1 Of neurons and engineers

Neurons are fascinating, wildly diverse devices. Some, like the giraffe primary afferent, have axons that are 15 feet long. Others, like the typical granule cell, have axons that are only about 100 microns long. Some, like the common pyramidal cells, produce and transmit stereotypical action potentials (neural spikes) down their axon. Others, like the retinal horizontal cells, communicate without such spikes. Some send action potentials at speeds over 400 km/h. Others transmit spikes at a mere 2 km/h. Some, like the purkinje cells, have about 200,000 input connections. Others, like the retinal ganglion cells, have only about 500 such connections. There are literally hundreds and hundreds—by some estimates thousands—of different *kinds* of neurons in the brain.[1] And, taken together their numbers are staggering: 10^{10} neurons in the human brain; at least 10^{13} synapses; 45 miles of fiber; 100 different kinds of neurotransmitters. Such facts simply impress most people, but they can be exceedingly daunting to those interested in explaining and understanding neural systems.

How are we to get an explanatory grip on any system that is this complex? In this book we pursue the idea that the quantitative tools typically associated with the field of engineering hold great promise for understanding complex neural systems. Many of these tools have their roots in theoretical physics, mathematics, and even pure logic, but it is typically in the hands of engineers that they are applied to physical systems like the brain. Once we consider brains as purely physical devices, it becomes perfectly natural to reach into the engineer's toolbox and grab ahold of information theory, control theory, signal processing theory, and a host of other such formalisms in order to understand, characterize, and explain the function of such systems. It is, after all, a *functional* understanding of the brain that neuroscience is after—at least part of what neuroscientists want to know is how neurons cooperate to give rise to complex behavior—and it is characterizing function that is the engineer's primary concern. Thus, the theoretical tools of engineering are likely well-suited to helping us figure out what all of those neurons are doing.

Of course, these tools cannot be *blindly* applied to characterize neurobiological systems. For instance, merely reproducing, at some abstract level, a function of the brain (such as playing chess) is not enough for neuroscience (although it might be for artificial intelligence). Neuroscientists are interested in knowing how *neurons* give rise to brain function. For their purposes, the vast amounts of knowledge they have culled in recent decades must be respected when trying to understand neural systems; such neurobiological constraints are some of the best constraints we have when trying to understand how brains give rise to behavior. To illustrate, it is standard practise in communications theory to assume that

1 There are at least 23 kinds of amacrine cells alone.

incoming signals are evenly sampled over the range of the signal. A number of optimal decoding strategies have been devised that take advantage of this particular kind of sampling. However, this kind of sampling is clearly *not* present in neural systems. Sampling densities often change over the range of the signal, and the particular sample points can be quite different even between two members of the same species. This is because neurons are diverse, unlike the typical components of artifactual communications systems. Thus, looking to the actual constraints respected (or not) by neurobiological systems is important for applying various mathematical tools in the appropriate manner.

In what follows, we are committed to devising a neuroscientifically respectable approach to modeling neurobiological systems. Being concerned with actual neural systems means embracing all of the 'messiness' of such systems. Such messiness exists because brains are *natural* physical systems whose functioning is exquisitely shaped by real-world environments. Unlike their artifactual counterparts, they have not been designed to function like theoretical computational systems such as Turing machines. This does not mean that computational theory isn't useful, it just means that it shouldn't be taken at face value. We need to adopt *and adapt* the engineer's tools in order to successfully quantify the function of neural systems. So, while brains are first and foremost *implemented* computational systems, and there are extensive tools that have been developed for quantifying such computational systems, brains respect a set of design constraints significantly different from those typically respected by engineered systems. Like mother nature, an engineer is a practical scientist who has to build things that work. But mother nature, unlike an engineer, does not rely on carefully manufactured, fault tolerant, and nearly identical parts.

These considerations will come as no surprise to some. Indeed, engineering tools have been successfully applied to neural systems for many years (Fitzhugh 1958). Information theoretical analyses of spike trains (Rieke et al. 1997), basis function characterizations of neural representations (Poggio 1990), and control theoretic characterizations of neural systems (Wolpert and Kawato 1998), have all resulted in sophisticated, biologically constrained computational models of neurobiological systems. However, much of this work has been carried out in relative isolation: those using information theory do not seem to be talking to those doing basis function research or using control theory, and vice versa. One of our main goals in this book is to provide a *synthesis* of these approaches to understanding the brain. Although determining how such seemingly disparate research programs are related often requires significant extrapolation, we are not trying to provide new tools. Rather, we are trying to articulate a *new way* to use them together, and we are hoping to make them available to *new users*.

In order to accomplish these goals we adopt the terminology of both engineers and neuroscientists, but we assume minimal background in either engineering or neuroscience on the part of the reader. So, when introducing our framework, or discussing specific

examples of its application, we provide: 1) the relevant neuroscientific details needed to understand the new way of applying these tools to neurobiological systems; *and* 2) the relevant technical details needed to introduce these tools to new users. We have taken this approach because we think the communication between neuroscientists and engineers needs to be strengthened. We think there are likely benefits to both parties: engineers may come to understand novel computational strategies and sharpen existing analytical tools; and neuroscientists may come to a more precise understanding of neurobiological function and be better able to direct future experimentation.

These are goals that can be accomplished without a perfect understanding of neurobiological systems; which is why we have some hope of success. As necessary, we make simplifications, approximations, and guesses that, in the long run, destine our models to function differently than the real neural systems they are intended to model.[2] These assumptions are generally due either to a lack of neuroscientific knowledge, or a limitation of current engineering tools. This is, we take it, further evidence that neuroscience and engineering are in an excellent position to be mutually beneficial. In particular, making assumptions has clear benefits to both engineers and neuroscientists as they can guide experiments to gain the facts that are missing, help determine what kinds of facts are the most useful, tell us what kinds of analytical tools are needed, and help us better understand how to apply the tools available.

In sum, what we present in this book is a principled, unified approach to understanding neurobiological systems that employs the quantitative tools of engineers, respects neuroscientific results, and should be of mutual benefit to neuroscientists and engineers. In the remainder of this chapter, we provide an overview of our approach. In the remainder of the book, we provide the details and examples that demonstrate how the approach described in this first chapter is supposed to work.

