Connectionism

Connectionism is an approach to the study of human cognition that utilizes mathematical models, known as connectionist networks or artificial neural networks. Often, these come in the form of highly interconnected, neuron-like processing units. There is no sharp dividing line between connectionism and computational neuroscience, but connectionists tend more often to abstract away from the specific details of neural functioning to focus on high-level cognitive processes (for example, recognition, memory, comprehension, grammatical competence and reasoning). During connectionism's ideological heyday in the late twentieth century, its proponents aimed to replace theoretical appeals to formal rules of inference and sentence-like cognitive representations with appeals to the parallel processing of diffuse patterns of neural activity.

Connectionism was pioneered in the 1940s and had attracted a great deal of attention by the 1960s. However, major flaws in the connectionist modeling techniques were soon revealed, and this led to reduced interest in connectionist research and reduced funding. But in the 1980s connectionism underwent a potent, permanent revival. During the later part of the twentieth century, connectionism would be touted by many as the brain-inspired replacement for the computational artifact-inspired 'classical' approach to the study of cognition. Like classicism, connectionism attracted and inspired a major cohort of naturalistic philosophers, and the two broad camps clashed over whether or not connectionism had the wherewithal to resolve central quandaries concerning minds, language, rationality and knowledge. More recently, connectionist techniques and concepts have helped inspire philosophers and scientists who maintain that human and non-human cognition is best explained without positing inner representations of the world. Indeed, connectionist techniques are now very widely embraced, even if few label themselves connectionists anymore. This is an indication of connectionism’s success.

Table of Contents

1. McCulloch and Pitts

In 1943, neurophysiologist Warren McCulloch and a young logician named Walter Pitts demonstrated that neuron-like structures (or *units*, as they were called) that act and interact purely on the basis of a few neurophysiologically plausible principles could be wired together and thereby be given the capacity to perform complex logical calculation (McCulloch & Pitts 1943). They began by noting that the activity of neurons has an all-or-none character to it – that is, neurons are either 'firing' electrochemical impulses down their lengthy projections (axons) towards junctions with other neurons (synapses) or they are inactive. They also noted that in order to become active, the net amount of excitatory influence from other neurons must reach a certain threshold and that some neurons must inhibit others. These principles can be described by mathematical formalisms, which allows for calculation of the unfolding behaviors of networks obeying such principles. McCulloch and Pitts capitalized on these facts to prove that neural networks are capable of performing a variety of logical calculations. For instance, a network of three units can be configured so as to compute the fact that a conjunction (that is, two complete statements connected by 'and') will be true only if both component statements are true (Figure 1). Other logical operations involving disjunctions (two statements connected by 'or') and negations can also be computed. McCulloch and Pitts showed how more complex logical calculations can be performed by combining the networks for simpler calculations. They even proposed that a properly configured network supplied with infinite tape (for storing information) and a read-write assembly (for recording and manipulating that information) would be capable of computing whatever any given Turing machine (that is, a machine that can compute any computable function) can.

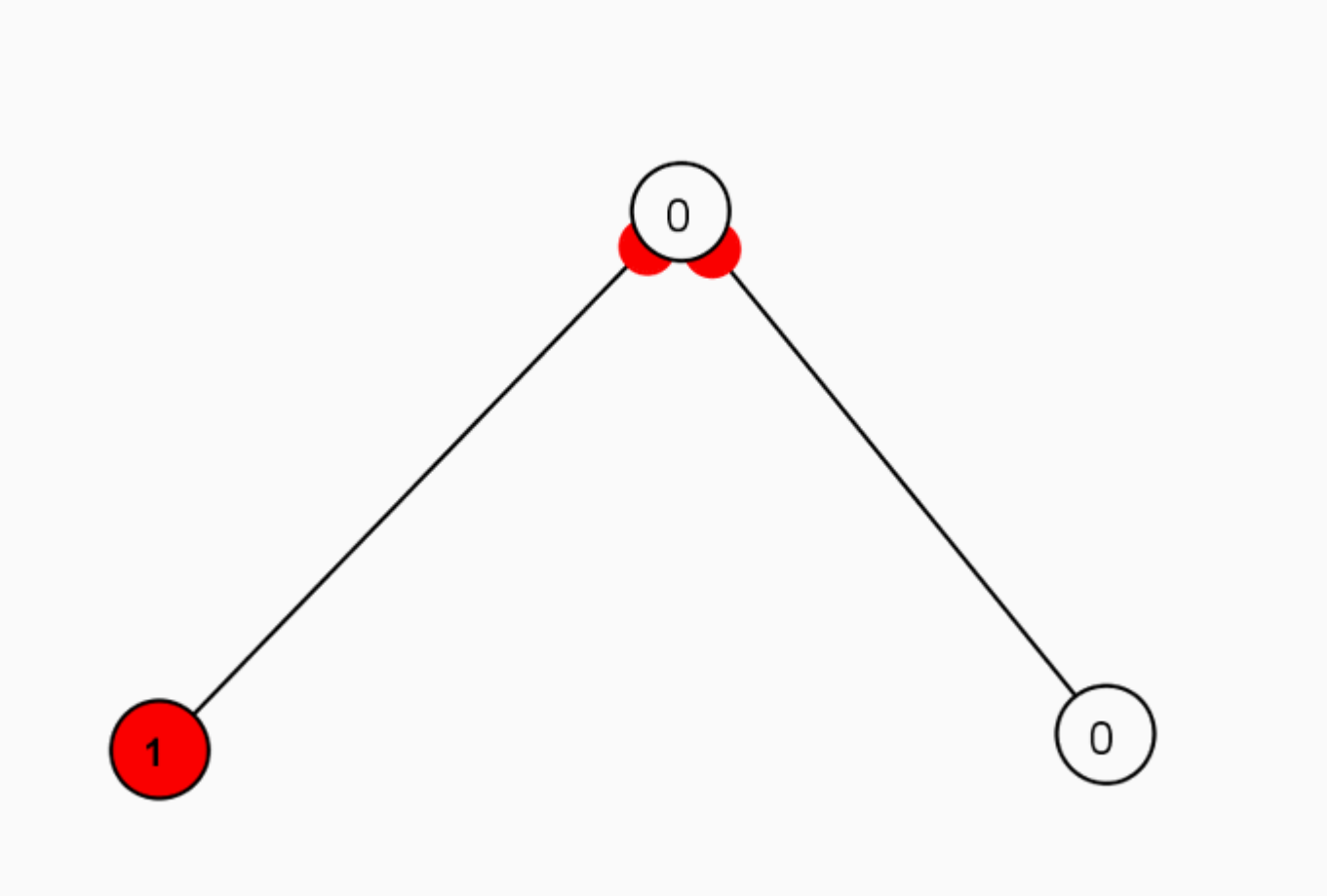

Figure 1: Conjunction Network We may interpret the top (output) unit as representing the truth value of a conjunction (that is, activation value 1 = true and 0 = false) and the bottom two (input) units as representing the truth values of each conjunct. The input units each have an excitatory connection to the output unit, but for the output unit to activate the sum of the input unit activations must still exceed a certain threshold. The threshold is set high enough to ensure that the output unit becomes active just in case both input units are activated simultaneously. Here we see a case where only one input unit is active, and so the output unit is inactive. A disjunction network can be constructed by lowering the threshold so that the output unit will become active if either input unit is fully active. [Created using Simbrain 2.0]

Somewhat ironically, these proposals were a major source of inspiration for John von Neumann's work demonstrating how a universal Turing machine can be created out of electronic components (vacuum tubes, for example) (Franklin & Garzon 1996, Boden 2006). Von Neumann's work yielded what is now a nearly ubiquitous programmable computing architecture that bears his name. The advent of these electronic computing devices and the subsequent development of high-level

programming languages greatly hastened the ascent of the formal classical approach to cognition, inspired by formal logic and based on sentence and rule (see Artificial Intelligence). Then again, electronic computers were also needed to model the behaviors of complicated neural networks.
For their part, McCulloch and Pitts had the foresight to see that the future of artificial neural networks lay not with their ability to implement formal computations, but with their ability to engage in messier tasks like recognizing distorted patterns and solving problems requiring the satisfaction of multiple 'soft' constraints. However, before we get to these developments, we should consider in a bit more detail some of the basic operating principles of typical connectionist networks.

