Oxford Handbooks Online

Connectionism and the Emergence of Mind

Stephen J. Flusberg and James L. McClelland

The Oxford Handbook of Cognitive Science *(Forthcoming)*
Edited by Susan Chipman

Online Publication Date: Nov 2014

Subject: Psychology, Cognitive Psychology
DOI: 10.1093/oxfordhb/9780199842193.013.5

[–] Abstract and Keywords

Connectionism is a computational modeling framework inspired by the principles of information processing that characterize biological neural systems, which rely on collections of simple processing units linked together into networks. These units communicate in parallel via connections of varying strength that can be modified by experience. Connectionist networks have a wide range of theoretical and practical applications because they exhibit sophisticated, flexible, and context-sensitive behavior that mirrors human cognitive performance in many domains, from perception to language processing. By emphasizing the commonalities underlying various cognitive abilities, connectionism considers how a basic set of computational principles might give rise to many different forms of complex behavior. Thus connectionism supports a novel way of thinking about the nature and origins of mental life, as the emergent consequence of a system based around principles of parallel processing, distributed representation, and statistical learning that interacts with its environment over the course of development.

Keywords: Connectionism, parallel distributed processing, artificial neural networks, learning, emergence

Our daily lives are characterized by remarkable feats of mental functioning. We effortlessly perceive and categorize the people and objects in our environment, remember past events while imagining possible futures, and communicate with friends and colleagues in speech and writing. A major goal for cognitive scientists is to figure out how these phenomena relate to the material world of atoms, molecules, brains, and bodies. Philosophers have described this as the problem of reconciling the *manifest image* of everyday experience with the *scientific image* of the universe as a material realm governed by physical laws (Dennett, 2013; Sellars, 1963). Although there is a general consensus that the brain plays a starring role in mental life, just how the activity of billions of brain cells (neurons) might support complex cognitive functioning remains something of a mystery. In this chapter, we provide a brief introduction to *connectionism*, a computational modeling framework that tries to address this issue by using artificial neural networks to simulate cognitive processes like perception, memory, language, and analogical reasoning. Our goal is to foster a general appreciation for how these models work and how they can enhance and even transform our understanding of the nature and origin of cognitive functioning.

Background and Overview

According to the philosopher Daniel Dennett, a notable scientist once opened a workshop with the following statement: “In our lab we have a saying: if you work on one neuron, that’s neuroscience; if you work on two neurons, that’s psychology” (Dennett, 2013). Although undoubtedly tongue-in-cheek, this comment underscores the desire of many researchers to ground an understanding of mental processes in terms of interactions between neurons in the brain. Indeed, scholars have long hypothesized that the pattern of connections among brain cells ought to have significant implications for mature psychological theories (e.g., Freud, 1895; James, 1890), and mathematical models in the connectionist tradition date back to the middle of the 20th century (e.g., McCulloch & Pitts, 1943; Rosenblatt, 1958). Connectionist modeling eventually rose to prominence in the 1980s, through a

convergence of cognitive psychologists, neuroscientists, and computational scientists, many of whom joined together to develop the parallel distributed processing (PDP) framework (Rumelhart, McClelland, & the PDP Research Group, 1986b). PDP models draw their inspiration from the functional organization of the brain, which is characterized by the parallel activity of many neurons, treated as simple processing units, wired together into intricate networks. The strength or *weight* of the connections between the units represents the knowledge that is stored in the network. This is because these weights, in combination with the input signals the network receives, determine the pattern of activation across the units as these signals are propagated through the system. It is the collective activity of the units that defines the functionality of the network. How this all works in practice, and the relationship between artificial and biological neural networks, will be explored in more detail in the following section.

Click to view larger

Figure 1 . Your knowledge of English and the surrounding letter context enable you to perceive the central letter as an "h" in "the" and the same character as an "a" in "cat."

That being said, most connectionist modelers favor the approach not simply because it is inspired by the brain, but because the models themselves are useful for addressing many types of psychological and computational questions (Anderson, 1977; McClelland, Rumelhart, & Hinton, 1986). Although connectionist models come in a wide range of configurations that differ in important respects, most of them naturally capture certain essential features of human thought and action. For instance, cognitive processing at all levels, from perception to memory to language comprehension, is known to be highly responsive to contextual factors (e.g., Bar, 2004; Oliva & Torralba, 2007). Figure 1 illustrates this idea in the domain of letter perception: although the words are easily read as "the cat," a closer look reveals that the "h" in "the" and the "a" in "cat" are exactly the same shape. It is the surrounding letters and your knowledge of English words that enables you to fluidly read the phrase and perceive the "correct" letters based on their context. Because units in PDP models can be influenced by multiple sources of information at the same time, PDP models are especially sensitive to the effects of context on cognition and behavior.

Furthermore, cognitive abilities do not show up fully formed but develop over time as a consequence of experience. An emphasis on learning makes PDP models ideal for addressing issues related to cognitive development, including the effects of different environmental inputs on psychological functioning (Elman et al., 1996; Munakata & McClelland, 2003). Finally, in cases of brain damage, many behaviors are not simply eliminated all together but rather show a gradual decline in functioning depending on where and how much damage has been inflicted. Because the information stored in a PDP model is often distributed across many units and connections, these models naturally capture this phenomenon, as well as the graded or probabilistic nature of knowledge representation (i.e., memory) more generally. Many of these properties of connectionist models are not shared by other popular approaches to cognitive modeling.

The key take-home message is that connectionist models are not just somewhat biologically plausible implementations of existing psychological theories; rather, they provide alternatives to other theories and offer a means of investigating a unique way of thinking about mental processing. In particular, the connectionist perspective suggests that complex cognitive functions are an *emergent* consequence of the dynamic interactions between much simpler processing elements. Importantly, the emergent properties of a system are those that cannot be reduced to the behavior of any of the simpler elements that make up the system in isolation. In psychology, therefore, familiar cognitive constructs like schemas, syntactic rules, analogical mapping, and executive functioning need not be thought of as distinctive knowledge structures or mechanisms. Instead, these processes can be understood as higher level, approximate descriptions of underlying network behavior; behavior that arises spontaneously out of a system that embodies a particular set of basic computational principles. As we

have alluded to already, these principles include an emphasis on distributed representations and parallel processing, as well as others like constraint satisfaction, pattern matching, and statistical learning. We suggest that this *emergentist* perspective (see McClelland, 2010; McClelland et al., 2010) is a powerful alternative to other computational approaches in the cognitive sciences, which tend to place a greater emphasis on structured, symbolic representations and task-specific processing machinery that must be explicitly coded into cognitive models. These ideas will be fleshed out throughout the chapter.

In the following section, we provide a broad overview of how connectionist models work, drawing attention to what we see as the key computational principles that characterize the PDP framework. To illustrate how sophisticated cognitive behavior can emerge in a connectionist model as a function of these principles, we review two influential models: the interactive-activation (IA) model (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982) and the simple recurrent network (SRN; Elman, 1990). Finally, we highlight several exciting future directions for the field and discuss how connectionist models can function as important tools for thinking about the nature of the mind and behavior. Along the way, we will decipher a cartographic parable, discover why errors are the key to learning, and, with any luck, start to demystify the exceptionally mystifying concept of emergence.

The Basic Philosophy and Mechanics of Connectionist Modeling

Maps, Models, and Reality

In a very short story by Jorge Luis Borges, the author describes an ancient empire that prized cartography above all other art forms (Borges, 1998). The cartographers in this land endeavored to create the most detailed and accurate atlases possible, which resulted in maps of provinces the size of cities. Eventually, they produced a map of the empire that was the size of the empire itself. The brief parable ends there, but we want to highlight two related themes that will help frame our discussion of connectionist modeling.

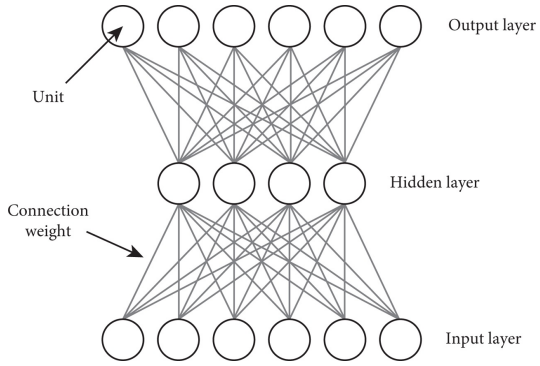
The first is the metaphor of scientific models as maps of reality. Maps are tools that can be used to navigate complex environments because they represent relevant features of the surrounding lands in a format that the map user can make sense of. Similarly, scientific models are tools that can be used to make sense of complex aspects of reality by representing them in a format that researchers can understand and employ in the service of fitting and predicting patterns of data. The second theme is the essential lesson of the story: maps must be *simplifications* of the territory they represent; otherwise, they cannot fulfill their role as navigational tools. The same reasoning applies to scientific models (which is why Borges titled his story “On Exactitude in Science”). By simplifying a complicated theory or domain down to a set of central principles and instantiating them in a formal model, the implications of these ideas become easier to get a handle on and the model can function as intended as a research tool (McClelland, 2009).

