

edited by Eric L. Schwartz

Computational Neuroscience

A Bradford Book
The MIT Press
Cambridge, Massachusetts
London, England

edited by Eric L. Schwartz

Computational Neuroscience

A Bradford Book
The MIT Press
Cambridge, Massachusetts
London, England

What Is Computational Neuroscience?

Patricia S. Churchland,
Christof Koch, and
Terrence J. Sejnowski

The expression "Computational neuroscience" reflects the possibility of generating theories of brain function in terms of the information-processing properties of structures that make up nervous systems. It implies that we ought to be able to exploit the conceptual and technical resources of computational research to help find explanations of how neural structures achieve their effects, what functions are executed by neural structures, and the nature of representation by states of the nervous system.

The expression also connotes the potential for theoretical progress in cooperative projects undertaken by neurobiologists and computer scientists. This collaborative possibility is crucial, for it appears that neither a purely bottom-up strategy nor a purely top-down strategy for explaining how the brain works is likely to be successful. With only marginal caricature, one can take the purely bottom-up strategy as recommending that higher-level functions can be neither addressed nor understood until all the fine-grained properties of each neuron and each synapse are understood. But if, as it is evident, some properties are *network* effects or *system* effects, and in that sense are *emergent* properties, they will need to be addressed by techniques appropriate for higher levels and described by theoretical categories suitable to that level. Assuming there are system properties or network properties that are not accessible at the single-unit level, then knowing all the fine-grained detail would still not suffice to explain how the brain works.

A purely top-down strategy is typified, again with minimal caricature, by its dismissal of the organization and structure of the nervous system as essentially irrelevant to determining the nature of cognition (Pylyshyn 1980; Fodor 1975). Advocates of this strategy prefer instead to find computational models that honor only (or at least primarily) psychological and computational constraints. One major reason for eyeing skeptically the purely top-down strategy is that computational space is consummately vast, and on their own, psychological and engineering constraints do not begin to narrow the search

space down to manageable proportions. Unless we go into the black box, we are unlikely to get very far in understanding the actual nature of fundamental cognitive capacities, such as learning, perceiving, orienting and moving in space-time, and planning.

There is an additional reason it may be inefficient to try to determine how the brain accomplishes some task, such as binocular depth perception, by taking a purely engineering approach: nervous systems are products of evolution. The brain's solutions to such problems may be radically different from what a clever engineer might design, at least for the reason that evolutionary changes are made within the context of a design and architecture that already is in place. Evolution cannot start from scratch, even when the optimal design would require that course. As Jacob (1982) has remarked, evolution is a tinkerer, and it fashions its modifications out of available materials, limited by earlier decisions. Moreover, any given capacity, such as binocular depth perception, is part of a much larger package subserving sensorimotor control and survival in general. From an engineering point of view, a design for an isolated stereoscopic device may look like a wonderfully smart design, but in fact it may not integrate at all well with the wider system and may be incompatible with the general design of the nervous system. For these and other reasons (Churchland 1986), neurobiological constraints have come to be recognized as essential to the project.

Although the name "computational neuroscience" may have emerged recently, the central motivation for the enterprise is by no means new. Even so, a great deal has happened between the publication of *Cybernetics* by Wiener in 1948 and the Computational Neuroscience meeting in Monterey in the spring of 1987. First, there has been a spectacular growth in anatomical and physiological information regarding nervous systems, and in techniques for extracting information. So remarkable and munificent have been the discoveries, that it has begun to seem that the time is ripe for generating empirically adequate computational theories at a variety of levels. No less dramatically, technological achievements in designing fast, powerful, and relatively inexpensive computing machines have made it possible to undertake simulation and modeling projects that were hitherto only pipe dreams. Finally, disappointment in conventional AI strategies (Good Old Fashioned AI, or GOFAI, as Haugeland [1985] calls it) for modeling cognition independently of neuro-

biological constraints has provoked theorists to reconsider what neuroscientists have said all along: the neuronal organization matters. Connectionists have begun to develop alternative strategies to GOFAI that are yielding models that are surprisingly powerful as well as more biologically plausible. (For example, Lehky and Sejnowski [1988] describe a neural network model that computes the shape of an object from its gray level intensities array; see also papers in Rumelhart and McClelland 1986 and Zipser and Andersen 1988.)

Is the Brain a Computer?