1.1 Explaining neural systems

Neural systems are amazingly adept at solving problems. Seagulls open shellfish by dropping them on rocks, or even in front of moving cars (Grandin and Deesing 1998). Bees excel at finding their way through natural environments, and communicating what they learned on the trip (Gould 1975). Rats have an excellent sense of direction, even in complete darkness (Redish 1999). Primates are so good at dealing with problems that they have learned to fashion tools to help themselves cope (McGrew 1992). Being interested in neurobiology means being interested in providing explanations of how these diverse

2 This too, may be considered in the spirit of engineering. As the apocryphal bumper sticker famously notes: "Engineers are approximately perfect".

kinds of nervous systems give rise to such interesting, proficient behaviors. Of course, neuroscientists and others have been providing such explanations, whether good or bad, for decades. One fairly typical aspect of these explanations is that they invoke the notion of *representation*. Representations, broadly speaking, serve to relate the internal state of the animal to its environment; they are often said to 'stand-in for' some external state of affairs. At least since Hubel and Wiesel (1962), neuroscientists have talked of neurons *representing* their environment. Current discussions in neuroscience continue to rely heavily on the notion (see, e.g., Desimone 1991; Felleman and Essen 1991).[3]

Neuroscientists are by no means the only scientists to use the term 'representation' in this way. Indeed, the notion of a 'mental representation' has a two thousand year history in Western thought (Watson 1995). The explanatory power of representations comes from the fact that they can be manipulated internally without manipulating the actual, external, represented object. The benefits of being able to do this range from simply saving energy to being able to generate life-saving predictions. In general, then, representations help solve problems. So, it is hardly surprising that this notion is used by neuroscientists to explain the behavior of neurobiological systems.[4]

While representations are an important ingredient of the kinds of explanations that neuroscientists are interested in, they are not the only ingredient. There would be little interesting behavior produced by a neurobiological system that was simply 'filled' with representations. Rather, we also need to understand how representations are manipulated, exploited, related, updated, and so on. That is, we need to characterize how representations are *transformed* such that they are useful. In order to recognize an object, for example, the retinal image must, in some sense, be compared to past experiences (hence *'re*-cognize'). Those past experiences are not stored as retinal images. Instead, they have been transformed into a form that aids recognition; irrelevant features are ignored, noise is eliminated, and metrics along various dimensions are extracted. As well, similar transformations must be carried out on the current retinal image in order for such a comparison to occur. All of this work can be understood in terms of transformations from one representation into another. Consider also a more behavior-oriented example; reaching for an object. In order to know where the object is, an animal must rely on information regarding the orientation

3 A search of PubMed abstracts using "representation and neurons" results in over two thousand hits (see `http://www.ncbi.nlm.nih.gov/PubMed/`).

4 Of course, representational explanations are not the *only* kinds of explanations that neuroscientists are interested in. There are numerous questions concerning how neurobiological systems develop, what kinds of chemicals affect neural functioning, how the basic biological mechanisms supporting neuron activity function, and so on. These kinds of questions may need different kinds of explanations. However, the focus of this book is on what is often called 'systems' neuroscience, the part of neuroscience concerned with explaining the behavior of the (average, adult) neurobiological system in terms of its components. In this context, representational explanations so far seem to be the best kind of explanation available.

of the eyes, head, and body. In this case we can think of the retinal image as presenting the initial 'eye-centered' information regarding the position of the object, which needs to be eventually translated into arm-centered information in order to give rise to the appropriate reaching action. Again, this kind of change of coordinate systems can be understood in terms of the transformation of internal, neural representations.

So, we take the central problem facing neuroscientists to be one of *explaining how neurobiological systems represent the world, and how they use those representations, via transformations, to guide behavior*. As a result, we have divided the book into two main parts. The first part, consisting of chapters 2–5, is concerned with characterizing neural representations. The second part, consisting of chapters 6–9, is concerned with understanding transformations of such representations. In the next two sections we provide a brief overview of our approach to this problem. As will become clear, despite this preliminary distinction between representation and transformation, they are intimately related.

1.2 Neural representation

The main problem regarding mental representation, both historically and for contemporary philosophers of mind, is to determine the exact nature of the representation *relation*; that is, to specify the relation between, and representationally relevant properties of, things 'inside the head' and things 'outside the head'. The traditional approach to solving this problem is to consider the data (broadly construed) available from metaphysics, introspection, psychology, and more recently neuroscience, in order to deduce: 1) the representation relation; and 2) ways of determining what things are representations and what things are represented. However, we do not take this approach. Rather, we define outright the representational relationship and its relata, and see if our definition does the explanatory work that is needed. We happen to think that, as a matter of fact, our definition of representation does speak to the problems of mental representation that arise when adopting the traditional approach, but showing that is beyond the scope of this book (see Eliasmith 2000 for a detailed discussion). So, we will not discuss, except in passing, how to determine what things are representations and what things are represented. Rather, we take the standard kinds of representational claims that neuroscientists make more-or-less at face value, and show how our account makes those claims precise. So, for example, we take it (at least initially) that neural firings represent the stimulus that causes them, and we presume that neural populations represent the external objects or variables that the population activity is correlated with. In both cases, we use our definition to give an explanatorily and predictively useful account of what it means to make those claims. And, in both cases, we adjust the claims to make them more theoretically sound.

In order to precisely define the representation relation we turn to the quantitative tools of engineering. Specifically, we believe that there is a close tie between neural representations as understood by neuroscientists and *codes* as understood by communications engineers (Reza 1961). Codes, in engineering, are defined in terms of a complimentary *encoding* and *decoding* procedure between two *alphabets*. Morse code, for example, is defined by the one-to-one relation between letters of the Roman alphabet, and the alphabet composed of a standard set of dashes and dots. The encoding procedure is the mapping from the Roman alphabet to the Morse code alphabet and the decoding procedure is its inverse (i.e., the mapping from the Morse code alphabet to the Roman alphabet).

Representations, in neuroscience, are seldom so precisely defined but there are some commonalities between this notion of a code and the typical use of the term 'representation'. For instance, the term 'representation' is usually used to denote one or more neural firings from one or more neurons. For example, a neuron is said to represent a face at a certain orientation if it fires most strongly when the animal is presented with a face at that orientation (Desimone 1991). However, neural representations tend to be *graded* representations; i.e., a neuron fires more or less strongly depending on the nearness of the stimulus to what is called its 'preferred' stimulus (that is, the stimulus that causes it to have the highest firing rate). Neuroscientists are interested in characterizing the relation between this graded representation and stimuli that evoke it. In fact, this standard description of neural behavior maps quite nicely onto the notion of *encoding* in engineering: neural firings encode properties of external stimuli.