Putting the “Neural” into Artificial Neural Network

Connectionist models are like simplified maps of cognitive systems inspired by the organization of the brain. They are not atlaslike maps of the nervous system but relatively abstract representations that seek to capture key functional features of neural information processing. Biological neurons are sophisticated cells that communicate with other neurons through a complex interplay of electrical and chemical signaling. When a neuron reaches a specific threshold of stimulation it sends a rapid electrical wave (action potential) down its axon. This results in the release of neurotransmitter chemicals into the small gap (synapse) that connects the terminal bud of the axon to the receiving end (dendrite) of the next cell in the chain. When the release of these neurotransmitters increases the likelihood that the postsynaptic neuron will *fire* (i.e., reach the threshold necessary for an action potential), it is called an *excitatory* connection. When it decreases the likelihood that the postsynaptic neuron will fire, it is called an *inhibitory* connection. These connections vary in *strength*, or extent, of the excitatory or inhibitory effect. Finally, each neuron may be connected to thousands of other neurons in a vast web of cellular activity, and all of these neurons are continually adjusting their activations in parallel so that they can influence one another at the same time.

Now take a look at the schematic connectionist network depicted in Figure 2. The circles represent the *units*, analogous to neurons. The lines linking the units represent the *connections* between the units, which are

analogous to the synapses linking biological neurons. In many models, the units are organized into distinct layers, as shown in Figure 2. The *input* layer receives information from the environment, analogous perhaps to lower level sensory regions of the brain (although this will depend on the particular model). The *output* layer denotes the "response" of the system after the input signals have propagated through the network, possibly analogous to a button press or perceptual judgment (depending on the task being simulated). The *hidden* layer is like the cortical brain regions that intervene between sensory input layers and response output layers. Figure 2 portrays a *feed-forward* network because activation travels in only one direction (from input to output, traversing the hidden layer). Brain networks likely include *feedback* and *recurrent* connections, and many models include them (we will consider such a network later), but you may be surprised by how much we can learn from a simple feed-forward network (and if you wonder how this could be so, think of the ancient cartographers!).



Click to view larger

Figure 2 . A schematic of a feed-forward, three-layer network. Activation in the network propagates from the input layer to the hidden layer to the output layer.

One tricky conceptual issue is the relationship between units and connections on the one hand and neurons and synapses on the other. It is tempting to think of units as corresponding to individual neurons or groups of neurons, and this can be useful for some purposes. However, it is ultimately better to adopt a more abstract perspective, in which the emergent, system-level consequences of activity in vast populations of neurons are simulated using much smaller populations of units (see Smolensky, 1986, for the first and still definitive treatment of this idea).

Activation Propagation

At any given moment in time, each unit in a network will have a particular level of *activation* that is represented by a real number. In some models, the range of possible unit activations is restricted to binary values, where an activation of 0 means the unit is "off" (analogous to the resting state of a neuron) and an activation of 1 means the unit is "on" (analogous to a neuron firing an action potential or a series of action potentials). In other models, the activation value of a unit is free to vary continuously, typically between 0 and 1, with higher levels of activation analogous to increases in the firing rate of a neuron. Each connection linking two units also has a particular real number value that signifies the strength and type of connection, with larger (absolute) weight values indicating greater connection strength. Positive weight values signify an excitatory connection, whereas negative values signify an inhibitory connection. In the brain, we imagine that activations of units are updated continually in time, just as the position of a moving object changes continually. In simulations, we break time up into discrete steps for methodological convenience. Sometimes these steps are very fine, closely approximating continuity. At other times, the steps are much coarser, sometimes simply corresponding to initial and final states at the beginning and end of processing.

How do signals actually propagate through the network? To begin with, all of the units are typically set to their resting state (usually near 0) or to a random activation near their resting state. Next, the network must receive information from the environment, which in practice means that one or more units in the input layer are turned "on" by the modeler (activation = 1). A simple algorithm is then used to compute the activation values for the rest of the units in the network at the next time step (it takes two time steps for input signals to reach the output layer in the three-layer, feed-forward model portrayed in Figure 1 because activation must first pass through the hidden layer). For a given unit, u_i, you must first determine the *net input* to that unit. To do this, multiply the current activation of each of the units that connects to u_i by the value of the weight connecting the units and then compute the sum of

these products. After that, you need to invoke a particular rule for converting the net input into the new activation value for u_i. Some models use a threshold rule, where u_i only becomes active if the net input exceeds a specific value. In other models, the net input might be fed through a mathematical function to come up with a continuous activation value. A *sigmoid* or *S-shaped* function is often used and can be thought of as a continuous version of a step function. Sigmoids neatly capture the fact that biological neurons cannot fire at rates above about 100 times per second and cannot fire at rates below zero.

Pattern Matching and Knowledge Representation

Because that is pretty much all there is to processing in connectionist networks, what can this simple type of system actually *do*? For starters, connectionist models are adept at pattern recognition or pattern matching, which in practice means taking in a specific input and yielding a specific output. In principle, a network can come to associate any arbitrary pattern of activation over the input layer with any pattern of activation over the output layer. This basic property has a wide range of powerful applications, which should become apparent as we continue to explore the way these models are used by researchers to understand cognitive processing. In essence, many different cognitive functions can be recast as the ability to detect, represent, and predict patterns of stimulation from the environment and to produce appropriate patterns of outputs.

Hopefully, it is already clear that the value of the weights in the network determines what output pattern will result from a given input signal; the "knowledge" of the association between a specific input and a specific output is "stored" in the connection weights. The use of quotation marks is deliberate here since these words are being used in a specific and atypical way. The knowledge represented in the weight matrix is *tacit* or *implicit* because the value of a weight simply determines how activations are filtered through the network, and information regarding multiple different associations may flow through the very same weights. In other words, the value of a weight or set of weights does not itself code or stand for anything in the world. This is very different from how a digital computer stores its "knowledge"; that is, through the use of stored arrangements of symbols that *explicitly* represent or stand for the information needed to guide processing. Connectionist models *do* represent the current content of mental states in a somewhat accessible form; when an input signal is propagated through the network, the resulting pattern of activation across all of the internal units can be thought of as a representation of the input pattern and of the thoughts and possible actions that it brings to mind. The key difference is that the knowledge that guides the formation of these patterns is not available for inspection, as it is in a standard computer program.

Maps, Models, and Reality Revisited

Now that we have reviewed the basic principles governing connectionist networks, we can start to address how these models may be used as tools for simulating and conceptualizing cognitive functioning. It will be helpful here to revisit the metaphor of scientific models as maps of reality. Most maps work by depicting important features of the environment in a simplified (and portable) format and transposing these representations onto a two-dimensional surface while preserving the spatial relationships between them. However, different types of maps preserve different features of the environment and different spatial relationships, depending on their intended function. Indeed, a subway map will help you get from Brooklyn to Queens using public transportation, but it won't be much use if you plan on driving. To use a real map, therefore, you have to figure out just *how* it maps onto the territory it represents.

In a similar vein, to use a connectionist model, you have to figure out how features of the model map onto the cognitive territory you intend to simulate and at what scale. This involves interpreting the cognitive *task* you care about in terms of the structural and functional properties of the model. Performance on the task should be framed as the behavior (the output or overall activation state) that results from a system (the network) that is stimulated by the environment (the input), possibly over the course of development (see the later discussion of learning). Be sure to keep in mind that this is a deliberate (and useful) oversimplification because no connectionist modeler believes that brains (or people) are just passively bombarded by environmental stimulation. Indeed, you should imagine that the system is (part of) a brain embedded in an organism that is actively exploring or interacting with the world since the same general principles will apply either way.

Diving a bit deeper, the pattern of activity across the input layer typically stands for the information the cognitive system is receiving from the environment. This could be anything from low-level sensory stimulation to higher level

perceptual or conceptual gestalts (e.g., words or objects). The complete set of patterns that a network is exposed to therefore signifies the total, simplified environment of the model. The pattern of activity across the output layer typically denotes the response or behavior of the system on a given "trial."

It is important to specify just how these units correspond to the task information you care about. Some models utilize *localist* representations, in which each individual unit stands for a single meaningful stimulus. This is a natural way of thinking about computational modeling, echoing classical approaches that treat representations as individual symbolic structures. Keep in mind, though, that even localist connectionist units do not function like the symbolic representations stored in your desktop computer, which can be shuttled around and manipulated by any number of different operations. Other connectionist networks utilize *distributed* representations, in which the relevant stimulus is represented by the *pattern* of activation across an entire layer of units. In this case, no single unit has any intrinsic meaning (or referent), and each unit may be an active participant in multiple different meaningful patterns. Some multilayer networks use localist input or output units for the sake of simplicity, but the learning process forces these networks to learn distributed representations over the hidden units (see the later discussion of the SRN, as well as Elman, 1990; Flusberg, Thibodeau, Sternberg, & Glick, 2010; Rogers & McClelland, 2004).

To make these ideas more concrete, we turn now to a discussion of a classic connectionist model with a wide array of applications: the interactive activation model (McClelland & Rumelhart, 1981; Rumelhart and McClelland, 1982).

