1
Introduction

1.1 For whom is this book?

This book is for undergraduates and beginning graduate students. Of course, other graduate students, postdoctoral students, and anyone concerned with the fate of Western Civilization should read this book as well. Why focus on undergraduates? Because the malleable mind of the undifferentiated stem-person of college age is likely to be capable of making conceptual leaps that more ossified brains cannot. A theme that will recur throughout the book is the difficulty of unifying the compute and the neuro.

There are two major barriers to grand unification. First, computer science (like math, physics, and engineering) is made up of grand unifiers and their all-encompassing schemes, while neuroscience (like the rest of biology) is boatloads of facts. The differentiated mind of the engineer cannot swallow so many facts without a unifying framework. The differentiated mind of the biologist knows too much and is distracted by the many facts that contradict any preliminary framework someone tries to build. The fabled undergraduate mind, however, is notoriously unburdened by facts and yet willing to accommodate them. It (that mind, whether he or she) also typically seeks big pictures and is willing to take the leaps of faith required to acquire them.

The second barrier: these are early days. The field is a newly emerging hybrid and is itself still undifferentiated (an undifferentiated field needs an undifferentiated mind). It is a field still driven more by passion and fashion than by cool reason, emerging like a star from a gaseous cloud (OK that's a bit fanciful). Anyway, some of my colleagues will read this book and say that I've missed the whole point of the field. I've said the same of their books. Specifically, many researchers in the field come from the aforementioned engineering, physics, and math tradition. Their efforts are directed at developing a theoretical framework and their work tends to be colored by the framework they have chosen. I, on the other hand, come from the bio side. As a consequence, this book is a largely atheoretical approach to a theoretical science. I present stuff that is either fun or interesting or important and maybe sometimes all three.

In this computational neuroscience funhouse, I have included biological facts, computer science facts, equations, and theories both true and false. In some cases, I have included false theories because I don't know that they're false, though I may suspect it. In other cases, I introduce unlikely hypotheses just to roll them around and play with them. This interplay of facts and ideas makes up much of the work (or play) of modeling. Through experiencing it, the student will get a better idea of how and why modeling is done. In the last chapter, in particular, I bring the reader to my own particular circle of purgatory — the hall of perpetual mystification. I've tried to illustrate the complexities and contradictions of the field without unduly confusing the reader.

1.2 What is in the book?

Although my target undergraduate students are pluripotential, they are not yet omniscient. Specifically, they are majoring in philosophy, physics, math, engineering, biology, zoology, psychology, physical education, or business administration. As a result they know a lot about some things and little or nothing at all about others. For this reason, I have tried to cover all the basic bases. I have relegated much of this to a final chapter (Chap. **??**), which can be read piecemeal as needed. In addition, many of the fundamentals have seeped into the main text as well. I have included a lot of basic computer science since an understanding of computational neuroscience requires a fairly sophisticated working knowledge of computers. This broad, blanket coverage means that certain chapters may seem trivial and others overly demanding for the particular student.

For the nonmathematical reader, the biggest challenge will be the advanced math topics, notably matrix algebra and numerical calculus. Though these topics are hard and can easily fill up a year of classroom instruction, I have tried to extract just the parts needed for present purposes and to make this accessible to anyone with mastery of high-school algebra. I have written out most equations in words so as to make them more accessible to anyone who is allergic to math. Additionally, the computer is a great leveler in this regard. Many once abstruse concepts in mathematics can now be quickly illustrated graphically. Computer in hand, this book can be enjoyed as a lighthearted romp through calculus, electrical engineering, matrix algebra, and other sometimes-intimidating topics.

For the nonbiological reader, the biggest challenge will be the profusion of facts and jargon words. This onslaught of information can be intimidating and discouraging. There is so much to know that it can be hard to know where to start. Often it is impossible to connect one set of facts to another set of facts. That is the goal of computational neuroscience. When you first encounter these facts, it will be without the benefit of such a model.

Although I have tried to write clearly and comprehensibly, I have also tried to use a lot of jargon. This can be annoying. I try to use jargon kindly and responsibly, to help the reader learn the words needed to read and converse knowledgeably in the many subfields that make up computational neuroscience. I have tried to always define jargon words immediately upon use in the text. As further assistance, I've provided a glossary. In addition to introducing jargon words, I also introduce some jargon concepts — touchstone ideas that are frequently referenced by people in a particular field. Having so much to introduce, concepts and phrases are sometimes mentioned, but not followed up on. They are presented to provide the reader with vocabulary and mental reference points for further reading or just plain thinking.

1.3 Do I need a computer for this book?

This book is meant to be read independently of any computer work. I have not put explicit exercises in the book but have made them available online (see below). One of the neat things about computational neuroscience is that it is so readily accessible. It is hard to get hold of the particle accelerators, centrifuges, and chimpanzees needed for most scientific study. But computers are everywhere, making computer-based research accessible to undergraduates and even to nonacademic folks. This is more true of computational neuroscience than it is of other computer modeling fields. To do weather prediction you need a supercomputer. A simple desktop PC will do for most of the material in this book. If the first stage of learning a field is to talk the talk by learning vocabulary, running the computer exercises will enable you to walk the walk as well.

