored at several neighboring locations to guarantee consistent retrieval. A similar framework has been proposed by Read, Nenov, and Halgren (1995) on the basis of Gardner-Meadwin's (1976) model of hippocampal functioning.
The idea of encoding memories through topological organization in the brain is further elaborated in the self-organizing feature map (SOFM) approach developed by Kohonen (1982) and Miikkulainen (1990). Self-organizing feature maps use an unsupervised, competitive learning algorithm. All input units are connected to cluster units which are organized in a two-dimensional topological grid (see Figure 13.6), which is actually a compressed representation of a multidimensional space. When an input is presented, the cluster unit that responds most strongly becomes the winner. The winning unit then decrements the units that are just outside its immediate neighborhood so that they are less likely to respond to a similar input when it is next presented. The pattern of inhibition follows the “Mexican hat” format found in cells of the visual cortex. In this way, two units that initially respond to the same set of inputs start to pull away from each other. As this process continues, the radius for each unit decreases and its specificity increases. MacWhinney (1996) found that a self-organizing feature map of 10,000 units was able to learn an array of 6,000 words with 99% accuracy. Thus, it seems that the SOFM architecture is well-suited for the learning of arbitrary associations such as words.

The success of feature maps in the learning of arbitrary associations, such as the sound-meaning associations involved in words, stands in marked contrast to the problems that backpropagation networks have with the same task. The backpropagation architecture is designed to detect patterns, rather than to encode arbitrary associations. When a backpropagation network is trained with a long list of English words, it will lose its ability to acquire new words after learning the first 700 words or so. Adding more hidden units to the network does not help at this point, since the limitation seems to be in the basic resolution of the weight space. The reason that backpropagation reaches saturation for learning new words is not because of the shortage of nodes, but because of problems with the basic algorithm. Backpropagation uses hidden units not as individual address spaces for individual lexical items, but as pattern detectors that search for commonalities between words. However, because words are really arbitrary associations between sounds and meanings, backpropagation is frustrated in its attempt to pick up meaningful or useful patterns. The SOFM architecture, on the other hand, can be used to simply throw a large number of only weakly associated memories onto a large feature map. As MacWhinney (1996) has found, feature maps and sparse distributed maps can learn items up to the size of the feature map. In this regard, they seem better suited to the task of lexical learning than does an architecture such as backpropagation.

Networks That Grow. In addition to the crosstalk problem that lies at the root of catastrophic interference, backpropagation networks also suffer from a problem with commitments to local minima during early phases of training. These networks tend to isolate the major patterns in the input early on and are often incapable of picking up secondary strategies that conflict with the basic patterns in the input. One way of solving this problem is to force the network to “start small.” By giving the network only minimal resources at first and allowing it to recruit new resources when the problem becomes more difficult, it is possible to force the network to treat basic statistical regularities as fundamental, while still learning higher-order regularities later.

Within the backpropagation framework, there have been quite a few recent proposals about how to add new units during learning (Azimi-Sadjadi, Sheedvash, & Trujillo, 1993; Fahlan & Lebiere, 1990; Frean, 1990; Hirata, Yamashita, & Hijiyama, 1991; Kadirkamanathan & Niranjan, 1993; Platt, 1991; Wynne-Jones, 1993). One of
these models is the “cascade correlation” approach of Fahlman and Lebière (1990) which adds units when error reduction is not otherwise possible. The network begins its existence with only input and output units and no hidden units. In this form, it is equivalent to a perceptron. During training, new hidden units are added to the net in an effort to continually reduce the error in the output. As we will see below, this expansion of computational space through recruitment allows cascade-correlation networks to solve developmental problems that stymie standard backpropagation networks.

The idea of adding new units to networks to increase their computational capacity can be found in many frameworks. Within the framework of Adaptive Resonance Theory or ART (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992; Carpenter, Grossberg, & Reynolds, 1991; Grossberg, 1987), recruitment is a basic part of network functioning. For example, Grossberg (1987) adds new units to a network when no current unit matches a new input within a certain level of tolerance. Blackmore and Miikkulainen (1993) present a self-organizing feature map (SOFM) approach to incremental grid growing that allows for the expansion of a feature map to correct for errors in the compression of high dimensional feature space onto the two-dimensional topological grid.

A crucial insight incorporated in the various recruitment models is the idea that, by starting off with minimal computational resources, the learning system is forced to deal first with the most general patterns in the data. In effect, the system can only deal with the first of the various factors that can be extracted by principal components analysis (PCA). Once this first factor is learned, the network finds that there is still some residual error and it recruits new units to extract additional regularities. However, these new units are then largely dedicated to a second aspect of the problem. In this way, the network comes closer to modeling the type of stage-like learning we see in the child.

**Recruitment versus Deletion**

Models that rely on the recruitment of new neurons have been criticized on the grounds that they go against facts of developmental neurobiology. We know that new neurons are not added after birth. In fact, development is more characterized by neuronal loss than by neuronal addition. Some theorists have seized on this fact to argue that, like the immune system, neural development works by generating a vast array of potential cognitive structures which are then weeded out during development (Changueux & Danchin, 1976; Edelman, 1987; Jerne, 1967). Extending the analysis to issues in human development, Siegler (1989), Piatelli-Palmarini (1989), and Campbell (1960) have argued for the importance of “blind variation and selective retention” in creative thought and cognitive development.

Recent work calls into question some of the assumptions of this selectionist approach. Although it is true that there is a rapid loss of both cells and dendritic branches during the first months of life, the period of loss does not continue through development. Reassessing earlier claims about ongoing losses in synaptic density, Bourgeois, Goldman-Rakic, and Rakic (1994) have found ongoing synaptogenesis in prefrontal cortex during development. These findings match up with reports of increased volume of frontal cortex during development (Dekaban & Sadowsky, 1978; Jernigan et al., 1991) which indicate that brain development is not fundamentally selectionist and that additional resources may well be recruited during learning. This is not to say that the actual mechanisms supporting recruitment have yet been identified, only that models that use recruitment cannot be excluded on neurological grounds.

**CONNECTIONISM AND DEVELOPMENT**

We now turn to an examination of connectionist models of specific developmental processes. We will first look at models of the various component processes in language development. Second, we will examine general issues in cognitive development and their realization in connectionist models of specific cognitive tasks. Third, we will look at models of additional issues in development, including motor development and early brain maturation.

**Connectionism and Language Development**

The learning of language is a complex process that extends over the course of many years and which relies on interplay between several complex cognitive processes. Some recent accounts of language learning (Lightfoot, 1989; Pinker, 1994) tend to focus rather exclusively on the learning of grammatical markings and syntax, but language learning is more than just the learning of a few rules of grammar. In fact, the child devotes far more attention to tasks such as word learning, concept acquisition, articularatory control, discourse structuring, and conversational maintenance. No symbolic or connectionist model has been formulated that can handle all levels of language learning, although MacWhinney (1978, 1982, 1987a, 1988) and Pinker (1984)
offered initial sketches in the symbolic framework. In the next sections, we will look at current work in connectionist models with an eye to discerning the shape of a new, more detailed, synthetic approach.

