Computing with Neural Circuits: A Model

Author(s): John J. Hopfield and David W. Tank

Source: *Science*, Aug. 8, 1986, New Series, Vol. 233, No. 4764 (Aug. 8, 1986), pp. 625-633

Published by: American Association for the Advancement of Science

Stable URL: https://www.jstor.org/stable/1698010

Computing with Neural Circuits: A Model

JOHN J. HOPFIELD AND DAVID W. TANK

A new conceptual framework and a minimization principle together provide an understanding of computation in model neural circuits. The circuits consist of nonlinear graded-response model neurons organized into networks with effectively symmetric synaptic connections. The neurons represent an approximation to biological neurons in which a simplified set of important computational properties is retained. Complex circuits solving problems similar to those essential in biology can be analyzed and understood without the need to follow the circuit dynamics in detail. Implementation of the model with electronic devices will provide a class of electronic circuits of novel form and function.

A COMPLETE UNDERSTANDING OF HOW A NERVOUS SYSTEM computes requires comprehension at several different levels. Marr (*1*) noted that the computational problem the system is attempting to solve (the problem of stereopsis in vision, for example) must be characterized. An understanding at this level requires determining the input data, the solution, and the transformations necessary to compute the desired solution from the input. The goal of computational neurobiology is to understand what these transformations are and how they take place. Intermediate computational results are represented in a pattern of neural activity. These representations are a second, and system-specific, level of understanding. It is important to understand how algorithms—transformations between representations—can be carried out by neural hardware. This understanding requires that one comprehend how the properties of individual neurons, their synaptic connections, and the dynamics of a neural circuit result in the implementation of a particular algorithm. Recent theoretical and experimental work attempting to model computation in neural circuits has provided insight into how algorithms can be implemented. Here we define and review one class of network models—nonlinear graded-response neurons organized into networks with effectively symmetric synaptic connections—and illustrate how they can implement algorithms for an interesting class of problems (*2*).

Early attempts to understand biological computation were stimulated by McCulloch and Pitts, who described (*3*) a "logical calculus of the ideas immanent in nervous activity." In these early theoretical studies, biological neurons were modeled as logical decision elements described by a two-valued state variable (on-off), which were organized into logical decision networks that could compute simple Boolean functions. The timing of the logical operations was controlled by a system clock. In studies of the "perceptron" by Rosenblatt (*4*), simple pattern recognition problems were solved by logical decision networks that used a system of feed-forward synaptic connectivity and a simple learning algorithm. Several reviews of McCulloch and Pitts and perceptron work are available (*5*). More recent studies have used model neurons having less contrived properties, with continuous dynamics and without the computerlike clocked dynamics. For example, Hartline *et al.* (*6*) showed that simple linear models with continuous variables could explain how lateral inhibition between adjacent photoreceptor cells enhanced the detection of edges in the compound eye of *Limulus*. Continuous variables and dynamics have been widely used in simulating membrane currents and synaptic integration in single neurons (*7*) and in simulating biological circuits, including central pattern generators (*8*) and cortical structures (*9*). Both two-state (*10, 11*) and continuous-valued nonlinear models (*12*) have been extensively studied in networks organized to implement algorithms for associative memories and associative tasks (*13*).

The recent work being reviewed here has been directed toward an understanding of how particular computations can be performed by selecting appropriate patterns of synaptic connectivity in a simple dynamical model system. Circuits can be designed to provide solutions to a rich repertoire of problems. Early work (*10*) was designed to examine the computational power of a model system of two-state neurons operating with organized symmetric connections. The inclusion of feedback connectivity in these networks distinguished them from perceptron-like networks, which emphasized feed-forward connectivity. Graded-response neurons described by continuous dynamics were combined with the synaptic organization described by earlier work to generate a more biologically accurate model (*14*) whose computational properties include those of the earlier model. General principles for designing circuits to solve specific optimization problems were subsequently developed (*15–17*). These networks demonstrated the power and speed of circuits that were based on the graded-response model. Unexpectedly, new computational properties resulted (*15*) from the use of nonlinear graded-response neurons instead of the two-state neurons of the earlier models. The problems that could be posed and solved on these neural circuits included signal decision and decoding problems, pattern recognition problems, and other optimization problems having combinatorial complexity (*15–20*).