1.3.1 Software

All of the figures in this book were put together using Neuron, a computer simulation program written by Mike Hines at Yale University. This program is freely available at *http://www.neuron.yale.edu*. Although Neuron is primarily designed for running the type of realistic simulations highlighted in the latter part of the book, it is flexible enough that I was able to use it for all the other simulations as well.

Software to produce all of the simulations and to run the emulator of Chap. **??** is available at these sites:

http://www.springer-ny.com/computer2brain
http://www.cnl.salk.edu/fctb
http://www.neuron.yale.edu/fctb

I will be pleased to consider additions or augmentations to this software, particularly if the contributor has already coded them.

Examples in the software are primarily presented through a graphical user interface. The reader or teacher who is interested in pursuing or presenting the subject in depth will want to become familiar with the Neuron program and with HOC, the Neuron programming language. This will allow the programs to be manipulated more flexibly in order to look at different aspects of a particular modeling problem.

Many of the examples presented in this book could also be readily programmed up in Matlab or Mathematica, or in other simulation programs such as Genesis or PDP++.

1.4 Why learn this now?

The genetic code was cracked in the mid-20th century. The neural code will be cracked in the mid-21st. Genetic science has given way to its applications in biotechnology and bioengineering. Neuroscience is still up-and-coming, the next big thing. Furthermore, as genetic manipulations and basic neuroscience add more raw facts to the broth, the need for meaning, structure, and theory becomes greater and greater. Enough information is coming together that the next generation of computational neuroscientists will make the leap into understanding. That means grant money, prizes, and fancy dinners! (If the movies are a guide, it also means evil robots, mind control, and dystopia, but let's not ruin the moment.)

There's such a variety of things to learn about in computational neuroscience that the student is in the position of the proverbial kid in the confectionery: so many problems to work on; so many amazing facts and theories from so many interrelated fields. Of course, this profusion of riches can also be frustrating. One doesn't know which gaudy bauble to pick up first, and, having picked one and discovered that it is not quite gold, strong is the temptation to drop it and pick up one that seems gaudier still.

1.5 What is the subtext?

In an age of ubiquitous computers, any topic can be discussed in their context, as attested to by the recent publication of *From Computer to Stain: Dry Cleaning in a Digital Age,* the inspiration for the title of this book. However, computational dry-cleaning is still just dry-cleaning done with computers. In fact, most of computational biology is just biology done with computers. Computational neuroscience is a little different. The computer itself represents the state of our knowledge about how complex information processing devices like the brain might work. For this reason, I have covered more computer science than would usually show up in a neuroscience book.

Present computer science curricula generally emphasize sophisticated abstractions that pull one away from the machine. Similarly, a branch of computational neuroscience has concerned itself with finding general principles of neural computation and has shied away from the messy meat of the brain. My contention is that the meat is the message for the brain and for the computer. Lovely abstract theories must grow out of an understanding of the machine. The most useful theories will be different for different machines. If there is a grand unified theory, it will stand abstract and austere away from the daily marketplace of synapses or transistors.

For this reason I have gone into some detail about the design and operation of an ancient computer, the PDP-8, a machine with the power of a modern pocket calculator. Such a simple machine can be readily described in a chapter. It also is small enough that one quickly runs into its limits and has to overcome them with programming tricks, commonly called "hacks." Hacking is now frowned upon in computer science, since it is mostly used to break into other people's machines. Biological evolution is one long history of hacking — using a piece of machinery for a new purpose and gradually working it into shape so that it seems to have been engineered from scratch for that purpose.

Understanding the process and products of evolution means understanding the problems of engineering with limited resources (and unlimited time). Programming a PDP-8 or rebuilding a diesel engine with pieces of scrap in a Third-World country requires ingenuity, ability to compromise, and willingness to make mistakes and start again. This process may leave us with a program or an engine with vestigial organs, tangled distribution routes, and inefficient procedures. Just as building the machine was a study in frustration, so examination of the machine will be a frustrating study that will also lead to dead-ends and back-tracking.

In this book, I repeatedly contrast my emphasis on the brain with the tendency of others in the field to focus on theory rather than detail. Perhaps I occasionally disparage these poor theoretical guys as cyborgs and hedgehogs. This is all in fun. Integration of theory and fact is a necessary goal in computational neuroscience. I try to give both their due in this book but I have not been successful in integrating them. The section titles — Computers, Cybernetics, and Brains — demonstrate this.

Our brains are full of contradictions but we learn to live with them. If we want to study the brain, we must be prepared for the kinds of ambiguities and occasional false leads that characterize life with our own brains.

