16

The Molecular Basis of Memory

The question of the physical basis of memory is one of the great questions in the life sciences. It has been recognized as such for more than a century. In Chapter 10, we quoted Christopher Koch to the effect that for over a century the leading hypothesis of both theoreticians and experimentalists has been that the physical realization of memory in the brain was an enduring change of some kind wrought by experience in the conductance of a synapse. This is the synaptic plasticity hypothesis. However, as Koch notes, it has proved extremely difficult to establish a convincing link between the behavioral manifestations of memory and its biophysical substrate. The upshot is that ". . . the notion that synaptic plasticity is the primary substrate of long-term learning and memory must at present be viewed as our most plausible hypothesis" (Koch, 1999, p. 308).

We could go farther and say that it is not only the most plausible hypothesis, it is the only hypothesis that has ever been seriously entertained by the community of researchers interested in the physical basis of memory. By our lights, that is the problem. The reason that after a century of determined effort by armies of investigators we still do not have a convincing story about the physical basis of memory is the plasticity hypothesis itself. It fundamentally misrepresents the nature of what it is we should be looking for.

The Need to Separate Theory of Memory from Theory of Learning

We argued in Chapter 11 that this misrepresentation of the nature of memory derives from a longstanding misrepresentation of the nature of learning within psychology. Because of its historical roots in the empiricist philosophy of mind, psychology has conceptualized learning as the remolding of a plastic brain by experience. On this view, learning and memory are one and the same process. Learning is the remolding process; and the remodeled brain embodies memory in the altered connections that experience has produced. That explains why Koch takes synaptic plasticity to be an hypothesis that explains *both* learning and memory.

On the view that we have argued for, learning is the extraction from experience of behaviorally useful information, while memory is the mechanism by which information is carried forward in time in a computationally accessible form. On this view, it is hard to see how one could have a single hypothesis about the physical basis of both learning and memory. The physical basis for one could not be the physical basis for the other. One could know with certainty what the mechanism was that carried information forward in time in a computationally accessible form, but have no idea what the mechanism was that extracted a particular piece of information from some class of experience. Computer scientists know the mechanisms that carry information forward in time within a computer, but this knowledge does not tell them how to compute a parsing of video input useful to a robot trying to recognize objects of a given kind. Conversely, one might believe with some certainty that one had correctly identified the neurobiological machinery that extracted a particular kind of information from a particular kind of experience, but have no idea how the nervous system was able to carry that information forward in time in such a way that it was accessible whenever it was needed in some later computation. This is pretty much the situation in which researchers who study mechanisms of sensory processing find themselves. Their focus is on the mechanisms by which the brain computes properties of distal stimuli from the sensory signals generated by proximal stimuli. They do not concern themselves with the question how the information thus extracted may be preserved for later use.

The Coding Question

The importance of correctly characterizing the nature of memory lies in the fact that this characterization determines what properties a proposed mechanism of memory must possess. If a memory mechanism is understood to be a mechanism that carries information forward in time in a computationally accessible form, then the first and most basic property that a proposed mechanism must possess is the ability to carry information. The synaptic plasticity hypothesis in any of its historically recognizable forms fails this first test. There is no way to use this mechanism in order to encode the values of variables in a computationally accessible form. That is why whenever the need arises in neural network modeling to carry values forward in time – and the need arises almost everywhere – recourse is had to reverberating activity loops. That is also why one can search in vain through the vast literature on the neurobiological mechanisms of memory for any discussion of the coding question. The question "How could one encode a number using changes in synaptic conductances?" has, so far as we know, never even been posed. And yet, if our characterization of the nature of memory is correct, then this is the very first question that should arise whenever suggestions are entertained about the physical identity of memory in the brain.