If we seek theories to explain in computational terms the function of some part of the nervous system—a network of neurons, or an individual neuron, or perhaps a system of networks—then that structure is seen as doing computation and hence as being a computer. But is this a justifiable perspective? If by "computer" we mean "serial digital computer," then the answer is No. For the analogy between the brain and a serial digital computer is an exceedingly poor one in most salient respects, and the failures of similarity between digital computers and nervous systems are striking. On the other hand, if we embrace a broader conception of computation (see below) then the answer is a definite Yes.

Of the dissimilarities between serial digital computers and nervous systems, perhaps the most important is this: nervous systems are not general purpose machines that are programmed up to be specialized machines; rather, they evolved to perform certain kinds of tasks. Unlike manufactured computers, nervous systems have plasticity—they grow, develop, learn, and change. Nervous systems have a profoundly different architecture from serial digital computers. In particular, they have parallel structure, and nervous systems appear to use a system of information storage that is very different from that employed in serial digital computers. In conventional silicon chips one node is connected, on average, to three to four other nodes, while by contrast one cortical neuron may receive input from hundreds or thousands of neurons and in turn project to thousands of other neurons in cortex. It is also important in this context to note that events in nervous systems happen in the millisecond (10^{-3} second) range, whereas events in silicon systems happen in the nanosecond (10^{-9} second) range. The individual com-

ponents making up the brain and those making up serial digital machines work on vastly different time scales. This makes it quite obvious that the parallel architecture is not a trivial but an absolutely critical aspect of computational design in nervous systems. Finally, much of the computing in nervous systems undoubtedly is not manipulation of symbols according to rules explicit in the system, in contrast to conventional AI programs run on a serial digital computer. (Rumelhart and McClelland 1986)

These dissimilarities do not imply that brains are not computers, but only that *brains are not serial digital computers*. Identifying computers with *serial digital computers* is neither justified nor edifying, and a more insightful strategy is to use a suitably broad notion of computation that takes the conventional digital computer as only a *special* instance, not as the *defining* archetype. (For a different view, see Pylyshyn 1984.)

It is useful, therefore, to class certain functions in the brain as computational because nervous systems *represent* and they respond on the basis of the representations. Nervous systems represent the external world, the body they inhabit, and, in some instances, parts of the nervous system itself. Nervous system effects can be seen as computations because at some levels the explanations of state transitions refer to abstract properties and the relations between abstract properties. That is, the explanations are not simple mechanical explanations that cite causal interactions between physically specified items; for example, the explanation of how a molecule is transported down an axon. Rather, the explanation describes the states in terms of the information transformed, represented, and stored.

The distinction between physical causation and computation also arises in other areas of biology. For example, the sequence of base pairs in DNA codes a wide variety of structures and functions, including the sequences of amino acids in proteins, regulatory signals, and developmental programs. We are still a long way from the complete account of how the information contained in the DNA guides regulation and development, but the activation and deactivation of different genes at different stages of development in different cells can be considered a very complex computational system whose goal is to produce a living creature.

In a most general sense, we can consider a physical system as a computational system just in case there is an appropriate (revealing) mapping between some algorithm

and associated physical variables. More exactly, a physical system computes a function $f(x)$ when there is (1) a mapping between the system's physical inputs and x , (2) a mapping between the system's physical outputs and y , such that (3) $f(x) = y$. This kind of arrangement is easily understood in a digital computer, though it is by no means confined to a machine of that configuration. A machine is taken to compute, for example, (PLUS 2,3) to give 5 as the output, by dint of the fact that its physical regularities are set up in such a way as to honor the abstract regularities in the domain of numbers. But that sort of property (physical states of a device mappable onto information-bearing states) can be achieved in many different kinds of architectural arrangements. Notice that although two pieces of wood sliding over each other may initially not be regarded as a computer, two pieces of wood with the proper representations is a slide rule, and a slide rule is a mechanical computer whose physical structure is set up so that certain mathematical relations are preserved. Given this basic account of computation, it is also useful to hypothesize what computations are performed by neurobiological components. For example, it appears that the neurons in the parietal cortex of primates compute head co-ordinates on the basis of retinal co-ordinates (Andersen and Mountcastle 1983; Zipser and Andersen, 1987, 1988), and that neurons in area VI of the visual cortex compute depth of a stimulus on the basis of binocular disparity (Poggio et al. 1985).