1.6 How is the book organized?

Computational neuroscience is a new field whose essential paradigms are still the subject of debate. For this reason, it is not possible to present the basic material with the conceptual coherence of an introduction to well-established fields like chemistry or physics. The field remains a hodge-podge of exciting ideas and remarkable facts, some of which cannot be neatly conjoined. Instead of progress in a neat sequence from one idea to another, this book will at times seem to jump back and forth from one thing to another. This is an inherent difficulty of trying to teach both the computational and biological approaches in a single text. In general, I try to fill in the gaps where they can be filled in and point them out where they remain unbridged.

The organization of the book is as follows. I start with a brief introduction to neuroscience, touching on many but not all of the subfields that may need to be considered. I then go top-down with a description of how computers work. I start by noting how computers represent information. I then go into still more detail about the bits-and-bytes level of computer function. From there, I switch from transistors to neural network units and explore the concepts of artificial neural networks. This will entail a comparison between transistors and neurons and an explanation of how the artificial neural network units represent a compromise position between the two. From there, I show how these units can be connected together into artificial neural networks. I further expand on the artificial neural network paradigm, showing the use of these networks to explain the retina of a simple sea creature, the horseshoe crab. I look at another simple brain system, but this time one found in humans, the brainstem reflex that stabilizes the eyes in the head when the head is moved. Then, in a more speculative vein, I go still higher in the brain, looking at how artificial neural networks can be used to emulate aspects of human memory. This will involve an explicit compare-and-contrast with computer memory design.

Following this, I turn bottom-up, more seriously exploring the biological concepts of nervous system function. I start with a detailed description of the neuron with some ideas of how the different parts of the neuron can be modeled. I then explore in greater detail the two major techniques of realistic neuronal modeling: compartment modeling and the Hodgkin-Huxley equations. I then look at an example of how artificial neural network models of learning can inform our understanding of the brain and how study of the brain leads us to reconsider the details of these artificial neural networks.

The final chapter covers some details of the mathematical and scientific approaches and techniques used in this book. It includes unit analysis, binary arithmetic, linear algebra, calculus, and electronics. Comfort with handling units and scientific notation is needed for finding your way around science. Knowledge of binary is important for finding your way around a computer. Linear algebra is useful for finding your way around a network. Calculus is important for assessing movement and change. Electronics is needed for understanding electrical signaling in neurons. In each case, the material has been presented graphically and algebraically to make the subject accessible to those who do not feel comfortable with mathematical notation. It is expected that many readers will be unfamiliar with some or all of these areas and will want to read these sections as the technique comes up in the main text.

Since there are many topics touched on that do not always relate cleanly to one another, I tried to provide additional guidance. Each chapter begins with a brief introduction entitled “Why learn this?” Similarly, each chapter ends with “Summary and thoughts,” meant to synthesize concepts and remind the reader of what was learned. This section is “... and thoughts,” rather than “... and conclusions” because in many cases the conclusions await.

Part I

Perspectives

2

Computational Neuroscience and You

CNE

A major goal of computational neuroscience is to provide theories as to how the brain works. Such mind–body theorizing has been a subject of philosophical, theological, and scientific debate for centuries. The new theories and taxonomies for organizing information about the brain will be built upon this historical foundation. It is valuable to see where we are starting.

2.1 Brain metaphors

Mechanical models or metaphors for the brain date back to the time when the brain first beat out the heart as leading candidate for siting the soul. Plato likened memory to the technique of imprinting a solid image onto a block of wax. Over the centuries, the nervous system has been compared to a hydraulic system, with pressurized signals coursing in and out; a post office, with information packets being exchanged; or a telephone switchboard, with multiple connecting wires to be variously assorted. Today, the digital computer, or sometimes the Internet, is cited as a model for brain function. Do these modern mechanisms hold greater promise than prior metaphors for helping us understand our most intimate organ?

In many ways, the brain is not much like the standard digital computer. Yet, both as a direct model of certain aspects of brain functioning and as a tool for exploring brain function, the computer enjoys many advantages over previous models. Take, for example, the post office. The difficulties of actually utilizing the postal service to test out the feasibility of a particular brain model must give one pause. (However, below I discuss a similar human-based system that was proposed more than a century ago as a calculating technique for weather prediction.) The telephone switchboard, on the other hand, is a considerably more manipulable organizational and technological artifact. In fact, the early analog computers of the 1930s and 1940s were, in appearance and in some functional aspects, aggrandized telephone switchboards. Although simple neural models were run on such machines, the technical difficulties of programming them made them far less useful than digital computers as a tool. However, the basic concepts of analog computing may be useful for understanding brain function.

2.2 Compare and contrast computer and brain

When we liken the brain to a computer, we mean several things. First, we mean that several definable computer actions are analogues of things that the brain appears to do. Such computer actions include memory, input/output, and representation. Second, we mean that computers have been used to do a variety of tasks that were previously believed to be exclusively the province of human intelligence: playing chess, reading books aloud, recognizing simple objects, performing logical and mathematical symbol manipulations. Finally, although no machine has yet passed the Turing test (a machine passes if it fools a conversation partner into thinking that it is a person), those who work intensively with computers develop a distinct sense of communicating or even communing with the machine.