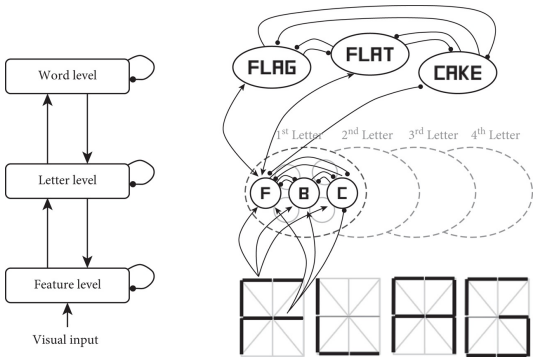
The Interactive Activation Model

Context in Perception

Psychologists have repeatedly shown that the surrounding context influences our ability to perceive things in the environment (see Figure 1). For example, people are faster to locate and recognize a familiar object when it is consistent with the background scene (Bar, 2004; Oliva & Torralba, 2007; Palmer, 1975). In other words, you're quicker to spot a toaster in the kitchen than in your bedroom. Similarly, people are faster and more accurate at perceiving letters when they are embedded in a real word than when they are embedded in a random string of letters (e.g., identifying the O in SPOT vs. TKOR; Reicher, 1969). These findings demonstrate that we bring knowledge of our past experiences to bear when we engage in perceptual tasks. Interestingly, when it comes to letter perception, we are also faster and more accurate at identifying a letter when it appears in a pronounceable nonword (e.g., BROT). This seems to suggest that we are using knowledge of orthographic rules in this task and not simply relying on familiarity with specific words we have encountered before. Does that mean we explicitly represent these rules in our minds and use them in the service of perceiving letters?

McClelland and Rumelhart (1981) developed the *interactive activation* (IA) model to help make sense of these findings and illustrate how they might emerge out of a system that does not explicitly instantiate any orthographic rules. The structure of their network was inspired by findings from the neuroscience literature showing that neurons in visual brain regions respond to perceptual information at varying levels of abstraction. Some cells respond chiefly to low-level perceptual features like lines and edges, whereas others respond to higher level perceptual gestalts like letters, words, and faces (Wandell, 1995). Importantly, feedback connections between these regions ensure they can mutually influence one another. In other words, perception seems to involve the simultaneous, interactive, parallel processing of incoming visual data at multiple hierarchical levels.

Model Architecture



Click to view larger

Figure 3 . On the left, a schematic of the interactive activation (IA) model of letter perception. On the right, a zoomed in, schematic close-up of part of the model (only a subset of the connections depicted). Arrows indicate excitatory connections while circles indicate inhibitory connections.

A schematic of the IA network is depicted in Figure 3, with the global architecture on the left and a close-up in more detail of one small bit of the network on the right. The model was designed to perceive four-letter strings using localist representations, but this architecture is somewhat different from the feed-forward networks we examined in the previous section. Starting at the highest level, each unit in the *word* layer represents one of 1,179 four-letter words in English. Each word unit has an inhibitory connection to every other word unit because you can only perceive one word at a time (and therefore if one word is strongly active, it should suppress the activation of every other word). Each unit in the *letter* layer represents an English letter at one of the four possible positions it might occupy in one of the words (i.e., there are 26 units for the first letter of the word, 26 units for the second letter of the word, etc.). Again, letter units send inhibitory connections to other letter units at their position since only one letter can be perceived at a given position at a time. Letter units project excitatory connections up to word units that are consistent with that letter at that position in the word, and word units do the same thing down to the letter layer. Thus, a letter unit representing “B” in the first position will have mutually excitatory connections to units representing BALL, BAKE, and BOWL. Letter units also have excitatory and inhibitory connections with units in the *feature* level, which represent the different line features that make up the block letters at each of the four possible positions. Again, these units inhibit one another when their mutual presence at a location would be inconsistent, and they excite one another when their mutual present would be consistent.

IA Model in Action

How is this network actually used to simulate the task of perceiving a letter at a particular position in a string? First, the units in the feature and letter layers are set to their resting activation value (just a bit below 0). The units in the word layer have resting levels based on their frequency in the English language to capture our prior knowledge about which four-letter strings we are likely to encounter (e.g., the resting level of unit representing the frequent word CALL would be very close to 0, whereas the resting level for the infrequent word RAPT would be farther below 0). Next, a letter string would be presented to the network, which means that units in the feature layer are turned on (referred to as “clamping”) to signify this visual input. Then, these activations are allowed to propagate through the network of excitatory and inhibitory connections one discrete step at a time (a general *decay* process is also included so that units’ activations tend to subside over time unless they are receiving a relatively greater amount of excitatory input). The “speed” at which the network “perceives” a letter in a given position is captured by measuring (or comparing) how many time steps it takes for units in the letter layer to reliably settle on the “correct” activation levels to match the visual input at that position. It is also possible to simulate conditions in which words or letters are masked or degraded and to obtain a measure of perceptual accuracy. Furthermore, slight variants of the model (using slightly different details of the activation and inhibition assumptions of the original) can be shown to closely approximate optimal use of sensory information and prior knowledge of words in perception (McClelland, 2013).

Importantly, this relatively simple network can capture all of the context effects on letter perception described earlier, including the advantage of identifying a letter in a pronounceable nonword. To get a sense for why this is so, consider what happens when the network is presented with a real word like FLAG and asked to identify the first letter. This string will initially produce bottom-up excitation of the F at the first position in the letter layer, the L at the

second position, and so on. This will then lead to the activation of words like FLAG, FLAT, and FLOG in the word layer because they are consistent in all or almost all of the letter positions (although the activation of FLAG will tend to dominate because it is the actual word being presented to the network). The activation of these word units then feeds back down to the letter layer, facilitating the activation of F at the first position. Thus, the network will quickly activate the F letter in the first position because of this combination of bottom-up, feature-driven processing and top-down, word-driven processing.

Now consider what happens when the network is presented with the random string FXZQ instead. Because none of the word-level units receives much excitation from this input, no word-level units will become active over the course of the first few time steps to facilitate the processing of the letter F in a top-down fashion. Thus, the network will be "slower" to perceive that F. Finally, consider what happens when a pronounceable nonword like FLIG is presented. Although this is not a real English word, pronounceable words tend to share multiple letters with real words (i.e., this is a statistical property of the language itself). Thus, when the network is presented with the string FLIG, units in the word layer representing FLAG, FLIP, and FLOG will actually receive some bottom-up activation because they share letters with this string at three positions. The activation of these word units will then feed back down to the letter layer and facilitate the perception of the F in the first position, mirroring human behavior on these sorts of trials. Of course, the network will show the same sort of effect for *non*pronounceable letter strings as well, as long as they happen to share multiple letters with real words (e.g., FLNG). Interestingly, Rumelhart and McClelland (1982) found that human participants also show this behavior, confirming a novel prediction made by the model.

Lessons Learned

There are four important lessons we wish to draw out from this example. First, as we have just explained, what allows the network to succeed at simulating context effects on letter perception is the integration of distributed information processed in parallel at multiple levels of abstraction. Thus, we can now understand a seemingly sophisticated perceptual process in terms of these core connectionist principles. Second, and in a related fashion, we can think about the network as automatically solving a *constraint satisfaction* problem that arises when the feature units are clamped at the beginning of a trial. The activation of any given unit is constrained by its relationships with (i.e., connections to) other units both within its layer and with the other layers it projects to. As activation propagates through the network, all of these constraints are acting simultaneously, which leads the network to eventually settle on a global activation state that best satisfies all of these mutual constraints. This is a general principle of parallel distributed processing that is a useful way of framing many cognitive tasks.

Third, the network can simulate the contextual benefits of pronounceable nonwords on letter perception without any explicit representation of orthographic rules whatsoever. That is, there is no unit or weight or any other feature of the network that explicitly codes for rules of orthography. This demonstrates that when a system behaves *as if* it is using a particular rule, this does not necessarily mean that it is explicitly representing that rule. In the present case, the appearance of rule-governed behavior was an *emergent* property of a system in which one of the constraints on network activity was knowledge of real English words (which tend to share letters at multiple positions with pronounceable nonwords). This is one of the key insights of the connectionist framework and a nice way to illustrate how complex behavior can emerge from a system of simple processing units operating on the basis of a few core computational principles. We can talk about the network as "knowing" orthographic rules, but this is really just an approximation or shorthand (and one that isn't entirely useful since it obfuscates the fact that both human subjects and the network show enhanced letter perception for certain nonpronounceable letter strings as well!).

Finally, the interactive activation architecture is not useful *only* for understanding letter perception. Indeed, it can be extended to capture findings from a variety of perceptual and cognitive domains that can be framed in terms of this sort of multiple constraint satisfaction problem. Here are a few examples: using this same set of principles, McClelland and Elman (1986) simulated the recognition of spoken words in their well-known TRACE model; Burton, Bruce, and Johnston (1990) simulated the recognition of faces; Freeman and Ambady (2011) modeled aspects of person perception in social psychology; Rumelhart et al. (1986c) simulated the perception of the ambiguous Necker cube; and McClelland (1981) and later Kumaran and McClelland (2012) applied the same ideas to the synthesis of information across items in memory.