A point that is perhaps only implicit in the foregoing characterization of computation should now be emphasized. A physical system is considered a computer when its states can be taken as representing states of some other system; that is, so long as someone sees an interpretation of its states in terms of a given algorithm. Thus a central feature of this characterization is that whether something is a computer has an interest-relative component, in the sense that it depends on whether someone has an interest in the device's abstract properties and in interpreting its states as representing states of something else. Consequently, a computer is not a natural kind, in the way that, for example, an electron or a protein or a mammal is natural kind. For categories that do delimit natural kinds, experimentation is relevant to determining whether a given item really belongs to the category, and there are generalizations and laws (natural laws) about the items in the categories and theories embodying the laws. Non-natural kinds differ in all these

respects, and typically have an interest-relative dimension. For example, 'bee' is a natural kind, but 'gem', 'dirt', and 'weed' are not.

Stones are considered gems depending on whether some social group puts special value of them; plants are considered weeds, depending on whether a given gardener happens to like them in his garden. Some gardeners cultivate Baby's Breath as a desirable plant; other gardeners fight it as a weed. There is no experiment that will settle for us whether Baby's Breath really is a weed or not, because there is no fact of the matter, only social or idiosyncratic conventions. Nor are there interest-independent generalizations about all weeds—there is nothing common to plants called weeds save that some gardener has an interest in keeping them out of his garden. Similarly, there is no intrinsic property common to all computers, just the interest-relative property that someone sees a value in interpreting a system's states as representing states of some other system, and the properties of the system support such an interpretation. (For more on natural kinds, see P. M. Churchland 1985.) Desk-top von Neumann machines have become prototypical computers because they are so common, just as dandelions are prototypical weeds, but these prototypes should not be mistaken for the category itself.

It may be suggested as a criticism of this very general characterization of computation that it is too general. For in this very wide sense, even a sieve or a threshing machine could be considered a computer, since they sort their inputs into types, and in principle one could specify the sorting algorithm. While this observation is correct, it is not so much a criticism but an apt appreciation of the breadth of the notion of computation. Conceivably, sieves and threshing machines could be construed as computers, if anyone has reason to care about the specific algorithm reflected in their input-output functions, though it is hard to see what those reasons might be. Still, an appropriately shaped rock can function as a sundial, and though this is a very simple computer, we do have reason to care about the temporal states that its shadow-casting states can be interpreted as representing.

Nonetheless, there is perhaps a correct intuition behind the criticism that is this. Finding a device sufficiently interesting to warrant the description 'computer' probably also entails that its input-output function is rather complex and unobvious, so that discovering what the function is reveals something important and perhaps unexpected

about the real nature of the device and how it works. Thus finding out what is computed by a sieve is probably not very interesting, and will not teach us much we did not already know. In contrast, finding out what is computed by the cerebellum will teach us a lot about the nature of that tissue, and how it works.

Their simplicity notwithstanding, simple computational arrangements can lead to interesting insights. For example, assuming that we can landscape a terrain appropriately, a marble rolling down a hill into a valley can find (compute) the local minimum of a nonconvex two-dimensional cost function just as well as the steepest descent algorithm working on a digital computer, and probably faster. There is an analogy between this simple computer and Hopfield networks, parallel networks that store information at local minima of energy functions similar to these landscapes (Hopfield 1982). Global optimization problems can be solved by adding noise to the system, effectively "heating up" the marble so that it can jump out of a local minimum (Hinton and Sejnowski 1983). Applied to a problem in vision, for example, states of a Boltzmann machine correspond to a three-dimensional interpretation of a two-dimensional image, and states with the lowest energy correspond to the best interpretation (Kienker et al. 1986; Sejnowski et al. 1986)

Further, this analogy between a ball rolling down to find a local minimum of a landscape and computation of minima by analog networks has been the basis for a proposal that most problems in early vision are carried out by analog networks minimizing convex and non-convex cost functions (Poggio and Koch 1985). The input is given by currents injected into the nodes of a resistive network and the final solution is recovered by measuring the voltage at each node (Koch et al. 1986). In other words, Kirchhoff's voltage and current laws are exploited directly for the computation, in contrast to standard VLSI circuits. If we do not know the exact mapping between a problem and the physical domain, however, it may seem rather strange to classify a marble rolling down a hill or a current flowing in a wire as computing.