**Word Learning**

Current research in sentence processing (Juliano & Tanenhaus, 1993; Trueswell & Tanenhaus, 1991, 1992) has stressed the importance of individual words as determiners of sentence-level processing. The central role of words in phonological development and auditory learning has been recognized for nearly two decades (Ferguson & Farwell, 1975). Even discourse processes and narrative structures are grounded on specific lexical constructions (Goldberg, 1995).

For symbolic models, lexical learning is a computationally trivial problem, since symbolic models have no trouble picking up arbitrary numbers of arbitrary associations. However, the symbolic view of word learning as mere association does not match up well with developmental data. We know that children can learn new words quickly on the basis of a single encounter (Carey, 1978; Dollaghan, Biber, & Campbell, 1995), but only a few new words can be picked up at the same time. If we present a child with dozens of new words at once, learning starts to fall apart. Moreover, the exact nature of the representation constructed by the child learner or the adult second language learner (Atkinson, 1975) is often heavily dependent on the context of presentation (Kay & Anglin, 1982).

Networks that use backpropagation and hidden units have exactly the opposite problem with modeling lexical learning. These networks typically cannot learn more than about 700 lexical forms. After this level, the hidden units are so fully invested in distinguishing phonological and semantic subtypes and their associations that there is simply no room for new words. Adding more hidden units doesn’t solve this problem, since all the interconnections must be computed and eventually the learning algorithm bogs down.

The problems faced by backpropagation are not general to all network models. Building on earlier models from Grossberg (1978, 1987), Houghton (1990), and Burgess and Hitch (1992), Gupta and MacWhinney (1996) have constructed a lexical learning model that uses three hierarchically-ordered layers, as illustrated in Figure 13.7. The lowest layer is a set of phonological units for the sounds of a word. Above this layer, is a set of phonological chunks representing the various syllable patterns of the language. On the top is a programmable level that controls the order of syllables within the word. Each new word is learned as a new node on the top layer and a series of weights on the connections from this layer to lower layers. Gupta and MacWhinney show that this model does a good job of accounting for a wide variety of well-researched phenomena in the literature on word learning, immediate serial recall, interference effects, and rehearsal in both adults and children (Gathercole & Baddeley, 1993).

A major limitation of the Gupta and MacWhinney model is its reliance on a rigid fixed level of items for the top-level word nodes. Once the initial number of nodes has been used up, the system would need to recruit a new node for each new word. However, simply inserting a fresh node into the model for each new word requires us to make excessively strong assumptions about neuronal plasticity, while also failing to capture the ways in which new words interact with old lexical structures. As we noted earlier in our discussion of the model of MacWhinney (1996), by relying on a fuller address space of the type proposed by Kanerva or Miikkulainen, problems with the learning of new lexical items can be minimized.

**Word Meaning**

The task of simulating the semantic aspects of word learning is extremely challenging, because of the open-ended nature of meaning. Connectionist models have found two
ways of dealing with this problem. One approach structures the world as a miniature perceptual system. This mini-world approach was developed first by Chauvin (1989) and echoed in a replication by Plunkett and Sinha (1992). The goal of the Chauvin model is to associate dot patterns with arbitrary labels. Learning is done using autoassociation. First the net is given an image and asked to activate a label, next it is given a label and asked to activate an image, and then it is given a trial with both image and label presented together. Plunkett and Sinha note that the cue validity of the label in their input corpus is higher than that of the image, since the label predicts a unique image, but an image does not predict a unique label. Unfortunately, it is difficult to see how a relation of this type can map onto the facts involved in real lexical structures where the opposite is usually the case.

Because the Chauvin/Plunkett-Sinha model uses backpropagation, it does not perform well in the basic lexical acquisition task. However, it does succeed in capturing some of the other phenomena associated with lexical learning. In particular, Plunkett and Sinha claim that their model captures these three phenomena:

1. **The prototype effect.** This effect replicates the original findings of Posner and Keele (1968; 1970) and has been observed in other connectionist models (McClelland & Rumelhart, 1985; Schyns, 1991), as well.

2. **The word learning spurt.** The model only starts producing labels after the first 20 epochs. At this point, Plunkett and Sinha view the onset of learning as similar to the “word spurt” that occurs in children. However, the cause of the delay in the model is the low validity of the image as a cue to the label and it is difficult to see how this structuring of lexical validities maps onto the real facts of lexical structures. Thus, it appears that the model is demonstrating the word learning spurt for the wrong reason.

3. **The superiority of comprehension over production.** Here, again, the reason for the superiority of comprehension is the higher cue validity of the label and it appears that the model is displaying the correct behavior for the wrong reasons.

Despite these limitations, the Chauvin/Plunkett and Sinha model serves as a useful starting point for thinking about comprehension-production relations in development (MacWhinney, 1990).

In another exploration within the mini-world framework, Schyns (1991) applied a Kohonen network to the task of learning three competing categories with prototype structures. The three categories were geometric patterns that were blurred by noise in order to create a prototype structure, although the actual prototypes were never displayed. The simulations showed that the network could acquire the patterns and demonstrate human-like categorization and naming behavior. When presented with a fourth new word that overlapped with one of the first three words, the system broke off some of the territory of the old referent to match up with the new name. Schyns interpreted this as evidence that the network was obeying the mutual exclusivity constraint of Markman (1989). However, the operation of his network can be understood even more clearly in terms of the forces of contrast and competition described by Clark (1987) and MacWhinney (1989).

An alternative approach to the development of word meaning focuses on the learning of small fields of real words. Three studies have been conducted to date. Shultz, Buckingham, and Oshima-Takane (1994) use the cascade-correlation algorithm to acquire the use of “you” and “me” from two types of input: child-directed speech and speech directed to a third party. The problem with learning the meaning of words like “you” and “me” is that the actual reference of the word is constantly changing. The best way to figure out the meanings of these words is to observe two other people using them. In this way, the child is able to see that “you” is used for the addressee and “me” for the speaker and that these words only have meaning in the context of the speaker-listener relation. Research by Oshima-Takane, Goodz, and Derevensky (in press) has shown that learning is faster when children are exposed to relatively more speech addressed to a third party, typically an older sibling, since this input makes the use of “you” clearer. By comparing input corpora with varying amounts of speech directed to a third party, Shultz et al. were able to model this effect.

Another study of meaning development by Li and MacWhinney (1996) used a standard backpropagation architecture to model the learning of reversion verbs that used the prefix “un-” as in untie or “dis-” as in disavow. The model succeeded in capturing the basic developmental stages reported by Bowerman (1982) and Clark, Carpenter, and Deutsch (1995) involving the production of errors such as unbreak or disbind. The network’s performance was based on its internalization of what Whorf (1938; 1941) called the “cryptotype” for the reversion which involved a “covering,
enclosing, and surface-attaching meaning” that is present in a word like untangle, but absent in a form such as unbreak. Whorf viewed this category as a prime example of the ways in which language reflects and possibly shapes thought.