A key conceptual link to other ideas in cognitive science was made by Rumelhart et al. (1986c). This article demonstrated that these same principles could help us think about the nature of *schemas* in cognitive processing. Traditionally, schemas were thought to be a particular class of cognitive structure that contained an organized body of information. For example, your schema of a kitchen would include knowledge of all of the furniture that typically goes into a kitchen. Thus, researchers might explain the fact that you are faster to locate a toaster in a kitchen than in a living room by claiming that your kitchen schema was activated (in the same way they might invoke knowledge of the rules of orthography to explain why you are faster at locating a letter in a pronounceable nonword). Rumelhart et al. showed that room schemas could instead be understood as emergent properties of a constraint satisfaction network that contained units representing different features typically found in rooms around the house.

Of course, the interactive activation architecture cannot explain every aspect of human cognition! The model as we have described it does not even address some of the core principles of connectionist modeling alluded to earlier in this chapter, such as distributed representations and a learning mechanism. As it turns out, those two properties can play a powerful rule in capturing other essential features of cognitive processing, an issue we turn to now.

Learning and Distributed Representation

Gaining (and Losing) Weight: Learning in Connectionist Networks

One important question is where the weight values in connectionist models come from in the first place. In some cases, like the IA model, the modeler stipulates the weight values to ensure the proper relationships between unit activations for the task being modeled. More commonly, however, the network is forced to *learn* the "correct" weight values during an initial training phase. In this way, a system that "knows" nothing at the outset may come to acquire that knowledge over a period of what we might think of as cognitive development. In these learning models, the connection weights are usually randomized before training to a range of small values so that the network starts out by treating every input pattern in roughly the same way (since activation will be weakly propagated for all inputs).

Hebbian Learning

A variety of different but related learning algorithms have been developed for use in artificial neural networks. The simplest one is *Hebbian learning*, inspired by neuropsychologist Donald Hebb's (1949) famous dictum that is commonly paraphrased as "neurons that fire together, wire together." In this case, whenever two connected units are active at the same time, the weight between them is strengthened in proportion to how strongly they were activated (usually, some way of reducing weights between weakly activated units is included as well). The precise degree to which the weight value is increased for each co-activation is determined by a parameter known as the *learning rate*. The learning rate is generally set to be quite small, which results in gradual development of the network and can help prevent new learning from simply overwriting previously learned information. A direct analog of this type of learning, known as long-term potentiation (LTP), was discovered in the mammalian brain less than 20 years after Hebb proposed it (Lømo, 1966).

Hebbian learning is a type of *unsupervised* learning that endows a network with the ability to adapt itself to the patterns it is exposed to in the environment (a form of *self-organization*). This yields some useful properties, such as the capacity to retrieve a complete memory from partial information. To understand how this works, consider a network consisting of just one layer of 100 units that are all connected to one another. Now imagine that whenever the network experiences a particular event in the environment, a dozen units turn on. Different events will lead to a different subset of units becoming active, but if the network encounters the same event again, the same 12 units will reactivate. Over time, the continued co-activation of these dozen units will lead to an increase in the strength of the weights connecting them. As a result, if you activate just a few of these units, the network will tend to reactivate the entire pattern representing the event (since the activations will propagate through the strengthened weights linking these units). This is analogous to our ability to retrieve a complex memory of a friend when we are exposed only to a simple cue like his name. We will revisit this phenomenon when we discuss the advantages conferred by using distributed representations.

Error-Driven Learning

A slightly more complicated learning algorithm requires comparing the activations at the output layer to a *target* activation pattern and using the discrepancy between the two, known as the *error signal*, to adjust the weights in order to reduce the magnitude of this error in the future. This error-driven learning algorithm is commonly known as the *delta rule* (delta, Δ, is the Greek letter that symbolizes *change* in math and science).

To make this procedure more concrete, consider an example of learning that most introductory students of psychology are familiar with: Pavlovian classical conditioning. In a typical study, a dog or other animal must learn to associate a neutral stimulus like a ringing bell or flash of light with an appetitive stimulus like food or an aversive stimulus like an electric shock. This is accomplished by repeatedly pairing the relevant stimuli. After several such pairings, the dog might drool when exposed to a flash of light that was paired with food and whimper when exposed to a ringing bell that was paired with a shock. A simple, error-driven learning network can capture this process quite nicely.[1]

Imagine a two-layer network with two units in each layer. The units in the input layer each represent a seemingly benign stimulus that may be present in the environment: a light and a bell (conditioned stimuli). The output units each represent another stimulus that our experimental animal actually cares about: a delicious bowl of meat and a painful shock (unconditioned stimuli). The "goal" of the network, if you like, is to activate the unit representing the appropriate unconditioned stimulus when one of the input conditioned stimuli units is activated. So, if you activate the light unit, the network should activate the food unit (but *not* the shock unit), and if you activate the bell unit, the network should activate the shock unit (but *not* the food unit). Because the connection weights are initialized to very low random values, at the start of training activating either the light or bell units will result in the weak activation of both the food and shock units. Therefore, when we activate the light unit and then present the food, there will be a large amount of error on the food unit, since we want this unit to be fully activated but it isn't at the moment (error is equal to the target activation of the output unit minus the actual output unit activation). To figure out how much to adjust the weight between the light unit and the food unit, simply multiply this error by the value of the learning rate parameter. Gradually, over the course of training, this connection weight will be strengthened by this procedure. At the same time, the weight between the light unit and the shock unit will be weakened because the target activation of the shock unit given an input of light is zero. The training phase itself would consist of individual episodes in which one of the input units is activated, the signal is propagated to the output layer, and the resulting error signals on the output units are used to adjust the weights. This process would then be repeated for the other input unit. This would constitute a single *epoch* of training. Because of the gradual nature of learning in most artificial neural networks, the entire training phase often consists of many epochs, continuing until the errors on the output units are substantially reduced or eliminated.

Frank Rosenblatt (1962) demonstrated that this simple error-driven learning process could be used to adjust the connection weights in two-layer networks (which he called *perceptrons*) to associate any arbitrary input and target output patterns. However, researchers later showed that these two-layer networks were inherently limited in their computational power (Minsky & Papert, 1969). In particular, if these networks have to learn to match multiple input and output patterns, there are certain sets they simply cannot get right because the error signals would keep pushing the weights in a direction that would work for one of the pattern pairs but *not* the other (mathematically speaking, these sets of patterns are not *linearly separable*).

Three-layer networks are much more powerful and can get around this problem. However, the basic delta rule cannot be used to train these multilayer networks since there are now two sets of weights that have to be adjusted; because there is no obvious target activation for the hidden layer units, there is no error signal available to adjust the weights between the input and hidden layers. Fortunately, researchers have discovered several learning algorithms that can be used to train multilayer networks using extensions of Hebbian learning and the delta rule (Hinton, 1989; Rumelhart et al., 1986a). For example, the *backpropagation algorithm* starts with the error at the output layer and sends a signal based on the error backward through the network toward the input to determine how to change each weight to reduce the error. The mathematically inclined reader is invited to check out the simple calculus that makes this possible (Rumelhart et al., 1986a). An early criticism of the backpropagation algorithm was that it could never be realized by a biological nervous system (this is in part due to the fact that it requires that error signals be sent backward through the network, something that neurons in your brain do not appear to do). However, is has now been shown that learning algorithms that are functionally equivalent to

backpropagation can be biologically plausible (e.g., O'Reilly, 1996).

One question you may have at this point is whether error-driven learning is at all plausible as a mechanism for human cognitive development. For example, is there any real-world analog of target activations that a biological agent could actually use? One source of target activations might be the supervised feedback we receive from other people in the environment: if a toddler sees a dog and screams "cat!" her caregiver may respond, "no, that's a *doggie*. Can you say doggie?" This type of scenario may be familiar and even plausible, but children seem to learn about vastly more in their environment than they are ever given explicit feedback on.

A more general source of target activations requires reconstruing the nature of learning as an attempt to improve *predictions* about what will be experienced in the environment (given what is currently being experienced). Consider once again the simple two-layer classical conditioning network described earlier. Recall that, at the start of training, activating the input unit that represents the light will lead to very little activation on the output units representing the meat and the shock because the weights are initialized to small random values. You can think about this in the following way: if you flash a light at the start of a conditioning experiment, the animal will register that fact but it will harbor no real *expectation* about what will happen next. However, when you repeatedly give the animal the food after turning on the light and shock it after ringing the bell, it will eventually learn that the light *predicts* food (and *not* shock) and the bell *predicts* pain (and *not* food). Dogs will readily acquire this information and adjust their expectations accordingly, signified by drooling for the light and whimpering for the bell. In terms of the learning model, the *actual* presence or absence of the food and shock in the environment serves as the relevant target activation. Thus, the error signal, or *prediction error*, is the difference between the network's expectation of what will happen (the activation over the output layer) and what actually does happen. On this view, then, learning is a process of reducing the prediction errors that continuously arise as a result of a cognitive agent attempting to predict what will happen next in its environment. Some researchers have even suggested that the minimization of prediction error may be a unifying principle of brain functioning (Clark, 2013; Friston, 2009), playing a fundamental role in cognitive development (McClelland, 1994).