However, in order to characterize the representation relation in a manner similar to the way communications engineers characterize codes, we must also identify the decoding procedure. This is a surprisingly natural constraint on representation in neural systems. If there is a neuron that initially seems to fire most strongly to a face at a certain orientation, then it must be *used* by 'downstream' processes in order for the claim that it actually *represents* that orientation to make any sense. If the previously described neural firing was, in fact, used by the system to determine eye orientation (and it just so happened that in the initial experiment face orientation and eye orientation covaried because the eyes were always facing forward), then the best case could be made for it representing eye orientation (given subsequent, better controlled experiments), and not face orientation. Of course, what it means for a downstream system to *use* these neural firings is that the downstream system can extract (i.e., *decode*) that particular information (e.g., eye orientation) from those neural firings. This holds as well for graded representations as for any other kind of representation. So decoding is as important to characterizing neural representations as encoding (we return to these issues in chapter 6).

Now, in order to fully characterize neural representation as a code, we need to specify the relevant alphabets. A problem arises for neural representation here because there are so

many different possible alphabets that can be identified. For instance, if we are interested in the representations of individual retinal ganglion cells, it seems that something like light intensities at certain retinal locations and spike trains of single neurons would be the relevant alphabets. In contrast, if we are interested an entire cortical area, like the primary visual cortex, it seems that something more like color, spatial frequency, intensity, etc. over the whole visual field and spike trains of large populations of neurons would be the relevant alphabets. The alphabets in these cases are extremely different. Nevertheless, if we are interested in a general understanding of neural representation (i.e., a general means of finding encoding and decoding rules), we *can* find general commonalities between these alphabets. Namely, we can understand these behaviors as relating *neural responses* (alphabet 1) and *physical properties* (alphabet 2).

In fact, we think it is possible to be a bit more specific. Neuroscientists generally agree that the basic element of the neural alphabet is the neural spike (see, e.g., Martin 1991, p. 334). However, it may be that the neural alphabets that are actually used include the average production rate of neural spikes (i.e., a rate code), specific timings of neural spikes (i.e., a timing code), population-wide groupings of neural spikes (i.e., a population code), or the synchrony of neural spikes across neurons (i.e., a synchrony code). Of these possibilities, arguably the best evidence exists for a combination of timing codes (see Rieke et al. 1997 for an overview) and population codes (see Salinas and Abbott 1994; Seung and Sompolinsky 1993; and Abbott 1994 for an overview).[5] For this reason, we take these two kinds of basic coding schemes to comprise the alphabet of neural responses (i.e., alphabet 1).

However, it is much more difficult to be any more specific about the nature of the alphabet of physical properties. We can begin by looking to the physical sciences for categories of physical properties that might be encoded by nervous systems. Indeed, we find that many of the properties that physicists traditionally identify do seem to be represented in nervous systems; e.g., displacement, velocity, acceleration, wavelength, temperature, pressure, mass, etc. But, there are many physical properties not discussed by physicists which also seem to be encoded in nervous systems; e.g., red, hot, square, dangerous, edible, object, conspecific, etc. Presumably, all of these latter 'higher-order' properties are *inferred* on the basis of (i.e., are the results of transformations of) representations of properties more like those that physicists talk about. In other words, encodings of 'edible' depend, in some complex way, on encodings of more basic physical properties like wavelength, temperature, etc. Given this assumption, our focus is (as it is in neuroscience, generally) on trying to understand how nervous systems encode the more basic physical properties. Eventually,

5 For a demonstration that rate codes are a specific instance of timing codes see Rieke et al. (1997). For empirical evidence that synchrony codes are not used by nervous systems see Kiper et al. (1996) and Hardcastle (1997).

however, we need to be able to explain how the property of being edible, or being an object is encoded by neurons. We think that this framework is flexible enough to help with this problem, but for the time being we focus our attention on characterizing more basic physical properties, where we believe successes can be more convincingly demonstrated.

This, then, is how 'neural representation' in neuroscience can be defined analogously to how 'codes' are defined in communications engineering. To recap, neurons encode physical properties into population-temporal neural activities that can be decoded to retrieve the relevant information. However, there are also important differences between engineered codes and neural representations. Perhaps most importantly, the former are *specified* whereas the latter are *discovered*. So there can be (and should be) a lot of debate concerning what is *actually* represented by a given neuron or neural population. Even without wading into the philosophical quicksand that surrounds such representational claims, we can discover something important about the proposed definition of representation. That is, there is bound to be some *arbitrariness* to the decoding we pick and thus the claims that we make regarding representation in neural populations. This is especially true at the beginning of our explorations of neural systems.

There are two sources of this seeming arbitrariness. First, since knowing what is represented depends in part on how it is subsequently used, it seems like we already have to know how the system works in order to know what it represents. But, of course, how the system works is precisely what we are trying to figure out when we are talking about representation. This problem seems much less of an obstacle once we realize that many difficult explanatory problems are resolved by making guesses about how things work and then testing those guesses (this is just the idealized 'scientific method'). So, although the 'fact of the matter' about what is represented will only be resolved once we have a fairly comprehensive understanding of what is actually represented in the brain, this should not be taken to be an insurmountable difficulty, or viciously circular. Specifically, it does not mean that our interim hypotheses are wrong, just that they might be (which is why they are *hypotheses*). In effect, what is 'really' represented is whatever is taken to be represented in a more-or-less complete, coherent, consistent, and useful theory of total brain function—a theory at least many decades away. But this does not mean that it is pointless to make representational claims now; in fact, making such claims is an essential step towards such a theory.

The second source of seeming arbitrariness stems from the fact that information encoded by a neural population may be decoded in a variety of ways. To see why, consider a neural population that encodes eye velocity. Not surprisingly, we can decode the information carried by this population to give us an estimate of eye velocity. However, we can also decode that same information to give us an estimate of a *function of* eye velocity (e.g., the square). This is because we can essentially 'weight' the information however

we see fit when decoding the population activity; different weightings result in different decoded functions. Since representation is defined in terms of encoding *and* decoding, it seems that we need a way to pick which of these possible decodings is the relevant one for defining *the* representation in the original population. We resolve this issue by specifying that what a population represents is determined by the decoding that results in the quantity that all other decodings are functions of. We call this the '*representational decoding*'. Thus, in this example, the population would be said to represent eye velocity because eye velocity and the square *of eye velocity* are decoded. Things are not quite so simple because, of course, eye velocity is also a function of the square of eye velocity. This problem can be resolved by recalling considerations brought to bear on the first source of ambiguity; namely that the right physical quantities for representation are those that are part of a coherent, consistent, and useful theory. Because physics (a coherent, consistent, and useful theory) quantifies over velocities (and not squares of velocities), so should neuroscience (as this renders science as a whole more coherent, consistent and useful).