Modeling is the work and play of computational neuroscience, as it is for much of physics, engineering, business, and applied mathematics. It's a tricky thing. To learn something about the thing being modeled, we need to reduce the model to the essentials. If we reduce too far, however, we may miss a critical component that is responsible for interesting properties. For the Wright brothers and other early aviators, the process of building a heavier-than-air flying machine was a task of bird emulation. To those who said that heavier-than-air flight was impossible, they could point to birds as a counterexample. As they evaluated the basic bird, it would have seemed clear that many aspects of bird design were not required for flight. For example, the beak seems quite clearly designed more for eating than for flying. However, the beak's aerodynamic design might still tell us something valuable about fuselage design. The importance of other bird features for flight would not have been as apparent. For example, it might a priori seem likely that wing beating was critical for flight. It is critical for small-creature flight but not for the flight of large birds or airplanes. Many early, misguided attempts were made to design a full-sized ornithopter (*e.g.,* that aircraft with flapping wings that beats itself to death). On the other hand, the Wright brothers had a key insight when they noticed that birds steered by tilting their body to the side (rolling) rather than by using a rudder like a boat.

When we model birds we know what we want to model. The function of interest is flying. We can focus on flying and ignore feeding, foraging, fleeing, etc. The brain, however, is doing many things simultaneously and using hidden processes to do them. Therefore, we can model a brain function, such as chess playing, and yet gain little or no insight into how the brain plays chess. The brain is utilizing unconscious properties that we are not aware of when we play chess. In this example, I would guess that an important underlying ability used in chess is the capacity of the brain to complete partial patterns. This ability is seen in the normal unawareness of the blind spot. It is also seen in the abnormal confabulatory tendency of demented or psychotic individuals to forge links between false perceptions so as to build an internally consistent, although irrational, story.

In this book, as we look in detail at how a computer works, and compare and contrast its functioning with that of the brain, a variety of differences will become apparent. We consider various brain features and wonder whether or not these are critical features for information process, for memory, or for thought. Certainly, many aspects of brain design are not critical for brain information processing but are there for other purposes: metabolism, growth and differentiation, cell repair, and general maintenance.

If we wanted to use a modern jet aircraft as a model to help us better understand birds and the phenomenon of flight, we would want to take note of similarities and differences that might clarify essential concepts. Both have wings; it seems reasonable to expect that wings are essential for heavier-than-air flight (note that helicopters are considered rotating wing aircraft). However, the wings are made of very different materials so there is apparently nothing critical in the design of feathers. Closer analysis would reveal that airplanes and large birds like albatrosses have similarly shaped wings (smaller birds and insects use different-style wings suited to their small size).

Using the computer as a model to understand the brain raises questions about similarities both in detail and in function. Airplanes fly like albatrosses but computers don't think like brains. Both brains and computers process information, but information processing may not be central to the process of thinking. Therefore, we will wish to explore not only differences from the bottom, differences in materials and design principles, but also differences from the top, differences in capacity and capability.

Starting with the manufacturing side, there are already a variety of differences that can be explored. Computers are made of sand and metal, while brains are made of water, salt, protein, and fat. The computer chip is built onto a two-dimensional matrix, while the brain fills three dimensions with its wiring. The time and size scales involved are also believed to be vastly different. Of course, this depends on exactly what is being compared to what. As we will see, typically a transistor in the computer is compared to a neuron in the brain. With this comparison, the time scales are about 1 ms for the neuron vs. 1 ns for the transistor (see BASUNI for discussion of units). The spatial scale is about 1 mm for the largest neuron vs. less than 1 μm for a modern CMOS transistor. Thus the neuron is much bigger and much slower. However, if it eventually turns out that the proper analogue for the transistor is the synapse, or a particular type of ion channel or a microtubule, then we would have to reevaluate this comparison.

Additional differences arise when one considers functional issues. Brains take hints; computers are remarkably stupid if given a slightly misspelled command or incomplete information. The digital computer has a general-purpose architecture that is designed to run many different programs. The brain, on the other hand, has dedicated, special-purpose circuits that provide great efficiency at solving particular problems quickly. Calculations on a digital computer are done serially, calculating step by step in a cookbook fashion from the beginning to the end of the calculation. The brain, on the other hand, performs many calculations simultaneously, using parallel processing. Digital computers use binary; transistors can take on only two values: 0 or 1. In this book, we utilize binary extensively, and consider its applicability to the brain. This may not be a fair approximation since the brain uses a variety of elements that take on a continuum of analog values.