Distributed Representation

In a previous section, we noted that many connectionist networks utilize distributed representations, in which a relevant stimulus is represented by the pattern of activation across an entire layer of units. There are several benefits to using a distributed coding scheme that makes it an especially powerful tool for thinking about cognitive processing (for an early review, see Hinton, McClelland, & Rumelhart, 1986). First, learning distributed representations naturally results in a system that can efficiently and automatically retrieve memories (specific patterns of activation across the units that have been encountered before) based on partial input cues (activating a subset of these units). Technically speaking, this is known as *content-addressable* memory. We observed this phenomenon earlier when we described an example of a single-layer network that self-organized via Hebbian learning, noting that it seems to capture something quite fundamental about how human memory works.

The second advantage is a built-in means of representing the similarity of different items, which plays an important role in cognitive processes like generalization and inference. In essence, two items are represented as similar to the extent that there is overlap in the activation patterns they produce over a set of distributed units. To pump your intuition on why this is the case, consider a network layer consisting of two units and imagine a graph where the x-axis represents the activation of one of the units and the y-axis represents the activation of the other unit. This two-dimensional graphical space can be used to plot all possible combinations of activation over those two units. Just as in a normal scatter plot, two points that are close together on the graph would be more similar than two points that are very far apart (i.e., in terms of unit activations). Thus, a simple geometric function (the distance between these points) can be used to determine how similar two patterns of activation are. This logic can be extended to an arbitrary number of dimensions, in which each dimension would correspond to another unit in the layer. Higher dimensional spaces allow a network to represent very complicated similarity structures that may be present in the input patterns, including hierarchical and contextual relationships (Elman, 1990; Rogers & McClelland, 2004; Saxe, McClelland, & Ganguli, 2013; Thibodeau, Flusberg, Glick, & Sternberg, 2013). Furthermore, these networks will naturally treat similar stimuli in a similar fashion (i.e., generalize based on similarity), which allows these models to make inferences about novel stimuli that were never presented during training.

A third advantage is that distributed representations are relatively robust and resistant to damage. Because an item

or event is represented across a whole layer of units, damaging a single unit or connection in the network will not eliminate all of the knowledge the network has about it. Although the overall performance of the network may suffer, it will still succeed to a certain degree. Indeed, performance will often gracefully degrade with increased amounts of damage, which mirrors research on the way brain damage affects human behavior. This is one of the reasons a majority of researchers assume that the brain itself relies on a distributed coding scheme (even though some studies have found that individual neurons often respond to very specific items, such as Jennifer Aniston; Quiroga, Reddy, Kreiman, Koch, & Fried, 2005; see Plaut & McClelland, 2010, for discussion). Indeed, it makes intuitive sense that whenever we encounter an item or event in the environment, we experience a complex pattern of activity across many thousands or millions of neurons. What's more, connectionist models have been effective at simulating and explaining the symptoms of a wide range of neurological disorders that result from damage or degeneration of the brain, including semantic dementia (e.g., Dilkina, McClelland, & Plaut, 2008) and acquired dyslexia (e.g., Plaut, McClelland, Seidenberg, & Patterson, 1996).

A final important point about learning in multilayer networks is that the similarity structure of the internal representations these networks learn is not simply a direct reflection of similarities in the inputs or target patterns used to train the network. Such *learned* similarity structure is an essential feature that allows these networks to capture abstract functional relations, such as those required to capture the structure of natural language, as we will see in the next section.

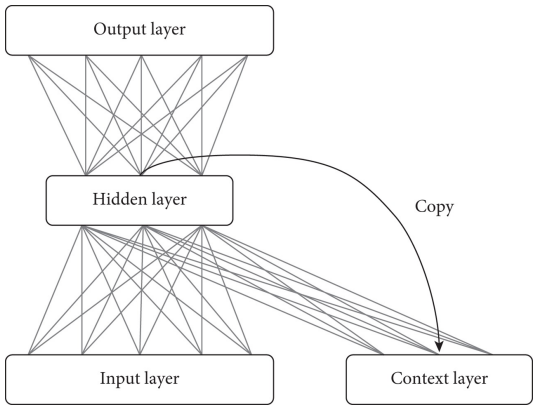
To understand how the principles of learning and distributed representation can result in surprisingly powerful cognitive behavior, we turn now to a discussion of another classic connectionist model, the simple recurrent network.

The Simple Recurrent Network

Temporal Patterns

When we look at a single word, we seem to experience it all at once, an observation that is built into the design of the IA model, in which a letter string input is processed in parallel at multiple levels of abstraction simultaneously. When we read a sentence or listen to spoken language, however, the event unfolds over time. Indeed, complex actions, musical compositions, board games, and many other activities have this sort of temporal quality. These episodes are not chaotic assemblies of random events, either; rather, they all possess a particular structure or texture. For example, the order of words in a sentence seems to conform to a set of patterns or rules defined by the syntax of the language (e.g., in English, a sentence must consist of a subject noun phrase and a predicate verb phrase, and these elements canonically occur in that order). Traditionally, researchers have therefore assumed that, to process language, we need to explicitly represent these syntactic rules and use that knowledge to properly parse what we are hearing or reading (Chomsky 1968; Pinker, 1984; 1999). If we did not represent the rules of grammar explicitly, how could we routinely generate and understand novel sentences? At this point, you will not be surprised to find out that a connectionist model might help shed light on this issue.

Model Architecture



Click to view larger

Figure 4 . A schematic of the simple recurrent network (SRN). It functions much like standard feed-forward network, except the pattern of activation over the hidden layer units is always copied in a one-to-one fashion to units in the context layer, so at time *t* the pattern of activation over the context units would be the same as the pattern of activation over the hidden units at time $t - 1$.

Elman (1990) created the SRN to determine whether (and how) connectionist models could learn complex patterns in sequences that unfold over time. A schematic of the model architecture is depicted in Figure 4. As you can see, the model looks somewhat like the three-layer, feed-forward networks we examined earlier. The units in the *input* layer receive information from the environment and propagate this activity to the units in the *output* layer, traversing the units in the *hidden* layer. However, the hidden units also send signals to another layer that consists of *context* units. The weights connecting the hidden layer to the context layer are fixed at full strength, so the activation over the hidden units is simply copied to the context layer at a one time-step delay. This means that at time $t + 1$, the activation over the context units is the same as the activation over the hidden units at time *t*. The context units then propagate their activity back to the hidden units via weights that are free to change over the course of training. This is where the notion of "recurrence" comes in since the hidden units integrate information about the current input state as well as their own previous activation state. You can think of this as sort of like a short-term memory for the network ("what was my internal experience a second ago?"), although that is only shorthand. In fact, because the network will experience a structured sequence of inputs, the activation of the context layer does not just provide information about the previous time step, but potentially about several previous time steps—and at the same time, predictions about future time-steps![2] To understand why, and what the consequences of this sort of architecture might be, we must examine the tasks that the network was designed to engage in.

Finding Structure in Time

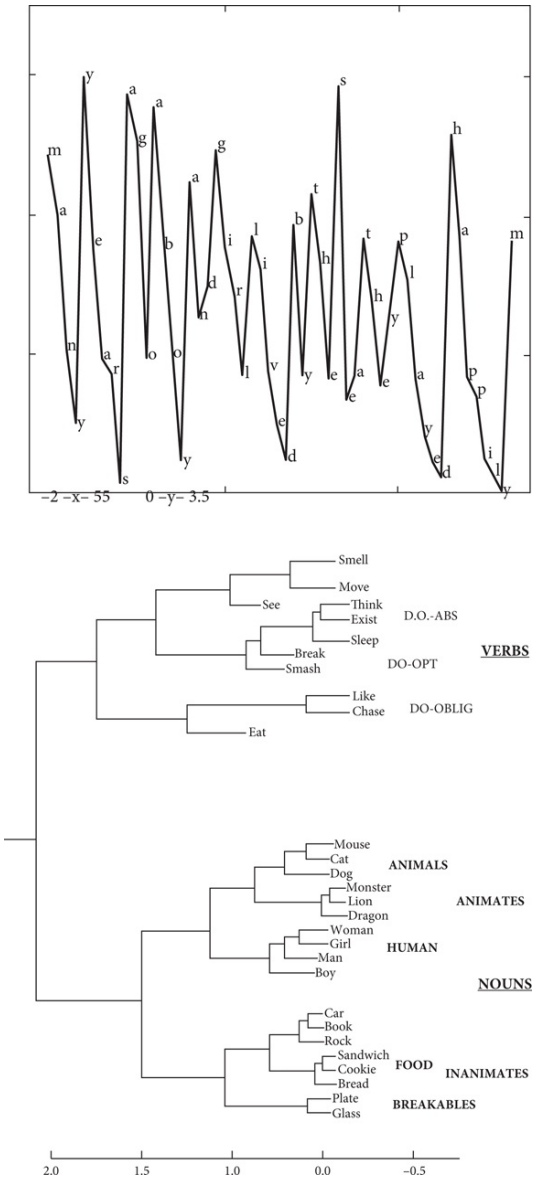
Elman presented the SRN with an ordered sequence of stimuli one at a time at the input layer and asked the network to respond with the *next* item in the sequence at the output layer. In other words, this model was explicitly set up as a *prediction* machine. The actual next item in the sequence would serve as the target activation pattern to calculate the prediction error signal, which was used to modify the weights via the backpropagation learning algorithm during training. In one simulation, the stimuli consisted of English letters, each of which was represented by a distributed pattern of activation over five input units. The network was presented with one letter at a time and tasked to predict the next letter in the sequence. The input sequences consisted of words arranged into sentences like:

MANYYEARSAGOABOYANDGIRLLIVEDBYTHESEA

When the network was presented with the first "A" to appear in this sequence, therefore, it was supposed to produce the representation for the letter "N" over the output units. When it was presented with the second "A" four time steps later, however, it was supposed to produce the letter "R" over the output units. In other words, the sentence *context* determined what the appropriate output should be for a given input letter. The network was trained on 200 sentences ranging from four to nine words long, drawing on a pool of 15 different words. This yielded a total of 1,270 words and 4,963 individual letter inputs during training.