It is important to realize that this is a well-posed question to which unequivocal answers can readily be suggested. Suppose, for example, that one were to suggest that the memory mechanism co-opted some of the molecular machinery by which phylogenetically acquired information is transmitted from generation to generation. If alterations in nucleotide sequences in either DNA or RNA were a proposed mechanism of memory, then the answer to the coding question would be immediately apparent: How to encode a number in a nucleotide sequence is no mystery. The mystery is how to construct machinery that could effect the encoding. We have no idea what the machinery might look like that would transcribe information from a spike train into a nucleotide sequence. How to gain access to the encoded number is also a mystery. We have no idea what the machinery might look like that would transcribe a nucleotide sequences into a spike train. But how it is that a nucleotide sequence could in principle encode a number is no mystery at all. Thus, such a proposal, wild as it is, passes the test that the synaptic plasticity hypothesis fails. And that is progress. Because there is no point in pondering how to transcribe information from spike trains into some enduring physical change if there does not appear to be any way for that enduring physical change to carry information. You cannot transcribe information into a medium that cannot hold information.

A second example of a possible memory mechanism that again answers the first question is a molecule like rhodopsin. Rhodopsin is a photon-activated molecular switch. It has two different configurations (isomeres). The absorption of a photon flips the switch from what might be regarded as its "open" (inactive) state to its "closed" (active) state. Both states are thermodynamically stable; no matter which state the switch is in, the system does not have to expend energy maintaining it in that state. This is highly desirable in a basic memory element from which one is to imagine constructing a memory system capable of storing large amounts of information. It stands in marked contrast to suggestions that memories are maintained in reverberating activity loops. One would have thought that the extravagant waste of energy in any such system would have been enough to have long ago taken such suggestions off the table, but, as we saw in Chapter 14, this is far from the case.

When we have settable molecular switches at our disposal, there is again no mystery about how to use them to encode a number. So this suggested mechanism passes the first test. It is not so hard to imagine transcription mechanisms from spike trains to settings of these switches and from those settings back to spike trains. The transcription from these switch settings to spike trains is what the first two stages of neural processing in the retina effect. We understand a lot about the mechanisms by which information about the number of photons captured by the molecular elements of the photosensitive array inside a rod are converted to the spike trains in ganglion cells that carry that information to the brain for further processing. It is also not hard to imagine how a spike train releasing transmitter onto a metabotropic receptor could set in motion an intracellular molecular cascade that changed the settings of intracellular molecular switches. Metabotropic receptors give external signals access to the cell's internal signal system, which is much richer and more complex than one would realize from most standard texts in neuroscience.

So, are we suggesting that the memory mechanism is a change in nucleotide sequences or some other ingenious adaptation of the molecular machinery that is already known to have an information-carrying function? Or are we suggesting the mechanism is a bistable molecular switch like rhodopsin? No. We refuse to make any specific hypotheses about the molecular basis of memory, because we have no faith in our ability to guess the answer to this profound mystery. We do think that both suggestions should be given some consideration, if for no other reason than that such consideration may lead to more promising or plausible alternative suggestions. At the very least, considering these suggestions brings the coding question into the center of attention, which is where it belongs in any discussion of the physical basis of memory.

A Cautionary Tale

In our reluctance to speculate about the molecular mechanism of memory we are greatly influenced by the history of molecular genetics. Before Watson and Crick (1953) deduced the structure of DNA, the gene was such a profound biochemical puzzle that a substantial minority of biochemists doubted its physical reality (Judson, 1980). They thought genetic theory was a physically unrealizable fantasy founded on a biochemically unrealizable entity, the gene. A gene was assumed to have two utterly mysterious properties: it could make a copy of itself and it could direct (not catalyze, but actually direct) the synthesis of other molecules. Because the properties of a gene assumed by geneticists made no biochemical sense, some biochemists simply refused to believe in its physical reality, despite what I think almost anyone in retrospect would argue was a very large body of evidence and theory arguing that there had to be such a thing. Even some geneticists were content to regard genes as just a convenient way of talking about the huge body of experimental facts that had grown up around the study of patterns of inheritance.