The various models of learning word meaning we have discussed so far all treat meaning as if it were a fixed set of elements for any given word. However, nothing could be farther from the truth. Virtually every common word in our vocabulary has many alternative meanings and shades of meanings. In extreme cases, such as the verbs “put” and “run,” the dictionary may list up to 70 alternative meanings. Typically, the choice of one meaning over another is determined by the other words in the sentence. For example, when we say that “the ball rolled over the table,” we are thinking of the word over as meaning across. However, when we say that “Jim placed the snuffer over the candle,” we are thinking of over as meaning covering. These competitions between the various alternative readings of words like “over” were discussed from a general connectionist perspective by MacWhinney (1989). Subsequently, an implemented connectionist model of the learning of the meanings of “over” by a network was developed by Harris (1990; 1994b). The Harris model is capable of taking new input test sentences of the type “the pin rolled over the table” and deciding on the basis of past learning that the meaning involved is across, rather than covering or above. It does this only on the basis of the co-occurrence patterns of the words involved, rather than on information from their individual semantics. Thus, it learns that combinations like ball, roll, and table tend to activate across without regard to facts such as knowing that balls are round and can roll or knowing that tables are flat and that rolling involves movement.

Inflectional Morphology

One of the most active areas of connectionist modeling has been the study of the child’s learning of the ways in which words change when they are combined with grammatical markings such as suffixes or prefixes. These markings are called inflections and the system that governs the use of these inflections is called inflectional morphology. A simple case of inflectional learning is the system of patterns that help us choose to say “bent” instead of “bended” as the past tense of the verb bend. Inflectional learning is also involved in the learning of the correct form of the German definite article that we examined earlier. There are now well over thirty empirical studies and simulations investigating this topic from a connectionist perspective. The majority of work on this topic has examined the learning of English verb morphology with a particular focus on the English past tense. The goal of these models is the learning of irregular forms such as went or fell, along with regular past tense forms such as wanted and jumped. Other areas of interest include German noun declension, Dutch stress placement, and German participle formation. This work has examined six core issues:

1. Cues versus rules. The most central issue addressed in this research is whether or not one can model the learning of inflectional morphology without using formal rules. Pinker (1991) has argued that irregular forms are indeed produced by connectionist networks, but that regular forms are produced by a regular rule. However, Pinker’s attempts to preserve a role for rules in human cognition runs into problems with the fact that even the most regular patterns or “rules” display phonological conditioning and patterns of gradience (Bybee, 1953) of the type that are easily captured in a connectionist network.

2. Phonological representation. Most current models of inflectional learning use a system for phonological representation like the one introduced by MacWhinney, Leinbach, Taraban, and MacDonald (1989). This system assigns each node a status on each of three coding systems. The first coding system indicates the position of the node in the syllable, the second indicates the position of the syllable in the word, and the third represents the presence or absence of a phonetic distinctive feature. Because this representational system relies on standard linguistic concepts, it addresses most of the concerns expressed by Pinker and Prince (1988) with earlier connectionist models of inflectional learning. An elaborated version of this same representational system can be found in Gasser (1991, 1992). Gasser’s models emphasize the serial quality of morphological formations by relying on predictive recurrent networks. The system Gasser proposes uses three separate recurrent subnetworks for phonemic structure, syllabic structure, and metrical structure. On the top level, the three levels would be integrated in terms of lexical items. However, exactly how this integration of separate recurrent subnetworks should occur remains unclear, since Gasser never fully implemented his model.

3. U-shaped learning. A major shortcoming of nearly all connectionist models has been their inability to capture
the patterns of overgeneralization and recovery from overgeneralization known as u-shaped learning. Empirical work by Marcus et al. (1992) has shown that strong u-shaped learning patterns occur only for some verbs and only for some children. The models of MacWhinney and Leinbach (1991) and Plunkett and Marchman (1991) showed levels of u-shaped learning in rough conformity with the patterns observed by Marcus et al. Moreover, Plunkett and Marchman showed that u-shaped learning levels could be affected by changes in the type and token frequencies of irregular verbs in the input.

4. Rote learning of irregulars. Although models like MacWhinney and Leinbach or Plunkett and Marchman succeed in demonstrating some u-shaped learning, this success is at least in part misleading. In order to correctly model the child’s learning of inflectional morphology, models must go through a period of virtually error-free learning of irregulars, followed by a period of learning of the first irregulars accompanied by the first overregularizations (Marcus et al., 1992). No current model consistently displays all of these features in exactly the right combination. MacWhinney (1996) has argued that models that rely exclusively on backpropagation will never be able to display the correct combination of developmental patterns and that a two-process connectionist approach will be needed. The basic process is one that learns new inflectional formations, both regular and irregular, by rote as items in self-organizing feature maps. The secondary process is a backpropagation network that uses the information inherent in feature maps to extract secondary productive generalizations.

5. The role of semantic factors. The first attempts to model morphological learning focused exclusively on the use of phonological features as both input and output. However, it is clear that the formation of past tense forms must also involve semantic factors. In English, the use of semantic information is associated with the irregular patterns of inflection. The idea is that, since we cannot access “went” by combining “go” and “-ed,” it might be that we can access it directly by a semantic route. Of course, this idea is much like that underlying the dual-route theory. In German gender, the role of semantic information is much clearer. Köpcke and Zubin (Köpcke, 1994; Köpcke & Zubin, 1983, 1984; Zubin & Köpcke, 1981, 1986) have shown that a wide variety of both phonological and semantic factors are used in predicting the gender of German nouns and their plural. Some of the features involved include: alcoholic beverages, superordinates, inherent biological gender, gemstones, body parts, rivers inside Germany, and light versus heavy breezes. Simulations (Cottrell & Plunkett, 1991; Gupta & MacWhinney, 1992; MacWhinney, 1996) have integrated semantic and phonological information in various ways. However, a better understanding of the ways in which semantic factors interact during word formation will require a more extensive modeling of lexical items and semantic features.

6. Extensions of irregular patterns to new words. Extending earlier work by Bybee and Slobin (1982), Prasad and Pinker (1993) examined the abilities of native English speakers to form the past tense for nonsense words like plink, plup, or ploth. They found that, the further the word diverged from the standard phonotactic rules for English verbs, the more likely the subjects were to form the past tense by just attaching the regular “-ed” suffix. Ling and Marinov (1993) noted that the original verb-learning model developed by Rumelhart and McClelland (1987) failed to match these new empirical data, largely because of its tendency to overapply irregular patterns. To correct this problem, Ling and Marinov created a nonconnectionist symbolic pattern associator which did a better job modeling the Prasad and Pinker data. However, MacWhinney (1993) found that the network model of MacWhinney and Leinbach (1991) worked as well as Ling and Marinov’s symbolic model in terms of matching up to the Prasad and Pinker generalization data.