A distinguishing feature of certain computational systems is their ease of programmability, and electronic computers are justly esteemed for this virtue. While it is impossible to adapt the marble-on-the-landscape computer to three or higher dimensional optimization problems, this would be a simple task for an optimization program running on an electronic computer. On the other hand,

different initial conditions can be achieved very simply in the marble-on-the-landscape computer, which demonstrates that its flexibility lies in a different domain. Ease of programmability addresses only one practical aspect of the prototypical computer, and although this aspect is of great importance for many purposes, it is irrelevant to the fundamental theoretical question of what it is to be a computer.

Given this preamble on the nature of computation, the question now is whether it is appropriate to describe various structures in nervous systems as computing. The summary answer is that it certainly is. The reason is that we need a description of what various structures are doing that specifies the algorithm and the abstract properties that it maps. Ions flowing along concentration and electrical gradients compute just as surely as do electrons and holes in an MOS circuit. In fact, both obey rather similar equations. What we do not yet know in most cases of interest in neurobiology is the relationship between the problem domain, for instance computing binocular depth, and the appropriate biophysical variables, for instance average firing frequency, timing of individual action potentials, and the occurrence of bursts.

On this point there is of course a major contrast between manufactured and biological computers. Since we construct digital computers ourselves, we build the appropriate relationship into their design. Consequently, we tend to take this mapping for granted in computers generally, both manufactured and evolved. But for structures in the nervous system, these relationships have to be empirically discovered. In the case of biological computers this may turn out to be very difficult since we typically do not know what is being computed by a structure, and intuitive folk ideas may be very misleading.

The relationship between electronic computers and the neurobiological computers that can be simulated on electronic computers may be misunderstood. Although nervous systems appear to be very different from serial digital computers, once we discover what their computational principles are, the computations can be simulated on serial digital computers. This is a purely formal point that can be put another way: so long as there is a systematic way (a nonmagical way) that a network performs its tasks, then in principle that way can be captured by an algorithm that formally specifies the relations between input and output. Since it is an algorithm, it can run on computers with a variety of different archi-

tectures, though not necessarily with comparable speed, elegance, or equivalent procedures.

In principle, any algorithm can be run on a Turing machine, which is the dominant *formal* model of computation, not to be confused with an actual piece of electronic hardware. Apart from telling us that these algorithms can *in principle* be implemented on computers, the observation about Turing machines tell us little else, since the structure of the algorithm and the actual operations needed to carry out these computations in the nervous system remain to be discovered. What it does tell us is that the brain does not perform any magical operations when it computes depth or moves an arm.

It is important to emphasize that different architectures may be input-output equivalent, but nevertheless be radically different in procedures and in the speed of arriving at results. The matter of time is especially worth raising in a biological context, since for organisms making a living in a world "red in tooth and claw," timing is of the essence. The organism's nervous system cannot afford the luxury of taking several days or even several minutes to decide whether and where to flee from a predator. If serial digital computers running GOFAI programs to perform tasks such as motor control and visual recognition had to compete in the actual space-time world with actual fauna—even humble fauna—they would not stand a chance. There is enormous evolutionary pressure to come up with fast solutions to problems, and this is a constraint of the first importance as we try to understand the computational principles that govern a brain. Biologically speaking, a quick and dirty solution is better than a slow and perfect one.

Levels

Marr's Three Levels

Discussions concerning computational theories and models designed to explain some aspect of nervous system function invariably make reference to the notion of levels. A simple framework outlining a conception of levels was articulated by Marr and Poggio (1976) and Marr (1982), and provided an important and influential conceptual starting point for thinking about levels in the context of computation by nervous structures. Marr's ideas drew upon the conception of levels in computer science, and accordingly he characterized three levels: (1) the com-

computational level of *abstract problem analysis*, wherein the task (e.g., determining structure from motion) is decomposed into its main constituents; (2) the level of the algorithm, which *specified a formal procedure* by which, for a given input, the correct output would be given, and the task could thereby be performed; (3) the level of *physical implementation* of the computation.

A central element in Marr's view was that a higher level was largely independent of the levels below it, and hence computational problems of the highest level could be analyzed independently of understanding the algorithm that executes the computation. For the same reason the algorithmic problem of the second level was thought to be solvable independently of an understanding of the physical implementation. Thus his preferred strategy was "top-down" rather than "bottom-up." At least this was the official doctrine, though in practice, downward glances figured significantly in the attempts to find problem analyses and algorithm solutions. Ironically, given his advocacy of the top-down strategy, Marr's work was itself highly influenced by neurobiological considerations, and implementational facts constrained his choice of problem and nurtured his computational and algorithmic insights. Publicly, advocacy of the top-down strategy did carry the implication, dismaying for some and comforting for others, that in solving questions of brain function the neurobiological facts could be more or less ignored, since they were, after all, just at the implementation level.