2. Parts and Properties of Connectionist Networks

Connectionist networks are made up of interconnected processing units which can take on a range of numerical *activation levels* (for example, a value ranging from 0 – 1). A given unit may have incoming connections from, or outgoing connections to, many other units. The excitatory or inhibitory strength (or *weight*) of each connection is determined by its positive or negative numerical value. The following is a typical equation for computing the *influence* of one unit on another:
$\text{Influence}_{iu} = a_i * w_{iu}$
This says that for any unit *i* and any unit *u* to which it is connected, the influence of *i* on *u* is equal to the product of the activation value of *i* and the weight of the connection from *i* to *u*. Thus, if $a_i = 1$ and $w_{iu} = .02$, then the influence of *i* on *u* will be 0.02. If a unit has inputs from multiple units, the*net influence* of those units will just be the sum of these individual influences.
One common sort of connectionist system is the two-layer feed-forward network. In these networks, units are segregated into discrete input and output layers such that connections run only from the former to the latter. Often, every input unit will be connected to every output unit, so that a network with 100 units, for instance, in each layer will possess 10,000 inter-unit connections. Let us suppose that in a network of this very sort each input unit is randomly assigned an activation level of 0 or 1 and each weight is randomly set to a level between -0.01 to 0.01. In this case, the activation level of each output unit will be determined by two factors: the *net influence* of the input units; and the degree to which the output unit

is sensitive to that influence, something which is determined by its *activation function*. One common activation function is the step function, which sets a very sharp threshold. For instance, if the threshold on a given output unit were set through a step function at 0.65, the level of activation for that unit under different amounts of net input could be graphed out as follows:

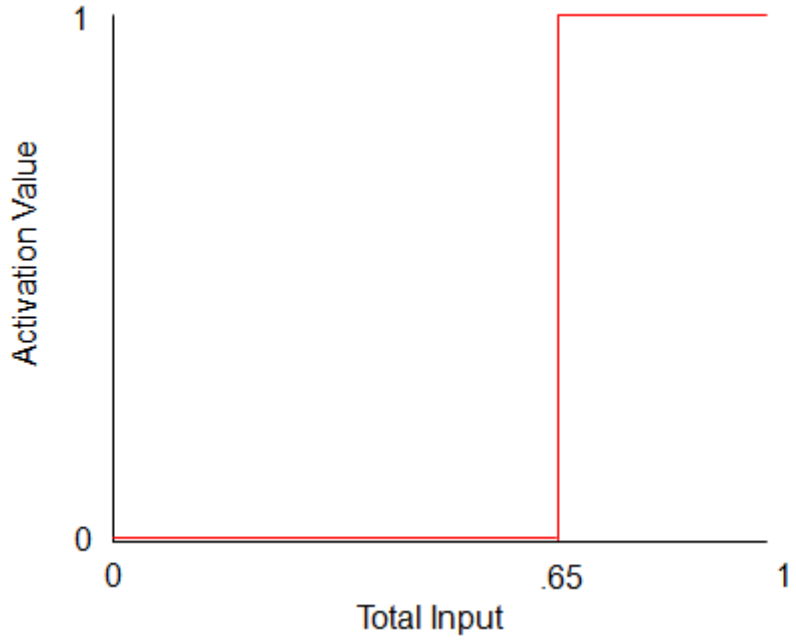

Figure 2: Step Activation Function

Thus, if the input units have a net influence of 0.7, the activation function returns a value of 1 for the output unit's activation level. If they had a net influence of 0.2, the output level would be 0, and so on. Another common activation that has more of a sigmoid shape to it – that is, graphed out it looks something like this:

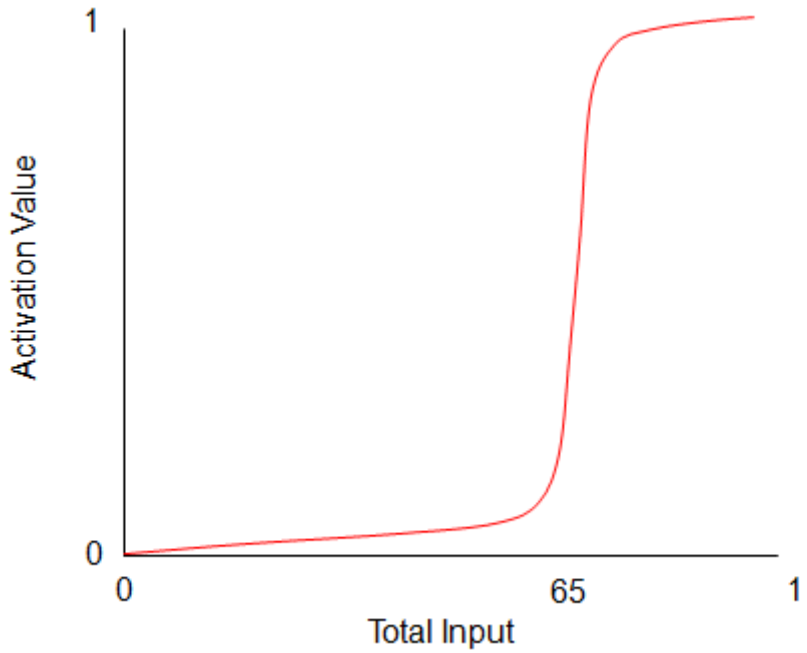

Figure 3: Sigmoid Activation Function

Thus, if our net input were 0.7, the output unit would take on an activation value somewhere near 0.9.

Now, suppose that a modeler set the activation values across the input units (that is, encodes an input *vector*) of our 200 unit network so that some units take on an activation level of 1 and others take on a value of 0. In order to determine what the value of a single output unit would be, one would have to perform the procedure just described (that is, calculate the net influence and pass it through an activation function). To determine what the entire *output vector* would be, one must repeat the procedure for all 100 output units.

As discussed earlier, the truth-value of a statement can be encoded in terms of a unit's activation level. There are, however, countless other sorts of information that can be encoded in terms of unit activation levels. For instance, the activation level of each input unit might represent the presence or absence of a different animal characteristic (say, "has hooves," "swims," or "has fangs,") whereas each output unit represents a particular kind of animal ("horse," "pig," or "dog,"). Our goal might be to construct a model that correctly classifies animals on the basis of their features. We might begin by creating a list (a *corpus*) that contains, for each animal, a specification of the appropriate input and output vectors. The challenge is then to set the weights on the connections so that when one of these input

vectors is encoded across the input units, the network will activate the appropriate animal unit at the output layer. Setting these weights by hand would be quite tedious given that our network has 10000 weighted connections. Researchers would discover, however, that the process of weight assignment can be automated.

3. Learning Algorithms

a. Hebb's Rule

The next major step in connectionist research came on the heels of neurophysiologist Donald Hebb's (1949) proposal that the connection between two biological neurons is strengthened (that is, the presynaptic neuron will come to have an even stronger excitatory influence) when both neurons are simultaneously active. As it is often put, "neurons that fire together, wire together." This principle would be expressed by a mathematical formula which came to be known as *Hebb's rule*:

Change of $\text{weight}_{iu} = a_i * a_u * \text{lrate}$

The rule states that the weight on a connection from input unit *i* to output unit *u* is to be changed by an amount equal to the product of the activation value of *i*, the activation value of *u*, and a learning rate. [Notice that a large learning rate conduces to large weight changes and a smaller learning rate to more gradual changes.] Hebb's rule gave connectionist models the capacity to modify the weights on their own connections in light of the input-output patterns it has encountered.

Let us suppose, for the sake of illustration, that our 200 unit network started out life with connection weights of 0 across the board. We might then take an entry from our corpus of input-output pairs (say, the entry for donkeys) and set the input and output values accordingly. Hebb's rule might then be employed to strengthen connections from active input units to active output units. [Note: if units are allowed to have weights that vary between positive and negative values (for example, between -1 and 1), then Hebb's rule will strengthen connections between units whose activation values have the same sign and weaken connections between units with different signs.] This procedure could then be repeated for each entry in the corpus. Given a corpus of 100 entries and at 10,000 applications of the rule per entry, a total of 1,000,000 applications of the rule would be required for just one pass through the corpus (called an *epoch* of training). Here, clearly, the powerful number-crunching capabilities of electronic computers become essential.

Let us assume that we have set the learning rate to a relatively high value and that the network has received one epoch of training. What we will find is that if a given input pattern from the training corpus is encoded across the input units, activity will propagate forward through the connections in such a way as to activate the appropriate output unit. That is, our network will have learned how to appropriately classify input patterns.