Phonology

In the area of speech processing, connectionist models have been developed primarily as ways of simulating aspects of adult word recognition. The recurrent backpropagation architecture has been used in word recognition models developed by Norris (1994), Waibel, Hanazawa, Hinton, Shikano, and Lang (1988), and Watrous, Shastri, and Waibel (1987). Recently, Markey (1994) has developed a realistic physical representation of the young child’s vocal apparatus and used it to model the development of phonetic and phonological skills. Markey’s model is able to capture some of the basic aspects of early phonological development. Hopefully, we will soon see additional models that will allow us to better understand how much of early phonological development is determined by the articulatory apparatus and how much by the structure of the words being learned.
Reading

Sejnowski and Rosenberg (1988) presented an entertaining demonstration of a system called NETalk that learned to read aloud. The system took as its input the orthographic representation of English words and was trained to produce computerized speech as its output. At first the network made only crude approximations to the sounds of the words and then moved phonologically closer and closer through training. In this regard, the network fails to actually capture the nature of early reading in the child where words are fully formed phonologically and the task is to extract enough cues to effect retrieval of the full word form (Simon, 1976; Simon & Simon, 1973). However, a positive aspect of the NETalk model is its ability to extract local graphemic linear dependencies that a beginning reader might use to derive the sound of a word.

A more complete picture of the development of early reading skills was provided in a backpropagation model developed by Seidenberg and McClelland (1989). An important quality of this model is that it emphasizes the ways in which both regular spellings such as hint or mint can be controlled by the same computational mechanism that also controls irregular spellings such as pint. In the traditional symbolic approach (Marshall & Newcombe, 1973), a distinction is made between rote storage for irregular forms and pattern-based storage for regular forms. This distinction motivates a dual-process or dual-route approach to reading. Seidenberg and McClelland show that one can model the learning and usage of both regulars and irregulars in a single model with a single set of processes.

The Seidenberg-McClelland model has been challenged on empirical grounds (Bub & Bub, 1992; Besner, Twilley, McCann, & Seergobin, 1990; Coltheart, Curtis, Atkins, & Haller, 1993). One problem with the model was its inability to acquire its own training set. However, by using a phonological input representation much like that developed by MacWhinney et al. (1989), Plaut, McClelland, Seidenberg, and Patterson (1995) were able to improve on the performance of the original model. A second problem with the model arose in connection with the modeling of data from neurological patients with deep dyslexia. For these patients, the model underestimated the sparing of high-frequency regular and irregular forms, as predicted by the dual-route model. Here, again, the revised coding system of Plaut et al. was able to improve on the performance of the original model.

In evaluating the status of the debate between single-route and dual-route accounts of reading and lexical processing, it is important to recognize that connectionist theory makes no specific commitment to the single-route concept. Moreover, it may be impossible to avoid some aspects of duality, even in the most homogeneous model. For example, Kawamoto (1993; Kawamoto & Zemblidge, 1992) has shown that subjects tend to produce incorrect pronunciations of irregulars more quickly than correct pronunciations. Thus, the pronunciation of pint to rhyme with hint is faster than the correct pronunciation of pint. Kawamoto models this effect using a ART-type model. At first, the large number of words with the regular “-int” shape activate a common pattern. If the subject produces a reading of the word at this time, it will be an error. A few milliseconds later, the slower connections to the irregular pronunciation start to dominate and the correct pronunciation will be produced. This is still a single mechanism, but the presence of two routes is simulated by contrasting pattern activations at different time points during the settling of the network.

Syntactic Classes

Psycholinguists working in the standard symbolic tradition (Chomsky, 1965; Fodor & Pylyshyn, 1988; Lachter & Bever, 1988) have pointed to the learning of syntax as a quintessential problem for connectionist approaches. One of the key abilities involved in the learning of syntax is the abstraction of syntactic classes or “parts of speech,” such as nouns, verbs, or prepositions. In the theory of universal grammar, these categories are innately given. However, their actual realization differs so much from language to language that it makes sense to explore accounts that induce these categories from the input data. Elman (1993) has presented a connectionist model that does just this. The model relies on a recurrent architecture of the type presented in Figure 13.7 above. The training set for the model consists of dozens of simple English sentences such as “The big dog chased the girl.” By examining the weight patterns on the hidden units in the fully trained model, Elman showed that the model was conducting implicit learning of the parts of speech. For example, after the word big in our example sentence, the model would be expecting to activate a noun. The model was also able to distinguish between subject and object relative structures, as in “the dog the cat chased ran” and “the dog that chased the cat ran.”

Even more interestingly, Elman found that the network only learned to pick up these positional expectations when
it began with a narrow perceptual window of two or three words. If the network started with too large a window, it could not focus on detection of the most basic determinants of syntactic positioning. Elman interpreted this contrast as underscoring the "importance of starting small." In many ways, this analysis is much like the one offered by Schultz for the importance of a learning algorithm that starts off with limited resources and only recruits new resources when it is unable to further reduce error.

**Lexical Segmentation and Masking**

In order to process sentences effectively, we need to be able to segment out words from the ongoing speech stream. Norris (1994) proposes a system called ShortList which uses a recurrent net of the type given in Figure 13.5 to process incoming phonemes left-to-right. A network of this type does fine with many words. However, it has trouble with words like catalog which have what Norris calls a "right context" problem. When processing the word catalog, a simple recurrent net would recognize the word cat and decide that this word had actually occurred, if it were not somehow forced to hold off and process further right context. In order to prevent this from happening, Norris suggests that there must be a short list of competitors that include words like cat, cattle, catalog, and others like them that will compete for full recognition of the input material. The ShortList implementation of this process uses a hand-wired word list. However, Miikkulainen (1993) has suggested that it would be possible to model this same process using self-organizing feature maps.

Once a word has been successfully detected, the sounds that activated it need to be masked out, in order to block multiple recognition of the same input by alternative competitors. Take a sentence like "I gave my cat a Miranda doll." Once the word cat has been selected, its component phonemes are "masked" in order to avoid the additional activation of the form "catamaran" on the basis of the string "cat a Miran." Once a word is fully recognized and its component sounds are masked, it must then begin to participate in higher level syntactic and semantic patterns. The exact nature of this conversion is not yet clear. There have been several suggestions regarding the nature of this short-term verbal memory:

1. As soon as words are linked together into conceptual clusters, they can be used to activate a unique underlying meaning that no longer requires verbal storage.
2. Before this linkage occurs, words may be retained in a phonological loop (Baddeley, 1986). This immediate rehearsal requires that words be present in a primarily articulatory form (Gupta & MacWhinney, 1994).
3. It is also possible that some additional mechanism operates on lexical items to encode their serial occurrence without reference to either meaning or sound. This could be done in terms of some additional episodic, possibly hippocampal, mechanism that stores activation levels of words prior to masking. A system of this type is close to the Competitive Queuing mechanism proposed first by Grossberg and then again by Houghton.

Further experimental work will be needed to decide which of these three mechanisms is involved at which points in the storage of short term verbal memories. However, there is already good (Gupta & MacWhinney, 1996) evidence that various neural mechanisms are available to support masking in the lexicon.