Two very different issues tended to become conflated in the doctrine of independence. One concerns whether, as a *matter of discovery*, one can figure out the algorithm and the problem analysis independently of facts about implementation. The other concerns whether, as a matter of formal theory, a given algorithm that is already known to perform a task in a given machine (e.g., the brain) can be implemented in some other machine that has a distinct architecture. Answers to these two questions may well diverge, and, as we argue, the answer to the first is probably No, while the answer to the second is Yes. So far as the latter is concerned, what computational theory tells us is merely that formal procedures can be run on different machines; in that sense and that sense alone, the algorithm is independent of the implementation. The formal point is straightforward: since an algorithm is formal, no specific physical parameters (e.g., "vacuum tubes", "Ca⁺⁺") are part of the algorithm. That said, it is

important to see that the purely formal point cannot speak to the issue of how best to *discover* the algorithm in fact used by a given machine, nor how best to arrive at the neurobiologically adequate task analysis (P. M. Churchland 1982). Certainly it cannot tell us that the *discovery* of the algorithms used by nervous structures will be independent of a detailed understanding of the nervous system. Moreover, it does not tell us that any implementation is as good as any other. And it had better not, since as discussed above, different implementations display enormous differences in speed, size, efficiency, elegance, etc. The formal independence of algorithm from architecture is something we can exploit to build other machines once we know how the brain works, but it is no guide to discovery if we do not yet know how the brain works.

The issue concerning independence of levels in top-down strategies marks a substantial conceptual difference between Marr and the current generation of connectionists and computational neuroscientists. In contrast to the doctrine of independence of computation from implementation, current research suggests that considerations of implementation play a vital role in the kinds of algorithms that are devised and the kind of computational insights available to the scientist. Knowledge of brain architecture, so far from being irrelevant to the project, can be the essential basis and invaluable catalyst for devising likely and powerful algorithms—algorithms that have a reasonable shot at explaining how in fact the neurons do the job.

Consequently, if we consider the problem of *discovering* the computational principles of the nervous system, it is clear that analyzing the task and devising the algorithm are not independent of the neurobiological implementation. Without benefit of neurobiological constraints, we will undoubtedly find ourselves exploring some region of computational space utterly remote from the region inhabited by biological machines. Such exploration could be fun for its own sake, and may lead to technological innovations, but given the vastness of computational space, it stands a negligible chance of helping explain how nervous systems work. Levels are not independent, and there is much more coupling and intercalation than was previously appreciated. This interdependence should not, however, be considered a disadvantage, but an advantage, because it means we can avail ourselves of guidance from all levels.

Levels of Organization in the Brain

Three levels of Analysis: How Adequate?

In the previous section we provisionally acquiesced in a background presumption of Marr's framework. That presumption treated computation monolithically, as a single kind of level of analysis. In the same vein, the framework treats implementation and task description/decomposition each as a single level of analysis. The presumption should not pass unchallenged, however, for on a closer look, there are difficulties, and these difficulties bear on how we conceive of levels.

The difficulties center around the observation that when we measure Marr's three *levels of analysis* against *levels of organization* in the nervous system, the fit is poor and confusing (Churchland and Sejnowski 1988). To begin with, there is organized structure at different scales:

