easily, and most children catch up quite quickly once the problem has been eliminated.

Sadly, not all impairments to language are so easily treated.

Almost anything that one can imagine going wrong with language can go wrong. And some things one would not imagine could go wrong do. The range of impairments is vast, and often, especially following brain injury, language is not the only faculty that has been impaired. It is all very well to describe, clinically and unemotionally, the nature of these impairments, but imagine waking up one day and discovering that it had happened to you. Or imagine growing up and losing out at school because you could not take notes easily, or could not understand the teacher, or could not express yourself adequately. Imagine your child having to grow up that way. None of us who are unimpaired can imagine what it is like to hear people speaking but be unable to understand them, or to know what we want to say but be unable to say it. For most of us, the ascent of our own personal Babel is unimpeded, and we should be thankful for that.

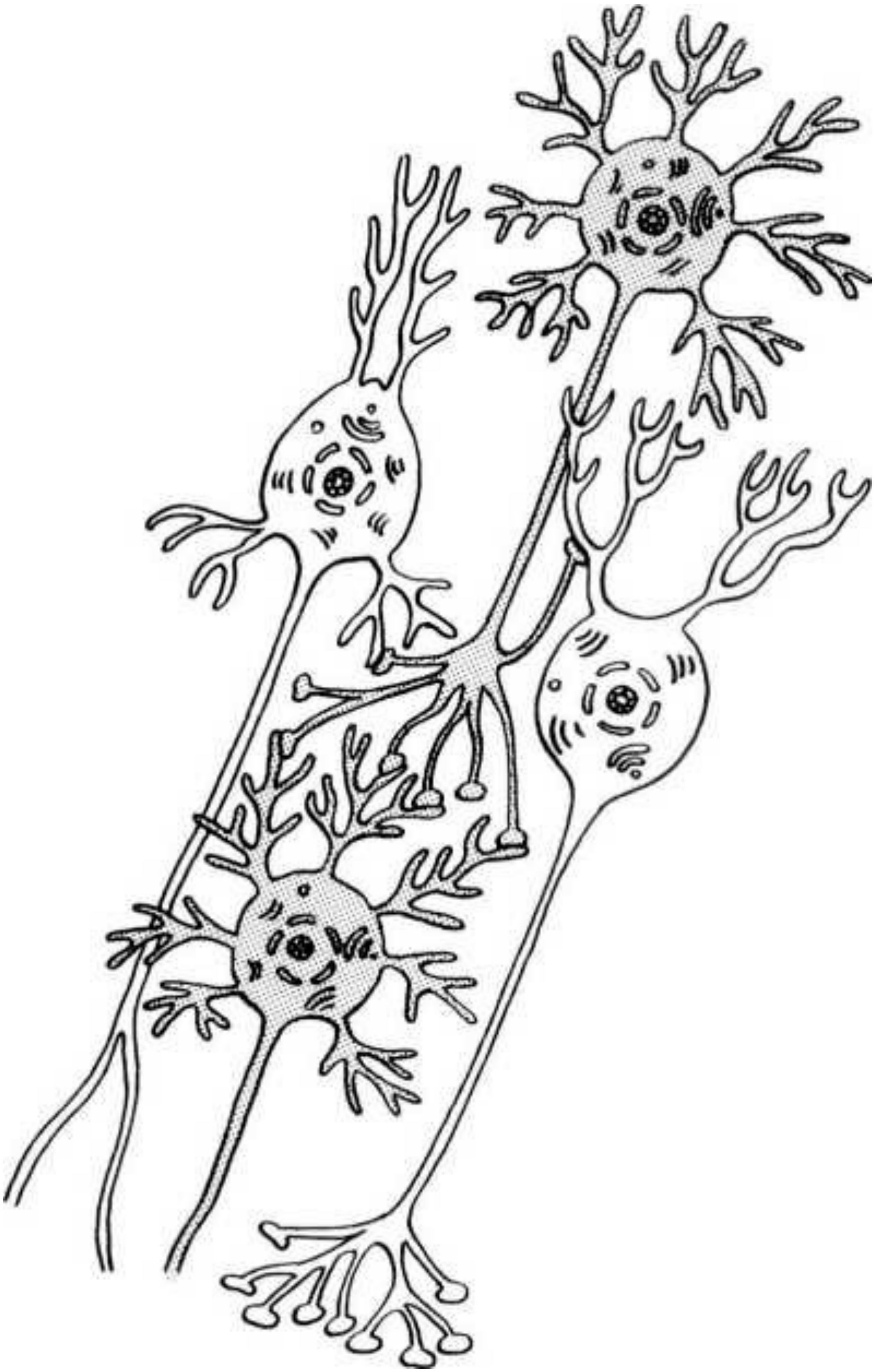
13

Wiring-up a brain

The average adult human brain weighs around 1.3 kg, and contains 10 billion or so nerve cells. Each nerve cell, or neuron, can connect to, and so stimulate, anything between a few hundred and perhaps 100 000 other nerve cells. And each neuron can itself receive connections from up to that same number again. Extend this to all 10 billion cells, and it is surprising that anything as vast and complex could work at all. But evidently it does. Because there is little else in the brain apart from neurons, we have no choice but to accept that they, and the manner in which they interconnect, are responsible for the mental feats of which we are capable. But there is rothing particularly special about neurons: if you stimulate one enough, it will stimulate the other neurons to which it is connected. Yet from this come our mental faculties. There is one further property of the brain that is crucial-even in an adult brain the wiring between the neurons is constantly changing. If it could not change, we could never learn.

We are not born knowing the language we shall end up using. We learn that language. Just as we learn about the world within which we shall use it. The meanings which we evoke with the words of our language are simply patterns of neural activity. These patterns reflect the accumulated experience of the contexts in which those words are used, and as such they have gradually changed with those experiences (see Chapter 9 for further details). But for patterns of neural activity to change, the patterns of neural connectivity and neural transmission that underlie those patterns of activity must also change. Ultimately, it is these changes that allow us to learn from experience. And learning from experience underlies just about everything we do. So how do these changes come about?