As a point of comparison, the mainstream approach to artificial intelligence (AI) research is basically an offshoot of traditional forms of computer programming. Computer programs manipulate sentential representations by applying rules which are sensitive to the syntax (roughly, the shape) of those sentences. For instance, a rule might be triggered at a certain point in processing because a certain input was presented – say, "Fred likes broccoli and Sam likes cauliflower." The rule might be triggered whenever a compound sentence of the form *p and q* is input and it might produce as output a sentence of the form *p* ("Fred likes broccoli"). Although this is a vast oversimplification, it does highlight a distinctive feature of the classical approach to AI, which is the assumption that cognition is effected through the application of syntax-sensitive rules to syntactically structured representations. What is distinctive about many connectionist systems is that they encode information through activation vectors (and weight vectors), and they process that information when activity propagates forward through many weighted connections.
In addition, insofar as connectionist processing is in this way highly distributed (that is, many processors and connections simultaneously shoulder a bit of the processing load), a network will often continue to function even if part of it gets destroyed (if connections are *pruned*). The same kind of *parallel* and *distributed* processing (where many processors and connections are shouldering a bit of the processing load simultaneously) that enables this kind of *graceful degradation* also allows connectionist systems to respond sensibly to noisy or otherwise imperfect inputs. For instance, even we encoded an input vector that deviated from the one for donkeys but was still closer to the donkey vector than to any other, our model would still likely classify it as a donkey. Traditional forms of computer programming, on the other hand, have a much greater tendency to fail or completely crash due to even minor imperfections in either programming code or inputs.

The advent of connectionist learning rules was clearly a watershed event in the history of connectionism. It made possible the automation of vast numbers of weight assignments, and this would eventually enable connectionist systems to perform feats that McCulloch and Pitts could scarcely have imagined. As a learning rule for feed-forward networks, however, Hebb's rule faces severe limitations. Particularly damaging is the fact that the learning of one input-output pair (an*association*) will in many cases disrupt what a network has already learned about other associations, a process known as catastrophic interference. Another problem is that although a set of weights oftentimes exists that would allow a network to perform a given pattern association task, oftentimes its discovery is beyond the capabilities of Hebb's rule.

b. The Delta Rule

Such shortcomings led researchers to investigate new learning rules, one of the most important being the *delta rule*. To train our network using the delta rule, we it out with random weights and feed it a particular input vector from the corpus. Activity then propagates forward to the output layer. Afterwards, for a given unit *u* at the output layer, the network takes the actual activation of *u*and its desired activation and modifies weights according to the following rule:

Change of $weight_{iu}$ = learning rate * ($desired_u$ - a_u) * a_i

That is, to modify a connection from input *i* to output *u*, the delta rule computes the product of the difference between the desired activation of *u* and the actual activation (the *error score*), the activation of *i*, and a (typically very small) learning rate. Thus, assuming that unit *u* should be fully active (but is not) and input *i* happens to be highly active, the delta rule will increase the strength of the connection from *i* to *u*. This will make it more likely that the next time *i* is highly active, *u*will be too. If, on the other hand, *u* should have been inactive but was not, the connection from *i* to*u* will be pushed in a negative direction. As with Hebb's rule, when an input pattern is presented during training, the delta rule is used to calculate how the weights from each input unit to a given output unit are to be modified, a procedure repeated for each output unit. The next item on the corpus is then input to the network and the process repeats, until the entire corpus (or at least that part of it that the researchers want the network to encounter) has been run through. Unlike Hebb's rule, the delta rule typically makes small weight changes, meaning that several epochs of

training may be required before a network achieves competent performance. Again unlike Hebb's rule, however, the delta rule will in principle always slowly converge on a set of weights that will allow for mastery of all associations in a corpus, *provided that such a set of weights exists*. Famed connectionist Frank Rosenblatt called networks of the sort lately discussed *perceptrons*. He also proved the foregoing truth about them, which became known as the *perceptron convergence theorem.*
Rosenblatt believed that his work with perceptrons constituted a radical departure from, and even spelled the beginning of the end of, logic-based classical accounts of information processing (1958, 449; see also Bechtel & Abrahamson 2002, 6). Rosenblatt was very much concerned with the abstract information-processing powers of connectionist systems, but others, like Oliver Selfridge (1959), were investigating the ability of connectionist systems to perform specific cognitive tasks, such as recognizing handwritten letters. Connectionist models began around this time to be implemented with the aid of Von Neumann devices, which, for reasons already mentioned, proved to be a blessing.

There was much exuberance associated with connectionism during this period, but it would not last long. Many point to the publication of *Perceptrons* by prominent classical AI researchers Marvin Minsky and Seymour Papert (1969) as the pivotal event. Minsky and Papert showed (among other things) that perceptrons cannot learn some sets of associations. The simplest of these is a mapping from truth values of statements *p* and *q* to the truth value of *p* XOR *q* (where *p* XOR *q*is true, just in case *p* is true or *q* is true but not both). No set of weights will enable a simple two-layer feed-forward perceptron to compute the XOR function. The fault here lies largely with the architecture, for feed-forward networks with one or more layers of *hidden units* intervening between input and output layers (see Figure 4) can be made to perform the sorts of mappings that troubled Minsky and Papert. However, these critics also speculated that three-layer networks could never be trained to converge upon the correct set of weights. This dealt connectionists a serious setback, for it helped to deprive connectionists of the AI research funds being doled out by the Defense Advanced Research Projects Agency (DARPA). Connectionists found themselves at a major competitive disadvantage, leaving classicists with the field largely to themselves for over a decade.

c. The Generalized Delta Rule

In the 1980s, as classical AI research was hitting doldrums of its own, connectionism underwent a powerful resurgence thanks to the advent of the *generalized delta rule* (Rumelhart, Hinton, & Williams 1986). This rule, which is still the backbone of contemporary connectionist research, enables networks with one or more layers of hidden units to learn how to perform sets of input-output mappings of the sort that troubled Minsky and Papert. The simpler delta rule (discussed above) uses an error score (the difference between the actual activation level of an output unit and its desired activation level) and the incoming unit's activation level to determine how much to alter a given weight. The generalized delta rule works roughly the same way for the layer of connections running from the final layer of hidden units to the output units. For a connection running into a hidden unit, the rule calculates how much the hidden unit contributed to the *total error signal* (the sum of the individual output unit error signals) rather than the error signal of any particular unit. It adjust the connection from a unit in a still earlier layer to that hidden unit based upon the activity of the former and based upon the latter's contribution to the total error score. This process can be repeated for networks of varying depth. Put differently, the generalized delta rule enables *backpropagation learning*, where an error signal propagates backwards through multiple layers in order to guide weight modifications.