2.3 Origins of computer science and neuroscience

Neuroscience and computer science came into being at about the same time and influenced each other heavily in their formative stages. Over time, the fields have diverged widely and have developed very different notions of seemingly shared concepts such as memory, cognition, and intelligence.

D.O. Hebb proposed over 40 years ago that a particular type of use-dependent modification of the connection strength of synapses might underlie learning in the nervous system. The Hebb rule predicts that synaptic strength increases when both the presynaptic and postsynaptic neurons are active simultaneously. Recent explorations of the physiological properties of neuronal connections have revealed the existence of long-term potentiation, a sustained state of increased synaptic efficacy consequent to intense synaptic activity. The conditions that Hebb predicted would lead to changes in synaptic strength have now been found to cause long-term potentiation in some neurons of the hippocampus and other brain areas. As we will see, similar conditions for changing synaptic strength are used in many neural models of learning and memory. These models indicate the great computational potential of this type of learning rule.

One difference between the neuroscience and computer science viewpoints has to do with the necessary adoption of a big-picture approach by computer scientists and a reductionist approach by many neuroscientists. These two approaches are typically called top-down and bottom-up, respectively. The top-down approach arises from an engineering perspective: design a machine to perform a particular task. If you're interested in intelligence, then design an artificial intelligence machine. The bottom-up perspective is the province of the phenomenologist or the taxonomist: collect data and organize it. Even granting that most U.S. science today is federally mandated to be hypothesis-driven, an essential element of biology is the discovery of facts. Hypotheses are then designed to fit these facts together. As outlined here, these positions are caricatures. Most biologists want to consider how the brain thinks, and many computer and cognitive scientists are interested in what goes on inside the skull.

In this book, we concern ourselves with many ideas that have been promulgated for understanding higher levels of nervous system function such as memory. However, it is important to note that much of the data on real nervous systems has been gathered from either the peripheral nervous systems of higher animals or from the nervous systems of invertebrates such as worms, leeches, and horseshoe crabs. The genesis of the action potential or neuron spike, one of the most important ideas to come out of computational study of the brain (Chap. **??**), was the result of studying the peripheral nervous system of the squid. The low-level source of much of our knowledge of the nervous system contrasts sharply with the ambition to understand the highest levels of mental functioning, and helps explain why some of the topics to be discussed may seem quite remote from human neural function, while other subjects will be very relevant but highly speculative.

2.4 Levels

The notions of bottom-up and top-down approaches to the problem of nervous system function, and the corresponding contrast between acknowledged facts at the lower level and uncertain hypotheses at the higher, lead naturally to hierarchical divisions. Two such divisions that are commonly used are called the levels of organization and levels of investigation. Each of these divisions into levels creates a hierarchy for brain research that leads between the reductionist bottom and the speculative top.

The levels-of-investigation analysis was historically a product of top-down thinking. This approach, pioneered by computationalists, starts at the top with the big-picture problem of brain function and drips down to the implementation in neurons or silicon. The levels-of-organization analysis was in part a reaction to this. By putting all of its levels on an equal footing, the levels-of-organization approach invited the investigator to start anywhere and either build up or hypothesize down.

2.4.1 Levels of organization

Levels of organization is fundamentally a bottom-up perspective. The basic observation that leads to this division of the knowledge comes from the "grand synthesis" that connected the physical with the vital world. Modern biology explains genetics and physiology in terms of the interactions of molecules. This allows connections to be made all the way from physics to physiology. Physics is the more fundamental science. The basic concepts of biology can be understood from the concepts of physics, while the converse is not the case. However, understanding biology directly from physics would be a hopeless task for two reasons. First, there is no way one can predict what would occur in a biological system using knowledge of atoms and electron orbits. Second, the conceptual leap from physics to biology is simply too great to be made without interposed models from other fields. Specifically, much of biology can be understood from cell biology, which can be understood from molecular biology, which can be understood from biochemistry, which can be understood from organic chemistry, which can be understood from physical chemistry, which can be understood from physics. In comparison with this known hierarchy of knowledge, the levels of organization of the nervous system remain tentative. Any hierarchy will likely embody a fundamental, trivial law: big things are built out of smaller things.

Following the scheme of others, we can build a hierarchy of levels of organization and levels of study. From smallest to largest:

study method	object of study
physics	ions
chemistry	transmitters and receptors
cell biology	neurons
computer science	networks
neurology	systems
psychology	behavior or thought

Although the general order of dependencies in the nervous system can be assumed to be based on size and the simple inclusion of one structure within another, the exact structures that are of functional importance are not clear. The ambiguity starts when one considers the appropriate items for anchoring the two ends. At the top, one can choose to regard either behavior or internal mental representation as the highest level suitable for scientific investigation. There is a long history of debate in the psychology literature between proponents of these two positions. Behavioralists believe that since physical movement is the only measurable evidence of nervous system function, this is the only appropriate area of high-level functional study. Other psychologists believe that putative internal representations of the external world are also suitable subjects of investigation, even though these cannot be measured directly. Computational approaches generally make the latter assumption, not only postulating internal representations but often making them the central question for further study.