One lesson learned from the study of these model circuits is that a detailed description of synaptic connectivity or a random sampling of neural activity is generally insufficient to determine how the circuit computes and what it is computing. As an introduction to the circuits we review, this analysis problem is illustrated on a simple and well-understood model neural circuit. We next define and discuss the simple dynamical model system and the underlying assumptions and simplifications that relate this model to biological neural circuits. A conceptual framework and minimization principle applied to the model provide an understanding of how these circuits compute, specifically, how they compute solutions to optimization problems. The design and architecture of circuits for two specific problems are presented, including the formerly enigmatic circuit used earlier to illustrate the analysis problem.

J. J. Hopfield is with the Divisions of Chemistry and Biology, California Institute of Technology, Pasadena, CA 91125. D. W. Tank and J. J. Hopfield are with the Molecular Biophysics Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974.

Understanding Computation in a Simple Neural Circuit

Let us analyze the hypothetical neural circuit shown in Fig. 1 with simulation experiments based on the tools and methods of neurophysiology and anatomy. The analysis will show that the usual available neurobiological measures and descriptions are insufficient to explain how even small circuits of modest complexity compute. The seven-neuron circuit in Fig. 1 is designed to compute in a specific way that will later be described. From a neurobiological viewpoint, the basic anatomy of the circuit contains four principal neurons (*21*), identified in the drawing as P_0, P_1, P_2, and P_3. Each neuron has an axon leaving the circuit near the bottom of the figure. The computational results of the circuit must be evident in the activity of these neurons. The one input pathway, from a neuron external to the circuit, is provided by axon Q. Neurons IN_1, IN_2, and IN_3 are intrinsic interneurons in the circuit.

In attempting to understand the circuit's operation, we simultaneously monitor the activity (computer simulated) in each of the seven neurons while providing for a controllable level of impulse activity in the input axon Q. Results from this experiment on the hypothetical circuit for several fixed levels of input activity are shown in Fig. 2A. The top trace represents our controlled activity in Q. In each time epoch this activity is progressively larger, as illustrated by the increasing number of action potentials per unit time. Although the activity of IN_3 is steadily rising as the activity in Q increases, the activities of the other neurons in the circuit are not simply related to this input. From these results we know what the output patterns of activity on the principal neurons are for specific levels of impulse activity on the input axon Q, but we cannot explain what computation the circuit is computing. Furthermore, we do not know how the structure and organization of the circuit has provided these particular patterns of neural activities for the different input intensities.

