Metaphors play a far more important role in science than many people realize. We are not only fascinated when we discover resemblances between phenomena that come from wildly different domains (atoms and solar systems, for example); these similarities often shape the way we think. Metaphors both extend but also limit our imagination.

Until recently, the metaphor that dominated the way we thought about the human brain was the digital computer. This is no coincidence: During the early days of what we now call computer science, in the 1940s and 1950s, engineers and mathematicians were very impressed by work by neuroscientists that suggested that the basic processing elements of the brain—neurons—were nothing more than binary on/off units. The first computers were actually designed to mimic with vacuum tubes what neuroscientists thought brains were doing. Thus, the metaphor of the brain-as-computer actually started the other way around: the computer-as-brain.

This metaphor has had an enormous impact in the theories that people have developed about many aspects of human cognition. Cognitive processes were assumed to be carried out by discrete operations that were executed in serial order. Memory was seen as distinct from the mechanisms that operated on it. And most importantly, processing was thought of in terms of symbolic rules of the sort that one finds in computer programming languages. These assumptions underlay almost all of the
important cognitive theories up through the 1970s, and continue to be highly influential today.

But as research within this framework progressed, the advances also revealed shortcomings. By the late 1970’s, a number of people interested in human cognition began to take a closer look at some of basic assumptions of the current theories. In particular, some people began to worry that the differences between digital computers and human brains might be more important than hitherto recognized. In part, this change reflected a more detailed and accurate understanding about the way brains work. For example, it is now recognized that the frequency with which a neuron fires—an essentially analog variable—is more important than the single on/off (or digital) pulse from which spike trains are formed. But the dissatisfaction with the brain-as-computer metaphor was equally rooted in empirical failures of the digitally based models to account for complex human behavior.

In 1981, Geoff Hinton and Jim Anderson put together a collection of papers (Parallel Associative Models of Associative Memory) that presented an alternative computational framework for understanding cognitive processes. This collection marked a sort of watershed. Brain-style approaches were hardly new. Psychologists such as Donald Hebb, Frank Rosenblatt, and Oliver Selfridge in the late 1940’s and 1950’s, mathematicians such as Jack Cowan in the 1960’s, and computer scientists such as Teuvo Kohonen in the 1970’s (to name but a small number of influential researchers) had made important advances in brain-style computation. But it was not until the early 1980’s that connectionist approaches made significant forays into mainstream cognitive psychology. Then, in 1981, David Rumelhart and Jay McClelland published a paper that described a
model of how people people read words. The model did not look at all like the traditional computer-based theories. Instead, it looked much more like a network of neurons. This paper had a dramatic impact on psychologists and linguists. Not only did it present a compelling and comprehensive account of a large body of empirical data, but laid out a conceptual framework for thinking about a number of problems which had seemed not to find ready explanation in the Human Information Processing approach. The publication, in 1986, a two-volume collection edited by Rumelhart and McClelland, and the PDP Research Group, called *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, served to consolidate and flesh out many details of the new approach (variously called PDP, neural networks, or connectionism).

This approach has stimulated a radical re-evaluation of many basic assumptions throughout cognitive science. One of the domains in which the impact has been particularly dramatic—and highly controversial—is in the study of language acquisition. Language is, after all, one of the quintessentially human characteristics. Figuring out just how it is that children learn language has to be one of the most challenging questions in cognitive science. But before turning to some of these new connectionist accounts of language acquisition, which is the main subject of this chapter, let us briefly define what we mean by connectionism.

**What is connectionism?**

The class of models that fall under the connectionist umbrella is large and diverse. But almost all models share certain characteristics.

Processing is carried out by a (usually large) number of (usually very simple) processing elements. These elements, called nodes or units, have a dynamics that is
roughly analogous to simple neurons. Each node receives input (which may be excitatory or inhibitory) from some number of other nodes, responds to that input according to a simple activation function, and in turn excites or inhibits other nodes to which it is connected. Details vary across models, but most adhere to this general scheme. One connectionist network is shown in Figure 1. This network is designed to take visual input in the form of letters, and then to recognize words—that is, to read.

There are several key characteristics that are important to the way these networks operate. First, the response (or activation) function of the units is often nonlinear. This means that the units may be particularly sensitive under certain circumstances but relatively insensitive under others. This nonlinearity has very important consequences for processing. Among other things, networks can sometimes operate in a discrete, binary-like manner, yielding crisp categorical behavior. In other circumstances, the system is capable of graded, continuous responses.

Second, what the system “knows” is, to a large extent captured by the pattern of connections—who talks to whom—as well as the weights associated with each connection (weights serve as multipliers).