What was striking was what happened when Elman graphed the prediction error (technically, the sum over the output units of the square of the difference between the target value and the network's output) for each letter in the sequence after the training phase. On the whole, the error signal was very high for letters appearing at the beginning of a word but gradually decreased for letters within that word (see Figure 5). When the first letter of the next word or sentence appeared, though, the error signal shot back up. Thus, prediction error accuracy can be thought of as a proxy measure of what sequences of letters the network perceived to be *words* in the input. Because each word appeared multiple times during training, the network was able to learn the patterns governing the relationship between the letters within these words and thus extract these nonrandom units out of a continuous stream of sequenced letter inputs. This is another case in which a particular type of knowledge (i.e., of word boundaries) emerges naturally out of a system based on distributed processing and representation that is simply learning to predict what will happen next. In recent years, research on infants and adults has shown that humans are in fact quite sensitive to these sorts of statistical relationships in sequences (e.g., Saffran, Aslin, & Newport, 1996).

In another simulation, Elman used a different distributed pattern to represent each of 29 different words as inputs and targets for the network. The words included different types of nouns (e.g., man, woman, cat, cookie) and verbs (e.g., think, move, break). He then generated 10,000 different sentences by combining the words according to one of a number of different simplified syntactic frames (e.g., noun + transitive verb + noun). The network was presented with all of these sentences one word at a time during training, and its goal was to predict the next word in each sentence. Once again, the network was able to learn the temporal patterns in the input sequence quite well. (The network's outputs could be understood as weighted averages of the possible successors of the input word in the given context: in similar simulations in which each word was represented by activating just a single input unit, Elman found that when it was given a noun at the input, the units representing verbs would become more active at the output layer [compared to nouns], and verbs that tended to appear after that particular noun during training would be the most active).



Click to view larger

Figure 5 . The graph on the left shows the prediction error signal for each letter in the simple recurrent network (SRN) letter sequence learning task after training. Note that letters with higher errors signal the beginnings of words, and error drops steadily as the word unfolds before shooting up again at the start of the next word. The graph on the right is a cluster diagram depicting the similarity relationships between the internal representations of the words in the SRN sentence learning task after training. Note that the network clusters words by both grammatical category (nouns vs. verbs) and meaning (animates vs. inanimates). From J. L. Elman, Finding structure in time. *Cognitive Science, 14*, 179–211. figure 6, page 194; right panel from J. L. Elman, Finding structure in time. *Cognitive Science, 14*, 179–211. figure 7, page 200. Reprinted

with permission from the Cognitive Science Society, Inc.

Additional analyses revealed just how much the network had learned about both the syntax and semantics of the linguistic environment. Elman examined the similarity structure of the hidden unit activation patterns associated with each input (we explored the logic of this sort of analysis earlier in our discussion of the advantages of distributed representations). Although the internal representations were unique for each word, the verbs formed one large cluster (i.e., their activation patterns were all somewhat similar to one another), while the nouns formed another cluster (see Figure 5). What's more, within each of these larger clusters, the network was sensitive to a surprising range of grammatical and semantic distinctions. For example, patterns for transitive verbs (those that are followed by an object, like *saw,* as in *John saw a dog*) formed one cluster while patterns for intransitive verbs (those that do not have objects, like *slept*) formed another, and patterns for animate nouns clustered separately from inanimate nouns. The network was even sensitive to the concept of *gender*, clustering *girl* and *woman* together and *boy* and *man* together. Thus, the network seemed to learn to differentiate the words according to both grammatical category *and* meaning.

Now, remember, the only thing the network was ever explicitly *doing* was attempting to predict the next word in a sentence, and, critically, none of the syntactic or semantic properties of words was explicitly represented in the patterns Elman used to train the network. Thus, these grammatical and semantic representations in the hidden unit activations emerged as a consequence of learning: that is, modifying the weights so that these patterns of activity appeared given a particular input word occurred simply through the process of error-driven learning. In a sense, then, the network has learned something about the grammatical categories and co-occurrences of members of these categories that structured the language to which it was exposed (e.g., nouns tend to be followed by verbs; transitive verbs are followed by object nouns, but intransitive verbs are not). And yet nowhere does it explicitly represent these categories or rules of syntax. Similarly, it has learned something about which words have similar or conceptually related meanings (e.g., woman and girl; man and boy) but only because these words tend to be used in similar *contexts* in the language, and thus they tend to *predict* similar words.

Lessons Learned

Research using the basic SRN architecture to simulate other (and more advanced) facets of language processing has been widespread in the past 20 years. As one key example (Elman, 1993), these models were able to learn how to exploit long-distance dependencies thought to require explicit representation of complex grammatical rules. That is, in the sentence "The boy who chased the girls like? ice cream," the model could learn that it should add a letter "s" to the verb "like" (where we have a "?") so that this verb agrees in number with the grammatically correct (but distant) noun phrase "the boy," rather than the nearby "the girls." Advanced versions of these models are now used in state-of-the-art machine language processing systems (Socher et al., 2013).

The architecture has also been extended to other domains that seem to depend on learning complex, structured sequences. Botvinick and Plaut (2004), for example, used this type of network to simulate the ability to learn and execute everyday sequences of actions like making a cup of tea. Routine actions like this are *hierarchically* structured: the larger action of "making tea" consists of smaller actions like "boiling water," which consists of smaller actions like "reach into the cupboard and grab the teapot," and so on. Traditional explanations for how we learn these actions invoked the formation of explicit action *schemas* (analogous to the room schemas we described earlier) that organize knowledge in a structured fashion. Botvinick and Plaut (2004) showed that an SRN model could learn these action sequences in a flexible and context-dependent way without explicitly representing this hierarchical information. In other words, "schematic" knowledge might simply be an emergent product of a cognitive agent learning about action sequences that have a particular type of structure.

The key lesson we wish to highlight in this section is as follows: connectionist models are exquisitely sensitive to the statistical structure in their environment (i.e., the items and sequences of items that they are trained with). They can extract this structure in a fairly straightforward manner, even when this information is embedded in temporally extended sequences, via a domain-general learning process that forms distributed representations of the input. This does not require any *explicit* representation of the rules that appear to govern regularities in the input. Nonetheless, over the course of development, a network may acquire *implicit* knowledge that approximates these sorts of rules (at least to the extent that this helps the network make predictions about the environment). Yet again, behavior indicating sensitivity to the regularities in a domain like language does not imply the existence of

knowledge of explicit rules that capture these regularities within the cognitive system. Behavior that conforms to certain regular patterns may be the emergent consequence of a more basic set of computational principles that includes statistical, error-driven learning and distributed representation.

Future Directions

Connectionist ideas have been floating around since the early days of psychological science, and, with ongoing methodological and theoretical advances, the future looks especially promising. On the technical side, new learning algorithms, training procedures, and network architectures will continue to enhance the power and versatility of PDP models. In recent years, for instance, researchers have made strides creating networks that can self-organize with additional layers of processing units, which increases their ability to simulate human levels of performance in pattern recognition tasks like object and letter perception (Hinton, 2007). Before these recent innovations, adding an additional layer to a network drastically affected training times, making such networks impractical for use in some settings, but novel training schemes are now available to wire up multiple layers relatively efficiently. Such models are, at present, the state of the art in machine perception—for example, they are being used in smartphone devices to recognize speech (Deng, Hinton, & Kingsbury, 2013). Continued progress in the development of new learning procedures promises to help bridge the gap between human and network performance, as well as yielding many practical applications in the realms of machine learning and artificial intelligence.