Neurons are a little like sunflowers. The flower corresponds to the body of the neuron. The stem is the main length of the neuron down which neural impulses are transmitted to a mass of roots. These connect to other neurons (via the equivalent of the sunflowers' `petals') to which those impulses are transmitted.



Despite the complexities of the neurochemical processes that underlie neural transmission, there are just three principles at work. The first, and most obvious, is that neurons send impulses to the other neurons to which they are connected. The rate at which impulses are sent corresponds, in a sense, to the `strength' of the signal. The second is that an impulse from one neuron can either make it more likely that another neuron will generate an impulse of its own, or less likely. Which of these it is depends on the kind of connection (it can be an excitatory connection, or an inhibitory one). The third principle is perhaps the most important. The connections can change in response to the surrounding neural activity-new ones can grow, especially in the first two to three years of life, although it happens in adulthood too; existing ones can die back (again, this is probably more common in younger brains); and the sensitivity of each connection can change, so that a neuron will need to receive either a stronger or a weaker signal across that connection before it generates its own impulse.

But from these three principles, are we any closer to understanding how a brain can wire itself up for language (or for anything else, come to that)? This is where artificial brains come in.

Inside an artificial neural network

Artificial neural networks exhibit the same three principles introduced in the preceding section. But the neurons are quite different. For a start, the most common neural networks (the `artificial' will be omitted from now on) are simulated. A computer program keeps track of which neurons there ought to be, what each should currently be doing, which should be connected to which, and so on. These simulated neurons are a lot simpler than their real counterparts. The purpose of these neural networks. is not to simulate the precise workings of the brain. What matters is that a signal is passed from one neuron to each of the others it connects to, or the fact that the sensitivity of a connection can change.

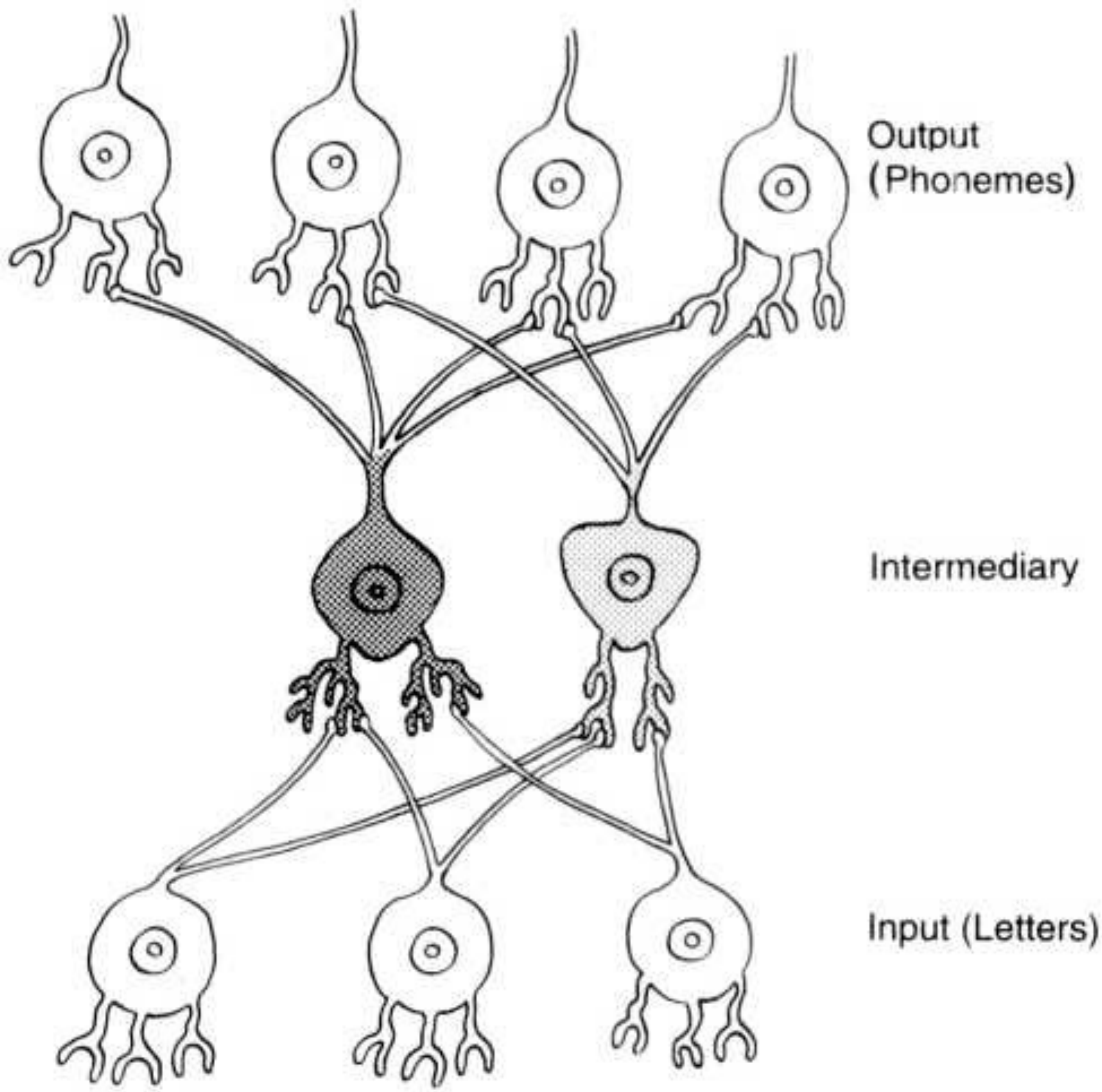
There is no attempt to model the process by which it changes.