Other biochemists were persuaded by the geneticists' data and theory that there must in fact be genes, that they were the sort of thing for which one could hope to have a biochemical account. They speculated about the biochemical nature of the gene. What impresses us is that, so far as we know, none of these speculations got close to the mark. The solution was beyond the power of pre-1953 biochemists to imagine. And yet, it was dazzlingly simple. When one looked at the structure suggested by Watson and Crick (1953), the scales fell from one's eyes. It was apparent that here in principle was the solution to both mysteries. The two strands, with their complementary sequences of nucleotides, immediately suggested how *this* molecule might make a copy of itself, as Watson and Crick noted in their famously coy one-sentence penultimate paragraph. It was also clear that this structure was in principle capable of encoding any information one liked, including, a fortiori, information about the sequence of amino acids in a protein. In fact, the structure immediately put the coding question at the center of the discussion for those alert to the profound importance of what Watson and Crick had achieved (Judson, 1980). As in the contemporary neuroscience of memory, the coding question can be said to have hardly existed prior to the revelation of the structure of DNA. Coding questions and answers thereto are now, of course, embedded in the foundations of molecular biology.

Why Not Synaptic Conductance?

In our experience, it is easier to persuade people that memory is a mechanism that carries information forward in time in a computationally accessible form – and that the coding question is indeed a question that must be answered – than it is to persuade them that it is, *therefore*, unlikely that changes in synaptic conductance are the mechanism of memory. The synaptic plasticity hypothesis has a truly formidable grip on people's imagination. They have great difficulty imagining that it could not be true. They always want to know, at this point in the discussion, why could the mechanism not be changes in synaptic conductance? The short answer is that, of course, it could be. But we think that it is not a likely possibility. Here is why.

First, if we go there, we have to give up the underlying idea that we can identify a change in synaptic conductance with the traditional notion of an associative bond. Associative bonds clearly cannot encode information. No associative theory ever propounded specifies encoding and decoding rules for associations, rules that would enable a reader who knew the encoding rule to deduce from the vector of association strengths the states of the world that created those association strengths. It is no accident that associative theories have always been anti-representational. They have always been so because associations were never conceived of in such a way that would enable them to function as symbols, that is, as entities that carry information forward in time by virtue of their enduring physical form.

Second, and closely related to the first point, we would have to change fundamentally the traditional conception of the architecture in which these putative memory elements (changeable synaptic conductances) are assumed to be embedded. If our story is going to be that there exists a mechanism that transcribes the information in a spike train into changes in synaptic conductances in such a way as to preserve that information, then the traditional considerations about the role of temporal pairing become irrelevant. The information in spike trains may very well reside in the intervals between the spikes (Rieke et al., 1997); indeed, we think it does reside there. But assuming that does not give us a story about how that information is transcribed into an enduring physical change that preserves the information in a computationally accessible form. There is no reason to assume that the process would bear any resemblance to the processes traditionally assumed to mediate changes in synaptic conductance. We would also have to assume that in order to access the information encoded in the synaptic conductances, the system probes the synaptic conductances with a read signal. We *can* assume all these things; indeed, if we are going to make plastic synapses the memory elements, we must. In our experience, one's enthusiasm for the idea diminishes when one contemplates the assumptions that must be made to make it work.

In the final analysis, however, our skepticism rests most strongly on the fact that the synapse is a circuit-level structure, a structure that it takes two different neurons and a great many molecules to realize. It seems to us likely for a variety of reasons that the elementary unit in the memory mechanism will prove to be a molecular or sub-molecular structural unit. Our rhodopsin-like switch molecule suggestion is a mechanism in which the basic memory element (a switch-like molecule) is a molecular-level unit, while our nucleotide-sequence suggestion is an example in which the element is a sub-molecular unit.

A Molecular or Sub-Molecular Mechanism?

