CONTENTS

1

Lifeless Brains

The Brain, in Theory

In modern mainstream culture, both popular and scientific, the brain is a sort of computer, a machine that processes information. It acquires data in the form of sensory signals, encodes them into some electrical format, then processes the data with neural algorithms. It broadcasts the information to specialized processing modules: the visual cortex for visual processing, the hippocampus for memory storage and retrieval, the prefrontal cortex for decision making and planning. Eventually, it outputs motor commands to the muscles. Obviously, the brain is not a conventional computer with transistors, hard drives, and USB ports, but a "biological computer" optimized by evolution. The goal of neuroscience, then, is to "reverse engineer" the brain, to understand its functional organization and biological implementation.

All these concepts are borrowed from the engineering domain. This source of inspiration predates the era of computers. In the seventeenth century, brains were likened to hydraulic mechanisms; in the nineteenth century, the nervous system was a telegraph (Cobb, 2020, 2021). In much of the twentieth century, the brain was a computer applying formal rules to mental symbols. Nowadays, the brain might be a neural network, but the kind that engineers run on massive computers with graphics cards: a vector of values updated by series of matrix multiplications, with parameters tuned to minimize a formally defined error. In fact, the modern neuroscience literature simultaneously embraces all of those engineering concepts: neurons are mechanisms (like hydraulic machines) that communicate with codes (like telegraphs); they compute (like computers) with parameters tuned by learning algorithms (like formal neural network models).

Theoretical neuroscience, the activity of building mathematical models of the nervous system, heavily borrows from engineering theories: computer science, signal processing, data analysis, optimization, information theory, control theory. In fact, the main subfield of theoretical neuroscience is called

computational neuroscience, which aims at understanding how (not whether) neurons compute.

Engineering concepts have indeed been very fruitful in understanding the logic of living beings, and of nervous systems in particular. For example, telegraph theory has been used to develop the biophysics of action potential propagation in the 1950s by Hodgkin, Huxley, Katz, and colleagues (Hodgkin, 1964), as axons share similarities with electrical wires. In fact, the theory of electrical propagation in neurons is traditionally called "cable theory" (Rall, 2011). Optimization principles have been shown to be relevant to understand the structure of living organisms (Rosen, 1967) and of nervous systems in particular (Sterling and Laughlin, 2017). Indeed, the structure of living organisms appears to be particularly efficient at various functions that are especially important for the survival of the organism, such as harvesting and saving energy. This is why biology has in turn been an inspiration for engineering.

But it is one thing to borrow relevant concepts from engineering to understand brains, and another entirely to claim that brains actually *are* engineered. Many general views on mind and consciousness are indeed based on a strict identification between brains and engineered devices (mostly computers). For example, since we are computers and computers are not conscious, then consciousness must be an illusion (eliminativism). Or conversely, since we are computers and we are conscious, computers must be conscious after all, so consciousness may actually be everywhere to different degrees (panpsychism). If intelligence is just an input–output mapping fitted on large amounts of data, then surely with more data and computing power, "artificial intelligence" will soon outrun human intelligence, leading the human species to extinction or slavery (an event called the "technological singularity"). If minds are algorithms, then we should be able to upload minds in a computer simulation, indefinitely extending our lives (transhumanism). In fact, we might already be living in a simulation right now, without knowing it. If not, since mind simulation would allow us to create an astonishing number of new happy human lives, we should make all possible efforts to ensure it happens (longtermism).

Yet, if we were to explicitly ask a modern neuroscientist whether the brain is actually an engineered device, she would certainly strongly object. Brains are not the result of intelligent design. This is a religious view of life that has been discredited by Darwinism. Why then are we to "reverse engineer" brains, if brains were not engineered in the first place?

This terminology is typically excused by adding that brains are engineered *by evolution*, not by God. But Darwin's insight is precisely that evolution is *not* a case of engineering. Engineering is the use of knowledge to solve technical problems. It presupposes an external mind that plans and assembles machines

according to a preexisting goal. But evolution has no goals, plans, or knowledge; in other words, it is not an engineer.

Thus, living organisms are not *really* engineered. Therefore, they are not really machines, which are engineered objects, and brains are not really computers, which are kinds of machines. Of course, there are features of machines and computers that are shared by living organisms and brains, which is why engineering concepts can be relevant in biology. But if the idea that we are the result of intelligent design is to be scandalous to a modern scientist, then surely this should at least make *some* difference to the way we conceive brains?