Artificial neurons are very much simpler than real ones. The computer works out how much stimulation a neuron should receive, and allocates that neuron a number to reflect how active it should be. Real neurons are similar-the rate at which they generate neural impulses, the strength of the signal, changes as a function of how much stimulation they receive. This number, the neuron's activation, is calculated on the basis of how active each of the neurons feeding into it is, and how `strong' each of these connections is. The connection strength reflects the fact that the sensitivity of real neural connections can vary. It is a little like the volume knob on an amplifier. The higher the volume, the louder the signal. But the difference is that in these artificial networks, the connection strength is just a number. If it is positive, it is an excitatory connection. If it is negative, it is an inhibitory connection. If it is zero, then that is the same as if there were no connection between those two neurons.

That, briefly, is the underlying physiology of a neural network. The fact that a computer keeps track of what is going on within an artificial neural network, and does all the working out, is immaterial. We could instead build an artificial neural system which consisted of physically distinct artificial neurons, with complex interconnections that enabled the neurons to, in effect, add, subtract, and multiply (equivalent interconnections exist within the brain). It is just much easier to have a computer simulate all this. But the important point is that the computer is simply doing what a real network could in principle do for itself.

How networks work

The easiest way to figure out how neural networks work is to go through an example. There are many different kinds, and we shall take as our example one that could be used, for instance, to learn letter-tosound correspondences. Typically, the neurons in these networks are separated into distinct groups. In our example we shall use just three such groups. One is going to act as the `eyes' of the network-patterns of activity across the neurons in that group will represent the letter that the network is `seeing'. Another group is going to represent the phonemes that the network is supposed to output (perhaps to a speech synthesizer). The third group will be intermediaries between the letter neurons and the phoneme neurons.



In real brains, there would probably be thousands of neurons in the chain, but the advantage of artificial neural networks is that they can be very much simpler than real brains. The only route from the letter neurons to the phoneme neurons in this example network is via the intermediary neurons. In principle, we could allow a more direct route, but we shall not do so here. Another thing we shall not allow in this example are direct connections from one letter neuron to another, or between the phoneme neurons.

That is the basic anatomy of the network. How does it work? We can start off by assuming that it has not yet learned anything. We must also assume that when the network `sees' a letter, the computer activates the letter neurons, giving each neuron a particular amount of activation. Each letter of the alphabet would have its own unique activation pattern. We shall return shortly to why the computer allocates one pattern, rather than another, to any one letter.

We can now work through what happens when the network sees an L. First, the letter neurons will each be activated, by different amounts, according to the pattern of activation that has been allocated to that letter. Next, the computer will look at each neuron in the intermediary set of neurons, work out how much stimulation each one is receiving from all the letter neurons that connect to it (taking into account each connection's strength), and activate it accordingly. For each neuron, it takes the activation value of all the neurons connecting to it, multiplies each of those values by the appropriate connection strength, adds the results of all these multiplications, enters the grand total into an equation which converts it to a number between 0 and 1, and sets the neuron's activation to this final value. Once it has done this for all the intermediary neurons, it does the same thing again for the phoneme neurons connected to the intermediary neurons. In this way, the pattern of activation across the letter neurons spreads, via the intermediary neurons, to the phoneme neurons.

The final pattern that develops across the phoneme neurons will mean absolutely nothing. The network has not learnt anything yet, and the connection strengths are all just random. So the activation pattern across the phoneme neurons would also be random. But if the network had learnt what it was intended to learn (and we shall come to how it would do this shortly), the pattern across the phoneme neurons when the network was seeing the letter L would have been a pattern that was supposed to correspond to the phoneme /1/. It would be a pattern that the computer had previously allocated to that phoneme, in much the same way that it had allocated one pattern to the letter L, another to M, and so on. The learning process would have taken a random set of connection strengths, and would have managed to change them so that a particular pattern across the letter neurons (the pattern for L) would spread through the network and cause a particular pattern across the phoneme neurons (the pattern for /1/).

There is an important consequence of this last fact. If the connection strengths start off as random, they will scramble up the pattern allocated to L when they transmit it to the intermediary neurons. Whether it is one pattern or another makes absolutely no difference--it will still be scrambled. But if the network can learn to change the connection strengths so that it activates a specific pattern across the phoneme neurons in response to a specific pattern across the letter neurons, even when the connection strengths started off as random, it would not matter what pattern was initially allocated to any individual letter, as long as it was different from the pattern allocated to any other letter. This is just as well when it comes to thinking about our own brains and the activation patterns that they start out with. As long as our senses are consistent, it does not matter what patterns of neural activity they cause, so long as the same sensation gives rise to the same pattern of activity, and there is some way of changing the neural connectivity from its initial (potentially random) state to its final (most definitely non-random) one. Exactly how this happens in the artificial case is described next.

How (some) networks learn

The way in which networks like the one in our example learn is surprisingly simple. But before seeing how they learn it is as well to consider what they learn. The sensitivity of each connection within the network determines the precise pattern of activation that forms across the phoneme neurons in response to a particular pattern of activation across the letter neurons. So if you want the network to produce a particular phoneme pattern in response to a particular letter pattern, you have to get the connections just right. And that is what networks learn to do. They can learn to set the sensitivities of their own connections so that, eventually, the network can encode many different letterphoneme pairings using the same set of neural connections.