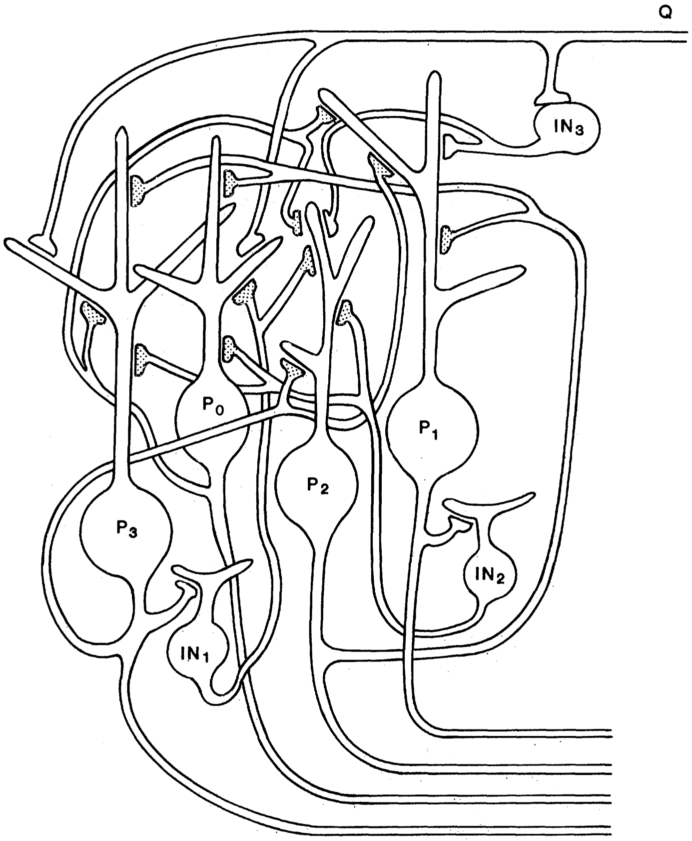

Fig. 1. "Anatomy" of a simple model neural circuit. Input axon Q has excitatory synapses (direct or effective) on each of the principal neurons P_0 through P_3. Each of these principal neurons has inhibitory synapses (direct or indirect) with all other principal neurons. Inhibitory synapses are shaded. IN_1 to IN_3, intrinsic interneurons.

Study of the synaptic organization of the connections between the neurons by electrophysiological or ultrastructural techniques could provide the numerical description of synaptic strengths shown in Table 1. The results of these experiments would show that each individual principal neuron P_i inhibits the other three principal neurons (P_j). There is either monosynaptic inhibition from P_i to P_j or polysynaptic inhibition by an excitatory synapse from P_i to an interneuron (IN_k), which then forms an inhibitory synapse with P_j (for example, the P_1-to-P_2 pathway in Fig. 1). This synaptic organization provides an "effective" inhibitory synapse between any two principal neurons; an action potential elicited in one principal neuron always contributes to inhibition of each of the others. Similar experiments measuring the strengths of the synaptic connections between the input axon Q and the P_i would show effective excitatory connections (Table 1). While the organization between principal neurons could be described classically as "lateral inhibition," the output patterns of activity in the P_i, shown in Fig. 2A for different input intensities, cannot be explained by this qualitative description.

Given the synaptic strengths in Table 1 and an appropriate mathematical description of the neurons, we can simulate the model neural circuit and produce the output activity patterns for the different inputs. Such detailed simulations can also be done for real neural circuits if the required parameters are known. In general, an ability to correctly predict a complex result that relies solely on simulation of the system provides a test of the simulation model, but does not provide an understanding of the result. Thus, despite our classical analysis of the simple neural circuit in Fig. 1, we still have no understanding of *why* these particular synaptic strengths (Table 1) provide these particular relations between input and output activity. Computation in the circuit shown in Fig. 1 can, however, be defined and understood within the conceptual framework provided by an analysis of dynamics in the simple neural circuit model we now discuss.

The Model Circuits and Their Relation to Biology

Neurons are continuous, dynamical systems, and neuron models must be able to describe smooth, continuous quantities such as graded transmitter release and time-averaged pulse intensity. In McCulloch-Pitts models, neurons were logical decision elements described by a two-valued state variable (on-off) and received synaptic input from a small number of other neurons. In general, McCulloch-Pitts models do not capture two important aspects of biological neurons and circuits: analog processing and high interconnectivity. While avoiding these limitations, we still want to model individual neurons simply. In the absence of appropriate simplifications, the complexities of the individual neurons will loom so large that it will be impossible to see the effects of organized synaptic interactions. A simplified model must describe a neuron's effective output, input, internal state, and the relation between its input and output.

In the face of the staggering diversity of neuronal properties, the goal of compressing their complicated characteristics is especially difficult. For the present, let us consider a prototypical biological neuron having inputs onto its dendritic arborization from other neurons and outputs to other neurons from synapses on its axon.