It is the main aim of this book to explore these differences, in particular in the context of making models of the brain. It appears indeed that, in mainstream neuroscience and cognitive science, the idea that we are not engineered is simultaneously an extremely important opinion to hold publicly as well as a theoretically insignificant fact. Hillary Putnam, a major philosophical figure of cognitivism, explicitly claimed that our biological nature is insignificant: "we could be made of Swiss cheese and it wouldn't matter" (Putnam, 1975).

To set the stage, I will briefly outline the main modern theoretical frameworks to think about brains and cognition, starting with *computationalism*.

Computationalism

Computationalism holds that cognition is a form of computation, seen as the manipulation of formal symbols with rules. Brains are said to *implement* such computation, where symbols are represented by the state of some neurons, while brain processes change neural states in such a way that the corresponding symbols are changed according to the formal rules of the computation. Usually, the relevant states are believed to be the firing activity of neurons (how many action potentials they fire per second). As we will see in chapter 8, this is problematic because activity is not a state, let alone a computational state. Unorthodox computational accounts propose instead that symbols are represented by stable molecules such as polynucleotides (Gallistel, 2017). Regardless of the physical basis of computational symbols, it is the computation that matters for cognition, not its implementation. This doctrine is known as *functionalism*. (See Zahnoun [2023] for a critique.) Brains merely support computations; how they do so is largely irrelevant to understand cognition.

This functionalist perspective comes from the fact that a computer is a machine, and what matters for the behavior of a machine is the functional specification of the components, not so much their material basis. An electric car is still a car, because the electric motor produces a rotating motion transferred to the wheels, even though it works differently from a combustion engine. Accordingly, computationalism relies on a distinction between hardware (the

brain) and software (the mind). Cognition is defined at the level of algorithms, while neurons only implement those algorithms. Thus, biological implementation is secondary for the understanding of cognition: the mind can run on any material support, as long as the functional organization of computational states, identified to mental states, is preserved. Thus, with some imagination, the brain could be made of Swiss cheese.

Computationalism developed in reaction to *behaviorism*, which was the dominant conceptual framework about brains in the first half of the twentieth century. Behaviorism saw behavior as nested reflexes adjusted by experience, strengthening or weakening associations. But as early cognitivists pointed out, behavior is highly structured and goal-directed, and appears to depend on abstractions rather on the details of proximal stimuli, just like computations. This is obviously so in human reasoning, but it is also a well-documented feature of animal behavior. For example, bees can recognize whether two objects are the same or different (Giurfa et al., 2001) and can count up to four (Dacke and Srinivasan, 2008). Many species such as ants can return to their nest in a straight path after foraging (Wehner, 2020), meaning that they implicitly integrate their own displacement—an ability called dead reckoning. This does not seem to be possible by the mere association of physical cues.

While the cognitivist critique of behaviorism is relevant, it was hardly new. Merleau-Ponty, a phenomenologist philosopher, already pointed out in *The Structure of Behavior* (1942) that behavior is made of actions, not reactions. An action is performed by an agent with certain goals, and therefore it depends both on the organism's internal state and on some abstract features of the situation—for example, whether the given pattern of light is identified as a source of food. Organisms do not respond automatically to proximal stimuli. Rather, behavior is anticipatory: actions are taken as a function of their expected consequences. Computation is indeed also directed toward a goal, which is its result (the thing that we compute), but that is hardly surprising, given that computation is a kind of behavior—the kind we try to emulate in computers. However, the converse assertion, that all behavior and cognition are computational, does not follow, as we will discuss in more detail in chapter 4. In the same way, it seems that we can store and retrieve memories just like a computer, but it is the computer that was built to mimic some features of human memory—indeed, the word *memory* originates from the mental domain, not the engineering domain. It does not follow that the computer literally remembers what you wrote when you open a text file.