Only one extra detail needs to be added to allow this to happenwhen a pattern is input to the network across the letter neurons, some thing has to be able to tell the network what pattern of activation it should output across the phoneme neurons. What happens is that the computer looks at the activation value of each of the output (phoneme) neurons and compares it with what it should have been if the right activation pattern had been produced. It then modifies the strength of each connection leading to that neuron by a very small amount that is dependent on how close it got to the right pattern. It does this for each connection leading to each of the output neurons. It then does the same thing for each connection leading to each of the intermediary neurons. It works out (in effect) what the activation pattern across these neurons should have been in order to get something a little more similar to the correct activation at the output neurons. It then works out how each one of these neurons did compared to how it should have done, and modifies the strength of each connection leading to each neuron, but again, only by a very small amount. It sounds complicated, but it just requires some fairly simple mathematics (which computers are good at). This whole process is repeated for each pairing of input and output patterns presented to the network, with each presentation causing the connection strengths to change very slightly. Eventually, as the network learns those input-output pairings, the changes to those connection strengths become even more slight, until they stop altogether when the network gets each pairing right.

David Rumelhart, Geoffrey Hinton, and Ronald Williams at the University of California in San Diego first described this learning procedure in the mid-1980s. Although it is very unlikely that real brains systematically modify the sensitivity of their neural connections in exactly the same way, we do know that those sensitivities do change (to the point where the connections may even disappear). The principles, if not the methods, are the same.

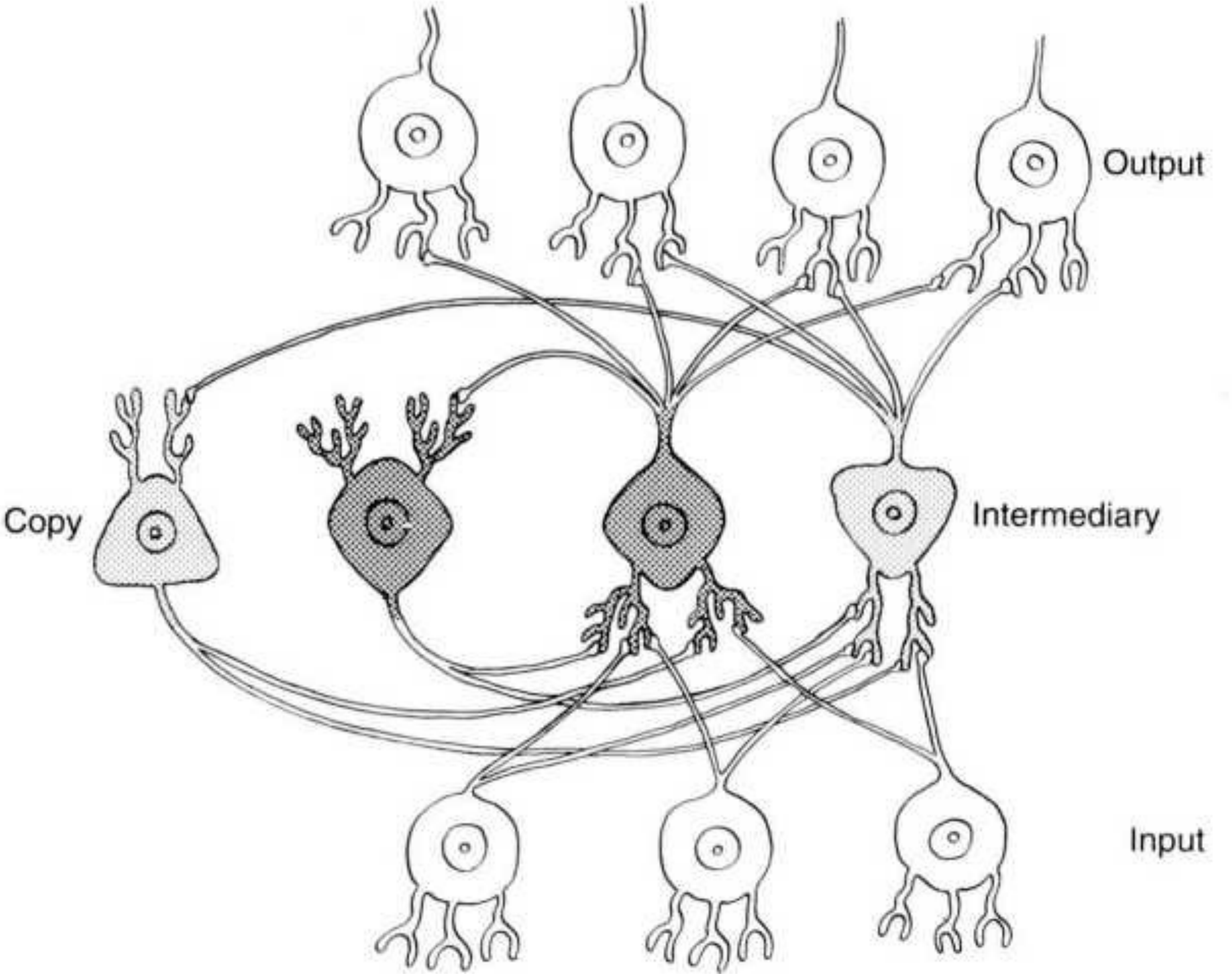
Coping with the sequential structure of language

The problem with language is that things happen one after the other. The neural network that we have been looking at is severely limited because, although it can learn to associate one thing with another, the things it learns about are static, unchanging patterns. What is needed is a network which can take a single pattern across one set of neurons (representing, for instance, the visual image of a word) and associate that with a sequence of patterns, one after the other, across another set (representing the sequence of phonemes that need to be uttered, one after the other, to say that word). Or, better still, a network which can take a sequence of patterns across one set of neurons (representing, perhaps, the incoming speech signal) and generate another sequence of patterns across another set of neurons (representing, perhaps, the developing meaning of that signal).

Our example network has a further limitation that is in some respects even more serious. In order to learn anything, it needs the equivalent of a teacher who can tell it that the current activation pattern is not quite right, and can tell it the correct pattern that it should be aiming for. But except for when we are explicitly taught to read, and are explicitly taught such things as letter-to-sound correspondences, nobody teaches us what the correct activation patterns should be in response to what we hear. Could we design a network which did not rely on an explicit teacher?