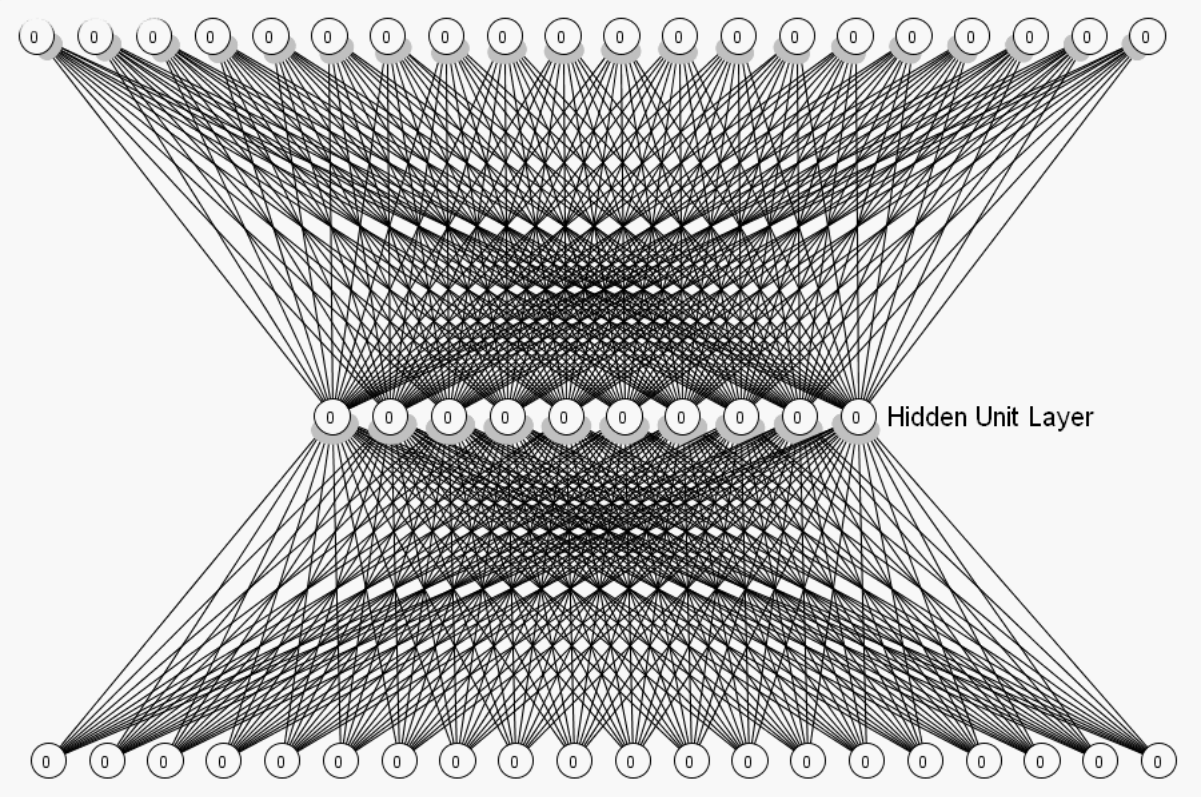

Figure 4: Three-layer Network [Created using Simbrain 2.0]

4. Connectionist Models Aplenty

Connectionism sprang back onto the scene in 1986 with a monumental two-volume compendium of connectionist modeling techniques (volume 1) and models of psychological processes (volume 2) by David Rumelhart, James McClelland and their colleagues in the Parallel Distributed Processing (PDP) research group. Each chapter of the second volume describes a connectionist model of some particular cognitive process along with a discussion of how the model departs from earlier ways of understanding that process. It included models of schemata (large scale data structures), speech recognition, memory, language comprehension, spatial reasoning and past-tense learning. Alongside this compendium, and in its wake, came a deluge of further models.

Although this new breed of connectionism was occasionally lauded as marking the next great paradigm shift in cognitive science, mainstream connectionist research has not tended to be directed at overthrowing previous ways of thinking. Rather, connectionists seem more interested in offering a deeper look at facets of cognitive processing that have already

been recognized and studied in disciplines like cognitive psychology, cognitive neuropsychology and cognitive neuroscience. What are highly novel are the claims made by connectionists about the precise form of internal information processing. Before getting to those claims, let us first discuss a few other connectionist architectures.

a. Elman's Recurrent Nets

Over the course of his investigation into whether or not a connectionist system can learn to master the complicated grammatical principles of a natural language such as English, Jeffrey Elman (1990) helped to pioneer a powerful, new connectionist architecture, sometimes known as an Elman net. This work posed a direct challenge to Chomsky's proposal that humans are born with an innate language acquisition device, one that comes preconfigured with vast knowledge of the space of possible grammatical principles. One of Chomsky's main arguments against Skinner's behaviorist theory of language-learning was that no general learning principles could enable humans to produce and comprehend a limitless number of grammatical sentences. Although connectionists had attempted (for example, with the aid of finite state grammars) to show that human languages *could* be mastered by general learning devices, sentences containing multiple center-embedded clauses ("The cats the dog chases run away," for instance) proved a major stumbling block. To produce and understand such a sentence requires one to be able to determine subject-verb agreements across the boundaries of multiple clauses by attending to contextual cues presented over time. All of this requires a kind of memory for preceding context that standard feed-forward connectionist systems lack.

Elman's solution was to incorporate a side layer of *context* units that receive input from and send output back to a hidden unit layer. In its simplest form, an input is presented to the network and activity propagates forward to the hidden layer. On the next step (or *cycle*) of processing, the hidden unit vector propagates forward through weighted connections to generate an output vector while at the same time being copied onto a side layer of context units. When the second input is presented (the second word in a sentence, for example), the new hidden layer activation is the product of both this second input *and* activity in the

context layer – that is, the hidden unit vector now contains information about both the current input and the preceding one. The hidden unit vector then produces an output vector as well as a new context vector. When the third item is input, a new hidden unit vector is produced that contains information about all of the previous time steps, and so on. This process provides Elman's networks with time-dependent contextual information of the sort required for language-processing. Indeed, his networks are able to form highly accurate predictions regarding which words and word forms are permissible in a given context, including those that involve multiple embedded clauses.

While Chomsky (1993) has continued to self-consciously advocate a shift back towards the nativist psychology of the rationalists, Elman and other connectionists have at least bolstered the plausibility of a more austere empiricist approach. Connectionism is, however, much more than a simple empiricist associationism, for it is at least compatible with a more complex picture of internal dynamics. For one thing, to maintain consistency with the findings of mainstream neuropsychology, connectionists ought to (and one suspects that most do) allow that we do not begin life with a uniform, amorphous cognitive mush. Rather, as mentioned earlier, the cognitive load may be divided among numerous, functionally distinct components. Moreover, even individual feed-forward networks are often tasked with unearthing complicated statistical patterns exhibited in large amounts of data. An indication of just how complicated a process this can be, the task of analyzing how it is that connectionist systems manage to accomplish the impressive things that they do has turned out to be a major undertaking unto itself (see Section 5).

b. Interactive Architectures

There are, it is important to realize, connectionist architectures that do not incorporate the kinds of feed-forward connections upon which we have so far concentrated. For instance, McClelland and Rumelhart's (1989) interactive activation and competition (IAC) architecture and its many variants utilize excitatory and inhibitory connections that run back and forth between the units in different groups. In IAC models, weights are hard-wired rather than learned and units are typically assigned their own particular, fixed meanings. When a set of units is activated so as to

encode some piece of information, activity may shift around a bit, but as units compete with one another to become most active through inter-unit inhibitory connections activity will eventually settle into a stable state. The stable state may be viewed, depending upon the process being modeled, as the network's reaction to the stimulus, which, depending upon the process being modeled, might be viewed as a semantic interpretation, a classification or a mnemonic association. The IAC architecture has proven particularly effective at modeling phenomena associated with long-term memory (content addressability, priming and language comprehension, for instance). The connection weights in IAC models can be set in various ways, including on the basis of individual hand selection, simulated evolution or statistical analysis of naturally occurring data (for example, co-occurrence of words in newspapers or encyclopedias (Kintsch 1998)).

An architecture that incorporates similar competitive processing principles, with the added twist that it allows weights to be learned, is the self-organizing feature map (SOFM) (see Kohonen 1983; see also Miikkulainen 1993). SOFMs learn to map complicated input vectors onto the individual units of a two-dimensional array of units. Unlike feed-forward systems that are supplied with information about the correct output for a given input, SOFMs learn in an *unsupervised* manner. Training consists simply in presenting the model with numerous input vectors. During training the network adjusts its inter-unit weights so that both each unit is highly 'tuned' to a specific input vector and the two-dimensional array is divided up in ways that reflect the most salient groupings of vectors. In principle, nothing more complicated than a Hebbian learning algorithm is required to train most SOFMs. After training, when an input pattern is presented, competition yields a single clear winner (for example, the most highly active unit), which is called the system's image (or interpretation) of that input.
SOFMs were coming into their own even during the connectionism drought of the 1970s, thanks in large part to Finnish researcher Tuevo Kohonen. Ultimately it was found that with proper learning procedures, trained SOFMs exhibit a number of biologically interesting features that will be familiar to anyone who knows a bit about topographic maps (for example, retinotopic, tonotopic and somatotopic) in the mammalian cortex. SOFMs tend not to allow a portion of the map go unused; they represent similar input vectors with neighboring units, which collectively

amount to a topographic map of the space of input vectors; and if a training corpus contains many similar input vectors, the portion of the map devoted to the task of discriminating between them will expand, resulting in a map with a distorted topography. SOFMs have even been used to model the formation of retinotopically organized columns of contour detectors found in the primary visual cortex (Goodhill 1993). SOFMs thus reside somewhere along the upper end of the biological-plausibility continuum.

Here we have encountered just a smattering of connectionist learning algorithms and architectures, which continue to evolve. Indeed, despite what in some quarters has been a protracted and often heated debate between connectionists and classicists (discussed below), many researchers are content to move back and forth between, and also to merge, the two approaches depending upon the task at hand.