Computationalism led to the development of symbolic artificial intelligence, also known as "good old-fashioned artificial intelligence" (GOFAI), in particular expert systems, which implemented logical inference on a base of rules gathered from experts. Those systems made spectacular progress in the

1960s to 1970s, raising high hopes, as recounted by Mitchell (2021). For example, in 1960, Herbert Simon predicted that "machines will be capable, within twenty years, of doing any work that a man can do." Skeptics, such as the philosopher Hubert Dreyfus (1978), explained that experts do not actually rely on rules: it is beginners who use rules to guide their learning process. This unpleasant rebuttal was dismissed, but expert systems were eventually abandoned in the 1980s.

Despite the failure of these approaches, the perspective introduced by computationalism has remained dominant: cognition is a form of computation, and neurons encode symbols used by the brain to compute.

One of the difficulties encountered by symbolic artificial intelligence was with perceptual tasks, such as identifying an object. To address this difficulty, a very different approach was introduced, which did not use symbolic rules: *connectionism*.

Connectionism

The precursor of all artificial neural network models is the binary neuron model of McCulloch and Pitts (1943). In that model, the neuron is seen as either active or inactive, symbolized by 0 or 1, a feature inspired by the all-or-none law of neural excitation. It receives inputs from other neurons, and its output activity is calculated as follows (figure 1.1): take the weighted sum of the activity of input neurons (weights are called *synaptic weights*), and output 1 if the sum exceeds a threshold (otherwise 0). This makes the neuron implement a logical function with *n* inputs and 1 output. One can then build more complicated logical functions by connecting neurons together. In fact, McCulloch and Pitts demonstrated that any logical function from *n* inputs to *m* outputs can be implemented with an appropriately wired neural network. Thus, the article was titled "A Logical Calculus of the Ideas Immanent in Nervous Activity."

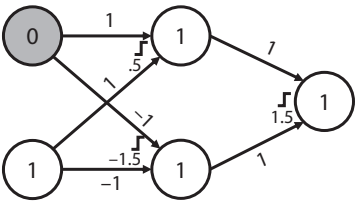

FIGURE 1.1. A network of binary neurons implementing the XOR operation. Binary inputs are multiplied by weights (on edges), and the output is 1 when the result is greater than a threshold.

Philosophically, the model of McCulloch and Pitts stands with classical computationalism (Dupuy, 2013): the state of each neuron represents a symbol with true or false value, and the model implements propositional calculus. Mental states are made of logical propositions. But in the 1950s and 1960s, Frank

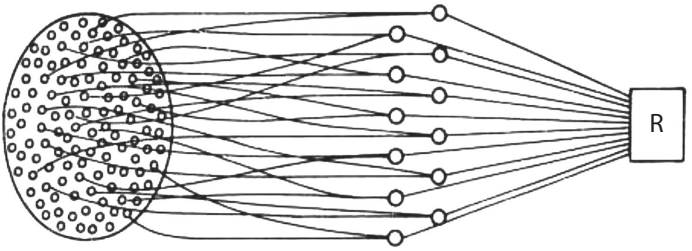

FIGURE 1.2. Rosenblatt's perceptron (from Rosenblatt, 1962).

Rosenblatt started to apply it to visual tasks, under the name "perceptron" (Rosenblatt, 1962; figure 1.2). There, the input variables represented light intensity at photoreceptors, the output represented the recognition of an object, and crucially, the synaptic weights were learned by association. The model did not implement logical inference anymore. Instead, Rosenblatt interpreted the model "in terms of probability theory rather than symbolic logic" and called his approach "connectionist" (Rosenblatt, 1958).

Despite initial interest in connectionism, it was abandoned a few years later in favor of symbolic approaches, when Minsky and Papert (1969) demonstrated the fundamental limitations of the perceptron. When expert systems were abandoned in the 1980s, there was a renewed interest in connectionism, triggered by the design of efficient learning algorithms for multilayer perceptrons, such as backpropagation (Rumelhart et al., 1986), still in use in modern artificial neural networks. Connectionism fell out of fashion again in the artificial intelligence community in the 1990s, in favor of more efficient statistical learning algorithms, such as support vector machines (Cortes and Vapnik, 1995). It was revived in the 2010s, when improvements in model design, software engineering techniques (such as automatic differentiation), as well as computing power and data availability led to impressive results in different areas, such as image processing (LeCun et al., 2015).