In the late 1980s, Jeffrey Elman, working at the University of California, San Diego, developed a neural network that addressed both these drawbacks. He borrowed an idea that had previously been developed by a colleague of his, Michael Jordan. Jordan had demonstrated that a very simple extension to our example network could learn to output a sequence of things, one thing after another, when given just a single input pattern. The network acted as if it had a queue, or buffer, containing activation patterns waiting to be output, with something directing which pattern should come next (see Chapter 10 for the application of queues in language production). Jordan's extension gave the network the equivalent of a memory for what it had output so far. Fortunately, his new network could use the same learning procedure as before. Elman extended Jordan's technique so that the network would also be able to take as input a sequence of things. Better still, he got rid of the need for any explicit teaching. This is how the `Elman net'works:

Imagine that at the first tick of a clock an activation pattern spreads from the input neurons to the intermediary neurons. At the second tick it spreads from the intermediary neurons to the output neurons. On the third tick, a new input (the next element in the sequence; feeds through from the input neurons to the intermediary neurons, and so on. Elman added an extra step. At the first tick, as before, activation would spread from the input neurons to the intermediary neurons. At the second tick, two things would happen. Activation would still spread from the intermediary neurons to the output neurons. But it would also spread to a new set of neurons (copy neurons) that were wired-up so that they would duplicate the activation pattern across the intermediary ones each copy neuron received activation from just one intermediary neuron, and a connection strength of one ensured that the copy neuron would take on the activation value of the intermediary neuron it was connected to. But each copy neuron had connections back to each of the intermediary neurons. So on the third tick, the intermediary neurons would receive both new activation from the input neurons, and a copy of the previous pattern of activation across those intermediary neurons. The pattern of activation across the intermediary neurons would therefore embody both the network's reaction to the new input, and the network's reaction to the previous input. And of course, that reaction was also a reflection of the reaction before that, and before that. The network therefore had a memory of how it had reacted to previous inputs, and how they were sequenced through time. And because it had a memory of its previous reactions, each output could be determined, in part, by that memory.



Because the Elman net has a memory for how it has reacted to previous input, it can either take a sequence of things in one order, and output one thing, or take those same things in a different order, and output something else (whether a static pattern or a sequence of changing patterns). We shall see an example of what the net was capable of in the next section. But a final innovation was Elman's realization that his network could be taught without an explicit teacher. He designed his network so that it would predict what its next input would be.

Neural networks and the prediction task

Whereas our earlier network could learn to associate one pattern with another, the Elman net can learn to associate a changing sequence of patterns with another changing sequence of patterns. The prediction task is just a variant on the idea that one sequence can be associated with another. Each element in an input sequence causes the network to output something. That output is compared with the next input, and the strengths modified according to the discrepancy. In principle, then, the Elman net can be trained to predict, on the basis of what it has seen so far of an input sequence, what the next element in that sequence is likely to be. But how useful is this? Surely we are not exposed to sequences of words, for instance, which allow us to predict with any certainty what the next element is going to be? In fact, we are. Here is what Elman did.

Using a limited vocabulary, Elman generated around 10 000 very short sentences. Each word, like each letter in our earlier example network, was allocated its own unique activation pattern. So a sentence, when presented to the network, would cause a sequence of activation patterns across the input neurons that corresponded to the sequence of words making up the sentence. These would spread through the network and cause a sequence of activation patterns across the output neurons. The network's task was to predict, after each word in the sentence, what the next word was going to be. Each output pattern would be compared against the pattern allocated to the next word in the input sequence, and the connection strengths changed (ever so slightly) according to how different the two patterns were. This was repeated many times for each of the 10 000 sentences. Not surprisingly, the network never managed to predict, with any great success, what the next word would ever be-something like `the boy' could be followed by any number of different verbs, for instance, but the network would not be able to predict which verb. So why all the excitement?

Elman knew full well that the network could not be expected to predict with any accuracy the next word in any sequence. But after repeated exposure to the many different sentences, it did none the less learn to output, at each step through the sequence, a complex pattern that represented the variety of different words that might occur next. If, for example, the boy ate a sandwich in one sentence, and a cake in another, the network would predict, after `The boy ate', that the next word would be `sandwich' or `cake'. It would do this by outputting the pattern of each one of these words simultaneously, superimposed on one another to form a composite pattern.

Elman was also interested in something else the network did. If a pattern of activation across the input neurons represented a particular word, what did the patterns of activation that developed across the intermediary neurons represent? With successive exposure to different sequences (or the same sequence, come to that), the learning procedure changed all the connection strengths linking the input neurons to the intermediary ones. The same input pattern would therefore lead to different intermediary patterns as the learning progressed. So different patterns evolved during (and as a consequence of) that learning. If activation patterns can be equated with representations, that means that the network had evolved its own, internal, representations. But of what?

In order to answer this question, Elman waited until the network had reached a point where the changes were very slight indeed, and, in effect, there was nothing more it could learn from successive exposure to the sentences it was being trained on - it was not getting any better. He then used a statistical procedure to analyse all the intermediary activation patterns that were produced in response to each input word. This allowed him to see which words caused similar patterns of activation across those intermediary neurons, and which caused different patterns. He found that all the nouns produced similar activation patterns, and the verbs did so too, but the two sets of patterns were quite distinct. Crucially, the patterns input to the network across the input neurons were just arbitrary patterns. Some might have been similar through chance, but others were quite different. But this did not stop the network from learning to distinguish between the noun patterns and the verb patterns.

In addition to being able to distinguish between nouns and verbs, the network also learned to distinguish between transitive and intransitive verbs ('chase' vs. `sleep'), and between animate nouns ('boy', `lion', `monster') and inanimate nouns ('sandwich', `plate', `glass'). It distinguished also between animals and humans, and between things that were edible and things that were not. In each case, words within one category would cause patterns of activation across the intermediary neurons which were similar, but which would be quite different from the patterns caused by words from another category. How could any of this come about?