5. Making Sense of Connectionist Processing

Connectionist systems generally learn by detecting complicated statistical patterns present in huge amounts of data. This often requires detection of complicated cues as to the proper response to a given input, the salience of which often varies with context. This can make it difficult to determine precisely how a given connectionist system utilizes its units and connections to accomplish the goals set for it.

One common way of making sense of the workings of connectionist systems is to view them at a coarse, rather than fine, grain of analysis -- to see them as concerned with the relationships between different activation vectors, not individual units and weighted connections. Consider, for instance, how a fully trained Elman network learns how to process particular words. Typically nouns like “ball,” “boy,” “cat,” and “potato” will produce hidden unit activation vectors that are more similar to one another (they tend to *cluster* together) than they are to “runs,” “ate,” and “coughed”. Moreover, the vectors for “boy” and “cat” will tend to be more similar to each other than either is to the “ball” or “potato” vectors. One way of determining that this is the case is to begin by conceiving activation vectors as points within a space that has as many dimensions as there are

units. For instance, the activation levels of two units might be represented as a single point in a two-dimensional plane where the y axis represents the value of the first unit and the x axis represents the second unit. This is called the *state space* for those units. Thus, if there are two units whose activation values are 0.2 and 0.7, this can be represented as the point where these two values intersect (Figure 5).

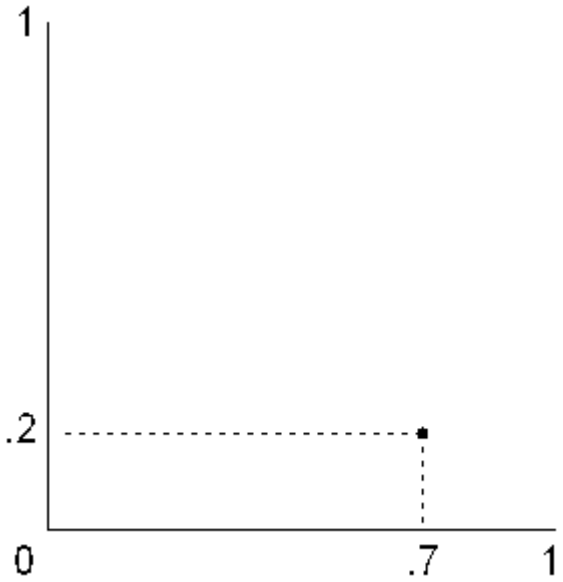

Figure 5: Activation of Two Units Plotted as Point in 2-D State Space

The activation levels of three units can be represented as the point in a cube where the three values intersect, and so on for other numbers of units. Of course, there is a limit to the number of dimensions we can depict or visualize, but there is no limit to the number of dimensions we can represent algebraically. Thus, even where many units are involved, activation vectors can be represented as points in high-dimensional space and the similarity of two vectors can be determined by measuring the proximity of those points in high-dimensional state space. This, however, tells us nothing about the way context determines the specific way in which networks represent particular words. Other techniques (for example, principal components analysis and multidimensional scaling) have been employed to understand such subtleties as the context-sensitive time-course of processing.

One of the interesting things revealed about connectionist systems through these sorts of techniques has been that networks which share the

same connection structure but begin training with different random starting weights will often learn to perform a given task equally well and to do so by partitioning hidden unit space in similar ways. For instance, the clustering in Elman's models discussed above will likely obtain for different networks even though they have very different weights and activities at the level of individual connections and units.

At this point, we are also in a good position to understand some differences in how connectionist networks code information. In the simplest case, a particular unit will represent a particular piece of information – for instance, our hypothetical network about animals uses particular units to represent particular features of animals. This is called a *localist encoding* scheme. In other cases an entire collection of activation values is taken to represents something – for instance, an entire input vector of our hypothetical animal classification network might represent the characteristics of a particular animal. This is a *distributed coding* scheme at the whole animal level, but still a local encoding scheme at the feature level. When we turn to hidden-unit representations, however, things are often quite different. Hidden-unit representations of inputs are often distributed without employing localist encoding at the level of individual units. That is, particular hidden units often fail to have any particular input feature that they are exclusively sensitive to. Rather, they participate in different ways in the processing of many different kinds of input. This is called*coarse coding*, and there are ways of coarse coding input and output patterns as well. The fact that connectionist networks excel at forming and processing these highly distributed representations is one of their most distinctive and important features.
Also important is that connectionist models often excel at processing novel input patterns (ones not encountered during training) appropriately. Successful performance of a task will often*generalize* to other related tasks. This is because connectionist models often work by detecting statistical patterns present in a corpus (of input-output pairs, for instance). They learn to process particular inputs in particular ways, and when they encounter inputs similar to those encountered during training they process them in a similar manner. For instance, Elman's networks were trained to determine which words and word forms to expect given a particular context (for example, "The boy threw the _____"). After training, they could do this very well even for sentence

parts they ha not encountered before. One caveat here is that connectionist systems with numerous hidden units (relative to the amount of variability in the training corpus) tend to use the extra memory to 'remember by rote' how to treat specific input patterns rather than discerning more abstract statistical patterns obtaining across many different input-output vectors. Consequently, in such cases performance tends not to generalize to novel cases very well.

As we have seen, connectionist networks have a number of desirable features from a cognitive modeling standpoint. There are, however, also serious concerns about connectionism. One is that connectionist models must usually undergo a great deal of training on many different inputs in order to perform a task and exhibit adequate generalization. In many instances, however, we can form a permanent memory (upon being told of a loved one's passing, for example) with zero repetition (this was also a major blow to the old psychological notion that rehearsal is required for a memory to make it into long-term storage). Nor is there much need to fear that subsequent memories will overwrite earlier ones, a process known in connectionist circles as *catastrophic interference*. We can also very quickly detect patterns in stimuli (for instance, the pattern exhibited by "J, M, P…") and apply them to new stimuli (for example, "7, 10, 13…"). Unfortunately, many (though not all) connectionist networks (namely many back-propagation networks) fail to exhibit one-shot learning and are prone to catastrophic interference.

Another worry about back-propagation networks is that the generalized delta rule is, biologically speaking, implausible. It certainly does look that way so far, but even if the criticism hits the mark we should bear in mind the difference between computability theory questions and learning theory questions. In the case of connectionism, questions of the former sort concern what sorts of things connectionist systems can and cannot do and questions of the latter address how connectionist systems might come to learn (or evolve) the ability to do these things. The back-propagation algorithm makes the networks that utilize them implausible from the perspective of learning theory, not computability theory. It should, in other words, be viewed as a major accomplishment when a connectionist network that utilizes only biologically plausible processing principles (, activation thresholds and weighted connections) is able to perform a cognitive task that had hitherto seemed mysterious. It constitutes a biologically plausible model of the underlying mechanisms regardless of

whether or not it came possess that structure through hand-selection of weights, Hebbian learning, back-propagation or simulated evolution.

6. Connectionism and the Mind

The classical conception of cognition was deeply entrenched in philosophy (namely in empirically oriented philosophy of mind) and cognitive science when the connectionist program was resurrected in the 1980s. Nevertheless, many researchers flocked to connectionism, feeling that it held much greater promise and that it might revamp our common-sense conception of ourselves. During the early days of the ensuing controversy, the differences between connectionist and classical models of cognition seemed to be fairly stark. Connectionist networks learned how to engage in the parallel processing of highly distributed representations and were fault tolerant because of it. Classical systems were vulnerable to catastrophic failure due to their reliance upon the serial application of syntax-sensitive rules to syntactically structured (sentence-like) representations. Connectionist systems superimposed many kinds of information across their units and weights, whereas classical systems stored separate pieces of information in distinct memory registers and accessed them in serial fashion on the basis of their numerical addresses.

Perhaps most importantly, connectionism promised to bridge low-level neuroscience and high-level psychology. Classicism, by contrast, lent itself to dismissive views about the relevance of neuroscience to psychology. It helped spawn the idea that cognitive processes can be realized by any of countless distinct physical substrates (see Multiple Realizability). The basic idea here is that if the mind is just a program being run by the brain, the material substrate through which the program is instantiated drops out as irrelevant. After all, computationally identical computers can be made out of neurons, vacuum tubes, microchips, pistons and gears, and so forth, which means that computer programs can be run on highly heterogeneous machines. Neural nets are but one of these types, and so they are of no essential relevance to psychology. On the connectionist view, by contrast, human cognition can only be understood by paying considerable attention to kind of physical mechanism that instantiates it.