Additional technical advances should help us understand how and why connectionist models work the way they do at a more precise, mathematical level of explanation. Connectionist models are sometimes criticized for being too opaque or mysterious because it is not always obvious how or why a particular networks works, just that the learning algorithm has led the model to converge on one possible solution to a problem. However, researchers have started to apply advanced analytic techniques to uncover the principles governing individual network behavior. For example, behavioral research has shown that semantic development in children follows a specific developmental trajectory: children learn to make broader conceptual distinctions (plant vs. animal) before they make finer ones (bird vs. fish), and they typically show sudden, stage-like transitions throughout this period rather than a gradual shift in performance. Rogers and McClelland (2004) showed that these and related findings could be captured by a particular class of feed-forward neural network. Recently, Saxe, McClelland, and Ganguli (2013) have formally demonstrated *why* the learning dynamics of these networks results in this particular pattern of development. In particular, they used mathematical analyses to link the learning dynamics of these models to the pattern of similarity relationships among the training patterns. For example, they showed that when the similarity relationships can be well-captured by a branching tree, learning will inevitably follow this particular developmental trajectory. This type of work promises to bring a new level of rigor and precision to the construction and use of connectionist networks in a variety of domains.

Related approaches to classifying network behavior will also help researchers identify the key similarities and differences between connectionism and other modeling frameworks in the cognitive sciences. This may help bridge some of the theoretical gaps that currently exist between camps and point the way toward a more unified understanding of cognition and behavior. For instance, connectionist models can be shown to approximate the behavior characteristic of probabilistic Bayesian models (McClelland, 2013), an increasingly popular approach that casts cognition as an optimal, probabilistic inference under uncertainty (for review, see Tenenbaum, Kemp, Griffiths, & Goodman, 2011). At the same time, however, connectionist networks seem to allow for more flexible and context-sensitive behavior and greater integration with what we know about brain structure and function (McClelland et al., 2010).

The behavior of connectionist models can also be described using the language of dynamical systems theory (DST), a mathematical framework that specifies the change in the state of a system over time (McClelland & Vallabha, 2009). Cognitive scientists have increasingly used the formalism of DST because it naturally captures the temporal dynamics of perception and action as they unfold in real time for physical agents embedded in physical environments while avoiding some of the philosophical pitfalls associated with traditional, representational theories of mind (Chemero, 2009; Thelen & Smith, 1994). Connectionist models not only display their own pattern of internal dynamics over time (which can be described by DST), but researchers can actually use these networks as control mechanisms or “brains” inside simulated or real-world robots. In this case, the network becomes just one element

of a larger dynamical system that includes the body and structure of the robot and the environment the robot is situated in (Beer, 2003; Chemero, 2009). Thus, connectionist models and dynamical systems approaches can be complementary (Spencer, Thomas, & McClelland, 2009).

Another exciting development is the use of connectionist methods to explain the nature and origin of some of the most sophisticated aspects of human cognition, from reasoning with metaphor and analogy to mathematical problem solving. Traditionally, these hallmarks of human intelligence have been offered as proof of domain-specific cognitive mechanisms that require the use of explicit, structured, symbolic representations. Recently, however, there has been a great deal of progress in our understanding of how these high-level abilities might emerge in a distributed connectionist network. For example, Leech, Mareschal, and Cooper (2008) showed that a connectionist model could learn to solve basic analogy problems by treating them as a form of relational priming. More recently, Thibodeau and colleagues (2013) adapted the network used by Rogers and McClelland (2004) to simulate semantic development in order to simulate our capacity to draw analogical inferences between two structurally similar domains (see also Kollias & McClelland, 2013). These demonstrations not only advance our understanding of the capabilities of connectionist models, but they also provide a new way of thinking about higher level cognitive functions in general. That is, some forms of analogical reasoning may not require specialized cognitive machinery but may arise spontaneously over the course of development due to the operation of a domain-general learning mechanism and a particular set of architectural constraints (see also Rogers & McClelland, 2008).

Flusberg and colleagues (2010) took a very similar approach to shed light on the mechanisms that support our ability to conceive of abstract concepts like time, justice, and the mind. We tend to use metaphors and analogies to talk about these abstract domains, borrowing language that is commonly used to describe more concrete aspects of experience (Lakoff & Johnson, 1980). For example, we use spatial language to talk about time, as when we say, "those *long* meetings are *close* together." Behavioral research has shown that priming people to think about space can impact their reasoning about time, suggesting that we really do think metaphorically in some sense (e.g., Boroditsky & Ramscar, 2002). Flusberg et al. (2010) used a connectionist model to show that this pattern of behavior is a natural consequence of a semantic system that integrates linguistic and perceptual information throughout learning but only when the specific concrete and abstract domains share some relational structure.

Part of the theoretical motivation for this work comes from the increasingly popular *embodied* approaches to cognitive science, which includes the dynamical systems ideas outlined earlier. Proponents of the embodied view suggest that cognition is grounded in sensory-motor processing and shaped and constrained by the physical structure of the agent and the environment it is situated in (Chemero, 2009; Clark, 1997; Gibbs, 2006). The fact that even abstract thinking seems to be grounded in more concrete physical experiences lends support to these ideas. We see the fusion of embodied and connectionist ideas as an important synthesis because the body of any agent will influence the information that the brain of the agent has access to. In other words, the input patterns that a neural network learns about are necessarily constrained by the physical structure of the agent itself. Taken together, these trends suggest that connectionism will prove to be an invaluable tool for researchers interested in understanding not just how the brain supports cognitive functioning, but how the brain, body, and world function together and evolve over the course of development.

Conclusion

In our introduction, we noted that cognitive scientists are interested in reconciling the *manifest image* of everyday experience, which includes mental phenomena like perceptions and memories, with the *scientific image* of the universe, which is characterized by physical phenomena like molecules and neurons (Dennett, 2013; Sellars, 1963). In this chapter, we have tried to demonstrate how connectionism offers a unique set of computational and theoretical tools for addressing this perennial issue.

Connectionist models are inspired by the principles of information processing that characterize the structure and function of the brain: namely, a large number of simple processing units linked together into complex networks that communicate in parallel via connections of varying strength that can be modified by experience. These networks can extract complex statistical patterns from their input and naturally settle on solutions to constraint satisfaction problems through the interactions of their constituent parts. These properties endow connectionist models with the ability to exhibit complex behavior that mirrors human cognitive performance in many domains, from perception

and memory to language processing and analogical inference. Although traditional approaches tend to treat these psychological functions as distinct processes that must be explicitly instantiated in a cognitive model, connectionism considers how a basic set of computational principles might give rise to many different forms of complex behavior. In other words, an emphasis is placed on the commonalities underlying various cognitive abilities rather than on their differences. Although it may appear as though behavior in a given domain is governed by a specially engineered process or set of rules, the connectionist lesson is that these rules need not be explicitly represented in the cognitive system.

A useful analogy can be made to the work of Charles Darwin. The structure and organization of the biological world has the appearance of design, and so it was long assumed that to explain this appearance we had to posit an intelligent designer (i.e., God). However, Darwin's theory of evolution by natural selection showed how the *appearance* of design could emerge from simpler processes of competition, variation, and heredity operating over the (long) course of history. It may sometimes be useful to talk about natural selection itself "designing" certain traits as if it were an engineer that could create an optimal solution, but a closer look at the microstructure of organisms reveals many elements that do not fit the design framework. Indeed, the evolutionary biologist Stephen Jay Gould argued that "poor" design elements in nature are some of the best evidence for evolution (Gould, 1980). For instance, the peculiar "thumb" that pandas use to grasp their food is not a finger at all (the panda already has five), but an enlarged radial sesamoid bone. Gould points out that it is far from an ideal solution, but one dictated by structures already available in the bears from which the panda descended and that its evolution likely occurred through a mutation that produced benign but unnecessary side effects.

In a similar way, a great deal of human behavior, from decision making to language comprehension, can often be described as being highly structured and governed by underlying rules. Thus, it was long assumed that we had to posit specialized internal mechanisms that explicitly instantiated the appropriate plans for each cognitive faculty or complex behavior. In fact, a closer look at human behavior often reveals that much of it is less rigidly rulelike than it appears at first blush, as we saw in the case of context effects in letter perception and the putative role of orthographic rules. Connectionist modeling is important because it shows us how complex, rulelike behavior can emerge from simpler processes interacting with one another in a mutually constraining fashion (often over a period of learning and development).

Some scholars, inspired as we are by Borges's mapmaking fable, might argue that it is in fact a useful and worthwhile simplification to describe individual cognitive functions or behaviors by appealing to domain-specific rules and principles. Although we agree this can sometime provide a useful simplification, if taken as literally correct, it can be misleading. One needs, for example, to then provide a mechanism for constructing rules or for replacing rules used at one stage of development with better ones used at later stages. If one sees the rules as approximate characterizations of the emergent properties of the underlying mechanism, a mechanism for constructing rules and replacing them with others may no longer be necessary. Thus, connectionist models support a novel way of thinking about the nature and origins of mental life as the emergent consequence of a system that is based on principles of parallel processing, distributed representation, and statistical learning that interacts with its environment over the course of development.

References

Anderson, J. A. (1977). Neural models with cognitive implications. In D. LaBerge & S. J. Samuels (Eds.), *Basic processes in reading perception and comprehension* (pp. 27–90). Hillsdale, NJ: Erlbaum.

Bar, M. (2004). Visual objects in context. *Nature Reviews: Neuroscience*, *5*, 617–629.

Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, *11*(4), 209–243.

Borges, J. L. (1998). On exactitude in science. In *Collected fictions* (p. 325). (A. Hurley, Trans.). London: Penguin.