The only information available to the network was in the form of activation patterns across its input neurons. For each word (that is, for each activation pattern), it had information about what had come beforehand in the sequence, and what had come after. And that is exactly the information that distinguishes nouns from verbs-they occur in different contexts. For instance, in the sentences that Elman used, verbs would be preceded by nouns, and nouns sometimes by verbs, but never by nouns. Similarly, certain nouns would occur in the context of certain verbs but not in the context of certain other verbs: inanimate nouns could only occur before certain kinds of verb, edible nouns after certain kinds of verb, and so on. In fact, all the distinctions that the network made were based solely on the fact that different kinds of word occurred in different kinds of context. The network's memory meant that it could `spot' that certain kinds of word occurred in certain similar kinds of context, whereas certain other kinds of word occurred in different kinds of context.

This still fails to explain how the network spotted anything at all. It would need some mechanism that would cause it to form representations that were defined by the contexts. This is where the prediction task comes in. The first step in the argument to explain all this is that the network did learn to predict which words could come next. And because the different words that can occur in the same position within a sentence must have the same syntactic category (e.g. noun, verb), the output patterns would necessarily come to reflect exactly those categories, with finer distinctions being made for subcategories that appeared in some contexts but not others-hence the distinction between animates and inanimates, transitives and intransitives, and so on. So the output neurons reflected the syntactic category of the next word in the input. But something must also have reflected, for the right category to be predicted, the syntactic categories that had come earlier in the sequence. That is what the intermediary neurons did. In Elman's sequences, the best predictor of the next word was the immediately preceding word (in fact, the current word being `seen' by the network), so the most obvious characteristic that was encoded by those neurons was the syntactic category of that word. In effect, this reflected the range of words that could occur in that position in the sentence.

The general principle at work here is that the intermediary neurons encode whatever property of the input sequences allows the correct (or best) predictions to be made at the output neurons. This happens because the connection strengths within the network change as a function of how good the prediction has been. If the intermediary neurons manage to encode a property of the input sequences that is highly predictive of the correct output, the strengths will be changed only very slightly, if at all. But if the intermediary neurons have failed to encode any property that is predictive of the correct output, the strengths will be changed quite substantially, across the many exposures that the network receives. And because the network's memory is encoded in those connection strengths, anything that is not predictive will, in effect, be forgotten.

So the Elman net does not use its memory to store a faithful reproduction of everything that it has ever encountered. If `sandwich' and `cake' had occurred in exactly the same contexts, they would have given rise to the same internal representations (that is, patterns of activation across its intermediary neurons). But in real language, different words tend to occur in different contexts, and Elman's simulations attempted to capture this. `Sandwich' and `cake' occurred in subtly different contexts and gave rise to subtly different representations, but whatever could be predicted by `sandwich' that was the same as whatever could be predicted by `cake' was encoded, by the network, in that part of the representation that was common to both words. This explains why all the words of the same syntactic category evoked, in Elman's network, similar activation patterns across the intermediary neurons-the overlap between the individually distinct patterns conveyed information that applied to each word in that category, or, in other words, the appropriate generalizations.

One final property of these networks: if the network sees a word like `the', it can predict that the next word will be a noun (e.g. `cake') or an adjective (e.g. `big'). So a composite pattern will be output that reflects both these possibilities. However, if, in the network's experience, it is more common for a noun to follow words like `the' than it is for an adjective to follow them, the pattern will reflect the difference in the frequency of occurrence. This is a straightforward consequence of the manner in which the connection strengths are changed slightly each time the network encounters each word. If there are lots of nouns in that position, they will pull the strengths in one direction. If there are lots of adjectives, they will pull in another. The final balance depends simply on how many pulls there are in each direction. So, not only does the output of the network reflect the range of predictions that are possible at each point in the sequence, it also reflects the likelihood of each of those predictions.

That, briefly, is the Elman net. We do not know whether the real neural networks operating in our brains do the same kinds of thing. Probably, they do not. But in all likelihood, some of the principles are the same. The remainder of this chapter will look at how networks exhibiting the same properties as the Elman net might explain some of the phenomena that the preceding chapters have introduced. Much of what follows is conjecture, but it is conjecture based on observable fact.

On the meaning of meaning

In the Elman net, the information that was stored about each wordwhether it was specific information or general information shared with similar words-was about the contexts in which each word could occur. `Cake' appeared after verbs like `eat', not `chase'. Dog' followed verbs like `chase', but not verbs like `eat'. Yet they both followed verbs. With this limited information, the network made subtle distinctions between edible things, inedible things, animals, and humans. It made distinctions that we might normally think of as having something to do with the words' meanings.

In Chapter 9, the meaning of something was defined as the knowledge about the contexts in which that something could occur. By this criterion the Elman net had acquired an element of meaning. It was very limited, because the only context it had available to it, and on which basis it could distinguish between different words, was the linguistic context. But imagine that the network could receive information more generally about the contexts in which those words would ordinarily be used. The network might have neurons that received information from a retina, or from an ear. The network would not know that these different inputs corresponded to different kinds of information, just as it did not know that the input in Elman's original simulations reflected words in a language (and just as our own neurons do not know what their input reflects). But the network still did a good job of categorizing the words in ways which, as defined by that language, were meaningful. With additional inputs, reflecting other aspects of the contexts in which those words would ordinarily be experienced, the network ought to be able to do an even better job. The nature of the prediction task means that only aspects of the context that are predictive of the current word would be encoded.

One of the puzzles in Chapter 4 was to explain how a child would know which aspects of the context to associate with the sounds he or she heard. That puzzle is effectively solved if the only aspects selected are those that are predictive, or predicted by, those sounds. And this, according to the descriptions given in Chapter 9, is exactly what is required in order to capture the meaning of something. In fact, meaning is nothing more than that very encoding. In principle then, even an artificial neural network could achieve such an encoding-it could achieve meaning.