**Bilingualism**

The study of bilingualism and adult second language learning is a particularly promising and challenging area for connectionist research. Recent research in second language acquisition (Dechert & Raupach, 1989; Fliege & Davidian, 1984; Hancin-Bhatt, 1994; Harrington, 1987; Johnson, 1989; MacWhinney, 1992; Omlin, 1989; Sasaki, 1994) has underscored the importance of transfer of first language skills to the learning of the second language. Because of its emphasis on pattern generalization, the backpropagation algorithm is well-suited to modeling transfer effects. In one of the first simulations designed to examine these issues, Gasser (1990) constructed an auto-associative network that used backpropagation training for the learning of basic word orders in second language learning. In one simulation, the network was first trained with a first language order of Subject-Verb (SV) and then exposed to a set of second language sentences with Verb-Subject (VS) order. In the other "mirror-image" simulation, the network began with VS in the first language and then shifted to SV in the second language. The network demonstrated a strong tendency to transfer the first language word order to the second language, particularly for words that were similar semantically. This type of lexically-based transfer for word order is exactly what one would expect for a strong pattern generalizing network. However, there is not yet any actual empirical data that would support the importance of this effect in real second language learning.

MacWhinney (1996) reports on unpublished work by Janice Johnson that adapts the architecture of the recurrent network shown in Figure 13.5 to the problem of second
language learning. The exact shape of the model is given again as Figure 13.8.

In these simulations, the input in pool C is a pattern that represents the status of the "current word" along the dimensions of animacy, number, case, agreement-marking, part-of-speech, and language. We assume that this information is available through the individual lexical item. Note that this highly structured form of the input differs radically from the raw word level input used by Elman (1993). Because of the highly structured shape of the input, this network performs much better than the Elman net as a sentence processor and interpreter. The task of the network is to assign the agent, object, and perspective roles to the correct words. In order to get these assignments right, the network must activate the correct output units in pool A. For example, the network can choose between activating a node that assigns the first noun as agent and a competing node that assigns the second noun as agent. Training involves the presentation to the network of sentences, one word at a time. For example, the input could be "the dog is chased by the cat." In this case, the network might begin by thinking that "the dog" is the agent. However, once the passive form of the verb is detected, the weight on this role assignment is decreased and the second noun is selected as agent instead. When the network is processing passive sentences, we find that it goes through an on-line reversal or "garden-path," first activating a choice of the first noun as agent and then reversing this activation to choose the second noun as the agent.

The network was trained initially with a wide variety of English sentence patterns. At the end of this initial training, it was performing well in the role assignment task for English. Then, the input was extended to include a full corpus of parallel sentences for Dutch. After a period of mixed training, the network then continued with Dutch-only training for a further period. The first basic finding of this research was that the exact weights of the various cues in the model matched up well with a large body of empirical research summarized in MacWhinney and Bates (1989). In particular, the model learned the basic English SVO pattern quickly and then continued to learn the VOS and OSV patterns found in adult speakers. The second finding was that, when learning Dutch, the model showed exactly the type of word order transfer effects reported by McDonald (1987, 1989) for the learning of Dutch by English speakers and the learning of English by Dutch speakers. Finally, the model also showed a clear tendency toward “catastrophic interference” if the period of mixed-language training was omitted. A more robust, general approach to the catastrophic interference problem in this network and others like it could be developed if the network were given a firmer grounding on the learning of syntactic patterns on the basis of generalization from particular lexical items, as we noted earlier.

Connectionism and Developmental Theory

Connectionism offers a fresh perspective on a variety of issues of ongoing concern to developmentalists, including the emergence of symbols and representations, the movement between developmental stages, and the role of nonlinearities in development.

Stages and Transitions

The simplest developing system is one that shows only one type of uniform change over time. For example, a falling ball undergoes only one type of transition during its downward movement. We can use Galileo's equation for acceleration to compute the distance traveled as a function of acceleration $a$ and time $t$. For this simple, uniform system, we have a clear rule that allows us to predict the state of the system at each time $t$.

More complex systems can go through a series of stages during state transitions. For example, a drop of rain can begin as cloud vapor, form into a droplet, freeze into hail, fall to the ground, and then melt into slush. Each of these state transitions delimit specific stages in the life of the droplet. In the human child, stages of this sort abound. For example, after learning the first word, children spend several months slowly picking up a few additional words. Then, suddenly, we see a rapid growth in vocabulary that has been called the vocabulary "burst." The vocabulary
burst does not emerge overnight, but builds over the course of several weeks. However, if we plot the size of the vocabulary on the y-axis and the child's age on the x-axis we will note a marked upward acceleration at the beginning of this period. Such changes indicate a stage-like quality in development.

Piaget has characterized the intellectual growth of the child in terms of four major epochs, each composed of several periods with some further divisions of the periods into subperiods. However, Piaget's characterization of these stages as invariant properties of human development is no longer widely accepted and few researchers are interested in developing simulations to account for the child's movement through the classical set of Piagetian stages. This is not to deny the reality of major qualitative changes in cognition as the child moves from infancy to adolescence. However, attempts to capture these changes across skill domains have not been successful. Because of this, connectionist models of stagelike transitions have tended to focus not on broad changes in cognition, but on local discontinuities within the development of specific skills. The areas that have been most closely investigated are the balance beam, velocity computation, and seriation.

**Balance Beam**

One of the clearest analyses of stage transitions in cognitive development is the case of the balance beam problem studied first by Piaget. Earlier we examined the production-system accounts for learning of the balance beam problem by Klahr and Siegler (1978). McClelland (1989, 1995) has noted that, although these production-system models provide a good description of the four balance beam rules discussed earlier, they tell us little about the forces that drive the child from one rule system to the next.

McClelland was able to construct a backpropagation model of the balance beam problem that used 20 input units. There were 10 positional units devoted to the 5 positions to the left of the fulcrum and the 5 positions to the right of the fulcrum. The 10 weight units were dedicated to represent the numbers of weights stacked up at a position with 5 units for the possible number of weights on the left and 5 units for the possible weights on the right. A given problem could be encoded with a total of 4 units turned on. For example, take a problem with 4 weights at a distance of 3 right and 5 weights at a distance of 2 left. The units turned on would then be 4-right-weight, 5-left-weight, 3-right-distance, and 2-left-distance. In order to bias the network toward reliance on the weight cue over the distance cue, McClelland included a large number of cases in which the distance cue was neutralized, thereby focusing the network's attention to the weight cue.

Using this type of representation, McClelland was able to model many aspects of the learning of this task. The network began with performance that relied on Rule 1 and moved on to learn Rule 2 and then Rule 3. It never acquired full use of Rule 4, but, McClelland argues, this is because some aspects of the use of Rule 4 in adults involve the application of full mathematical analysis. However, the network was able to capture aspects of the rather subtle "torque distance effect" detected in studies by Ferretti and Butterfield (1986) and Wilkening and Anderson (1982). These studies have shown that subjects perform best and most consistently on balance beam problems when it is clear perceptually that one side has an overwhelming combination of weight and distance in its favor. When the balance between the two sides is closer numerically, decisions are less consistent. Torque distance effects indicate that subjects are not simply applying an all-or-none rule, but are performing a type of cue-weighting that is much like that conducted inside a neural network.