Third, rather than using symbolic representations, the vocabulary of connectionist systems consists of patterns of activations across different units. For example, to present a word as a stimulus to a network, we would represent it as a pattern of activations across a set of input units. The exact choice of representation might vary dramatically. At one extreme, a word could be represented by a single, dedicated input unit (thus acting very much like an atomic symbol). At the other extreme, the entire ensemble of input units
might participate in the representation, with different words having different patterns of activation across a shared set of units.

Given the importance of the weighted connections in these models, a key question is, What determines the values of these weights? Put in more traditional terms, Who programs the networks? In early models, the connectivity was laid out by hand, and this remains the case for what are sometimes called “structured” connectionist models. However, one of the exciting developments that has made connectionism so attractive to many was the development of algorithms by which the weights on the connections could be learned. In other words, the networks could learn the values for the weights on their own—they could be self-programming. Moreover, the style of learning was through induction. Networks would be exposed to examples of a target behavior (for example, the appropriate responses was to a set of varied stimuli). Through learning, the network would adjust the weights in small incremental steps in such a way that over time, the network’s response accuracy would improve. Hopefully, the network would also be able to generalize its performance to novel stimuli, thereby demonstrating that it had learned the underlying generalization that related outputs to inputs (as opposed to merely memorizing the training examples).

(It should be noted that the type of learning described above—so-called “supervised learning”—is but one of a number of different types of learning that are possible in connectionist networks. Other learning procedures do not involve any prior notion of “correct behavior” at all. The network might learn instead, for example, the correlational structure underlying a set of patterns.) Now let us turn to some of the
interesting properties of these networks and the ways in which they offer new accounts of
language acquisition

Learning the past tense of English: Rules or associations?

The study of language is notoriously contentious, but until recently, researchers
who could agree on little else have all agreed on one thing: that linguistic
knowledge is couched in the form of rules and principles. (Pinker & Prince, 1988)

We have, we believe, provided a distinct alternative to the view that children
learn the rules of English past-tense formation an any explicit sense. we have
shown that a reasonable account of the acquisition of past tense can be provided
without recourse to the notion of a ‘rule’ as anything more than a description of
the language. (Rumelhart & McClelland, 1986)

In 1986, Rumelhart and McClelland published a paper that described a neural network
that learned the past tense of English. Given many examples of the form
“walk→walked”, the network not only learned to produce the correct past tense for those
verbs to which it had been exposed, but for novel verbs as well—even including novel
irregular verbs (e.g., “sing/sang”). Impressively, the network had not been taught explicit
rules, but seemed to have learned the pattern on which the English past tense is formed
through an inductive process based on many examples. Rumelhart and McClelland’s
conclusion—that the notion of “rule” might help describe the behavior of children as
well as networks, but play no role in generating that behavior—generated a storm of
controversy and has given rise to hundreds of experiments (with children) and
simulations (with neural networks) as these claims and the counter-claims (e.g., by Pinker
& Prince, in the above citation) have been refined and tested.

The particular example of the past tense was particularly significant, because the
pattern that many children display in the course of learning the past tense of English
verbs had long been interpreted as evidence that children were learning a rule. Cazden (1968) and Kuczaj (1977) were among the first to notice that at very early stages of learning, many children are relatively accurate in producing both the “regular” (add +ed to make the past) and “irregular” (“sang”, “came”, made”, “caught”) verbs. Subsequently, as they learn more verbs, some children appear to go through a stage where they indiscriminately add the “+ed” suffix to all verbs, even irregulars that they have previously produced correctly (e.g., “comed” or “camed”). A reasonable interpretation is that at this point, the child has discovered the rule “add +ed”. The errors arise because learning is incomplete and they have failed to note that there are some verbs to which the rule does not apply. So they over-generalize the “+ed” pattern inappropriately. Ultimately, of course, the children do then pass to a third stage in which these exceptions are handled correctly. Voilà: a rule caught in the process of being learned.