The first relevant consideration is that it is clearly possible to implement this function at the sub-molecular level, given how much of the requisite machinery is already realized at the sub-molecular level in DNA and RNA. A large share of the very successful effort that has driven the ever-increasing power of computers has been the relentless reduction in the physical resources required to store a bit. This reflects the fundamental role that storing and accessing information plays in computation – the central theme of this book. For something as basic as memory, the simpler, more compact and less energetically costly the realization of this basic function is, the better it is. To our mind, it would be more than a little curious if a basic function that could be better implemented at the lowest possible level of structure (the sub-molecular) were found to be implemented instead at the circuit level, an implementation requiring orders of magnitude more physical resources.

The second, closely related consideration is the evident and profoundly puzzling speed of neural computation. How it is possible for the nervous system to compute the complex functions that it does compute as fast as it does, given that signals travel eight orders of magnitude more slowly in the nervous system than they do in a conventional computer? In computation, most of the signal flow is to and fro between memory, where the symbols reside when not entering into computations, and the processing machinery that implements the primitive two-argument functions. We believe that this aspect of the structure of a conventional computer is dictated by the ineluctable logic of physically realized computation. Given that signals travel slowly in neural tissue, the only way to minimize the time consumed in transferring information from memory to the processing machinery and back again is to place both the memory machinery and the processing machinery as close together as is physically possible. This is where Feynman's (1959) famous argument that "There is plenty of room at the bottom" comes into play. When interpreted in a neurobiological context, we take the force of his argument to be that sub-molecular and atomic structures are many orders of magnitude smaller than the cellular or circuit structures. Insofar as functions and structures that are imagined to be implemented at the level of cellular or circuit structure can in fact be implemented at the level of molecular structure, there is a gain in functional capacity of many orders of magnitude. One can accomplish much more in much less space – and in much less time, because much less time is wasted transmitting signals over needlessly long distances.

Bringing the Data to the Computational Machinery

Here, we remind the reader why we think the architecture of any powerful computing machine must be such as to bring the data to the computational machinery.

Complex functions can be realized by the composition of functions of two variables, but they cannot be realized by the composition of functions of a single variable. Thus, a computing machine must physically realize some two-argument functions. To physically implement a two-argument function, that is, to make a machine that gives the requisite output for each pair of input values, the values of the two input variables must converge in space and time on the processing machinery that generates the output for those two particular inputs. In order that two numbers be added, they must converge on machinery that can generate their sum.[1] In a powerful computing device, the number of different variables whose values might have to be summed is essentially infinite. Values cannot practically be stored in memory in such a way that they are all physically adjacent, each with every other, *and* with processing machinery capable of generating a sum. Because the two values that may need to be summed cannot generally be physically adjacent in memory, the computing machinery cannot be brought to where they are. They are not in one place; they are in two different places. Therefore, the architecture of the computing machine must make provision for values to be retrieved from physically different locations in memory and brought to the processing machinery that implements the primitive two-argument functions.

The question arises whether a distributed representation of the values stored in memory does not invalidate the above argument. The idea behind a distributed representation is that the same set of memory elements (e.g., plastic synapses) is used to represent all of the values stored in a memory, with the state of every element contributing to the representation of every value. In such a representation, the different values are not located at different places within the memory. It is not the case that some memory elements are used to represent some values, while other memory elements are used to represent other values. The representation of every value is distributed across all the memory elements. So, contrary to what we said above, the values are stored in such a way that every value is physically at the same location as every other value. They are all of them everywhere.

Such representations are possible. Encryption procedures generate them. An encryption procedure treats the bit patterns in successive words of memory as one huge bit pattern and performs on that pattern a hard-to-invert arithmetic operation (e.g., multiplication by a large prime) that generates a new bit pattern that is unintelligible. Only if one knows the prime by which the set of values was multiplied is it possible to recover the original set of values (decrypt the encrypted memories). The essential point for present purposes is that the recovery of any single one of the values in the original set of bit patterns depends on knowing every bit in the bit pattern generated by the encryption. Thus, there is a clear sense in which every bit (hence, every memory element) in the encrypted representation participates in the representation of every value in the encrypted set of values.