Shultz, Schmidt, Buckingham, and Mareschal (1995) extended McClelland's model by using the cascade correlation variation of the backpropagation algorithm. Shultz et al. argue that static backpropagation networks with only a few hidden units can succeed at modeling the first stages of development, but are unable to reach higher levels of performance, because their weights become too closely tuned to solving the basic levels of the problem. This was true for McClelland's balance beam model, which learned Rules 1, 2, and aspects of 3, but was unable to learn Rule 4. However, using the cascade correlation framework, Shultz et al. were able to model successful learning of all four rules.

These models make two important points. First, both the McClelland and the Shultz et al. models show that connectionist models can provide good accounts of perceptual aspects of learning such as the torque distance effect. Second, Shultz's model shows that static networks that begin life with abundant numbers of extra hidden units may fail to perform the type of architectural decomposition of a problem space that is required for successful mastery. Models that start out small and are forced to recruit new units when they run out of steam are more likely to be able to focus first on the core of a problem and then add the details as elaborations of this core.

**Other Physical Coordinations**

In addition to their work on the balance beam problem, Shultz and his colleagues have developed connectionist...
models for three other types of physical coordinations. These include the learning of seriation, potency-resistance, and velocity-distance-time relations. Mareschal and Shultz (1993) developed a model of seriation that attempts to simulate the developmental progression reported by Piaget (1965). The model's task is to order a series of six sticks, each with a different length, so that the shortest is on the left and the longest is on the right. This is done by placing one stick in position at a time. The network is composed of two independent modules—a "which" module and a "where" module (Jacobs, Jordan & Barto, 1991). The "which" module is given the task of deciding which stick to move at a given point in the problem. The "where" module is given the task of deciding where to position each stick in terms of one of six possible spatial positions.

The results for these simulations of seriation learning match up closely with the empirical findings reported by Piaget. In stage 1, performance is close to chance. In stage 2, the network forms pairs and triplets that are correctly ordered, but the whole array is not correct. In stage 3, the whole array is ordered, but through largely trial-and-error repetition of subgroup ordering. In stage 4, seriation is performed correctly with previous analysis.

Buckingham and Shultz (1994) developed a model of the learning of the relations inherent in the physical relations expressed by these equations: \( d = vt \), \( v = d/t \), and \( t = d/v \). These equations relate distance, velocity, and time through multiplicative relations. Wilkening (1981) found that children tend to progress through three levels of information integration in learning these relations. First, they relate each quantity only to itself. Second, they take into account the effect of the other two determining variables, but employ subtraction or addition instead of the correct division or multiplication rules. Third, they acquire the correct division or multiplication rules. Buckingham and Shultz (1994) were able to capture this three-stage developmental sequence in a neural network model. As in the other simulations reported by Shultz et al. (1995), the movement through these stages was facilitated by use of the cascade-correlation algorithm which tends to force simple solutions at early periods, but allows for the recruitment of additional resources to solve problems in more complex ways later on. In order to reach the more extreme values required by the multiplicative rule, weights have to first move through a set of values that match the additive rule. As additional units are recruited, these weights move closer to approximating a multiplicative relation.

Finally, Shultz and his colleagues have also studied the learning of resistance-potency relations. When a force with a given potency goes directly against a force with a certain resistance, the resultant force is computed by subtracting the two vectors. However, when a ramp is included in the physical system, the sum of the two vectors is computed by division, rather than subtraction. Shultz et al. (1995) were able to simulate the learning of both types of computations and showed that the subtractive relations were learned earlier than the division relations. Again, these effects seem to emerge from basic facts about the process of weight changes in neural networks.

**Attachment**

Van Geert (1991) developed a dynamic systems model designed to model growth curve developments in both vocabulary acquisition and the formation of attachment relations. One particularly interesting aspect of his model is the analysis he provides for the interaction between two competing developmental strategies. Van Geert shows how a variety of growth curves can arise from the competition and that the shapes of these curves depend on the internal stability of the two separate processes. In a system with optimally sensitive parenting, attachment grows steadily over time to reach a ceiling level. In a system with insensitive parenting, attachment grows weakly to reach a lower, but steady state. In a system with inconsistently sensitive parenting, the resulting attachment behavior of the child is extremely variable and unstable. These patterns of growth match up well with empirical data on the development of attachment under conditions of consistent and inconsistent parenting (Ainsworth, Blehar, Waters, & Wall, 1978; Belsky, Rovine, & Taylor, 1984).

**Connectionism and Brain Development**

Connectionist theory is extremely rich in terms of its implications for brain development. The first major area for which connectionism is relevant is brain development during embryogenesis. Here, connectionist models suggest that the commitment and inductance of particular neural areas to particular functions is driven by connections between areas and sensorimotor functions. The idea is that the shape of the brain emerges under the real physical constraints of the sensory and motor systems to which it is linked, rather than out of response to some abstract genetic blueprint for a set of disembodied innate ideas. To consider an example of how this works, consider the development of columns in the visual cortex (Hubel & Weisel, 1963). Miller, Keller, and Stryker (1989) have formulated network models that show how this columnar organization can arise.
from competitive interactions between signals from the two eyes. In general, it may be true that patterns of connectivity in the brain arise from the competition between signals arriving from different sensory systems and signals being sent to motor processes (Walsh & Cepko, 1992, 1993).

Connectionism may also help us to understand some of the mysteries of brain development during infancy and early childhood. Work on children with perinatal brain lesions (Aram & Eisele, 1992; Dennis, 1980; Feldman, 1993; Feldman, Janosky, Scher, & Wareham, 1994; Thal et al., 1991) has demonstrated the remarkable ability of the young brain to acquire normal language functioning after even the most severe early lesions. How the brain reorganizes to achieve this dynamic response is one of the great challenges facing developmental psychology and it is one in which connectionist modeling can play an important role.

Recent constructivist accounts of brain development (Montague & Sejnowski, 1994; Quartz & Sejnowski, 1995) point out some possible mechanisms for changes in brain function, even after major damage. These models note that the continual refinement of patterns of connectivity is driven by local mechanisms, including dendritic growth, synaptogenesis, myelination, and changes in membrane potential. In a constructivist model of the brain new synaptic connections are viewed as emerging through the action of nitrous oxide. When a cell fires, it broadcasts nitrous oxide to nearby cells and encourages the development of projections in the direction of the gradient of diffusion. A mechanism of this type fits in well with ideas about topological organization which we discussed earlier such as the self-organizing feature map models of Kohonen and Miikkulainen or the sparse distributed memory of Kanerva. However, a full account of reorganization after early brain damage may require more than just the local reorganization offered by these models.