But is this really what is happening? Rumelhart and McClelland’s model also demonstrated a similar U-shaped performance as it learned the past tense. Yet their network did not seem to be learning a rule per se. Instead, the network was using analogy to discover patterns of behavior. During the very early stages, the network did not know enough verbs for this process to be very effective, and so its performance was very conservative—almost a matter of rote learning. As more verbs were acquired, the pattern of “add +ed” that was common to the majority of verbs took hold, and the network started generalizing that pattern across the board. It was only with additional learning that the network was able to learn both the general pattern as well as the sub-patterns (“sing/sang”, “ring/rang”, etc.) and outright exceptions (“is/was”, “have/had”).
This alternative account does not, of course, prove anything about what real 
children do. But it does provide an alternative and very different account of what had 
been taken to be the paradigm example of rule-learning by children. So it is no surprise 
that this model generated a storm of controversy. Steven Pinker and Alan Prince wrote a 
detailed and highly critical response in which they questioned many of the method-
ological assumptions made by Rumelhart and McClelland in their simulation, and 
challenged Rumelhart and McClelland’s conclusions. This then spurred many others to 
develop connectionist models that corrected some of the weaknesses of the original 
model, and also to help provide a better understanding for the underlying principles that 
guide learning in connectionist networks (e.g., Daugherty & Seidenberg, 1992; Hare, 
Elman, & Daugherty, 1995; Plunkett & Marchman, 1991, 1993; to name but a few).

In response, proponents of the symbolic rule account have argued that perhaps 
children might employ something like a connectionist-like system to learn irregular 
verbs, but that the regular verbs are produced by a distinct rule-based mechanism (e.g., 
Marcus, Brinkmann, Clahsen, Wiese, Woest, & Pinker, 1993; Marcus, 1995; Prasada, 
Pinker, & Snyder, 1990). This has become known as the “dual mechanism” account. 
Furthermore, these researchers cite neurological and genetic data that they say argues for 
two such separable systems in humans (Jaeger, Lockwood, Kemmerer, VanValin, 
Murphy, & Khalak, 1996; Gopnik & Crago, 1991). Connectionists, who favor a single 
mechanism approach, argue that a network can in fact produce the full range of behaviors 
which characterize regular and irregular verbs (Marchman, Plunkett, & Goodman, 1997; 
Nakisa & Hahn, 1996; Plunkett & Nakissa, 1997)—including historical changes in the
way the English past tense was formed (Hare & Elman, 1995), as well as the neurological
data (Elman & Hare, 1997; Joanisse & Seidenberg, 1999).

This debate continues today, and although the acrimony is at times excessive,
there is no question that the issues have been sharpened through this debate, and we have
developed a far more detailed picture of this corner of language than we had before.

**Learning the concept “word”: Studies with infants and networks**

Clearly, knowledge of vocabulary cannot be innate: A child born in Singapore must be
able to learn a different word for “milk” than a child born in Tibet. But how do infants
even know that there are such things as words in the first place? After all, spoken
language rarely brackets words with signposts that say “here is a word.” In fluent speech,
for example, words are not separated by pauses or silence, and all the infant hears is a
continuous stream of unsegmented noise. But many theories of acquisition depend
crucially upon prior knowledge of such primitive concepts as word, or morpheme, or
more abstract categories such as noun, verb, or phrase (e.g., Berwick & Weinberg, 1984;
Pinker, 1984). Rarely is it asked how a language learner knows about these concepts in
the first place. Often, it is assumed they are innate.

Yet in fact, there is considerable debate among linguists and psycholinguists
about what are the basic representations that are used in language. It is commonplace to
speak of basic entities such as phoneme, morpheme, or word. Surprisingly, these
constructs have no clear or uncontroversial definition. Furthermore, in what are called
polysynthetic languages (e.g., many Eskimo languages) things that might in those
languages be called a word would, in English, be considered a phrase or even entire
sentence. Thus, there is a fundamental question about how even the concept word might be learned—let alone, the vocabulary of a language itself.

One connectionist model that investigated this question began by making two simple assumptions. First, the architecture of the network would have feedback loops so that the network had a basic kind of memory (memory not for the literal past, but for the network’s only internal prior states). Such a network (called a “simple recurrent network”) is shown in Figure 2. Second, the network would be given a sequence of encoded stimuli, and after each new stimulus, asked to predict what might come next.

—Insert Figure 2 about here—

The encoded stimuli were actually numeric codes that stood for letters, and the whole series of stimuli were drawn from a children’s story. But instead of seeing “Many years ago, a boy and a girl lived by the sea…”, the network saw a continuous stream of numbers (each number standing for a different letter). There were no breaks (spaces) between words. And the network wasn’t told what letters the codes stood for.

Of course, short of memorizing the story—which was too long for memorization to be a feasible strategy—one would not expect the network to perform perfectly in its prediction. On the other hand, consider what a person would do if confronted with the first two letters of a word, for example, “Ma_”, and asked to predict what came next. She would know that there is a limited range of possibilities. The next letter is almost certainly a consonant, and “n”, “t”, “s” are far more likely than “v”, given the vocabulary of English. And this turns out to be exactly what the network learns. It discovers the distributional properties of these encoded stimuli (which, remember, it doesn’t know are letters) and then it predicts stimuli in a way that reflects their conditional probabilities,
given the context. If one graphs the network’s performance, measuring its error when predicting each letter, the result looks something like what is shown in Figure 3.