According to connectionism, cognition arises from the interaction of many neurons, seen as simple stereotypical input–output devices. Learning consists in modifications of the association strength between pairs of neurons, summarized by a single parameter. Thus, connectionism is explicitly associationist, and therefore conceptually closer to behaviorism than to computationalism. Cognition is not logical calculus anymore, but a form of calculation more akin to linear algebra. Furthermore, the activity of neurons in inner layers is not associated to mental symbols anymore, but rather to intermediate computational variables.

These differences remain a major source of mutual criticism between the two approaches. On one hand, (symbolic) computational models are

essentially incapable of dealing with real sensory inputs, such as images. On the other hand, connectionist models have great difficulties dealing with relational tasks, such as deciding whether an image contains two identical objects (Kim et al., 2018), or with compositional tasks (Dziri et al., 2023), or generally tasks that rely on abstraction (Lewis and Mitchell, 2024).

Despite these differences, computationalism and connectionism are conceptually related in many ways. Indeed, the model of McCulloch and Pitts was an explicit inspiration of John von Neumann's work on the electronic computer in 1945 (von Neumann, 1993), as well as of Rosenblatt's first connectionist model. Both computationalism and connectionism see cognition as a form of computation, consisting in applying a series of elementary operations to an input. This means in particular that cognition is an input–output process, which takes data and maps it to a response. As Hendriks-Jansen pointed out already in 1996, "most of the connectionist systems that have been built to date are models in which the inputs and outputs are assigned by the programmer following analysis of a particular task domain" (Hendriks-Jansen, 1996)—this is still the case at the time of writing.

This view preserves the behaviorist concept of the stimulus: behavior is made of responses to stimuli, except that there is now "cognition" between perception and action—the "classical sandwich" model of cognition, as Susan Hurley put it (Hurley, 2001). Indeed, in standard experimental sensory neuroscience, neural activity is almost invariably reported as a response to stimuli, and activity unrelated to the stimulus is called "noise"—as opposed to the autonomous activity of the organism. This is obviously the experimenter's perspective.

The way computations are performed differs greatly between classical computationalism and connectionism. Indeed, in a deep neural network model that identifies faces, neurons of the hidden layers do not represent anything in particular. This is why a common complaint about modern artificial networks (deep learning in particular) is that they are not *explainable*: we cannot easily explain what they do because the results of intermediate calculations are not meant to be interpretable as symbols. However, neurons of the output layer do *represent*: in a face recognition model, their activity represents the occurrence of a particular face. Therefore, the output remains symbolic, just like in classical computationalism. Furthermore, since these output symbols must be the inputs to some other computational networks—for example, those responsible for uttering the name of the face—connectionism still generally commits to a symbolic view of cognition, both for inputs and outputs. There are still "neural representations" or "neural codes" of mental content, in the form of the activity of specific neurons or groups of neurons, but not all neurons encode; only those at the input and output of designated cognitive

functions. Thus, classical connectionism has a somewhat confused view of the symbolic nature of cognition.

Connectionism also preserves the hardware/software distinction at the heart of computationalism. In this case, software is the set of synaptic weights. Neurons are input–output devices with a few knobs, but are otherwise rigidly specified. This is a key requirement of modern connectionist models, where the tuning of synaptic weights relies on formal differentiation of neural input–output functions, as we will see in chapter 4.

Thus, although connectionism describes cognition in terms of the operation of "neurons," the biological nature of neurons, or of the organism, plays exactly no role, just like in classical computationalism. The facts that the organism lives and that brains develop (as opposed to being assembled) are peculiarities of "implementation" with no theoretical significance.

Because biology is just implementation, both computationalism and connectionism start from the cognitive problem being solved and then try to figure out how the brain might solve it. This approach is typically called "top-down"—the top being the mind. This is of course in line with the engineering mindset: first, we describe what the machine should do; second, we design its functional organization; third, we implement the functional description by assembling components with the right specifications. This is essentially what David Marr, a pioneer of computational neuroscience, proposed as the methodology for modeling brains (Marr, 1982): start with the "computational level," the task that the model is supposed to achieve; then describe the "algorithmic/representational level," the algorithm that solves the task, at an abstract level; and finally worry about the "implementation level," how the algorithm is realized in the brain.