While these considerations may seem distant from empirical neuroscience, they play an important role in specifying what is meant by claims that a neuron or neural population is representing the environment; claims that empirical neuroscientists make all the time. So, throughout the course of the book we revisit these issues as various examples and analyses are presented that provide greater insight into the nature of neural representation.

In sum, there are good reasons to think that neuroscientists can and should rigorously define the notion of neural representation along the lines that engineers have defined the notion of codes.[6] Specifically, both encoding and decoding are important for defining neural representation, and the relevant alphabets are neural activities and physical properties. There are important differences between neural representation and engineered codes that raise important theoretical issues regarding the nature of neural representation. Nevertheless, considerations from engineering provide an excellent starting point for characterizing neural representation. In the next two sections we present more detail on how coding theory can play this role.

1.2.1 The single neuron

To adopt an engineering perspective regarding neural function, we can begin by asking: What kind of physical devices are neurons? Fortunately, there is some consensus on an answer to this question; neurons are electro-chemical devices. Simpler yet, the behavior of single neurons can be well-characterized in terms of their *electrical* properties. In fact, detailed, quantitative, and highly accurate models of single cell behavior have been around

6 We are by no means the first to suggest this (see, e.g., Fitzhugh 1958; Rieke et al. 1997; Abbott 1994; Seung and Sompolinsky 1993).

for about 50 years, and continue to be improved (see Bower and Beeman 1995 for a history and some recent models). By far the majority of vertebrate neurons can be understood as physical devices that convert an 'input' voltage change on their dendrites into an 'output' voltage spike train that travels down their axon. This spike train is the result of a highly nonlinear process in the neuron's soma that relies on the interaction of different kinds of voltage-gated ion channels in the cell membrane. These spike trains then cause further electrical changes in the dendrites of receiving (i.e., postsynaptic) neurons.

The dendrites themselves have active electrical properties similar to those in the soma, which result in a current flowing to the soma from the dendrites.[7] The soma voltage itself is determined by the current flowing into the soma from the dendrites, as well as active and passive membrane resistances and a passive membrane capacitance. Typically, the passive membrane time constant in the soma (determined by the passive resistance and capacitance) is on the order of about 10 ms. This time constant effectively characterizes the time-scale of the 'memory' of the soma with respect to past signals. Thus, the signal generation process only 'remembers' (or is sensitive to) dendritic inputs that occurred in the very recent past—perhaps as long ago as 200 ms, but more typically in the tens of milliseconds.[8] So, considered as electrical devices, neurons have highly nonlinear input processes (at the dendrites), highly nonlinear output processes (in the soma, resulting in voltage spike trains), and a fairly short signal memory (in both dendrites and at the soma).

Because neurons can be characterized as electrical devices that transmit signals in this way, it is fairly natural to analyze their behavior using the tools of signal processing theory. In other words, given this characterization neurons can be directly analyzed as information processing devices. Adopting this perspective leads to the insight that neurons are effectively *low-precision* electrical devices that transmit about 1–3 bits of information per neural spike, or a few hundred to a thousand bits per second (see section 4.4.2). This means that modeling their output using 16 bit floating point numbers updated at hundreds of *mega*hertz—as those in the artificial neural net community tend to do—is not a good way to characterize real, *neurobiological* representation. We are concerned with neural representation only from this combination of a biological and engineering perspective; i.e., understanding real neurons as electrical devices.

---

7 Although we begin by assuming that dendritic contributions are linear, we discuss a means of modeling the likely dendritic nonlinearities in section 6.3.

8 Of course, the dendrites themselves have a longer memory, commonly characterized by a synaptic weight (which can change due to learning). However, even in the dendrites, there is a sense in which the memory of recent electrical input decays within about 200 ms. Specifically, if we distinguish the synaptic weight (which changes slowly) from the postsynaptic current (which changes rapidly as spikes are received), as most models do, then the memory of activity is short (i.e., on the order of the time constant of the postsynaptic current). Because synaptic weights change slowly, they can be considered fixed on the time-scale of the synaptic currents and somatic spikes. Given the large difference in time scales, synaptic weights can therefore be considered separately from dendritic postsynaptic currents. In particular, the weights can be treated as a simple multiplier, or gain, on a stereotypical postsynaptic response.

Nevertheless, it is useful to examine how real neurons, considered as information processors, are analogous to transistors on silicon computer chips. Like transistors, neurons are: 1) electrical devices; 2) highly nonlinear; and 3) signal/information processors. These similarities make it very natural to use the kinds of mathematical tools that have been used by engineers in the past to understand transistor signal processing in order to understand neurons. However, we must bear in mind that, unlike transistors, neurons: 1) have very short memories; 2) output voltage spikes; 3) are heterogeneous; and 4) are biological (i.e., not manufactured). These differences again make it clear that we have to be careful exactly how we apply engineering tools to understanding neurobiology. In chapter 4, we discuss a means of characterizing neurons as signal processors with these differences in mind and using both simple and realistic neural models. Because of the diversity of models we consider, it becomes clear that our approach does not rely on the functioning of specific kinds of neurons (or neural models). This helps make the approach a general one. And, more importantly, the insights gained by understanding single neurons as information processors are essential for putting them together to build realistic, large-scale models.

1.2.2 Beyond the single neuron

Despite our discussion of single neuron function, the focus of this book is definitively *not* on the analysis of single neurons (*c.f.* Koch 1999; Wilson 1999b; Rieke et al. 1997). Rather, it is on understanding how groups or populations of neurons can be characterized as working together to support neural representation and transformation. To this end, we begin our discussion of representation with population-level considerations (in chapter 2) before worrying about the details of information processing in individual neurons (in chapter 4). After developing these two aspects separately, we combine them to give a general description of 'population-temporal' representation (in chapter 5). As we show, describing population coding more-or-less independently of the specifics of individual neuron function provides the flexibility to model many levels of representational detail concurrently in a single model.

Because our focus is on population representation, we go to some length to show that having to depend on highly nonlinear processing elements (i.e., the neurons), does not necessarily result in difficult-to-analyze representations. This result is not too surprising if we again consider the analogy between neurons and transistors. Binary representations in computers rely on encoding signals into 'populations' of highly nonlinear processing elements (i.e., the transistors). Nevertheless, these signals are decoded using a simple linear decoder (see section 2.1.1). Similarly, we show that considering populations of nonlinear neurons can result in good, simple, linearly decodable representations (see section 4.3.2). Furthermore, as more neurons are included in the representation, it becomes even better; but only if the resulting population is *heterogeneous* (i.e., has a range of neuron properties).