Action potentials initiate near the soma and propagate along the axon, activating synapses. Although we *could* model the detailed synaptic, integrative, and spike-initiating biophysics of this neuron, following, for example, the ideas of Rall (*7*), the first simplification we make in our description of the neuron is to neglect electrical effects attributable to the shape of dendrites and axon. (The axon and dendrite space-constants are assumed to be very large.) Our model neuron has the capacitances and conductances of the arborization added directly to those of the soma. The input currents from all synaptic channels are simply additive; more complex interactions between input currents are ignored. Membrane potential changes are assumed to arrive at the presynaptic side of synapses at the same time as they are initated at the soma. The second simplification is to deal only with "fast" synaptic events. When a potential fluctuation occurs in the presynaptic terminal of a chemical synapse, a change in the concentration of neurotransmitter is followed (with a slight delay) by a current in the postsynaptic cell. In our model neurons we presume this delay is much shorter than the membrane time constant of the neuron.

These two suppositions on time scale mean that when a change in potential is initiated at the soma of cell j, it introduces an effectively instantaneous conductance change in a postsynaptic cell i. The amount of the conductance change depends on the nature and strength of the synapse from cell j to cell i.

Biological neurons that produce action potentials do so (in steady state) at a rate determined by the net synaptic input current. This current acts indirectly by charging the soma and changing the cell potential. A characteristic charging or discharging time constant is determined by the cell capacitance C and membrane resistance R. The input current is "integrated" by the cell RC time constant to determine a value of an effective "input-potential," u. Conceptually, this potential u is the cell membrane potential after deletion of the action potentials. Action potentials (and postsynaptic responses in follower cells) are then generated at a rate dependent on the value of u. Dependencies of firing rates on input currents (and hence u) vary greatly, but have a generally sigmoid and monotonic form (Fig. 3A), rising continuously between zero and some maximum value (*22*). The firing rate of cell i can be described by the function $f_i(u_i)$. For processing in which individual action potentials are not synchronized or highly significant, a model that suppresses the details of action potentials should be adequate. In such a limiting case, two variables describe the state of neuron i: the effective input potential u_i and the output firing rate $f_i(u_i)$. The strength of the synaptic current into a postsynaptic neuron j due to a presynaptic neuron i is proportional to the product of the presynaptic cell's output $[f_i(u_i)]$ and the strength of the synapse from i to j. In our model, the strength of this synapse is represented by the parameter T_{ij}, so that the postsynaptic current is given by $T_{ij} f_j(u_j)$. The net result of our description is that action potentials have their effects represented by continuous variables, just as the usual equations describing the behavior of electrical circuits replace discrete electrons by continuous charge and current variables.

Table 1. Effective synaptic strengths for the circuit in Fig. 1.

Post-synaptic neuron	Presynaptic neuron				
	P_0	P_1	P_2	P_3	Q
P_0		−2	−4	−8	+1
P_1	−2		−8	−16	+2
P_2	−4	−8		−32	+4
P_3	−8	−16	−32		+8

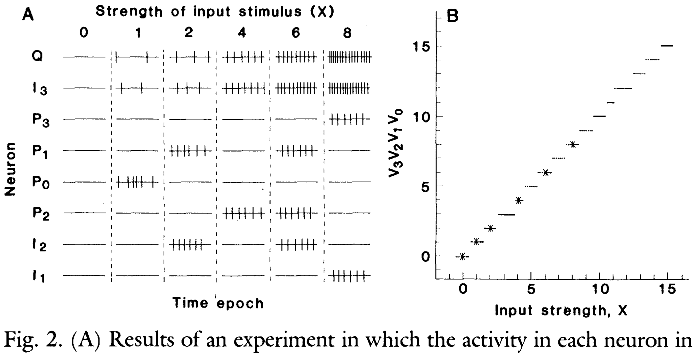

Fig. 2. (A) Results of an experiment in which the activity in each neuron in the circuit of Fig. 1 was simultaneously recorded (by simulation) as a function of the strength of the input stimulus on axon Q. The strength of the input stimulus is indicated by the numbers above each time epoch. (B) A selective rearrangement of the data in (A) illustrating the analog-binary computation being performed by the circuit. The digital word $V_3V_2V_1V_0$ is calculated from the records.