Of course, this methodology makes perfect sense for artificial intelligence, since in that case, we are indeed engineering the models. In neuroscience, an alternative kind of methodology, which brands itself as more empirical ("data-driven"), consists in measuring the different components of the brain as well as the way they are assembled. This kind of approach is often called "bottom-up." It is in fact also inspired from engineering, because parts of a living organism are conceived as parts of a machine.

Bottom-Up Neuroscience

An example of a bottom-up approach in neuroscience is the Human Brain Project, which aimed at simulating an entire brain based on systematic large-scale measurements of the properties of neurons and synapses. In this case, the neuron models are not classical connectionist models with abstract variables such as the "activity" of a neuron, but biophysical models taking the form of

dynamical systems with measurable variables, such as the membrane potential. Those models were obtained from electrophysiological measurements in animals. In the Human Brain Project, the measurements were statistical: models of typical neurons, and average connectivity between brain areas.

Other bottom-up projects rely on more systematic measurements. For example, a technical approach known as *connectomics* aims at systematically measuring the detailed synaptic connections between neurons in an entire region or in the whole brain. The graph of connections is called the *connectome*. According to its strongest supporters, connectomics should bring a decisive contribution to the understanding of brains and cognition. For example, Morgan and Lichtman (2013) assert that "it might not be so unrealistic to hope that in staring into such a map we might get a glimpse of the human mind," and Seung (2012) claims that you literally *are* your connectome. Of course, this is simply the expression of connectionism in its most radical form: cognition is essentially specified by the connections between neurons.

Thus, bottom-up approaches also often embrace some variation of connectionism, as well as the general framework of computationalism—in particular, its terminology. That is, brains are described as implementing computations, processing information, and so on. But in contrast with top-down approaches, models of the brain are established by measurement, independently of what the brain is supposed to achieve. Function is assumed to follow from those measured properties. The implicit assumption is that, like in a machine, the properties of parts are independent of the system in which it is embedded (the "top"), and of what that system does. First come the parts with their specified properties, and then they are assembled according to a plan.

But of course, this analogy with machines is fragile, because in a living organism, parts always grow within a functional system, and so the relation between "bottom" and "top" is circular, not unidirectional. As we will see in chapter 3, this explains why the hopes of bottom-up approaches have not been realized so far.

Brains Beyond Engineering

The Neurocomputational Patchwork

In practice, models of neuroscience (as opposed to artificial intelligence) do not strictly adhere to either computationalism or connectionism in their classical form but instead borrow concepts from both approaches. In the same way, modeling is rarely purely top-down or bottom-up. For example, bottom-up approaches generally use properties of the "top" as constraints, although this is rarely acknowledged (as we will see in chapter 3). Conversely, connectionist

approaches often take inspiration from structural peculiarities, such as the modular organization of the brain or the presence of dendrites. Some parts of brain and mind studies are dominated by connectionism, such as systems neuroscience, while others are dominated by classical computationalism, such as cognitive science. The most empirically driven models of neuroscience are in fact dynamical systems that are neither symbolic nor connectionist (such as the Hodgkin-Huxley model).

Thus, brain theory consists of a heterogeneous patchwork of approaches and models. Nonetheless, they share a common terminology borrowed from engineering, in particular computer science: brains and neurons compute, implement algorithms, encode objects and properties, represent and process information, and so on. For example, Sydney Brenner, who pioneered the neurogenetic study of *C. elegans*, a microscopic worm with 302 neurons, describes his approach as follows:

> Behaviour is the result of a complex set of computations performed by nervous systems and it seems necessary to decompose the problem into two: one is concerned with how the genes specify the structure of the nervous system, the other with questions of how nervous systems work to produce their outputs. (Brenner, 1973)

Unlike bees, *C. elegans* cannot count, and so far, no one has found hints of symbolic representations in its neurons. Thus, Brenner meant "computation" in a much broader sense than classical computationalists do. Apparently, it is not just that the animal *can* compute, but *all* behavior results from a kind of computation implemented by the nervous system. This is typical of modern neuroscience literature, a view that I shall refer to as *neurocomputationalism*: neurons are conceived as formally specified input–output devices that compute and implement the algorithms of cognition. But what is meant exactly by "compute," "implement," and "algorithms" is often rather vague, and indeed may differ substantially between approaches.