In other words, population activity can be linearly decoded to give an increasingly accurate indication of what was originally (nonlinearly) encoded, with an increasing number of heterogeneous neurons.

There are two important results here. First, it is extremely fortunate that we are able to extract the information that was nonlinearly encoded using a linear decoder because that allows many of the tools of linear signals and systems theory, a very well-developed and understood field in engineering, to be at our disposal. Second, the often underemphasized property of the heterogeneity of neural systems become central. Given this perspective, the heterogeneity of neural populations can be explained from a functional point of view; in fact, heterogeneity becomes indispensable for a good representation (see section 7.5). This has significant consequences for both experimentalists and theoreticians. Specifically, this result shows that it is more appropriate for experimentalists to report the *distributions* of neuron response properties, rather than presenting a few 'typical' (i.e., best) neurons in detail. It is more appropriate because it is the distribution that provides insight into the kind of representation employed by, and the function of, the system under study (see sections 7.4 and 7.3). For theoreticians this result means that assuming that every neuron in a population is identical is going to give significantly misleading results, despite such assumptions making the system mathematically simpler to handle.

The sensitivity of neural representation to population-level properties like heterogeneity and the number of neurons suggests that it is most useful to think of neural representation in terms of populations, rather than in terms of single neurons. Thus, we argue that it the fundamental unit of signal processing in the nervous system is the simplest neural population (a neuron pair), rather than the single neuron (see section 4.3.2).

Adopting this perspective on neural representation has some useful pragmatic results. In particular, focusing on population coding permits us to consider a given model with varying degrees of detail. We can, for instance, build a simulation using only population representations, ignoring the details of individual neurons. Or, we can build the same simulation using neural-level representations and including whatever degree of biological detail is appropriate. Or, we can build that same simulation using *both* kinds of representation concurrently.[9] This flexibility is useful because it addresses an important need in neuroscience: "Ideally, we wish to be able to move smoothly between levels of models and to understand how to reduce systematically more complex models into simpler forms that retain their essential properties" (Marder et al. 1997, p. 143). Our analysis of representation addresses this need by concurrently supporting various levels of representation; from highly abstract population representations to the single cell representations of detailed con-

9 The simulation package that we have released with this book supports such multi-level simulations. It can be found at `http://compneuro.uwaterloo.ca`.

ductance models. When formalizing these representational levels, we discuss how to define a representational hierarchy that spans the levels of biological structure from single neurons through networks and maps to brain areas. To preview, this hierarchy begins with scalar representations, like those found in nuclei prepositus hypoglossi for controlling horizontal eye position (sections 2.3, 5.3, and 8.2). It then incorporates slightly more complex vector representations, like those found in motor cortex for controlling arm motion (section 2.5) and those found in vestibular nucleus for representing angular velocity and linear acceleration of the head (sections 6.5). Lastly, we use the hierarchy to characterize functional representations, like those found in lateral intraparietal cortex (sections 3.4 and 8.3). Despite our not providing examples of the hierarchy past this specific representational level, we show how it is straightforward to generalize these examples to more complex representations (like vector fields). While it is unlikely that there is a precise relation between such a representational hierarchy and biological structure, being able to build a general and flexible hierarchy proves useful for characterizing such structure at many different levels.

1.3 NEURAL TRANSFORMATION

In part II of this book we show that our characterization of neural representation paves the way for a useful understanding of neural transformation. This is largely because transformation, like representation, can be characterized using linear decoding. However, rather than using the *representational decoder* discussed earlier, we use what we call a *transformational decoder*. The contrast between these two kinds of decoders lies in the fact that, when performing a transformation on encoded information, we are attempting to extract information *other* than what the population is taken to represent. Transformational decoding, then, is not a 'pure' decoding of the encoded information. So, for example, if we think that the quantity x is encoded in some neural population, when defining the *representation* we identify the decoders that estimate x (i.e., the information that is taken to be 'primarily' encoded by that population). However, when defining a *transformation* we identify decoders that estimate some function, $f(x)$, of the represented quantity, x. In other words, we find decoders that, rather than extracting the signal represented by a population, extract some transformed version of that signal.

Defining transformations in this way allows us to use a slight variation of our representational analysis to determine what transformations a neural population can, in principle, support (see section 7.3). This allows us to determine how well a given neural population can support the transformations defined by a particular class of functions. This can be very important for constraining hypotheses about the functional role of particular neural populations that are observed in a neurobiological system. We show that neurons with

certain response properties support particular transformations better than others. This is a good reason to think that populations with those properties are involved in computing certain transformations rather than others. In other words, this approach enables a good, quantitative characterization of the functional potential of sets of neurons.

Furthermore, defining transformations in this way permits us to *analytically* find connection weights in a biologically plausible network. So, rather than having to train a fully connected network using a learning rule (which might prove difficult if not impossible for large-scale simulations), we can define the representations we take to be in the system and the transformations that those representations undergo, and then directly find the weights to implement those transformations. This is beneficial in that it allows us to test explicit hypotheses about the function of a particular neural system, without having to worry about how to train that system to perform some function. As well, there is a significant practical benefit, in terms of computational savings, in not having to simulate large training regimes for complex models. Notably, this does not mean that learning is somehow antithetical to our approach (we discuss the relation of this approach to learning in chapter 9), it merely means that we do not *need* to rely on training to have interesting, biologically plausible models.

However, this is not the whole story about neural transformation. Transformations, as we have discussed them so far, are just like computing some function of x. However, this kind of static computation of a function is not, alone, a good description of the kinds of transformations neurobiological systems typically exhibit. Animals have evolved in a dynamic environment and are themselves dynamic systems. So, it is essential to be able to characterize the *dynamics* of the transformations that neural populations support. Again, engineers have a number of useful quantitative tools for describing dynamic systems. In particular, modern control theory has been successfully used by engineers to describe a huge variety of dynamic systems, both natural and artificial. In order to apply these same techniques to neural systems, we must identify the 'state vector' (i.e., the real-valued vector that tracks the system's internal variables) of the system. In chapter 8 we argue that neural representations can play this role. Given our analyses of neural representation, which include vector representation, this should come as no surprise. However, because neurons have intrinsic dynamics dictated by their particular physical characteristics, we must also *adapt* the standard control theory toolbox for understanding neurobiological systems (see section 8.1). Once this is done, we can directly apply the techniques for analyzing complex dynamic systems that have been developed by, and for, engineers.