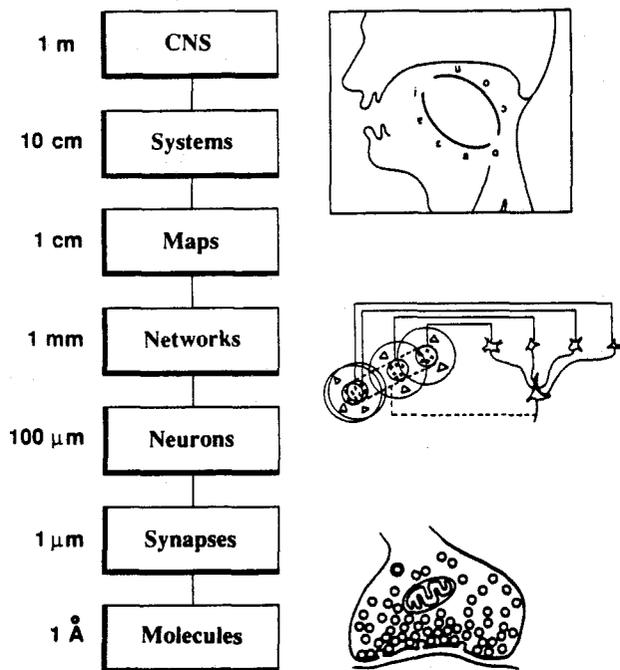


Figure 1. Structural levels of organization in the nervous system. The spatial scale at which anatomical organization can be identified varies over many orders of magnitude, from molecular dimensions to that of the entire central nervous system. The schematic diagrams on the right illustrate (bottom) a typical chemical synapse, (middle) a proposed circuit for generating oriented receptive fields in visual cortex, and (top) part of the motor system that controls the production of speech sounds.

molecules, membranes, synapses, neurons, nuclei, circuits, networks, layers, maps, and systems (figure 1). At each structurally specified stratum we can raise the computational question; what does that organization of elements do? What does it contribute to the wider, computational organization of the brain? In addition, there are physiological levels: ion movement, channel configurations, EPSPs, IPSPs, action potentials, evoked potential waves, behavior, and perhaps other intervening levels that we have yet to learn about and that involve effects at higher anatomical levels such as circuits or systems.

The range of structural organization implies, therefore, that there are many levels of implementation, and that each has its companion task description. But if there is a ramifying of task descriptions to match the ramified structural organization, this diversity will probably be reflected in the ramification of *algorithms* that characterize how a task is accomplished. This in turn means that the notion of *the* algorithmic level is as oversimplified as the notion of *the* implementation level.

Note also that the same level can be viewed computationally (in terms of functional role) or implementationally (in terms of the substrate on which the function runs), depending on what questions you ask, and on whether you look up or down. For example, an action potential, from the point of view of communication between neurons, might be considered an implementation, since the neurons really only care about the presence of a binary event. However, from a lower level—the point of view of ionic distribution—the action potential is computational, since it is the result of integrating many sources of information, and this information is carried by the train of action potentials. In the next section we shall explore in more detail the question of levels in the nervous system, and now this bears upon the project of discovering computational models to explain nervous system function.

Philosophical Observations on Levels of Organization

Which really are the levels relevant to explanation in the nervous system is an empirical, not an a priori, question. The answers to the question will slowly emerge as neuroscience and computational theories co-evolve, as theory and experiment converge on robust hypotheses. As things stand, there is a variety of ways of addressing the question of levels of organization.

The range of spatial scales over which the nervous system has been explored covers more than eight orders of magnitude, from molecular dimensions measured in angstroms to fiber tracts that span centimeters. Organizational principles emerge at each spatial level that are directly relevant to the function of the nervous system (figure 1). A few of these principles will be summarized here to serve as concrete examples.

Sensory information tends to be organized in spatial maps. The image of the world on the retina is mapped onto subcortical structures such as the superior colliculus and lateral geniculate nucleus, and from there to a proliferation of more than twenty maps in visual cortex. The generality of this principle throughout the somatosensory, auditory, and visual systems was noted by Adrian (1953), who suggested that even the olfactory system may be organized in this way. Often these maps are arranged in sheets, as in the superior colliculus where maps of the visual, auditory, and somatic spaces are stacked in adjacent layers such that stimuli from corresponding points in space lie above each other.

Topographic maps and laminae are special cases of a more general principle, which is the exploitation of geometry in the design of information processing systems (Mead 1987; Yeshurun and Schwartz, this volume). Spatial proximity may be an efficient way for biological systems to get together the information needed to solve rapidly difficult computational problems. For example, it is often important to compute the differences between similar features of a stimulus at nearby locations in space. By maintaining neighborhood relationships the total length of the connections needed to bring together the signals is minimized.

Most of our information about the representation of sensory information is based on recording from single neurons. This technique is revealing but it is also confining insofar as it biases us toward thinking about the cellular level rather than the subcellular or circuit levels. As we learn more about the complexity of subcellular structures, such as voltage-dependent processing in dendrites and spines (see the contributions of Koch and Poggio, Shepherd, Perkel, and Rall to this volume), our view of the computational capability of the neuron may change. We are especially in need of techniques that would allow us to monitor populations of neurons (Llinas 1985). Simulations of information processing in neural networks may

aid us in understanding this level (Sejnowski and Lehky 1988; Zipser and Andersen 1988).