This terminology is not decorative: it is a theoretical commitment that forms the scaffold of reasoning about brains as well as of model building. Because the precise meaning of those words is often left unspecified, this scaffold is fragile and often incoherent. (Do neurons compute in the sense of connectionism, in the sense of computationalism, or do they just do something useful?) And because brains are not actually engineered, this scaffold is often poorly fitted to the subject, as we will see. In this book, we will explore the meaning of those words, and the extent to which they make sense when talking about brains, including computation (chapter 4), representations and codes (chapter 5), information (chapter 6), prediction (chapter 7), and implementation (chapter 8).

First, I want to make it clear why this choice of words is indeed a theoretical commitment about how brains work.

Of Words and Theories

Many words we use to talk about brains come from our ordinary human experience. For example, we say that neurons communicate, or send messages. These words originate from our social experience as a speaking species. We use them for neurons because we recognize some features of communication in the biological phenomenon: neurons of the retina produce electrical spikes (*action potentials*) that are specific to the image being presented, and this sequence of spikes then travels along the axon, unchanged, up to the axonal terminals, as if they were Morse code messages being delivered through the nerves, from one neuron to the next. On the other hand, we know very well that the receiving neuron does not literally "read the message," neither does it imagine the image that the message is supposed to stand for. There are features of messages that seem relevant to describe the electrical activity of neurons, and others that are not. When we say that spikes are messages, we focus on those features that we find relevant. This point about language was made eloquently by Lakoff and Johnson in their classic book *Metaphors We Live By* (Lakoff and Johnson, 1980): "What metaphor does is limit what we notice, highlight what we do see, and provide part of the inferential structure that we reason with." For this reason, choosing a particular word from another domain is a theoretical commitment. We will discuss communication metaphors in more detail in chapter 5, in the context of neural codes.

The most important engineering metaphor in biology is the machine metaphor. In the modern view, living beings are machines, and brains are computers, which are kinds of machines. We know this is a theoretical commitment because the idea that living beings are machines is supposed to be an *insight*. We know quite well what machines are in real life. If we were to point out a rabbit to a ten-year-old and say, "look at this machine," she would certainly object that it is not a machine but an animal. A machine is something made by humans to do something useful for them, it is not autonomous, it does not grow, it does not feed, and it does not feel. A ten-year-old, as well as most adults, would certainly put machines and animals in different categories. Thus, when the biologist Jacques Monod insists in *Chance and Necessity* (Monod, 1970) that *actually* a living organism is a molecular machine, he wants to convey something important and not obvious about life. It is not just a decorative term but a theoretical claim.

What was so important to Monod? Mainly, he wanted to oppose *vitalism*, according to which organisms live thanks to a nonphysical vital fluid. By

claiming that living organisms are machines, he meant that biological matter follows the same ordinary laws of physics and chemistry as inert matter, and, like machines, it is by virtue of its organization that the organism does what it is supposed to do (living and reproducing), not thanks to a special substance. A machine is made of components interacting together in certain ways so as to support the function of the machine. In the same way, a biological organism consists of a functional arrangement of organs—the digestive system, the circulatory system, and so on—in the service of the maintenance and reproduction of the organism. Thus, Monod's theoretical claim is that living organisms are goal-directed functional organizations of ordinary matter.

This is not a trivial claim at all. Surely, we can recognize parts such as organs in animals, but those are very unlike the parts of machines. Organs grow, for example. At the microscopic level, the molecular content of a cell changes in composition, number, and localization, at timescales of milliseconds to years. It is not so obvious how this molecular maelstrom can be conceptualized as components to which we can assign functions, like the functional diagram of a machine.

In fact, Monod also meant that living organisms are machines in the sense that their processes are essentially mechanical—that is, that their parts follow deterministic local interactions between discrete elements, mostly based on shape, like the solid macroscopic objects of our ordinary experience. Monod used the word "clockwork." This is the idea of the standard "key-and-lock" concept of molecular biology, according to which the shape of a protein determines its function. However, the claim that living processes are essentially mechanical in this narrow sense is demonstrably false, as Daniel Nicholson has clearly argued (Nicholson, 2019), and as we will see in the next chapter. A common example in the brain is the action potential, which is produced by spatially separated ionic channels that interact nonspecifically at a distance.