Although these sorts of differences seemed fairly stark in the early days of the connectionism-classicism debate, proponents of the classical conception have recently made great progress emulating the aforementioned virtues of connectionist processing. For instance, classical systems have been implemented with a high degree of redundancy, through the action of many processors working in parallel, and by incorporating fuzzier rules to allow for input variability. In these ways, classical systems can be endowed with a much higher level of fault and noise tolerance, not to mention processing speed (See Bechtel & Abrahamson 2002). We should also not lose sight of the fact that classical systems have virtually always been capable of learning. They have, in particular, long excelled at learning new ways to efficiently search branching problem spaces. That said, connectionist systems seem to have a very different natural learning aptitude – namely, they excel at picking up on complicated patterns, sub-patterns, and exceptions, and apparently without the need for syntax-sensitive inference rules. This claim has, however, not gone uncontested.

a. Rules versus General Learning Mechanisms: The Past-Tense Controversy

Rumelhart and McClelland's (1986) model of past-tense learning has long been at the heart of this particular controversy. What these researchers claimed to have shown was that over the course of learning how to produce past-tense forms of verbs, their connectionist model naturally exhibited the same distinctive u-shaped learning curve as children. Of particular interest was the fact that early in the learning process children come to generate the correct past-tense forms of a number of verbs, mostly irregulars ("go" → "went"). Later, performance drops precipitously as they pick up on certain fairly general principles (for example, adding "-ed") and over-apply them even to previously learned irregulars ("went" may become "goed"). Lastly, performance increases as the child learns both the rules and their exceptions.

What Rumelhart and McClelland (1986) attempted to show was that this sort of process need not be underwritten by mechanisms that work by applying physically and functionally distinct rules to representations.

Instead, all of the relevant information can be stored in superimposed fashion within the weights of a connectionist network (really three of them linked end-to-end). Pinker and Prince (1988), however, would charge (inter alia) that the picture of linguistic processing painted by Rumelhart and McClelland was extremely simplistic and that their training corpus was artificially structured (namely, that the proportion of regular to irregular verbs varied unnaturally over the course of training) so as to elicit u-shaped learning. Plunkett and Marchman (1993) went a long way towards remedying the second apparent defect, though Marcus (1995) complained that they did not go far enough since the proportion of regular to irregular verbs was still not completely homogenous throughout training. As with most of the major debates constituting the broader connectionist-classicist controversy, this one has yet to be conclusively resolved. Nevertheless, it seems clear that this line of connectionist research does at least suggest something of more general importance – namely, that an interplay between a structured environment and general associative learning mechanisms might in principle conspire so as to yield complicated behaviors of the sort that lead some researchers to posit inner classical process.

b. Concepts

Some connectionists also hope to challenge the classical account of concepts, which embody knowledge of categories and kinds. It has long been widely held that concepts specify the singularly necessary and jointly sufficient conditions for category membership – for instance, "bachelor" might be said to apply to all and only unmarried, eligible males. Membership conditions of this sort would give concepts a sharp, all-or-none character, and they naturally lend themselves to instantiation in terms of formal inference rules and sentential representations. However, asWittgenstein (1953) pointed out, many words (for example, "game") seem to lack these sorts of strict membership criteria. Instead, their referents bear a much looser *family resemblance* relation to one another. Rosch & Mervis (1975) later provided apparent experimental support for the related idea that our knowledge of categories is organized not in terms of necessary and sufficient conditions but rather in terms of clusters of features, some of which (namely those most frequently encountered in

category members) are more strongly associated with the category than others. For instance, the ability to fly is more frequently encountered in birds than is the ability to swim, though neither ability is common to all birds. On the prototype view (and also on the closely related exemplar view), category instances are thought of as clustering together in what might be thought of as a hyper-dimensional semantic space (a space in which there are as many dimensions as there are relevant features). In this space, the prototype is the central region around which instances cluster (exemplar theory essentially does away with this abstract region, allowing only for memory of actual concrete instances). There are clearly significant isomorphisms between concepts conceived of in this way and the kinds of hyper-dimensional clusters of hidden unit representations formed by connectionist networks, and so the two approaches are often viewed as natural allies (Horgan & Tienson 1991). This way of thinking about concepts has, of course, not gone unchallenged (see Rey 1983 and Barsalou 1987 for two very different responses).

c. Connectionism and Eliminativism

Neuroscientist Patricia Churchland and philosopher Paul Churchland have argued that connectionism has done much to weaken the plausibility of our pre-scientific conception of mental processes (our *folk psychology*). Like other prominent figures in the debate regarding connectionism and folk psychology, the Churchlands appear to be heavily influenced by Wilfrid Sellars' view that folk psychology is a theory that enables predictions and explanations of everyday behaviors, a theory that posits internal manipulation to the sentence-like representations of the things that we believe and desire. The classical conception of cognition is, accordingly, viewed as a natural spinoff of this folk theory. The Churchlands maintain that neither the folk theory nor the classical theory bears much resemblance to the way in which representations are actually stored and transformed in the human brain. What leads many astray, say Churchland and Sejnowski (1990), is the idea that the structure of an effect directly reflects the structure of its cause (as exemplified by the homuncular theory of embryonic development). Thus, many mistakenly think that the structure of the language through which we express our thoughts is a clear indication of the structure of the thoughts themselves. The Churchlands think that connectionism may afford a glimpse into the

future of cognitive neuroscience, a future wherein the classical conception is supplanted by the view that thoughts are just points in hyper-dimensional neural state space and sequences of thoughts are trajectories through this space (see Churchland 1989).

A more moderate position on these issues has been advanced by Daniel Dennett (1991) who largely agrees with the Churchlands in regarding the broadly connectionist character of our actual inner workings. He also maintains, however, that folk psychology is for all practical purposes indispensible. It enables us to adopt a high-level stance towards human behavior wherein we are able to detect patterns that we would miss if we restricted ourselves to a low-level neurological stance. In the same way, he claims, one can gain great predictive leverage over a chess-playing computer by ignoring the low-level details of its inner circuitry and treating it as a thinking opponent. Although an electrical engineer who had perfect information about the device's low-level inner working could in principle make much more accurate predictions about its behavior, she would get so bogged down in those low-level details as to make her greater predictive leverage useless for any real-time practical purposes. The chess expert wisely forsakes some accuracy in favor of a large increase in efficiency when he treats the machine as a thinking opponent, an intentional agent. Dennett maintains that we do the same when we adopt an intentional stance towards human behavior. Thus, although neuroscience will not discover any of the inner sentences (putatively) posited by folk psychology, a high-level theoretical apparatus that includes them is an indispensable predictive instrument.

On a related note, McCauley (1986) claims that whereas it is relatively common for one high-level theory to be eliminated in favor of another, it is much harder to find examples where a high-level theory is eliminated in favor of a lower-level theory in the way that the Churchlands envision. However, perhaps neither Dennett nor McCauley are being entirely fair to the Churchlands in this regard. What the Churchlands foretell is the elimination of a high-level folk theory in favor of another high-level theory that emanates out of connectionist and neuroscientific research. Connectionists, we have seen, look for ways of understanding how their models accomplish the tasks set for them by abstracting away from neural particulars. The Churchlands, one might argue, are no exception. Their view that sequences are trajectories through a hyperdimensional

landscape abstracts away from most neural specifics, such as action potentials and inhibitory neurotransmitters.

d. Classicists on the Offensive: Fodor and Pylyshyn's Critique

When connectionism reemerged in the 1980s, it helped to foment resistance to both classicism and folk psychology. In response, stalwart classicists Jerry Fodor and Zenon Pylyshyn (1988) formulated a trenchant critique of connectionism. One imagines that they hoped to do for the new connectionism what Chomsky did for the associationist psychology of the radical behaviorists and what Minsky and Papert did for the old connectionism. They did not accomplish that much, but they did succeed in framing the debate over connectionism for years to come. Though their criticisms of connectionism were wide-ranging, they were largely aimed at showing that connectionism could not account for important characteristics of human thinking, such as its generally truth-preserving character, its productivity, and (most important of all) its systematicity. Of course they had no qualms with the proposal that vaguely connectionist-style processes happen, in the human case, to implement high-level, classical computations.