The various connectionist models described in this section represent only a first step toward resolving some of the enduring issues in cognitive development. Bechtel and Abrahamsen (1991) outline the further potential of such models, including: (a) a new interpretation of the distinction between maturation and learning; (b) a computational instantiation of the distinction between accommodation and assimilation; (c) an account of context effects (in which minor task variations have large effects on preschooler’s performance [Gelman, 1978]); and (d) explanations of many of the phenomena and anomalies associated with stages and transitions.

**FUTURE DIRECTIONS**

Having presented a description of the two principle approaches to computational modeling of cognitive development, we close with a discussion of their similarities, differences, and current inadequacies. Three themes run through this final discussion. One is that the two approaches are not as distinct as their practitioners often claim. The second is that—for all of their accomplishments—both approaches must solve some very difficult remaining problems. The third theme is that such challenges can only be met by infusing computational techniques into the training of the next generation of cognitive developmentalists.

**Comparing Connectionist and Production-System Models**

Although connectionist forays into cognitive development are often accompanied by the dismissal of “symbolic” approaches as unsuited to the task, we wonder whether the differences are as substantial as are sometimes claimed. Connectionist models are usually proposed as radically different from production-system architectures, and more neurally plausible. However, one can ask where the fundamental differences lie: in the parallelism of the processing, in the distributed knowledge, or in the connectivity of that knowledge?

1. Parallelism can not be the source of the difference, because during the “match” or “recognize” phase of a production system’s recognize-act cycle, the condition side of all productions are matched in parallel with all the elements in working memory. In some systems, working memory is defined as the set of elements in a very semantic memory that are above some threshold, so the match process is massively parallel and the connectivity between working memory elements and the production is dynamic and potentially unbounded.

2. What about distributed knowledge? The extent to which knowledge is distributed or modularized in a production system depends entirely upon the grain size that elements or productions are supposed to capture. Thus, the actual implementation of this parallel match occurs in a serial Von Neumann machine. But so too do the implementations of the learning algorithms in PDP models. This microlevel of implementation is not regarded as part of either theoretical stance.
single production might represent a very explicit and verbalizable rule; it might represent a small piece of processing for a complex, implicit piece of knowledge; or it might represent a complex pattern of cue associations much like those found in connectionist models (Ling & Marinov, 1993). Similarly, in PDP models, the individual element can represent knowledge at any grain size: from an individual neuron, to an assembly of neurons, to the word "neuron." There is nothing inherent in either formulation that specifies what this grain should be, until additional constraints are imposed on the model. Such constraints might include attempting to match production-system cycles to human reaction times, or the connectivity of connectionist models to neural connectivity.

3. Another purported difference between PDP models and production-system models is the gradualism of the former and the abruptness of the latter. But as evidenced by some of the models described earlier, one can create a production-system architecture with continuously varying strengths of productions—hence production systems can exhibit gradualism. Conversely, the higher order derivatives of different learning functions in connectionist systems can assume large values. Given the appropriate grain size on a performance window, such models would appear to be undergoing discontinuous changes (cf. Newell's 1973 classic analysis of process-structure distinctions in developmental psychology).

These many points of similarity have also been noted by advocates of the connectionist approach. Bechtel and Abrahamsen summarize some of these areas of potential overlap and rapprochement:

Most of the modifications incorporated in the most recent symbolic models have narrowed the gap between symbolic and network models. . . . 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 (Holland, 1975; Koza, 1992), knowledge compilation and "chunking" of rules into larger units (Anderson, 1983; Newell, 1990), and ways of applying old knowledge to new problems, such as (Falkenheimer, Forbus, & Gentner, 1989).

. . . 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. (Bechtel & Abrahamsen, 1991, pp. 18–19)

**Problems Facing Computational Models**

**Scalability**

To date, both symbolic and subsymbolic models of cognitive development have focused on highly circumscribed domains, and within those domains, on small scale exemplars of the domain. For all of the work on connectionist models of language, no one has yet been able to construct a complete connectionist model of language acquisition. For example, developmental neural networks are often constrained to well-defined topics such as the acquisition of the English past tense (Cottrell & Plunkett, 1991), or learning German gender (MacWhinney et al., 1989). The toy model approach often reduces large problems such as question answering (St. John, 1992) or word sense disambiguation (Harris, 1994a) to small problems by using only a few dozen sentences or words in the input corpus. In fact, there is not even a reasonably complete account for smaller skill domains such as word learning or syntactic development. For all of the work on Piagetian and other types of problem solving, no one has constructed a production system or a neural net that performs the full range of tasks encountered by a normal 5-year-old child. In essence, all of the work so far has been on toy versions of larger domains.

Computational modelers argue, either explicitly or implicitly, that in principle, such models could be expanded substantially with no major theoretical modifications. But could they? Here, the plausibility of the claim varies according to the approach, with the symbolic models having the better track record. Although there are no large scale developmental production systems, there do exist several
very large production systems that start with a few hundred initial "hand-coded" productions and go on to learn over 100,000 productions. Domains include both AI-type tasks and cognitive models. (See Doorenbos, 1995 for a review and evaluation of several such large-scale production systems.)

With respect to scaling up connectionist systems, there are grounds for skepticism. For example, in the language learning domain, when one attempts to add additional words or sentences to many of the connectionist language models, their performance begins to degenerate. One of the major challenges for computational models then, is a direct attack on this scalability problem.

Ad Hoc Assumptions about the Environment

Another problem facing both connectionist and production-system models is the lack of a principled, data-constrained theory of the effective environment in which such models operate. For many models, the "training" to which they are exposed is based on arbitrary, unprincipled, ecologically ungrounded assumptions about the environmental inputs that the child receives. Until we have better ways of measuring the actual properties of patterns in the effective environment, we cannot really claim that our models are being properly constrained by real empirical data.

Fortunately, there are two promising research avenues that may soon begin to alleviate this problem. The first avenue is the development of rich computerized databases. In the area of language development the Child Language Data Exchange System (CHILDES) database (MacWhinney, 1995) has collected transcript data from dozens of major empirical projects. These transcripts contain both the language input to the child and the child's developing conversational competence. More recently, these data are being supplemented by digitized audio and video records that give researchers access to the full richness of the original interactions. Because this database is computerized according to a standardized format, it is possible to use a wide variety of computer programs for search and analysis of patterns in both the input and the child's productions. Increasingly, simulations of language learning are being based on properties of the input as computed from the CHILDES database and similar computerized sources.

A second promising development is the growth of microgenetic studies. This research is designed to capture developmental processes as they occur by looking at fine-grained moment-to-moment changes in cognition and behavior. Kuhn (1995) has applied microgenetic techniques to the study of scientific reasoning, and Siegler and Crowley (1991) and Alibali (1993) have applied this methodology to the study of strategy development in mathematics. However, the technique can be used equally well with basic behaviors such as walking (Adolph, 1995) or reaching (Thelen & Smith, 1994). Because microgenetic methods have such a fine-grained level of analysis, they collect quantities of data that are rich enough to support interesting tests of connectionist (MacWhinney & Leinbach, 1991), symbolic (Marcus et al., 1992), and dynamic systems (van der Maas & Molenaar, 1992) approaches to cognitive development.