—Insert Figure 3 about here—

An immediate pattern is obvious here: The network’s ability to predict letters depends on where in the word the letter is. Word-initial letters are difficult to predict (because virtually any letter might occur), whereas after a few letters, the constraints limit the possible letters and the network’s predictions improve.

As far as the network is concerned, of course, it doesn’t need to know about things such as letters or words. The network is merely trying to anticipate what could come next, and is taking advantage of the statistical regularities in the sequence. However, the fact that there are some sequences (letters internal to a word) that are more predictable than others (letters between words), provides evidence that the network could then use to infer the existence of the basic units we call words. All that is needed is for the network to be aware of its own performance, and notice that there are strings of sequences which seem to go together, and others that don’t. The strings that go together are, of course, what we know to be words. (Notice that in the above example, “aboy” would be segmented as a single word. This is because this phrase is very common and the network has not yet broken the sequence into two words. Interestingly, young children also make similar errors, treating common sequences as if they were units.)

This simulation suggests a strategy that infants might use to learn the basic units of their language. In 1996, empirical evidence was reported that suggested that infants in fact do use such a strategy (Saffran, Aslin, & Newport, 1986). In the Saffran et al. study, 8-month old infants were exposed to nonsense speech made up of three-syllable “words”
that were strung together with no breaks (e.g., “bidakupadotigolabubidaku”). After listening to this sequence for only a few minutes, infants were able to discriminate between two new sequences, one of which was made up of a new combination of these “words” vs. another that used the same syllables permuted in a different order. The behavior of these infants is very much like what one would expect if there were using statistical regularities in a way similar to the prediction network above.

More recently, other researchers have found that young infants are able to learn regularities in artificially generated sequences that are even more complex—as if the rudiments of simple grammar might be learned through statistical induction (Gomez & Gerken, 1999; Marcus, Vijayan, Rao, & Vishton, 1999). The interpretation of these data are quite controversial. Marcus and his colleagues, for example, argue that the results show that infants are using algebraic, symbol-processing (Gomez and Gerken, on the other hand, make no such claims). But Seidenberg and Elman (1999a, 1999b) have demonstrated that neural networks show exactly the same kind of performance as the infants, and clearly are not using symbolic machinery to learn the patterns.

**Learning the unlearnable: Recursion and “the importance of starting small”**

The last example of connectionist models of language acquisition to be discussed concerns a phenomenon that many have held provides unequivocal evidence for innate knowledge of linguistic structure—yet which we now know from connectionist modeling is learnable. The phenomenon is what has been called “recursion”, and it turns up in virtually every sentence we produce that has more than a few words.
Consider the phrases such as “John,” “the old man,” “the curious yellow cat,” or “the mouse that ate the cheese.” These are all examples of what are called noun phrases. Notice in this last example, “the cheese” itself is a noun phrase. Thus, noun phrases may contain other noun phrases (if you remember from high school English, “who ate the cheese” is called a “relative clause”) and in fact, there is no principled limit to how far this process can be carried (e.g., “the mouse that ate the cheese that the child dropped on the floor… ”). Such embedding of one linguistic unit inside another is called “recursion,” and refers to the possibility that a category may be defined in terms of itself.

The Rumelhart and McClelland verb learning simulation that we began with dealt with issues in morphology (e.g., verb inflections), but soon other connectionist simulations were developed which modeled syntactic and semantic phenomena. All of those simulations, however, involved sentences of pre-specified (and limited) complexity. In 1988, Jerry Fodor and Zenon Pylyshyn wrote a paper in which they called attention to this shortcoming, and argued that the deficiency was not accidental. They claimed that connectionist models were in principle unable to deal with the kind of unbounded recursion or to represent complex structural relationships (constituent structure) in an open-ended manner.

In a simple recurrent network, the internal (or “hidden”) units feedback on themselves (see Figure 2). This provides the network with a kind of memory. The form of the memory is not like a tape recorder, however; it does not literally record prior inputs. Instead, the network has to learn itself how to encode the inputs in the internal state, such that when the state is fed back it will provide the necessary information to carry out whatever task is being learned.
In 1988, Servan-Schreiber, Cleeremans, and McClelland demonstrated that such a network could be trained on an artificial language which was generated by something called a Finite State Automaton (FSA; an FSA is the simplest possible digital computer; the most powerful type of computer is a Turing Machine). The recurrent network’s task was simply to listen to each incoming symbol, one at a time, and to predict what would come next. In the course of doing this, the network inferred the more abstract structure of the underlying artificial language.