i. Reason

Unlike Dennett and the Churchlands, Fodor and Pylyshyn (F&P) claim that folk psychology works so well because it is largely correct. On their view, human thinking involves the rule-governed formulation and manipulation of sentences in an inner linguistic code (sometimes called*mentalese*). [Incidentally, one of the main reasons why classicists maintain that thinking occurs in a special 'thought language' rather than in one's native natural language is that they want to preserve the notion that people who speak different languages can nevertheless think the same thoughts – for instance, the thought that snow is white.] One bit of evidence that Fodor frequently marshals in support of this proposal is the putative fact that human thinking typically progresses in a largely truth-preserving manner. That is to say, if one's initial beliefs are true, the subsequent beliefs that one infers from them are also likely to be true. For instance, from the belief that the ATM will not give you any money and

the belief that it gave money to the people before and after you in line, you might reasonably form a new belief that there is something wrong with either your card or your account. Says Fodor (1987), if thinking were not typically truth-preserving in this way, there wouldn't be much point in thinking. Indeed, given a historical context in which philosophers throughout the ages frequently decried the notion that any mechanism could engage in reasoning, it is no small matter that early work in AI yielded the first fully mechanical models and perhaps even artificial implementations of important facets of human reasoning. On the classical conception, this can be done through the purely formal, syntax-sensitive application of rules to sentences insofar as the syntactic properties mirror the semantic ones. Logicians of the late nineteenth and early twentieth century showed how to accomplish just this in the abstract, so all that was left was to figure out (as von Neumann did) how to realize logical principles in artifacts.

F&P (1988) argue that connectionist systems can only ever realize the same degree of truth preserving processing by implementing a classical architecture. This would, on their view, render connectionism a sub-cognitive endeavor. One way connectionists could respond to this challenge would be to create connectionist systems that support truth-preservation without any reliance upon sentential representations or formal inference rules. Bechtel and Abrahamson (2002) explore another option, however, which is to situate important facets of rationality in human interactions with the external symbols of natural and formal languages. Bechtel and Abrahamson argue that "the ability to manipulate *external* symbols in accordance with the principles of logic need not depend upon a mental mechanism that itself manipulates *internal* symbols" (1991, 173). This proposal is backed by a pair of connectionist models that learn to detect patterns during the construction of formal deductive proofs and to use this information to decide on the validity of arguments and to accurately fill in missing premises.

ii. Productivity and Systematicity

Much more attention has been pain to other aspects of F&P's (1988) critique, such as their claim that only a classical architecture can account for the productivity and systematicity of thought. To better understand

the nature of their concerns, it might help if we first consider the putative productivity and systematicity of natural languages.

Consider, to start with, the following sentence:

(1) "The angry jay chased the cat."

The rules governing English appear to license (1), but not (2), which is made from (*modulo*capitalization) qualitatively identical parts:
(2) "Angry the the chased jay cat."

We who are fluent in some natural language have knowledge of the rules that govern the permissible ways in which the basic components of that language can be arranged – that is, we have mastery of the syntax of the language.

Sentences are, of course, also typically intended to carry or convey some meaning. The meaning of a sentence, say F&P (1988), is determined by the meanings of the individual constituents and by the manner in which they are arranged. Thus (3), which is made from the same constituents as (1), conveys a very different meaning.

(3) "The angry cat chased the jay."

Natural language expressions, in other words, have a combinatorial syntax *and* semantics.
In addition, natural languages appear to be characterized by certain *recursive* rules which enable the production of an infinite variety of syntactically distinct sentences. For instance, in English one such rule allows any two grammatical statements to be combined with 'and'. Thus, if (1) and (3) are grammatical, so is this:
(4) "The angry jay chased the cat and the angry cat chased the jay."

Sentence (4) too can be combined with another, as in (5) which conjoins (4) and (3):

"The angry jay chased the cat and the angry cat chased the jay, and the angry cat chased the jay."

Earlier we discussed another recursive principle which allows for center-embedded clauses.

One who has mastered the combinatorial and recursive syntax and semantics of a natural language is, according to classicists like F&P (1988), thereby capable in principle of producing and comprehending an infinite number of grammatically distinct sentences. In other words, their mastery of these linguistic principles gives them a *productive* linguistic competence. It is also reputed to give them a *systematic* competence, in that a fluent language user who can produce and understand one sentence can produce and understand systematic variants. A fluent English speaker who can produce and understand (1) will surely be able to produce and understand (3). It is, on the other hand, entirely possible for one who has learned English from a phrase-book (that is, without learning the meanings of the constituents or the combinatorial semantics of the language) to be able to produce and understand (1) but not its systematic variant (3).

Thinking, F&P (1988) claim, is also productive and systematic, which is to say that we are capable of thinking an infinite variety of thoughts and that the ability to think some thoughts is intrinsically connected with the ability to think others. For instance, on this view, anyone who can think the thought expressed by (1) will be able to think the thought expressed by (3). Indeed, claims Fodor (1987), since to understand a sentence is to entertain the thought the sentence expresses, the productivity and systematicity of language *imply* the productivity and systematicity of thought. F&P (1988) also maintain that just as the productivity and systematicity of language is best explained by its combinatorial and recursive syntax and semantics, so too is the productivity and systematicity of thought. Indeed, they say, this is the only explanation anyone has ever offered.

The systematicity issue has generated a vast debate (see Bechtel & Abrahamson 2002), but one general line of connectionist response has probably garnered the most attention. This approach, which appeals to functional rather than literal compositionality (see van Gelder 1990), is most often associated with Smolensky (1990) and with Pollack (1990), though for simplicity's sake discussion will be restricted to the latter.

Pollack (1990) uses recurrent connectionist networks to generate compressed, distributed encodings of syntactic strings and subsequently uses those encodings to either recreate the original string or to perform a systematic transformation of it (e.g., from "Mary loved John" to "John loved Mary"). Pollack's approach was quickly extended by Chalmers (1990), who showed that one could use such compressed distributed representations to perform systematic transformations (namely moving from an active to a passive form) of even sentences with complex embedded clauses. He showed that this could be done for both familiar and novel sentences. What this suggests is that connectionism might offer its own unique, non-classical account of the apparent systematicity of thought processes. However, Fodor and McLaughlin (1990) argue that such demonstrations only show that networks can be *forced* to exhibit systematic processing, not that they exhibit it naturally in the way that classical systems do. After all, on a classical account, the same rules that license one expression will automatically license its systematic variant. It bears noting, however, that this approach may itself need to impose some ad hoc constraints in order to work. Aizawa (1997) points out, for instance, that many classical systems do not exhibit systematicity. On the flipside, Matthews (1997) notes that systematic variants that are licensed by the rules of syntax need not be thinkable. Waskan (2006) makes a similar point, noting that thinking may be more and less systematic than language and that the actual degree to which thought is systematic may be best accounted for by, theoretically speaking, pushing the structure of the world 'up' into the thought medium, rather than pushing the structure of language 'down'. This might, however, come as cold comfort to connectionists, for it appears to merely replace one competitor to connectionism with another.

7. Anti-Represenationalism: Dynamical Stystems Theory, A-Life and Embodied Cognition

As alluded to above, whatever F&P may have hoped, connectionism has continued to thrive. Connectionist techniques are now employed in virtually every corner of cognitive science. On the other hand, despite what connectionists may have wished for, these techniques have not come close to fully supplanting classical ones. There is now much more of a peaceful coexistence between the two camps. Indeed, what probably seems far more important to both sides these days is the advent and promulgation

of approaches that reject or downplay central assumptions of *both* classicists and mainstream connectionists, the most important being that human cognition is largely constituted by the creation, manipulation, storage and utilization of representations. Many cognitive researchers who identify themselves with the dynamical systems, artificial life and (albeit to a much lesser extent) embodied cognition endorse the doctrine that one version of the world is enough. Even so, practitioners of the first two approaches have often co-opted connectionist techniques and terminology. In closing, let us briefly consider the rationale behind each of these two approaches and their relation to connectionism.

Briefly, dynamical systems theorists adopt a very high-level perspective on human behavior (inner and/or outer) that treats its state at any given time as a point in high-dimensional space (where the number of dimensions is determined by the number of numerical variables being used to quantify the behavior) and treats its time course as a trajectory through that space (van Gelder & Port 1995). As connectionist research has revealed, there tend to be regularities in the trajectories taken by particular types of system through their state spaces. As paths are plotted, it is often as if the trajectory taken by a system gets attracted to certain regions and repulsed by others, much like a marble rolling across a landscape can get guided by valleys, roll away from peaks, and get trapped in wells (*local* or *global minima*). The general goal is to formulate equations like those at work in the physical sciences that will capture such regularities in the continuous time-course of behavior. Connectionist systems have often provided nice case studies in how to characterize a system from the dynamical systems perspective. However, whether working from within this perspective in physics or in cognitive science, researchers find little need to invoke the ontologically strange category of representations in order to understand the time course of a system's behavior.