At the small end of the organizational scale, most investigators would consider the concentrations of ions and neurotransmitters and their channels and receptors to be the smallest pieces of nervous system that are worth paying any attention to. A dissenter from this view is the physicist Roger Penrose, who believes that the underlying basis of neural function will lie in quantum mechanics and that it's necessary to study the subatomic realm.

In between quantum mechanics and behavior, there is still more room for debate both as to which levels are relevant, and as to which levels can be adequately built on a previous level without further investigation at an intermediate level. To go back to the physics-to-biology spectrum described above, the conceptual jump from the concepts of physics to the concepts of organic chemistry would not be possible without the intermediate concepts developed by physical chemistry. This is because the representations of electron orbitals and chemical bonds used in physical chemistry provide conceptual links between the detailed equations describing electron orbitals used in physics, and the schematic stick diagrams used for bonds in organic chemistry. Similarly, the neuroscience levels of organization suggests that neurons can be adequately described by taking account of properties at the level of transmitters and receptors. That's probably not going to turn out to be true. It's likely that intermediate-sized ultrastructural components of the neuron such as spines, dendrites, and synapses may have their own critical properties that cannot be understood without independent study of these structures in themselves.

As we move up the scale, higher levels of neural organization are less well understood and can be farmed out somewhat arbitrarily to various interested specialty areas. Much study of networks has come out of computer science, but the organization of networks is also studied in mathematics by geometry and topology. The level of cortical columns is not shown in this diagram. It is unclear whether this level would go below or above the level of the network. I gave *systems* to neurology, a clinical field that subdivides brain function into motor, sensory, and various cognitive systems based on changes seen with brain damage. Engineers mean something different when they study *systems* in the field called "signals and systems." *Systems* neuroscience has yet another connotation, referring to neurophysiological techniques related to investigating the origins of perception and behavior.

2.4.2 Levels of investigation

The levels-of-investigation approach comes from David Marr, a computationalist who produced some very influential early models of different brain areas. This viewpoint is from the top down. The top level is the level of problem definition (this was called the computational-theoretic level by Marr). Marr suggested that understanding any particular brain function requires that we first understand what problem the brain is solving. Problem in hand, we can deduce what additional information the brain would need to solve it. The next level is that of algorithm definition. An algorithm is like a cookbook recipe, defining a step-by-step approach to accomplish some task. The third and final level is the level of implementation, where the algorithm is finally translated into machinery, whether neural or silicon, that can actually perform the task.

Marr's three levels of problem, algorithm, and implementation are the current approach a software engineer would take in designing a big program (*e.g.,* a word processor or a Net browser) using a modern computer language. If writing a Web browser like Netscape or Explorer, for example, we would first define the problem — delivering information from remote sites to a user in a user-palatable form. We would then write algorithms that would simply assume that we have or can develop the underlying tools needed for the subsidiary processes. For example, a basic algorithm for processing a Web page would be 1) request the page, 2) wait and accept the data, 3) confirm that a full data set was received, 4) parse the data to determine content type, and 5) parse fully to present in a graphical form on the screen. Individual steps would then be implemented. It is important to avoid considering details of implementation in working out the algorithm since we are interested in readily porting our browser between machines that use different low-level implementations.

This Marr trinity of problem, algorithm, and implementation can be collapsed into the familiar concepts of software and hardware. A problem is provided. Algorithms are written into the software. The software is compiled so as to run on a computer — the physical implementation level. A software engineer using modern computing machinery doesn't routinely run into the limits of what the machine can do. The Marr top-down approach is ideal in this engineering environment. However, as will be discussed in the next chapter, when the limitation of the machine becomes part of the problem, another engineering approach is needed.

2.5 New engineering vs. old engineering

Over time, science and technology have advanced from being based on everyday commonplace observations to being based on sophisticated theories. Similarly, engineering has moved away from the tinkerer or hacker mentality toward reasoned conceptual approaches to technical problems. Working from theory, rather than empirically, the modern engineering approach is close to David Marr's notions of levels of investigation: from problem to method to implementation.

Modern building design is predicated on principles of tension and stress. By contrast, the great cathedrals of Europe were largely built using rules of thumb and intuition born of experience. Sometimes they fell down. Similarly, computer science has given up ad hoc hacking and developed tools and theories to allow software design problems to be addressed from basic principles.

From one perspective, Marr's insistence on first defining the problem is unavoidable. Until we know that the brain can do a certain thing, we cannot study it. A blind man who has never had sight, and had not spoken with someone who has, would have an impossible task trying to study vision based simply on being told that it represented an alternative to hearing. On the other hand, insistence on an initial problem definition can lead to what has been called premature definition. Fondly held hypotheses can be blinders that preclude appreciation of new facts that could shed light on the problem. This risk is particularly great in the general area of brain/mind studies, where the appeal to intuition is hard to resist.