Describing dynamics in this manner allows us to address issues that have proven very difficult for other approaches: "A great challenge for the future is to understand how the flexible modulation of motor circuits occurs without the loss of their essential stability" (Marder et al. 1997, p. 147). We attempt to meet this specific challenge in section 8.5.

More generally, because the stability (and various other high-level properties) of control systems is well understood, we believe that this approach to neural dynamics can help quantify our understanding of the dynamics of a wide range of neurobiological systems in an interesting, new way (see section 8.4).

1.4 THREE PRINCIPLES OF NEURAL ENGINEERING

To this point in the chapter we have outlined, in very general terms, the approach that we develop in the remainder of the book. In this section, we consolidate these previous considerations by clearly stating the guiding assumptions of our approach. Much of our argument for these 'principles of neural engineering' is of the proof-is-in-the-pudding variety. That is, throughout the course of the book we provide numerous examples of detailed models of a wide variety of neurobiological systems that were constructed based on these principles. But, our goal is not to simply provide examples, rather it is to demonstrate how to *use* these principles. That is, we are interested in describing a framework for understanding and simulating neurobiological systems in general. Thus we provide not only this set of guiding principles, but also a *methodology* for applying these principles. In order to demonstrate this methodology, we follow it for each of the examples we present. It is important for our purposes that not only are a large number and variety of examples provided, but that they are also built with consistent assumptions and with a consistent methodology. Given our discussion to this point, these principles should be unsurprising:

Three principles of neural engineering

1. Neural representations are defined by the combination of nonlinear encoding (exemplified by neuron tuning curves) and weighted linear decoding (see chapters 2, 3, 4, and 5).
2. Transformations of neural representations are functions of variables that are represented by neural populations. Transformations are determined using an alternately weighted linear decoding (i.e., the transformational decoding as opposed to the representational decoding; see chapters 6 and 7).
3. Neural dynamics are characterized by considering neural representations as control theoretic state variables. Thus, the dynamics of neurobiological systems can be analyzed using control theory (see chapter 8).

We take these three principles to be excellent guiding assumptions for the construction of biologically plausible, large-scale simulations of neurobiological systems. While it is premature to state these principles more quantitatively, later on we will be in a position to do so (see section 8.1.4). In addition to these main principles, there is an important addendum which guides our analyses of neurobiological systems:

Addendum

4. Neural systems are subject to significant amounts of noise. Therefore, any analysis of such systems must account for the effects of noise (see sections 2.2, and 5.2).

We do not consider this addendum to be a principle because, rather than being a claim about how to *explain the functioning* of neurobiological systems, it is a claim about how to *analyze* such systems. Nevertheless, it is essential for articulating the principles in detail. In the next four sections, we briefly discuss the strengths of, and possible concerns with, these principles and the addendum.

1.4.1 Principle 1

Principle 1 emphasizes the importance of identifying both encoding and decoding when defining neural representation. Moreover, this principle highlights the central assumption that, despite a *nonlinear* encoding, *linear* decoding is valid (see Rieke et al. 1997, pp. 76–87). As discussed in detail by Rieke et al., a nonlinear response function like that of typical neurons is, in fact, unrelated to whether or not the resulting signal can be linearly decoded.[10] That is, the nature of the input/output function (i.e., encoding) of a device is independent of the decoder that is needed to estimate its input. This means that a nonlinear encoding could need a linear *or* nonlinear decoding, and vice versa. This is because the decoding depends on the conditional probability of input given the output *and* on the statistics of the noise (hence our addendum). Perhaps surprisingly, linear decoding works quite well in many neural systems. Specifically, the additional information gained with nonlinear decoding is generally less than 5%.

Of course, nonlinear decoding is able to do as well or better than linear decoding at extracting information, but the price paid in biological plausibility is generally thought to be quite high (see, e.g., Salinas and Abbott 1994). Furthermore, even if there initially seems to be a case in which nonlinear decoding is employed by a neural system, that decoding may, in the end, be explained by linear decoding. This is because, as we discuss in section 6.3, nonlinear transformations can be performed using *linear* decoding. Thus, assuming linear decoding at the neuron (or sub-neuron, see section 6.3) level can well be consistent with nonlinear decoding at the network (or neuron) level. So, especially in combination with principle 2, linear decoding is a good candidate for describing neural decoding in general.

It is important to emphasize that analyzing neurons as decoding signals using (optimal) linear or nonlinear filters does not mean that neurons are presumed to *explicitly use* opti-

10 During this same discussion, Rieke et al. mention that there are certain constraints on when linear decoding will work. In particular, they claim that there can be only a few, preferably one, spike(s) per correlation time of the signal. However, we have found that this not the case (see section 4.3.3).

mal filters. In fact, according to our account, there is no directly observable counterpart to these optimal decoders. Rather, the decoders are 'embedded' in the synaptic weights between neighboring neurons. That is, coupling weights of neighboring neurons indirectly reflect a particular population decoder, but they are not identical to the population decoder, nor can the decoder be unequivocally 'read-off' of the weights. This is because connection weights are determined by both the decoding of incoming signals *and* the encoding of the outgoing signals (see, e.g., section 6.2). Practically speaking, this means that changing a connection weight both changes the transformation being performed and the tuning curve of the receiving neuron. As is well known from work in artificial neural networks and computational neuroscience, this is exactly what should happen. In essence, the encoding/decoding distinction is not one that neurobiological systems need to respect in order to perform their functions, but it is extremely useful in trying to *understand* such systems and how they do, in fact, manage to perform those functions.

1.4.2 Principle 2

The preceding comments about representational decoders apply equally to transformational decoders. This should be no surprise given our prior discussion (in section 1.3) in which we noted that defining a transformation is just like defining a representation (although with different decoders). However, we did not previously emphasize the kinds of transformations that can be supported with linear decoding.

It has often been argued that nonlinear transformations are by far the most common kind of transformations found in neurobiological systems (see, e.g., Freeman 1987). This should not be surprising to engineers, as most real-world control problems require complex, nonlinear control analyses; a good contemporary example being the remote manipulator system on the international space station. This should be even less of a surprise to neuroscientists who study the subtle behavior of natural systems. As Pouget and Sejnowski (1997) note, even a relatively simple task, such as determining the head-centered coordinates of a target given retinal coordinates, requires nonlinear computation when considered fully (i.e., including the geometry of rotation in three dimensions). Thus, it is essential that we be able to account for *nonlinear* as well as linear transformations. In section 6.3 we discuss how to characterize nonlinear transformations in general. We provide a neurobiological example of a nonlinear transformation (determining the cross product) that allows us to account for a number of experimental results (see section 6.5). Thus we show that assumptions about the linearity of decoding do not limit the possible functions that can be supported by neurobiological systems.