The FSA language, however, lacked the hierarchical structure shown in Figure 2. In another set of simulations, Elman (1991, 1993) showed that simple recurrent networks could also process sentences containing relative clauses, which specifically involve hierarchical/compositional relations among different sentence elements. Weckerly and Elman (1992) showed that these networks also exhibit the same performance asymmetry in processing different types of embedded sentences that humans do. Their networks, like people, find sentences such as (1) more difficult than sentences such as (2a and 2b).

(1) The mouse the cat the dog scared chased ran away.

(2a) The dog scared the cat that chased the mouse that ran away.

(2b) Do you believe the report the stuff they put in coke causes cancer is true?

Understanding exactly how such networks work is an interesting problem in its own right, and there are strong connections between their solutions and the dynamical systems we discuss below. What was interesting from the viewpoint of language acquisition, however, was something entirely unanticipated.

Initial attempts to train simple recurrent networks on an artificial language that contained sentences with recursion did not in fact succeed. Dismaying, the network’s
performance was far worse than would have been expected given prior results with
simple sentences. In order to see where the problem might lie (perhaps the network might
deal with a limited form of recursion, for example), a new training regime was tried in
which the network was given only simple sentences. This network did fine. So the same
network was then given additional sentences to learn, a small number of which were
complex (i.e., contained recursion). Again, the network did fine. In fact, as the
complexity of the sentences was gradually increased, the network easily assimilated the
new examples. By the end, the network was able to process the original set of complex
sentences that it had failed to learn from when they were the initial dataset. The crucial
difference was that when the network was hand-fed simple sentences prior to the
complex ones, it was then—and only then—able to progress to the more complex forms
later.

But it’s not necessary to spoon-feed the network in this way. In a second
simulation, at the outset of training, artificial noise was injected into the network every
few words, simulating the effect of having a very limited working memory. The
network, however, was exposed from the start to complex sentences. Over time, the noise
was gradually decreased, as if the network’s working memory were gradually improving
with age. By the time the network had “grown up” (had adult-like working memory), it
was able to fully process the complex sentences. The key to both simulations (described
in greater detail in Elman, 1993) turned out to be that by focusing on the simple sentences
first, the network was able to learn basic facts about language, such as grammatical
category, noun-verb agreement, etc., that provided the necessary scaffolding for learning
about more complex sentences. The problem with starting with those more complex
sentences is that, lacking knowledge of the fundamentals, the patterns that are created by recursion were too difficult for the network to identify.

This result is interesting from several points of view. First, the fact that the network was able to learn complex grammar shows that these kinds of structures can be learnable by example, and without need symbolic rules. Innate knowledge that is specifically linguistic is not necessary. Second, the networks demonstrate “the importance of starting small.” This strategy suggests that what may be special about children’s ability to learn languages may not be due to any special mechanism that they possess as children (e.g., a “Language Acquisition Device” of the sort hypothesized by Chomsky). The starting small hypothesis, which is similar to Newport’s (1990) “less is more” proposal, is that children’s language learning abilities are rooted in something entirely different. It is children’s processing limitations that make language learning possible. By having a restricted working memory, children may be able to process only simple patterns; these patterns then provide a crucial foundation for learning subtler generalities. Viewed this way, what is “innate” that makes language possible has nothing to do specifically with language. Instead, it is—paradoxically—a maturational limitation on working memory that interacts with a general purpose learning mechanism that makes it possible to learn complex sentences.

Of course, the language that the networks described here have learned is still quite limited, and there are clearly many other aspects of human cognitive and social development that are necessary to learn language. The lesson of the connectionist models that have been described here are just that rather simple learning algorithms may be far more powerful than were previously recognized. And what makes human language
possible may turn out not to be the evolution of a separate Language Organ, as envisaged by Chomsky, but rather a number of fairly small tweaks and twiddles in the cognitive capacities (including things such as the developmental timetable for working memory) that we share with our non-human cousins. It is from the complex interaction of these many small changes that language emerges.
FIGURE LEGENDS

Figure 1: A neural network that reads letters and recognizes words. When a letter is detected, the corresponding letter node activates all words that contain it (lines with arrows). Since only one word can be present at a time, word nodes compete with inhibitory connections (lines with filled circles).

Figure 2: Simple recurrent network. Groups of nodes are shown as rectangles. The feedback connection between Hidden Units and Context Units provides the network with memory.

Figure 3. Plot of network error in “predict the next letter” task, using a simple recurrent network (after Elman, 1990).
REFERENCES