The problem is the unintelligibility of the distributed representation. The values thus represented are not accessible to computation. Indeed, the goal of the encryp-

[1] In what follows, we continue to use the sum function as a stand-in for the set of primitive two-argument functions implemented by the computational hardware.

tion is to render them inaccessible to computation. We believe (but we cannot prove) that there is no way to realize primitive two-argument functions within a distributed representation of their arguments. The question is this: Does there exist a physically realizable distributed representation of a large number of distinct values such that one can send into that representation two probe signals to activate an arbitrarily chosen pair of the values therein represented, and another signal that specifies a two-argument function of those values, and get out the value of the specified function for the specified input values? We believe that this is impossible, because, for one thing, we do not see how it can be possible to simultaneously activate two and only two of the values represented in the distributed representation.

We can imagine that the distributed representation is such that when we send in one probe signal (input vector) we get one signal out from the memory (one output vector) and when we send in a different probe (a different input vector), we get out a different output vector. That, however, is not what is required. If that is the only capability we have, then in order to realize a function of two values stored in that distributed memory, we need to first extract one value, hold it in a memory register while we extract the second value, then bring the two sequentially extracted values to the machinery that can realize the specified two-argument function, and then put the value thus obtained back into the distributed representation, changing in the process the value of every memory element(!). Thus, even though values in a distributed memory are "all in the same place," this scheme nonetheless requires that they be separately extracted and brought to the machinery that realizes a two-argument function.

It is difficult (for us at least) to imagine how two and only two different values in a distributed representation could be simultaneously activated. Presumably, their activation would require the activation of the memory elements that encode them. But in a distributed representation, every memory element participates in the representation of every value. Thus, a probe for any value must activate every memory element. How can the activation states of one and the same set of memory elements *simultaneously* represent *two and only two* different values arbitrarily chosen from among the large number of values whose representation is distributed across that entire set of memory elements?

This last question brings us back to a point we stressed in Chapter 10 and again in Chapters 14 and 15, namely that in schemes in which plastic synapses within neural networks are thought to be the memory elements, the values of the synapses themselves do not (and, except under highly constrained circumstances) cannot encode the values of variables. What they encode are not the values themselves, but rather a procedure that will generate different values *given different input vectors*. In such schemes, the memory elements do not themselves represent values. All that the connection weights represent are procedures for generating different values (output vectors) given different input probes. This is another manifestation of the idea that learning is a rewiring that alters behavior but does not represent the experienced world. That's why these schemes have a finite-state architecture. As we have just noted, their architecture does not realize a system in which there is no need to bring the arguments to the machinery that implements the functions of those arguments. Indeed, the look-up tables that generally implement two-argument functions in neural network computations (see, for example, Chapter 14) do get signals from memory (the moving bumps of activity) and send signals back to those moving bumps.

As has been pointed out by others (Fodor & Pylyshyn, 1988), the great weakness of neural network models is their inability to offer a generally effective solution to the problem of compositionality. In our terms, the problem is their inability to provide compact procedures that implement functions of two arguments without pre-specification of the possible arguments. As we have attempted to show, this problem appears over and over again when one contemplates contemporary neural network models for things like dead reckoning and spatial and temporal maps, data structures for which there is strong behavioral evidence in insects and in vertebrates far removed from humans. Why try to use such an architecture to implement a complex computation like parsing, when it cannot plausibly be used to carry out even extremely simple computations like dead reckoning or vector addition and subtraction?