Boroditsky, L., & Ramscar, M. (2002). The roles of body and mind in abstract thought. *Psychological Science*, *13*(2), 185–188.

Botvinick, M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review*, *111*, 395–429.

Burton, A. M., Bruce, V., & Johnston, R. A. (1990). Understanding face recognition with an interactive activation model. *British Journal of Psychology*, *81*(3), 361–380.

Chemero, A. (2009). *Radical embodied cognitive science*. Cambridge, MA: MIT Press.

Chomsky, N. (1968). *Language and mind*. New York: Harcourt, Brace & World.

Clark, A. (1997). *Being there: Putting brain, body, and world together again*. Cambridge, MA: MIT Press.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*, 181–204.

Deng, L., Hinton, G. E., & Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In *IEEE International Conference on Acoustic Speech and Signal Processing* (ICASSP 2013).

Dennett, D. C. (2013). *Intuition pumps and other tools for thinking*. New York: W. W. Norton & Company.

Dilkina, K., McClelland, J. L., & Plaut, D. C. (2008). A single-system account of semantic and lexical deficits in five semantic dementia patients. *Cognitive Neuropsychology*, *25*(2), 136–164.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*, 71–99.

Elman, J. L., Bates, E., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking innateness: Connectionism in a developmental framework*. Cambridge, MA: MIT Press.

Flusberg S. J., Thibodeau, P. H., Sternberg, D. A., & Glick, J. J. (2010). A connectionist approach to embodied conceptual metaphor. *Frontiers in Psychology*, *1*, 197.

Freeman, J. B., & Ambady, N. (2011). A dynamic interactive theory of person construal. *Psychological Review*, *118*, 247–279.

Freud, S. (1895). Project for a scientific psychology. In J. Strachey (Ed.), *The standard edition of the complete psychological works of Sigmund Freud*. London: The Hogarth Press and the Institute of Psycho-Analysis.

Friston, K. (2009). The free-energy principle: A rough guide to the brain? *Trends in Cognitive Sciences*, *13*, 293–301.

Gibbs, R. W. (2006). *Embodiment and cognitive science*. Cambridge, UK: Cambridge University Press.

Gould, S. J. (1980). *The panda's thumb: More reflections in natural history*. New York: W. W. Norton.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological approach*. New York: John Wiley & Sons.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence*, *40*(1), 185–234.

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in Cognitive Sciences*, *11*(10), 428–434.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 77–109). Cambridge, MA: MIT Press.

James, W. (1890). *Principles of psychology*. New York: Holt.

Kollias, P., & McClelland, J. L. (2013). Context, cortex, and associations: A connectionist developmental approach to verbal analogies. *Frontiers in Psychology*, *4*, 857. doi: 10.3389/fpsyg.2013.00857

Kumaran, D., & McClelland, J. L. (2012). Generalization through the recurrent interaction of episodic memories: A model of the hippocampal system. *Psychological Review*, *119*, 573–616.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. Chicago, IL: University of Chicago Press.

Leech, R., Mareschal, D., & Cooper, R. (2008). Analogy as relational priming: A developmental and computational perspective on the origins of a complex cognitive skill. *Behavioral and Brain Sciences*, *31*, 357–378.

Lømo, T. (1966). Frequency potentiation of excitatory synaptic activity in the dentate area of the hippocampal formation. *Acta Physiologica Scandinavica*, *68*(Suppl. 277), 128.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

McClelland, J. L. (1981). Retrieving general and specific information from stored knowledge of specifics. *Proceedings of the Third Annual Conference of the Cognitive Science Society*, 170–172.

McClelland, J. L. (1994). The interaction of nature and nurture in development: A parallel distributed processing perspective. In P. Bertelson, P. Eelen, & G. d'Ydewalle (Eds.), *International perspectives on psychological science* Vol. 1: *Leading themes*. London: Erlbaum.

McClelland, J. L. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science*, *1*(1), 11–38.

McClelland, J. L. (2010). Emergence in cognitive science. *Topics in Cognitive Science*, *2*, 751–770.

McClelland, J. L. (2013). Integrating probabilistic models of perception and interactive neural networks: A historical and tutorial review. *Frontiers in Psychology*, *4*, 503.

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., & Smith, L. B. (2010). Letting structure emerge: Connectionist and dynamical systems approaches to understanding cognition. *Trends in Cognitive Sciences*, *14*, 348–356.

McClelland, J. L., & Elman, J. L. (1986). The TRACE model of speech perception. *Cognitive Psychology*, *18*, 1–86.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*, 375–407.

McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). The appeal of parallel distributed processing. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 3–44). Cambridge, MA: MIT Press.

McClelland, J. L., & Vallabha, G. (2009). Connectionist models of development: Mechanistic dynamical models with emergent dynamical properties. In J. P. Spencer, M. S. C. Thomas, & J. L. McClelland (Eds.), *Toward a unified theory of development: Connectionism and dynamic systems theory reconsidered* (pp. 3–24). New York: Oxford.

Minsky, M., & Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. Cambridge, MA: MIT Press.

Munakata, Y., & McClelland, J. L. (2003). Connectionist models of development. Contribution to a special issue on Dynamical Systems and Connectionist Models. *Developmental Science*, *6*(4), 413–429.

Oliva, A., & Torralba, A. (2007). The role of context in object recognition. *Trends in Cognitive Sciences*, *11*, 520–527.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural computation*, *8*(5), 895–938.

Palmer, S. E. (1975). The effects of contextual scenes on the identification of objects. *Memory and Cognition*, *3*, 519–526.

Pinker, S. (1984). *Language learnability and language development*. Cambridge, MA: Harvard University Press.

Pinker, S. (1999). *Words and rules*. London: Weidenfeld & Nicolson.

Plaut, D. C., & McClelland, J. L. (2010). Locating object knowledge in the brain: A critique of Bowers' (2009) attempt to revive the grandmother cell hypothesis. *Psychological Review*, *117*, 284–288.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, *103*, 56–115.

Quiroga R. Q., Reddy, L., Kreiman, G., Koch, C., & Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, *435*, 1102–1107.

Reicher, G. M. (1969). Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of Experimental Psychology*, *81*, 274–280.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning II: Current research and theory* (pp. 64–99). New York: Appleton-Century-Crofts.

Rogers, T. T., & McClelland, J. L. (2004). *Semantic cognition: A parallel distributed processing approach.* Cambridge, MA: MIT Press.

Rogers, T. T., & McClelland, J. L. (2008). Precis of semantic cognition: A parallel distributed processing approach. *Behavioral and Brain Sciences*, *31*, 689–749.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.

Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Washington, DC: Spartan Books.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986a). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* Vol. 1: *Foundations* (pp. 318–362). Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. The context enhancement effect and some tests and extensions of the model. *Psychological Review*, *89*, 60–94.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group. (1986b). *Parallel distributed processing: Explorations in the microstructure of cognition. Volume I*. Cambridge, MA: MIT Press.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986c). Schemata and sequential thought processes in PDP models. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing* (Vol. 2, pp. 7–57). Cambridge, MA: MIT Press.

Saffran, J. R., Aslin, R. N., & Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science*, *274*(5294), 1926–1928.

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Learning hierarchical category structure in deep neural networks. In M. Knauff, M. Paulen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th annual meeting of the Cognitive Science Society* (pp. 1271–1276). Austin, TX: Cognitive Science Society.

Sellars, W. (1963). Philosophy and the scientific image of man. In *Science, perception and reality*. London: Routledge & Kegan Paul, 35–78.

Smolensky, P. (1986). Neural and conceptual interpretations of parallel distributed processing models. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition.* Vol. 2: *Psychological and biological models* (pp. 390–431). Cambridge, MA: MIT Press/Bradford Books.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631–1642). Seattle, WA: Association for Computational Linguistics. **http://www.aclweb.org/anthology/D13-1170**

Spencer, J. P., Thomas, M. S. C., & McClelland, J. L. (2009). *Toward a unified theory of development: Connectionism and dynamic systems theory reconsidered*. New York: Oxford.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, *331*(6022), 1279–1285.

Thelen, E., & Smith, L. B. (1994). *A dynamic systems approach to the development of cognition and action*. Cambridge, MA: MIT Press.

Thibodeau, P., Flusberg, S. J., Glick, J. J., & Sternberg, D. A. (2013). An emergent approach to analogical inference. *Connection Science*, *25*(1), 27–53.

Wandell, B. A. (1995). *Foundations of vision*. Sinauer Associates.

Notes:

([1]) Rescorla and Wagner (1972) trained what amounts to a two-layer localist networks using the delta rule to simulate a wide range of findings in the classical conditioning literature that could not be easily understood using more traditional theories.

([2]) Simple recurrent networks seem to embody this statement from T. S. Eliot: “Time present and time past are both perhaps present in time future, and time future contained in time past.” From Eliot, T. S. (1944). Burnt Norton, I:1–3. In T. S. Eliot, *Four Quartets,* London, UK: Faber & Faber.

Stephen J. Flusberg

Stephen J. Flusberg, Purchase College, State University of New York

James L. McClelland

James L. McClelland, Stanford University