Who did what, and to whom

All this talk of neural activation, the encoding of experience, and prediction, is a far cry from the earlier talk (in Chapter 8) of participants, roles, and the assignment of one to the other. On the face of it, it looks as if we have ended up with an analysis of how we derive meaning that is quite different from that earlier role-assignment approach. In fact, we have simply ended up with a different vocabulary for describing the same process.

One of the puzzling properties of the way in which we go about assigning roles (to use that vocabulary) is that we apparently assign them without waiting for the grammatical information that would unambiguously signal which assignments should be made. We assume that, in the sequence `The woman that Bertie presented the wedding ring ... the woman is being given the wedding ring even before we encounter the grammatical information, later on in the sentence, that would tell us whether this was right. It need not be. The sentence could be `The woman that Bertie presented the wedding ring to his fiancee in front of was his cousin'-yes this is a difficult sentence, but if we blindly obeyed the principles of grammar, we should not assign any role to the woman until we reached the gap between `of and `was'. It looks from the evidence (see Chapter 8) as if there is some sort of need to allocate each participant a role as soon as one becomes available. We are even prepared to make preliminary role assignments which must subsequently be revised. Why? This is where the more recent talk of neural encoding and prediction comes in.

When we encounter a sentence like `A balding linguist ate a very large fish', our experience of similar linguistic contexts ('an X Y'd') correlates with our experience of X doing the Y'ing (as opposed to X being Y'd). When the verb `ate' is encountered in this sentence, a pattern of neural activity ensues which reflects this experience. And in so doing, it reflects, in effect, the assignment of the `eater' role to the linguist. When `a very large fish' is encountered, the ensuing pattern of neural activity reflects the assignment of the `being eaten' role to the fish. So that is how the neural equivalent of role-assignment works. But each pattern of neural activity also constitutes a prediction of what the successive patterns will be. In a world in which linguists ate fish 75% of the time, and spaghetti the remaining 25%, the patterns of neural activity after `ate' would reflect these differences. They would, in effect, predict one assigmment as being more likely than the other, before the sentence unambiguously signalled which was the correct assigmment. Of course, we do not live in such a world, but this example demonstrates that if, after `ate', there was a strong likelihood that one thing, rather than another, would fill the being-eaten role, then this would be reflected in the pattern of neural activity at that point.

In sequences of the form `The X that . . .' (as in `The woman that. . .'), the X will be assigned a role by something later on in the sentence, in the relative clause. The X could in principle fill any of the roles that become available when the next verb is encountered. And because the predictions that are made at each step in a sentence reflectwhat is possible given our experience, it follows that the neural activity evoked by the sequence `The woman that Bettie presented the wedding ring ...' will reflect the possibility that the woman is the recipient of the wedding ring. And because, in our experience, the thing that occupies the same position as `the woman' in this sentence is almost always assigned one of the roles associated with the next verb, the possibility that the woman is the recipient in this case is very strong. The neural activity would reflect the strength of this possibility. Itwould reflect, in effect, that particular role-assigmment, before the point in the sentence that would unambiguously signal that this was the correct role assigmment. The relationship between meaning, prediction, and experience, makes such `early' role assigmments an inevitability.

Time flew like an arrow

The link between prediction and meaning ensures that only certain aspects of our experience (or a network's) become encoded as the meaning of something. It also ensures that when a particular combination of contextual factors is encountered, certain predictions will bemore likely, and so more influential, than others. This has consequences for the way in which ambiguities are resolved. An example from Chapter 7 involved eating pizza with your friends, your fingers, your favourite topping, your favourite wine, or your customary enthusiasm. The image that is conjured up by hearing that your friend ate pizza with his favourite film star probably does not involve that film star being poured into a glass, being used to cut through the pizza, or being sprinkled on top. We are usually quite unaware of these other possibilities. Why? Because past experience prevents the corresponding predictions from being made.

Chapter 7 ended with the observation that many factors can influence our interpretation of ambiguous sentences. Sometimes it might be the plausibility of the role-assigments. At other times it might be the frequency of the occurrence in the language at large of one kind of interpretation rather than another (or perhaps, of one kind of grammatical structure rather than another). At other times it might be the fit with the context. Each of these different factors will cause the network (real or artificial) to predict some aspect of its future input.

These factors are influential only insofar as they are predictive. Some factors may be more predictive than others. For example, the frequency of occurrence of a particular syntactic sequence in the language at large may be much more predictive of what will happen next than any other factor. But on occasion, and depending on the context, some other factor may be more predictive. The patterns of activation that encode these predictions will be superimposed one on the other, and depending on the precise circumstances (the preceding input) one pattern may dominate. What counts is not the kind of information that constitutes each factor, but simply how predictive it is.

Words and how we found them

An Elman-like network with sufficiently rich input could derive meaning from sequences of words. At least, that is the conjecture. But why should the linguistic input to this hypothetical network be confined just to sequences of words? As far as the original Elman net was concerned, it was simply experiencing different activation patterns across its input neurons. It could not know what those patterns correlated with in the world beyond its neurons. In Elman's original experiments, the activation patterns across the input neurons did represent whole words. But, although we perceive what we hear as sequences of words, each word is itself a sequence of phonemes, and phonemes can themselves be broken down into subtly different acoustic patterns. If an Elman-like net was given these acoustic patterns as input, what would happen?