In sum, we do not know what the physical mechanism of memory is. Moreover, we refuse to conjecture, except by way of illustration, what it might be. We refuse because we believe that we are unlikely to conjecture the correct answer in the face of our present level of ignorance. We are mindful that biochemists were to unable to conjecture the answer to the molecular structure of the gene in the face of much more abundant and relevant evidence, such as, for example, the Chargaff ratios (see Judson, 1980). What we do know is some of the properties that the mechanism must possess. It must be capable of encoding the values of variables. That is, it must be possible to see how, at least in principle, the proposed physical change could carry forward in time a number. We focus on the carrying of a number, because any system that can carry a number can carry any kind of information. It must achieve very high information density (bits per cubic micron). It must have low volatility; hence, a low use of energy to maintain the physical changes that carry the information forward in time. It must be capable of entering either directly or via transcription into physically realized compact procedures for implementing two-argument functions. That is a more detailed set of specifications than has heretofore guided the search for the physical basis of memory. It much more strongly constrains viable hypotheses.

Is It Universal?

Finally, bringing our book full circle, we consider an assumption implicit in our discussion so far, which is that it is reasonable to suppose that the mechanism of memory is universal. Our reasons for supposing that it is rest squarely on a point of Shannon's that we discussed in the opening pages. The function of memory in a computing device is to enable the present to communicate with the past. By carrying information forward in time in a computationally accessible form, memory makes possible the composition of functions, which is at the heart of computation. It makes the results of functions executed earlier accessible as arguments of current functions. Shannon began by pointing out that in considering the communication of information, one need only consider the probability distribution on the set of possible messages. One need not consider the semantics of those messages, what they are about. Put simply, information is information: in the end, it is all bits.

The modern communication and information technology industry bears witness to the truth of this insight. In designing a storage medium, such as a video disk, engineers are concerned only with how many bits it can hold, not whether those bits will encode text messages or images or sound streams. Similarly, in auctioning off or purchasing a segment of the electromagnetic spectrum, the government and industry are concerned only with how many bits can be sent per unit of time, not with the content of those bit streams.

The insight also finds striking confirmation in the fact that nucleotide sequences encode for both the amino acid sequences of proteins and promoters, the sites where transcription factors bind to DNA so as to initiate the transcription of one or more genes. The distinction between promoter encoding and amino-acid sequence encoding is roughly the distinction between encoding the data on which a program operates and the program itself. Conceptually these are very different. As we have explained at length, program code and data code play very different roles in the operation of a computing device. And, promoter sequences and so-called coding sequences play very different roles in the machinery by which inherited information finds expression in organismic structure. But, in both cases, the essence of the matter is the carrying of information. For that purpose, it is a logico-mathematical truth that if you can carry any kind of information, you can carry every kind of information.

The universality of the action potential as the medium for carrying information from one place to another within nervous systems is a further illustration of Shannon's insight. The action-potential mechanism is not domain- or modality-specific. Whenever information must be carried from one place to another, the same mechanism is used. Because the function of memory is to carry information from one place in time to a later place in time, we see no more reason to suppose that different mechanisms are required for different kinds of messages here than to suppose that different kinds of action potentials are required or different kinds of DNA.

In stressing this, we stress a central message of our book: Memory is a simple function. Simple as it is, however, it is absolutely indispensable in a computing device. If the computational theory of mind, the core assumption of cognitive science, is correct, then brains must possess a mechanism for carrying information forward in time in a computationally accessible form. This elementary insight will someday transform neuroscience, because, as of this writing, neuroscience knows of no plausible mechanism for carrying a large amount of information forward over indefinitely long intervals in a form that makes it accessible to the mechanisms that effect the primitive two-argument functions that are at the heart of computation.

Until the day comes when neuroscientists are able to specify the neurobiological machinery that performs this key function, all talk about *how* brains compute is premature. It is premature in the same way that talk of how genes govern epigenesis was premature prior to an understanding of what it was about the structure of a gene that enabled it to carry inherited information forward in time, making it both copyable and available to direct the creation of organic structures. Symbolic memory is as central to computation as DNA is to life.