Temporal Aspects of Levels

Nervous systems are dynamic, and physiological observations at each structural level can also be arranged in hierarchy of time scales. These scales reach from tens to hundreds of microseconds in the case of gating of single ionic channels to days or weeks for biophysical and biochemical events underlying memory, such as long-term potentiation. Action potentials and synaptic potentials of single neurons can be measured on a time scale of milliseconds using intracellular single-electrode recording. Recently, optical techniques have been used to record the activity in populations of neurons in cerebral cortex with a time resolution of about 100 msec and a spatial resolution of about 100 μm . The measurement of blood flow with PET scanning has been used to observe the average activity throughout the brain, averaged over minutes with 5 mm resolution. Even longer-term changes of behavior can be monitored and correlated with changes in the biochemical state of the nervous system. The range of time scales that can be studied varies over 12 orders of magnitude from microseconds to days.

The Status of Levels

Each of the structural levels—molecules, membranes, synapses, neurons, nuclei, circuits, layers, columns, maps, and systems—is separable conceptually but not detachable physically. What is picked out as a level is actually a boundary imposed on the structure we study using such techniques as are available to us, in order to try to understand the phenomena at hand. In the brain, they are all part of one, integrated, unified biological machine. That is, the function of a neuron depends on the synapses that bring it information, and, in turn, the neuron processes information by virtue of its interaction with other neurons in local circuits, which themselves play a particular role by virtue of their place in the overall geometry of the brain. This is a cautionary, if obvious, observation, generated by the appreciation that in a sense the postulation of levels is artificial from the point of view of the functioning brain, even though it may be scientifically indispensable if we are to make progress. And until we understand how the brain works, it is always an open question whether we have put our boundaries in the most revealing place.

Concluding Remarks

A scientific field is defined primarily by its problem space and its successful large-scale theories. Until there are such theories in computational neuroscience, the field is defined mostly by the problems it seeks to solve, and the general methodology and specific techniques it hopes will yield successful theories. Accordingly, within the framework outlined here, we can say quite simply that the ultimate aim of computational neuroscience is to explain how the brain works. Since nervous systems have many levels of organization, theories appropriate to these levels will have to be generated, and the theories should integrate, level by level, into a coherent whole.

No single neural model can be expected to span all the levels cited, and an essential feature at one level of organization may be an insignificant detail at another. For example, the identity of a neurotransmitter is an essential feature in studying receptor binding, less important when studying circuits, and of marginal significance at the systems level. The multiplicity of levels of organization is a feature not only of neuroscience, but also of physics and chemistry, where explanations of phenomena on distinct levels of organization are more developed. Assuming a comprehensive "theory of the brain" does emerge, it will involve establishing a successive and overlapping chain of reasoning from the lowest levels to the highest, encompassing the various spatial, temporal, structural, and computational levels. Only then will we have achieved our ultimate aim.

It is rare for a model to jump over many levels, and the most successful models typically link only neighboring levels. For example, the Hodgkin-Huxley ionic model of the initiation and propagation of action potentials related macroscopic current measurements to the microscopic kinetics of ion channels, although such channels were only hypothetical at that time. This example also demonstrates that it may be necessary to make assumptions beyond the data available at one level in order to reach a better understanding at that level. It is only with the advent of single channel patch clamp recording techniques developed decades after the Hodgkin-Huxley model that their assumptions regarding the nature of microscopic ion channels could be verified.

As knowledge accumulates at the cellular and molecular levels, it is tempting to incorporate all that is known into

a model that aims to reproduce as much of the nervous system as possible from the bottom up. The problem with this approach is that a genuinely perfect model, faithful in every detail, is likely to be as incomprehensible as the nervous system itself. Although such a model may mimic the brain, it is debatable whether any true understanding has been achieved. However, as noted above, the exclusively top-down approach has its own special problems.