The complexity of the unconscious workings of the many subdivisions of brain function makes them resistant to an introspective, intuitive understanding of what is going on behind the scene. This can make it impossible to frame the problem correctly. For example, Marr believed that vision primarily performed the task of taking the two-dimensional retinal representation of the world and re-creating a three-dimensional internal model of the world, just as one might look at a family photograph and determine that your cousin was standing to the side and in front of your aunt. This definition of the *problem* of vision seems intuitively reasonable to a sighted person. This clearly describes something that the brain can do and that needs to be done under some circumstances. However, this turns out to be a bad start for studying vision in the brain. As it turns out, any single *problem definition* for vision will turn out to be a bad choice, since the brain does not simply have one type of vision but instead utilizes many different types of vision that are processed simultaneously.

Marr's is a sophisticated engineering approach to vision. Given the complexity of brain and our limited intuition, a naive engineering approach is more reasonable. A congenitally blind man starts with no clue as to where to begin studying this mysterious phenomenon called "vision." He might therefore start asking questions about different things that vision can do. "Can it detect objects behind other objects? Can it detect motion? Can it determine shapes?" This set of questions would put the blind-man-explaining-vision in the proverbial blind-men-with-elephant situation (each feels a different body part and each has a different idea of what an elephant is). The blind man might conclude that vision was not one thing but was made up of separate detecting systems that handled various detection tasks. This sort of piecemeal understanding would bring him closer to the underlying mechanism of visual perception in the brain than was Marr with his sighted person's intuition of a unitary process. Although the blind man would have no appreciation of the personal experience (the qualia) of seeing, he would have some notion of how the brain actually performs the task. Lacking access to the myriad unconscious processes of our brain, we are nonetheless blessed with the illusion of introspection. Sight makes us blind to vision.

Modern, sophisticated engineering takes place in big laboratories. Discovery proceeds in reasoned steps according to quarterly plans. The old engineering was, to paraphrase Edison, all inspiration and perspiration. This is the engineering of tinkerers and hackers, who take discarded bits and pieces of machinery and cobble them together so as to make them do things they were not originally meant to do. When it doesn't work, the tinkerer bends and hammers and makes it work. The workman's ideal is always to have the right tool for the job. The tinkerer is cursed with the wrong tools, the wrong materials, the wrong job.

This discrepancy between available material and the exigencies of the task is also the plight of the evolutionary process. For example, a gill is not well suited to life on land. Since water must be moved across the gill in order to replenish oxygen, the gill needs to be exposed externally. However, gills must also be kept wet at all times. Keeping wet becomes a problem when we move the gill onto dry land. Although the modern lung looks great, early versions would have been crude hacks that managed to barely satisfy these contradictory needs: invaginating the gill to prevent drying, while exposing it enough to prevent asphyxiation.

2.6 The neural code

The brain denies the philosopher's, the mathematical modeler's, and the guy-on-the-street's desire for clarity and simplicity. Since there is no single overarching task for the brain to do, different facets of brain function must be studied separately. This does not necessarily mean that there are no unifying principles. There may well be basic neural codes that are used throughout the animal kingdom. However, the discovery of neural codes will no more free us from the need for further research into different brain areas than the discovery of the genetic code revealed all the functions of all enzymes and structural proteins.

The analogy between the search for the genetic code and the search for a neural code has been highlighted by Francis Crick, discoverer of the former and pursuer of the latter. To doubters, he points out that the quest for a simple genetic code seemed quixotic to anyone who considered that the complexity of the natural design encompasses enzymes and organs, growth and development. Of course, the discovery of a simple genetic code did not in any way provide an understanding of all the things that are coded for. It did, however, provide a powerful new tool for exploring these things. Similarly, the discovery of neural code or codes will not tell us how any part of the brain works, but will enable us to start to understand what we see when we amplify electrical signals from different parts of the brain.

Several neural signals are well established. However, some of these signals probably carry no information at all, while other signals carry information that is not used by the brain or body. For example, the electroencephalogram (EEG) is a very well studied signal that is emitted by the brain. There is information in the EEG that permits an outside observer to determine whether a brain is awake or asleep or even, after some signal processing, whether the brain is hearing clicks or seeing a flashing checkerboard. These field signals are generally an epiphenomenon, a side effect that has no functional relevance. These signals are not used within the brain under normal circumstances, and are too weak to be used for telepathy, no matter how close you put the two heads together. There are some cases where the field is used or misused. Some neurons in the goldfish communicate internally with such field effects. Field effects are used to communicate between individuals in the weakly-electric fish (the strongly electric fish use their fields to stun prey). Field effects are also responsible for pathological signaling in cases of epilepsy and multiple sclerosis. However, in general the EEG can be considered an information-carrying neural signal that is not used internally as a neural code.