This result will not be surprising to researchers familiar with current computational neuroscience. It has long been known that linear decoding of nonlinear 'basis functions' can be used to approximate nonlinear functions (see section 7.4). Nevertheless, our analysis

sheds new light on standard approaches. Specifically, we: 1) show how observations about neural systems can determine *which* nonlinear functions can be well-approximated by those systems (section 7.3); 2) apply these results to large-scale, fully spiking networks (section 6.5); and 3) integrate these results with a characterization of neural dynamics and representation (section 8.1.3).

1.4.3 Principle 3

As noted in section 1.3, we can adapt standard control theory to be useful for modeling neurobiological systems by accounting for intrinsic neuron dynamics. There are a number of features of control theory that make it extremely useful for modeling neurobiological systems. First, the general form of control systems, captured by the state-space equations, can be used to relate to the dynamics of non-biological systems (with which engineers may be more familiar) to the dynamics of neurobiological systems. Second, the engineering community is very familiar with the state-space approach for describing the dynamic properties of physical systems, and thus has many related analytical tools for characterizing such systems. Third, modern control theory can be used to relate the dynamics of 'external' variables, like actual joint angles, to 'internal' variables, like desired joint angles. This demonstrates how one formalism can be used to span internal and external descriptions of behavior.

Adopting this perspective on neural dynamics allows us to develop a characterization of what we call a 'generic neural subsystem'. This multi-level, quantitative characterization of neural systems serves to unify our discussions of neural representation, transformation, and dynamics (section 8.1.3).

Given our previous discussion regarding the importance of nonlinear computations, a focus on standard control theory, which deals mainly with linear dynamics, may seem unwarranted. However, contemporary nonlinear control theory, which may prove more valuable in the long run, depends critically on our current understanding of linear systems. Thus, showing how linear control theory relates to neurobiological systems has the effect of showing how nonlinear control theory relates to neurobiological systems as well. In fact, many of the examples we provide are of nonlinear dynamic systems (see sections 6.5 and 8.2).

1.4.4 Addendum

There are numerous sources of noise in any physical system, and neurobiological systems are no exception (see section 2.2.1). As a result, and despite recent contentions to the contrary (van Gelder and Port 1995), neural systems can be understood as essentially finite (Eliasmith 2001). This is important, though not surprising, because it means that

information theory is applicable to analyzing such systems. This ubiquity of noise also suggests that knowing the limits of neural processing is important for understanding that processing. For instance, we would not expect neurons to transmit information at a rate of 10 or 20 bits per spike if the usual sources of noise limited the signal-to-noise ratio to 10:1, because that would waste valuable resources. Instead, we would expect information transmission rates of about 3 bits per spike given that signal-to-noise ratio as is found in many neurobiological systems (see section 4.4.2).

These kinds of limits prove to be very useful for determining how *good* we can expect a system's performance to be, and for constraining hypotheses about what a particular neural system is *for*. For example, if we choose to model a system with about 100 neurons (such as the horizontal neural integrator in the goldfish), and we know that the variance of the noise in the system is about 10%, we can expect a root-mean-squared (RMS) error of about 2% in that system's representation (see section 2.3). Conversely, we might know the errors typically observed in a system's behavior and the nature of the relevant signals, and use this knowledge to guide hypotheses about which subsystems are involved in which functions. Either way, information regarding implementational constraints, like noise, can help us learn something new about the system in which we are interested.

1.5 METHODOLOGY

As noted previously, our central goal is to provide a general *framework* for constructing neurobiological simulations. We take this framework to consist of two parts: the guiding *principles* just outlined; and a *methodology* for applying those principles, which we describe next. So our interests are practical as well as theoretical. To this end, we have written a software package in MatLab® that can be used with this methodology (and applies the principles). Nearly all of the models discussed in the remainder of the book have been implemented with this package. The package, examples, documentation, and some extras are available at `http://compneuro.uwaterloo.ca`.

We present the methodology in three stages: system description; design specification; and implementation.

1.5.1 System description

The main goal of this first step is to describe the neural system of interest in such a way that the principles outlined in section 1.4 are directly applicable to it. In particular, available neuroanatomical data and any current functional understanding should be used to describe the architecture, function, and representations of the neural system. This description should include at least:

1. basic interconnectivity between subsystems (i.e., what is connected to what);

2. neuron response functions (i.e., distributions of neuron parameters evident in the relevant populations);

3. neuron tuning curves (i.e., distributions of encoding functions of the relevant populations);

4. subsystem functional relations; and

5. overall system behavior.

Of course, if this information was easily available, we might have little reason to construct a model. As a result, many of these details will probably be hypotheses. The point is to make explicit the assumptions that inform the model. Then, any differences between model function and the function of the modeled system may be traced to these assumptions.

Importantly, the last two functional descriptions (4. and 5.) need to be expressed in mathematical terms. In particular, the relevant neural representations need to be specified in such a way that they can be used to write explicit transformations that perform the specified functions. The goal here is to translate the functional description provided in neurobiological terms into a description in mathematical terms. So, for example, we might describe a particular neural subsystem as acting as working memory for spatial locations. To provide a mathematical description, we must identify a represented variable (e.g., x), units (e.g., degrees from midline), and a description of the dynamics that captures the notion of memory (e.g., $\dot{x}(t) = 0$, i.e., x stays the same over time; see section 8.3). In this case, there is a natural correspondence between time derivatives equaling zero and memory, which we have exploited. In most cases, the translation is much more involved. If this subsystem is part of a larger neural system that we are interested in simulating, then a similar procedure must be carried out for each subsystem. Control theory-like diagrams can be very useful for performing this decomposition.

Essentially, this step requires a rigorous formulation of hypotheses we may have regarding the function of the system we are interested in. Admittedly, this mathematical description may be highly abstract (e.g., describing a swimming eel as instantiating a kind of sine function; see section 8.5). But that is acceptable so long as the description is complete. It is not necessary to hypothesize about the functioning of every individual neuron or neuronal group, so long as we have a hypothesis about the overall behavior. In fact, our framework is intended to be a means of determining what the likely role of neurons or groups of neurons *is* at these 'lower' levels. Any differences between the resulting simulation and known neurobiological properties of these lower levels (e.g., connectivity, tuning curves, etc.) should help improve the higher-level hypothesis. Given the improved higher-level formulation, predicted neural properties can again be examined,

and so on. Thus, this methodology supports a 'bootstrapping' approach to understanding neural systems.