At this stage in our understanding of the brain, it may be fruitful to concentrate on models that suggest new and promising lines of experimentation, at all levels of organization. In this spirit, a model should be considered a provisional framework for organizing possible ways of thinking about the nervous system. The model may not be able to generate a full range of predictions owing to incompleteness; some assumptions may be unrealistic simplifications, and some details may even be demonstrably wrong. Nevertheless, as long as the model captures some useful kernel of truth that leads to new ways of thinking and to productive experiments, the model will have served its purpose, suggesting an improved model even as it is itself disproved. A corollary to this is that a very elaborate and sophisticated model that does not translate well into an experimental program is a sterile exercise when compared to something rougher that leads to productive research directions.

The convergence of research in computer science, neuroscience, psychophysics, and cognitive psychology makes these truly exciting times. The hope that some of the major mysteries of mind-brain function will finally be explained has become less pie-in-the-sky romanticism and more a palpable project that can be actively and experimentally pursued. Even so, the brain is awesomely complex; experiments are typically very difficult to do; there are ethical barriers to studying the normal human brain, and there may be nontrivial individual differences at many levels of organization. The answers may well elude us for some time to come.

References

- Adrian, E. D. (1953) The mechanism of olfactory stimulation in the mammal. *Adv. Sci. (London)* 9: 417-420.
- Andersen, R. A., and V. B. Mountcastle (1983) The influence of the angle of gaze upon the excitability of the light-sensitive neurons of the posterior parietal cortex. *J. Neurosci.* 3: 532-548.
- Churchland, P. M. (1982) "Is thinker a natural kind?" *Dialogue* 21: 223-238.
- Churchland, P. M. (1985) Conceptual progress and word/world relations: In search of the essence of natural kinds. *Can. J. of Philos.* 15: 1-17.
- Churchland, P. S. (1986) *Neurophilosophy: Toward a Unified Science of the Mind-Brain*. Cambridge, MIT Press.
- Churchland, P. S., and T. J. Sejnowski (1988) Neural representations and neural computations. In: *Biological Computation*, ed. L. Nadel. Cambridge, MIT Press.
- Fodor, J. A. (1975) *The Language of Thought*. New York: Crowell. (Paperback edition, 1979, Cambridge, Harvard University Press.)
- Haugeland, J. (1985) *Artificial Intelligence: The Very Idea*. Cambridge, MIT Press.
- Hinton, G. E., and T. J. Sejnowski (1983) Optimal perceptual inference. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- Hopfield, J. J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci.* 79: 2554-2558.
- Jacob, F. (1982) *The Possible and the Actual*. Seattle, University of Washington Press.
- Kienker, P. K., T. J. Sejnowski, G. E., Hinton, and L. E. Schumacher (1986) Separating figure from ground with a parallel network. *Perception* 15: 197-216.
- Koch, C., J. Marroquin, and A. Yuille (1986) Analog neuronal networks for early vision. *Proc. Acad. Natl. Sci.* 83: 4263-4267.
- Lehky, S., and T. J. Sejnowski (1988) Network model of shape-from-shading: Neural function arises from both receptive and projective fields. *Nature* 333: 452-454.
- Lainas R. R. (1985) Electrotonic transmission in the mammalian central nervous system. In *Gap Junctions*, ed. Michael E. L. Bennett and David C. Spray. Cold Spring Harbor Laboratory.
- Marr, D. (1982) *Vision*. San Francisco, Freeman.
- Marr, D., and T. Poggio (1976) Cooperative computation of stereo disparity. *Science* 194: 283-287.
- Mead, C. (1987) *Analog VLSI and Neural Systems*. Reading, Mass., Addison-Wesley.
- Poggio, G., B. C. Motter, S. Squatrito, and Y. Trotter (1985) Responses of neurons in visual cortex (V1 and V2) of the alert macaque to dynamic random-dot stereograms. *Vision Res.* 25: 397-406.
- Poggio, T., and C. Koch (1985) Ill-posed problems in early vision: from computational theory to analogue networks. *Proc. R. Soc. Lond. B* 226: 393-323.
- Pylyshyn, Z. W. (1984) *Computation and Cognition: Toward a Foundation for Cognitive Science*. Cambridge, MIT Press.
- Rumelhart, D. E., and McClelland, J. L. (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MIT Press.
- Sejnowski, T. J., P. K. Kienker, and G. E. Hinton (1986) Learning symmetry groups with hidden units: Beyond the perceptron. *Physica* 22D: 260-275.
- Zipser, D., and R. Andersen (1988) A backpropagation network that simulates response properties of a subset of posterior parietal neurons. *Nature* 331: 629-684.