By trying to predict what will come next, an Elman-like net will learn to encode information that is predictive of the kinds of context in which an input sequence might occur. If it is a sequence of phonemes (or of the acoustic patterns that make up each phoneme) the network will learn to encode information about the contexts in which those sequences might ordinarily occur. It would be able to predict the range of phonemes (or equivalent) that could continue a sequence-in effect, the range of words that are compatible with the sequence of phonemes heard so far. And as the network `heard' progressively more of any one sequence, the number of possible continuations would fall, and the network would progressively activate patterns that reflected more strongly the remaining predictions. But there is more. With the right kinds of input (more than just the linguistic input), and sufficient exposure to the language, the network could in principle learn not simply what the next phoneme was likely to be, but more generally it could learn about the context in which that entire sequence would occur-in effect, the meaning of the word composed of those phonemes. So as more of each word was heard, the network's internal activation patterns would reflect more strongly the meaning of each word that was still compatible with the input. Eventually, when no other continuation was possible, they would reflect just the one meaning.

The description given in Chapter 6 of this process included the idea that a sequence of phonemes stimulates a neural circuit much like a sequence of numbers opens a mechanical combination lock, with successive tumblers falling into place one number after another. This analogy is in fact inappropriate. The different neural circuits are not physically separable in the same way that different combination locks are. One can think of each word that is input to the network as activating a separate neural circuit, but it is the same neurons each time, just with different activation patterns across them. It is the same connections also.

The process by which we recognize spoken words is complicated, as we saw in Chapter 6, by the fact that the same word will often be pronounced using different phonemes, depending on the surrounding words. So the sequence corresponding to `Hand me that thin book' might on occasion sound more like `hameethathimboo'. One possible solution, mentioned in that chapter, was that we use rules which define the circumstances in which a phoneme of one kind should be interpreted as a phoneme of another. We would then recover the meaning of the re-interpreted sequence-something that sounded like `thim' would be re-interpreted as `thin' if the following phoneme had been a /b/ (as in `book'). A rule of this kind is nothing more than a statement of the contextual conditions in which a particular meaning should be associated with one sequence of phonemes rather than another. This is exactly the kind of thing that networks can learn. If `thim' had been experienced in exactly the same contexts as had been experienced with `thin', except that the following phoneme was a /b/, the network would inevitably activate a pattern across its intermediary neurons that reflected this experience. As long as `thim' was encountered before a

/b/, the network would activate a pattern across its intermediary neurons that was, to all intents and purposes, the same as that for `thin'.

To the extent that linguistic rules are simply generalizations about the contexts in which one can predict one thing to happen or another, a network exhibiting the same properties as an Elman net ought to be able to encode information that is equivalent to those rules. In fact, it is difficult to see how else a set of rules could be encoded within the neural circuitry.

Words and what we learnt to do with them

In this final section, we move away from what an Elman-like net could in principle do, back to what Elman's net actually did.

Many linguists and psycholinguists believed that the acquisition of grammatical knowledge would not be possible if the only input to the learning device was the language itself. The basic problem was that grammatical knowledge was believed to exist as rules about the relative positioning of syntactic categories (things like `noun' and `verb') within the sentences of the language. But if you did not know about syntactic categories, how could you generate these rules? Even if you knew about which syntactic categories existed in your language, how would you know which words belonged to which category? The only solution to this problem (so the argument went) was to assume that some of the knowledge that was necessary for learning about grammar was innate (see Chapter 4). Several researchers suggested instead that because those linguistic rules were simply generalizations about which words could occur where in the sentence, all the child needed to do wa; calculate the equivalent-in other words, calculate the individual positions of each word relative to each other word in the language. The problem, at the time, was how the learning device would avoid learning irrelevant facts. Knowing that a noun might appear four words before a verb, or seven words after `a' or `the' would not be very helpful. In any case, it would be impossible to store in memory every possible fact about every possible position in which every possible word could occur. The Elman net demonstrated a simple solution.

The only information that Elman's network encoded was information that was predictive of the next word. In effect, it did simply calculate the position of each word relative to each other. It kept the information that was predictive, and discarded the information that was not. It encoded the information it kept as a combination of information that was specific to each word, and information that constituted generalizations that applied to whole groups of words. And as we saw earlier in this chapter, by developing those generalizations, the network developed the equivalent of syntactic categories and knowledge about which order they could appear in. It acquired grammar.

It might appear that there is little these networks cannot do. But an inherent problem with many artificial neural networks is that they can be quite unpredictable. Their mathematical properties do not guarantee that two networks with, for example, different numbers of neurons, will behave in exactly the same way. Much of the previous discussion is therefore limited to speculation until such time as the appropriate networks are built, and trained on the appropriate kinds of input. But even if this happened, and they did all the things we hoped they could do, there would still be a lot they could not do. They would not play. They would not have the same drives or desires as their human counterparts. They would not sit on the kitchen floor and lick the cake mixture off the spoon. They would not interact with their environment. They would not develop physically. Would that matter, though? To the extent that these are all parts of the child's experience, yes. To the extent that artificial neural networks are being built which mimic aspects of neural development, or which can learn to interact with their environment in appropriate ways, perhaps not.

Elman's prediction task is appealing because as tasks go, it has obvious evolutionary advantages-watching a moving object, catching a fly, chasing a beetle, and fleeing a foe all require the organism to predict, from one moment to the next, what is about to happen. But although it has obvious appeal, is it really enough to explain the phenomena that we have been dealing with here? Probably not. Even a frog can track a moving object, catch a fly, chase a beetle, or flee a cat. But can it talk? Does it understand? True, it does not grow up in the same environment that we do, but attempts to bring up even chimpanzees in the home environment have failed to produce a human chimpanzee. Their language is at best limited, and there is considerable controversy surrounding the claim that such chimpanzees can acquire even the smallest rudiments of grammatical knowledge. Perhaps what differs across the species is the sophistication and subtlety of prediction that each species is capable of. At the least, the difference must have something to do with differences in the sophistication, subtlety, and development, of their neural networks. It is therefore instructive to consider the principles, predictive or otherwise, that might be shared between natural and artificial neural networks. To echo a by-now familiar theme, the methods may well be different, but the principles may well be similar. And if we understand those principles, we necessarily understand better whatever lies at Babel's summit.