In sum, the main purposes of this step are to: 1) identify the relevant neurobiological constraints; 2) identify the represented system variables; and 3) rigorously relate those variables to one another in such a way that the observed behavior results.

1.5.2 Design specification

The purpose of the design specification is to further delineate the real-world limitations that are known, or assumed to be present, for the neural system. As a result of the addendum, we must be explicit about the operating conditions (e.g., noise) that the system is subject to. This second step in the methodology explicitly demands stating the implementational constraints. So, given the representation for each variable as determined in the system description, the dynamic range, precision, and signal-to-noise ratio for each degree of freedom of those variables must be specified. Similarly, the temporal and dynamic characteristics (e.g., bandwidth, power spectrum, stable regimes, etc.) must be described. Ideally, these specifications would be made on the basis of available data. However, they may also be parameters to be manipulated during the implementation stage of the methodology. Adopting this route allows us to ask questions like: How much noise could such a system tolerate? How good does the encoding of individual neurons have to be under various noise conditions? What is the minimum allowable bandwidth for the system to function properly? and so on.

Although the precise specifications may not seem important, they can significantly affect the final model that is generated (see section 2.3.3). This goes to show the significance of implementational constraints for building good neurobiological models. If, for example, the signal-to-noise ratio must be extremely high for a particular variable, there will have to be many neurons dedicated to representing that variable (or fewer highly precise neurons). Conversely, if we have good estimates of the number of neurons in a particular system, we can use that information to determine possible design specifications.

In sum, the main purpose of this step is to precisely specify the implementational constraints on the model for each represented variable identified in the previous step.

1.5.3 Implementation

The third step of the methodology involves generating and running the model itself. Given the system description and design specification, this step combines them to determine the appropriate decoding rules, and hence synaptic weights, needed to implement the desired behavior.

Because the original system description may be framed in terms of high-level neural representations, it is often possible to simulate some parts of the model at the level of

those representations (i.e., without simulating every individual neuron's function) while simulating other parts of the model with more realistic spiking neurons. As well, the amount of detail in the model of individual neurons (e.g., a rate model, spiking model, or conductance model) can vary from one part of the model to another. The computational savings of these variations in detail can be significant for large-scale models. In many cases, large-scale models could not be simulated on available hardware without this kind of control over the amount of detail incorporated into various parts of the model.

In general, the purpose of the implementation step is to run numerical experiments on the model. These experiments may take the form of simply changing the input to the network, or they might involve changing system properties defined in the design specification. The implementation stage thus supports performing an in-depth analysis of the model's behavior (e.g., stability analysis, sensitivity to noise, etc.). The results of such experiments and analyses can be used to inform revisions to either of the two previous steps. In the end, the results of such numerical experiments often suggest neurobiological experiments to pursue in the system being modeled (see, e.g., 6.5 and 8.3).

In sum, the main purpose of the final step of the methodology is to apply the principles of neural engineering outlined previously to embed the system description into a plausible neural model, and to analyze and experiment with the resulting simulation.

1.5.4 Discussion

The three main steps of the methodology and their objectives are summarized in table 1.1. These steps provide a 'recipe' for generating simulations of neurobiological systems. However, applying these steps to real systems is seldom a straightforward task. Although we have presented the methodology as consecutive steps, it is often necessary in practice to iterate over these steps (i.e., 'bootstrap' from an initial guess to a final model). Often, the reason for such interplay between steps is the preponderance of gaps in our knowledge about the system we are modeling. Of course, one of the greatest benefits of a good simulation can be determining precisely where those gaps lie.

There are a number of ways in which detailed simulations constructed using this methodology can help fill these gaps. For one, such simulations can both fine-tune and test hypotheses about neural function. More importantly, they can help predict properties of systems based on partial information. For example, if we think a given system performs some function, we can use this methodology to make predictions about what distributions of neurons there should be and what kinds of dynamic properties they should have (see section 6.5). In addition, constructing models using control theory makes it possible to build in top-down constraints (e.g., stability) observed in the real system, giving insight into how those constraints might be met in neurobiological systems (see section 8.5). Notably, the effects of bottom-up constraints (e.g., cellular properties) can be studied at the same

Table 1.1
Summary of the methodology for generating neurobiological models.

Step 1	*System description*
	- Identify the relevant neurobiological properties (e.g., tuning curves, connectivity, etc.).
	- Specify the representations as variables (e.g., scalars, vectors, functions, etc.).
	- Provide a functional description including specification of subsystems and overall system architecture.
	- Provide a mathematical description of system function.
Step 2	*Design specification*
	- Specify the range, precision, and signal-to-noise ratio for each variable.
	- Specify the temporal and dynamic characteristics for each variable.
Step 3	*Implementation*
	- Determine the decoding rules for implementing the specified transformations.
	- Determine which parts of the model are to be simulated to which degrees of detail.
	- Perform numerical experiments using resulting simulation.

time, by varying the parameters of the single neuron model being used. As a result, this approach can serve to unify the often antagonistic top-down and bottom-up perspectives on how to best understand neurobiological systems.

The challenges that arise in applying this methodology can vary significantly from system to system. This will become apparent as we explore the numerous examples in the remainder of the book. The variation stems from many sources: different systems are of varying degrees of complexity; different systems have distinct dynamics and are thus more or less sensitive to different implementational constraints; and, perhaps most importantly, there are unequal amounts of neuroscientific detail about different systems. Nevertheless, the principles we have outlined in section 1.4 and the methodology we have outlined here prove to be useful tools for generating biologically plausible, yet computationally tractable models.

1.6 A POSSIBLE THEORY OF NEUROBIOLOGICAL SYSTEMS

To this point, we have presented what we have called a 'framework' that consists of a set of three principles and a corresponding methodology. We have chosen these terms because they are neutral regarding the scientific status of this approach. Nevertheless, in this section we explore the possibility that the three principles can be properly called a theory of neurobiological systems. Note that the practical utility of the framework itself is independent of whether this claim is found convincing.

Recently, a consensus has begun to develop about the state of theories in neuroscience; there aren't any. Or at least there aren't any good ones (Churchland and Sejnowski 1992;