Hybrid Models

By now the reader has come to appreciate the degree to which connectionist models focus on low-level cognition, leaving the more complex aspects of cognitive performance to full symbolic models. There are not yet connectionist models of processes such as the learning of double-digit addition, gaining expertise in solving the Tower of Hanoi, or solving cryptarithmetic problems. Is it possible that neural networks are only appropriate as models of perception and low-level aspects of language and cognition? If so, it would make sense to graft together models that use neural networks for low-level tasks and production systems for high-level tasks.

There are reasons to believe that it would be premature to explore the construction of hybrid models of this type. Before we start building Centaurs and mermaids, we should complete our exploration of more complex, multi-componenental neural network models. By linking up systems for arbitrary pattern association such as SOFM or SDM with other modules that use backpropagation or ART to extract regularities and patterns, we can increase the power of our models, while retaining the connectionist framework. When we look at the complex architecture of processing types implemented in brain structures such as the hippocampus, thalamus, and cerebellum, we realize that neuronally plausible connectionist models of tomorrow will make the simple backpropagation models of today seem primitive indeed.

Once this basic exploration of complex connectionist architectures has been completed, it may be propitious to examine the ways in which connectionist models implement algorithms developed in symbolic models such as SOAR, IBL, or ACT-R. A detailed example of close computational equivalence between a low-level symbolic model and
structured connectionist model can be found in the dialog between Ling and Marinov (1993) and MacWhinney (1993).

Why Compute?

Why should someone interested in cognitive development be concerned about computational models of the sort described in this chapter? The primary justification for focusing on such systems is the claim that self-modification is the central question for cognitive developmental theory. We are convinced that in order to make major theoretical advances, it will be necessary to formulate computational models at least as complex as the systems described here.

As we noted previously, early commentators on computational models often faulted them for being insufficiently attentive to the issue of self-modification. Such criticism strikes us as misplaced and ironic. While it is easy to find developmentalists who fault computational models, it is even easier to find criticisms of the entire field of developmental psychology for its inability to deal adequately with transition and change:

I have asked some of my developmental friends where the issue stands on transitional mechanisms. Mostly, they say that developmental psychologists don’t have good answers. Moreover, they haven’t had the answer for so long now that they don’t very often ask the question anymore—not daily, in terms of their research. (Newell, 1990, p. 462)

Is this too harsh a judgment? Perhaps we can dismiss it as based on hearsay, for Newell himself was not a developmental psychologist. But Newell’s comments simply echoed an earlier assessment from one of the central figures in the field:

... serious theorizing about basic mechanisms of cognitive growth has actually never been a popular pastime, ... It is rare indeed to encounter a substantive treatment of the problem in the annual flood of articles, chapters, and books on cognitive development. The reason is not hard to find: Good theorizing about mechanisms is very, very hard to do. (Flavell, 1984, p. 189)

Even more critical is the following observation on the state of theory in perceptual development from one of the area’s major contributors in recent years:

Put simply, our models of developmental mechanisms are disappointingly vague. This observation is rather embarrassing because the aspect of perceptual developmental psychology that should set it apart from the rest of perceptual psychology is the explanation of how development occurs, and such an explanation is precisely what is lacking. (Banks, 1987, p. 342)

It is difficult to deny either Newell’s or Bank’s assertions that we don’t have good answers, or Flavell’s assessment of the difficulty of the question. However, the good news is that the question is no longer being avoided: many developmentalists have been at least asking the right questions recently. In the past decade or so, we have seen Sternberg’s (1984) edited volume Mechanisms of Cognitive Development, MacWhinney’s (1987b) edited volume Mechanisms of Language Acquisition, and Siegler’s (1989) Annual Review chapter devoted to transition mechanisms. So the question is being asked.

And the answers are, increasingly, coming in the form of computational models. Only a few of the chapters in the 1984 Sternberg volume specify mechanisms any more precisely than at the flow-chart level, and most of the proposed “mechanisms” are at the soft end of the information-processing spectrum. However, only five years later, Siegler (1989) in characterizing several general categories for transition mechanisms (neural mechanisms, associative competition, encoding, analogy, and strategy choice) was able to point to computationally-based exemplars for all but the neural mechanisms (Bakker & Halford, 1988; Falkenheiner et al., 1989; Holland, 1986; MacWhinney, 1987a; Rumelhart & McClelland, 1986; Siegler, 1988). The recent Simon and Halford (1995) book, consisting entirely of computational models of developmental processes, provides a clear indication of this trend toward “hardening the core” (Klahr, 1992).

The advantage of such computational models is that they force difficult questions into the foreground, where they can be neither sidetracked by the wealth of experimental results nor obscured by vague characterizations of the various “essences” of cognitive development. The relative lack of progress in theory development—noticed by Banks, Flavell, and Newell—is a consequence of the fact that, until recently, most developmental psychologists have avoided moving to computationally-based theories, attempting instead to attack the profoundly difficult question of self-modification with inadequate tools. In contrast, computational models render these issues into a form sufficiently specific that it is possible to assess
Theoretical progress (see Mareschal & Shultz, 1996, for a cogent example).

The Future of Computational Models of Cognitive Development

That brings us to our final topic: The education of future cognitive developmentalists. As this book goes to press, the conceptual and technical skills necessary for computational modeling of developmental phenomena are taught in only a handful of graduate programs. However, we see the current situation as analogous to earlier challenges to the technical content of graduate training. When other kinds of computational technology that are now in common use—such as statistical packages—were first being applied to psychological topics, journal articles invariably included several pages of description about the technique itself. Writers of those early articles correctly assumed that their readers needed such background information before the psychological issue of interest could be addressed. Today, writers of papers using analysis of variance, or path analysis, or logistic regression simply assume that their readers have had several courses in graduate school learning the fundamentals.

Similarly, in the early years of computer simulation, the necessary resources of large “main frame” computers were limited to very few research centers, and exposure to computational modeling was inaccessible to most developmentalists. Even today, very few developmental psychologists have had any training with computational models, and only a handful of computational modelers have a primary interest in cognitive development. Nevertheless, as evidenced by the work described in this chapter, the intersection of these two areas of research is growing. Moreover, with the increasing availability of powerful workstations, the proliferation of computer networks for dissemination of computational models, the increasing number of published reports on various kinds of computationally-based cognitive architectures, the appropriate technology and support structures—such as summer workshops—are becoming widely accessible. All of these activities will increase the pool of appropriately trained developmentalists.

Even then, mastery of these new tools for computational modeling will not be easy. Nevertheless it appears to be a necessary condition for advancing our understanding of cognitive development. As Flavell and Wohlwill (1969) noted nearly thirty years ago: “Simple models will just not do for developmental psychology.”

Acknowledgments

We thank Robert Siegler and Chris Schunn for comments on earlier versions.

References