Researchers in artificial life primarily focus on creating artificial creatures (virtual or real) that can navigate environments in a fully autonomous manner. The strategy generally favored by artificial life researchers is to start small, with a simple behavior repertoire, to test one's design in an environment (preferably a real one), to adjust it until success is achieved, and then to gradually add layers of complexity by repeating this process. In one early and influential manifesto of the 'a-life' movement, Rodney Brooks claims, "When intelligence is approached in an incremental

manner, with strict reliance on interfacing to the real world through perception and action, reliance on representation disappears" (Brooks 1991). The aims of a-life research are sometimes achieved through the deliberate engineering efforts of modelers, but connectionist learning techniques are also commonly employed, as are simulated evolutionary processes (processes that operate over both the bodies and brains of organisms, for instance).

8. Where Have All the Connectionists Gone?

There perhaps may be fewer today who label themselves "connectionists" than there were during the 1990s. Fodor & Pylyshyn's (1988) critique may be partly responsible for this shift, though it is probably more because the novelty of the approach has worn off and the initial fervor died down. Also to blame may be the fact that connectionist techniques are now very widely employed throughout cognitive science, often by people who have very little in common ideologically. It is thus increasingly hard to discern among those who utilize connectionist modeling techniques any one clearly demarcated ideology or research program. Even many of those who continue to maintain an at least background commitment to the original ideals of connectionism might nowadays find that there are clearer ways of signaling who they are and what they care about than to call themselves "connectionists." In any case, whether connectionist techniques are limited in some important respects or not, it is perfectly clear is that connectionist modeling techniques are still powerful and flexible enough as to have been widely embraced by philosophers and cognitive scientists, whether they be mainstream moderates or radical insurgents. It is therefore hard to imagine any technological or theoretical development that would lead to connectionism's complete abandonment. Thus, despite some early fits and starts, connectionism is now most assuredly here to stay.

9. References and Further Reading

a. References

- Aizawa, K. (1997). Explaining systematicity, *Mind and Language*, **12,** 115-136.
- Barsalou, L. (1987). The instability of graded structure: Implications for the nature of concepts. In U. Neisser (Ed.), *Concepts and conceptual development: Ecological and intellectual factors in categorization*. Cambridge, UK: Cambridge University Press, 101-140.
- Bechtel, W. & A. Abrahamsen. (1991). *Connectionism and the mind: An introduction to parallel processing in networks*. Cambridge, MA: Basil Blackwell.
- Bechtel, W. & A. Abrahamsen. (2002). *Connectionism and the mind: An introduction to parallel processing in networks, 2nd Ed*. Cambridge, MA: Basil Blackwell.
  - Highly recommended introduction to connectionism and the philosophy thereof.
- Boden, M. (2006). *Mind as machine: A history of cognitive science*. New York: Oxford.
- Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence*, 47, 139-159.
- Chalmers, D. (1990). Syntactic transformations on distributed representations. *Connection Science, 2*, 53-62.
- Chomsky, N. (1993). On the nature, use and acquisition of language. In A. Goldman (Ed.), *Readings in the Philosophy and Cognitive Science*. Cambridge, MA: MIT, 511-534.
- Churchland, P.M. (1989). *A neurocomputational perspective: The nature of mind and the structure of science*. Cambridge, MA: MIT.
- Churchland, P.S. & T. Sejnowski. (1990). Neural representation and neural computation. *Philosophical Perspectives*, 4, 343-382.
- Dennett, D. (1991). Real Patterns. *The Journal of Philosophy*, 88, 27-51.
- Elman, J. (1990). Finding Structure in Time. *Cognitive Science*, 14, 179-211.
- Fodor, J. (1987). *Psychosemantics*. Cambridge, MA: MIT.
- Fodor, J. & B. McLaughlin. (1990). Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work, *Cognition, 35***,** 183-204.
- Fodor, J. & Z. Pylyshyn. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28, 3-71.
- Franklin, S. & M. Garzon. (1996). Computation by discrete neural nets. In P. Smolensky, M. Mozer, & D. Rumelhart (Eds.) *Mathematical perspectives on neural networks* (41-84). Mahwah, NJ: Lawrence Earlbaum.
- Goodhill, G. (1993). Topography and ocular dominance with positive correlations. *Biological Cybernetics*, 69, 109-118 .
- Hebb, D.O. (1949). The Organization of Behavior. New York: Wiley.

- Horgan, T. & J. Tienson (1991). Overview. In Horgan, T. & J. Tienson (Eds.) *Connectionism and the Philosophy of Mind*. Dordrecht: Kluwer.
- Kintsch, W. (1998). *Comprehension: A Paradigm for Cognition*. Cambridge: Cambridge University Press.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.
- Marcus, R. (1995). The acquisition of the English past tense in children and multilayered connectionist networks.*Cognition*, 56, 271-279.
- Matthews, R. (1997). Can connectionists explain systematicity? *Mind and Language*, 12, 154-177.
- McCauley, R. (1986). Intertheoretic relations and the future of psychology. *Philosophy of Science*, 53, 179-199.
- McClelland, J. & D. Rumelhart. (1989). Explorations in parallel distributed processing: A handbook of models, programs, and exercises. Cambridge, MA: MIT.
    - This excellent hands-on introduction to connectionist models of psychological processes has been replaced by: R. O'Reilly & Y. Munakata. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. Cambridge, MA: MIT. Companion software called *Emergent*.
- McCulloch, W. & W. Pitts. (1943). A logical calculus of the ideas immanent in nervous activity *Bulletin of Mathematical Biophysics*, 5:115-133.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.
- Miikkulainen, R. (1993). *Subsymbolic Natural Language Processing*. Cambridge, MA: MIT.
    - Highly recommended for its introduction to Kohonen nets.
- Minsky, M. & S. Papert. (1969). Perceptrons: An introduction to computational geometry. Cambridge, MA: MIT.
- Pinker, S. & A. Prince. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition,* 28, 73-193.
- Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence*, 46, 77-105.
- Plunkett, K. & V. Marchman. (1993). From rote learning to system building: Acquiring verb morphology in children and connectionist nets. *Cognition,* 48, 21-69.
- Rey, G. (1983). Concepts and stereotypes. *Cognition,* 15, 273-262.
- Rosch, E. & C. Mervis. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7, 573-605.

- Rumelhart, D., G. Hinton, & R. Williams. (1986). Learning internal representations by error propagation. In D. Rumelhart & J. McClelland (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1. Cambridge, MA: MIT, 318-362.
- Selfridge, O. (1959). Pandemonium: A paradigm for learning. Rpt. in J. Anderson & E. Rosenfeld (1988),*Neurocomputing: Foundations of research*. Cambridge, MA: MIT, 115-122.
- Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence*, 46, 159–216.
- van Gelder, T. (1990). Compositionality: A connectionist variation on a classical theme. *Cognitive Science*, *14*, 355-384.
- van Gelder, T. & R. Port. (1995). *Mind as motion: Explorations in the dynamics of cognition*. Cambridge, MA: MIT.
- Waskan, J. (2006). *Models and Cognition: Prediction and explanation in everyday life and in science*. Cambridge, MA: MIT.
- Wittgenstein, L. (1953). *Philosophical Investigations*. New York: Macmillan.

b. Connectionism Freeware

- *BugBrain* provides an excellent, accessible, and highly entertaining game-based hands-on tutorial on the basics of neural networks and gives one a good idea of what a-life is all about as well. *BugBrain* comes with some learning components, but they are not recommended.
- *Emergent* is research-grade software that accompanies O'Reilly and Munakata's *Computational explorations in cognitive neuroscience* (referenced above).
- *Simbrain* is a fairly accessible, but somewhat weak, tool for implementing a variety of common neural network architectures.
- *Framsticks* is a wonderful program that enables anyone with the time and patience to evolve virtual stick creatures and their neural network controllers.

Author Information

Jonathan Waskan
Email: waskan@illinois.edu
University of Illinois at Urbana-Champaign
U. S. A.