Thus, by claiming that living organisms are machines, Monod makes three assertions, corresponding to three features of machines. The first is that, like machines, living organisms are made of ordinary matter, following the same laws of physics and chemistry as inert matter. This is fairly consensual. The second is that living organisms are organized like machines, with parts arranged so as to ensure the function of the whole system. This is questionable or at least ambiguous (what are "parts"? what is "function"?). The third is that living processes are mostly mechanical, essentially deterministic local interactions between discrete objects (he had proteins and nucleic acids in mind). This is demonstrably false.

This illustrates several important points about words and theories. First, the choice of engineering words is a theoretical commitment. When we use the word *machine* to designate living beings, we refer to some features of

machines that we think are shared by living beings. This is a convenient way to make theoretical claims about how living beings work. These claims may or may not be correct or may need to be substantiated. Second, strict identification as in "organisms are machines" or "neurons compute" is a great source of confusion. In what sense are living organisms machines? Are they made of parts? Assembled? Are they mechanical? Are they lawful? Are they engineered? These are very different claims. If one needs to carefully explain in what exact sense organisms are machines, and if different people pick different features, then organisms are not actually machines. They are somewhat similar, and somewhat different. This acknowledgment is crucial for conceptual clarification.

Biological Cognition

Cognition is a property of (at least some) living organisms. Perception, cognition, agency, free will, and consciousness are all biological phenomena. Even though we might try to replicate those phenomena in artifacts, the primary empirical source remains biology. Yet, strikingly, the study of cognition appears to be a branch of computer science rather than of biology. This dismissal of biology is even explicitly embraced by classical cognitive scientists, a view known as *functionalism*—biology is just "implementation." In neuroscience, the standard terminology of brain theory largely refers to a nonbiological world, the world of machines made by humans—computation, implementation, algorithms, codes, optimization. . . . Ironically, scientists have abandoned the idea that living organisms have been designed by God, only to adopt a model of the living based on artifacts made by an engineer. Thus, Monod ridicules vitalism as some sort of magical belief, but then identifies living organisms with machines, those artifacts made by humans for a purpose using knowledge and planning. Is this a scientific view on life, or monotheism rejecting paganism?

The idea that animals result from intelligent design is scandalous to a scientist. Yet, it appears to make very little difference to the way we think about brain and mind. On the contrary, I assert that a proper understanding of life, beyond engineering preconceptions, is crucial to an understanding of its cognitive properties.

Why are living organisms compared to machines in the first place, rather than to any complex physical system like the climate? The reason is that machines are goal-directed, like living organisms. But the goals of machines are just the goals of their engineers, and therefore the machine view does not actually address the issue of goals, which means that the choice of the machine metaphor has no ground. As we will see in the next chapter, the reason why

living organisms have goals is not because they are machines, but because they are precarious entities that must exchange matter and energy with their environment in order to maintain themselves. Cognitive properties are rooted in these facts of life, not in their presumed mechanistic nature.

Living organisms must feed. They have no material persistence. They develop by division. They evolve with no plan or direction. They are autonomous. This book explores the consequences of these facts of life for the understanding of brains, cognition, and behavior. I will start by presenting a modern view of life in the next chapter. In the rest of the book, we will use these lessons of life to revisit the standard concepts of brain theory. In chapter 3, I will question the reductionist preconceptions of "bottom-up" (*reverse-engineering*) approaches. In chapter 4, I will argue that brains are not *biological computers* in any useful sense. In chapter 5, I will explain that *neural codes* (or *neural representations*) are a misleading engineering concept, which does not stand empirical scrutiny, and which is theoretically incoherent when applied to brains. In chapter 6, I will show that the neuroscientific concept of *information* is problematic in a biological setting, because it is framed as what the engineer can recover from a signal, and the engineer always uses preexisting knowledge in addition to the signal. In chapter 7, I will argue that anticipation is the core property that theories of cognition try to explain, but that its common identification with prediction is mistaken. Instead, I will develop an account of anticipation as the exploitation of regularities, rooted in the precarious nature of life. In chapter 8, I will show that the concept of *implementation* introduces a biased view of the organization of brain processes, mirroring the way *we* make devices rather than accounting for the autonomy of life. I will end the book on an alternative view of organisms and brains as colonies of living entities, and outline what it implies for the development of brain theory.

INDEX