Various signals are used directly by the brain and therefore can be considered to be codes. For example, the rate of spiking of neurons carries information that determines how powerfully a muscle will contract. This is a code that has been cracked: the nerve tells the muscle "squeeze ... squeeze harder." It appears likely that similar rate coding is also used in the central nervous system. Rate coding has also been suggested to be the primary code in parts of sensory cortex. Neurons in visual cortex spike fastest when presented with oriented bars of a certain configuration, and auditory cortex neurons will spike faster in response to particular sound frequencies.

There are an enormous number of electrical and chemical signals that influence neuron firing. Many of these can be considered to have a coding function as well. Most neurons use chemical synapses to communicate. The presence of neurotransmitter is a coding signal at these synapses. Synapses are typically viewed as passive information conduits connecting complicated information-processing neurons. An alternative view is that a synaptic complex may itself be a sophisticated information processor. Neurotransmitter concentration may vary and be a relevant signal in some cases. Within the postsynaptic cell, ions and molecules function as second and third messengers in cascades of chemical reactions. These chemical reactions can be very rapid. It may be that sequences of chemical reactions are as important as electrical activity for neural information processing.

2.7 The goals and methods of computational neuroscience

Conferences in computational neuroscience often feature energetic debates about what constitutes the correct approach to the problem of understanding brain function. Generally, it's biologists against computationalists, bottom-uppers versus top-downers. To caricature, the most rabid biologists believe that a model that deviates from the details of known physiology is inadmissibly inaccurate. Meanwhile, the computer scientists, physicists, and mathematicians feel that models that fail to simplify aggressively do not allow any useful generalizations to be made. The view presented here is one of compromise. Both perspectives are in part correct. Leaving out biological details will lead to models that can no longer make the connections with physiological experiment. However, failure to simplify at all can produce models that may not generalize at all. For example, it is possible to model a specific experiment with such fidelity to detail that one just has another copy of the experiment. Such a model would not be able to generalize so as to explain other related experiments in the same brain area. Also duplicating the system will not by itself give you any insight into how the system works.

In addition to the inherent intellectual tension between dry computers and wet biology, there are also historical tensions between traditional applied mathematics and the newer computational approaches. Traditionally, applied mathematics and theoretical physics were done with paper and pencil. The resulting formulations embedded complex physical phenomena in simple attractive formulae that could be disseminated by T-shirt and coffee cup. The Maxwell equations and $E = mc^2$ are examples that have been translated into both of these media. Although these equations are mysterious to most people, their elegance and aesthetic appeal is evident. They look like a key to the mysteries of the universe. Computer modeling, on the other hand, has little of the elegance and none of the generality of the traditional great equations. Although it is possible that neuroscience may someday yield clear-cut defining equations of this sort, it seems to me more likely that it will not. Just as with wet biology experiments, the results of computer simulations are rarely definitive and perhaps never canonical in the way of the great physics equations.

Computer modeling or simulation can be considered to be experimental mathematics. Simulations are themselves so complex that they must be studied by using virtual experiments to try to understand them. The simulated complex system, like the original, shows emergent behaviors whose origins and implications are not immediately obvious and must be explored experimentally. Traditional mathematics provides clean translations of reality. Simulation provides an alternative reality with advantages of manipulability and accessibility.

Simulation is used to assess large sets of complex mathematical formulae that cannot be solved by traditional analytic (paper-and-pencil) means. Since the single simulation never unequivocally represents the biology, it is often necessary to cross-check results among several simulations that represent the same system with different levels of detail or scale or simply with different choices for undefined parameters.

On the bright side, simulation also produces a variety of very nice benefits. Simply transforming a notion about how something works into an explicit computer model requires a complete accounting for all system parameters. Compiling this list often reveals basic, critical aspects of the system that are not known. Sometimes this is simply because no one ever bothered to look. Additionally, running computer simulations permits one to test specific questions about causality that can only be guessed at in paper-and-pencil modeling. Finally, working with computer simulations provides a way of getting a very intimate view of a complex system. The next time you take a commercial airliner flight, consider that this may be your pilot's first flight in this aircraft type, since many airlines now do all step-up training on a simulator. Just as flight simulators provide an intuitive feel for flight, neural simulators can provide intuition and understanding of the dynamics of neural systems. If I swim with the neurons long enough, maybe I'll learn to think like a neuron.

I have presented this brief history of brain thoughts partly to present my own view and place it in perspective. The view in this book is particularly contrasted with Marr's ideal separation of ends from means. The present evolutionary view of implementation entangled with task is comparable to the mainstream programming practices of Marr's era. In Chap. **??**, I present basic computer science through exploration of computer practices from that era, when hacking was necessary to perform complex computations despite hardware limitations. These practices have been lost with the growing power of computers and increasing sophistication of programming tools.

This book focuses on the interface between task and machine, where tricks and shortcuts, hacks in software parlance, are used to optimize a function on a particular architecture. The assumption is that the brain uses a thousand tiny hacks, each cleverly evolved to do some little task very well.