Many neurons, both central and peripheral, show a graded response and do not normally produce action potentials (*23*). The presynaptic terminals of these graded-response neurons secrete neurotransmitters, and hence induce postsynaptic currents, at a rate dependent on the presynaptic cell potential. The effective output of such cells is also a monotonic sigmoid function of the net synaptic input. Thus the model treats both neurons with graded responses and those exhibiting action potentials with the same mathematics.

We can now describe the dynamics of an interacting system of N neurons. The following set of coupled nonlinear differential equations results from our simplifications and describes how the state variables of the neurons (u_i; $i = 1,\ldots,N$) will change with time under the influence of synaptic currents from other neurons in the circuit.

$$C_i\frac{du_i}{dt} = \sum_{j=1}^{N} T_{i,j}\, f_j(u_j) - \frac{u_i}{R_i} + I_i \qquad (i = 1,\ldots,N) \qquad (1)$$

These equations might be thought of as a description of "classical" neurodynamics (*12, 14*). They express the net input current charging the input capacitance C_i of neuron i to potential u_i as the sum of three sources: (i) postsynaptic currents induced in i by presynaptic activity in neuron j, (ii) leakage current due to the finite input resistance R_i of neuron i, and (iii) input currents I_i from other neurons external to the circuit. The time evolution of any hypothetical circuit, defined by specific values of T_{ij}, I_i, f_i, C_i, and R_i, can be simulated by numerical integration of these equations.

Some intuitive feeling for how a model neural circuit might behave can be provided by considering the electrical circuit shown in Fig. 3B, which obeys the same differential equation (Eq. 1). The "neurons" consist of amplifiers in conjunction with feedback circuits composed of wires, resistors, and capacitors organized to represent axons, dendritic arborization, and synapses connecting the neurons. The firing rate function of our model neurons $[f_i(u_i)]$ is replaced in the circuit by the output voltage V_i of amplifier i. This output is $V_i = V_i^{\max}\, g_i(u_i)$, where the dimensionless function $g_i(u_i)$ has the same sigmoid monotonic shape (Fig. 3A) as $f_i(u_i)$ and a maximum value of 1. $V_i^{\max}$ is the electrical circuit equivalent of the maximum firing rate of cell i. The input impedance of our model neuron is represented in the circuit by an equivalent resistor ρ_j and an input capacitor C_j connected in parallel from the amplifier input to ground. These components define the time constants of the neurons and provide for the integrative analog summation of the synaptic input currents from other neurons in the network. To provide for both excitatory and inhibitory synaptic connections between neurons while using conventional electrical components, each amplifier is given two outputs—a normal (+) output and an inverted (−) output of the same magnitude but opposite in sign. A synapse between two neurons is defined by a conductance T_{ij}, which connects one of the two outputs of amplifier j to the input of amplifier i. This connection is made with a resistor of value $R_{ij} = 1/|T_{ij}|$. If the synapse is excitatory ($T_{ij} > 0$), this resistor is connected to the normal (+) output of amplifier j. For an inhibitory synapse ($T_{ij} < 0$), it is connected to the inverted (−) output of amplifier j. Thus, the normal and inverted outputs for each neuron allow for the construction of both excitatory and inhibitory connections through the use of normal (positive valued) resistors. The circuits include a wire providing an externally supplied input current I_i for each neuron (Fig. 3B). These inputs can be used to set the general level of excitability of the network through constant biases, which effectively shift the input-output relation along the u_i axis, or to provide direct parallel inputs to drive specific neurons. As in Eq. 1, the net input current to any neuron is the sum of the synaptic currents (flowing through the set of resistors connecting its input to the outputs of the other neurons), externally provided currents, and leakage current.

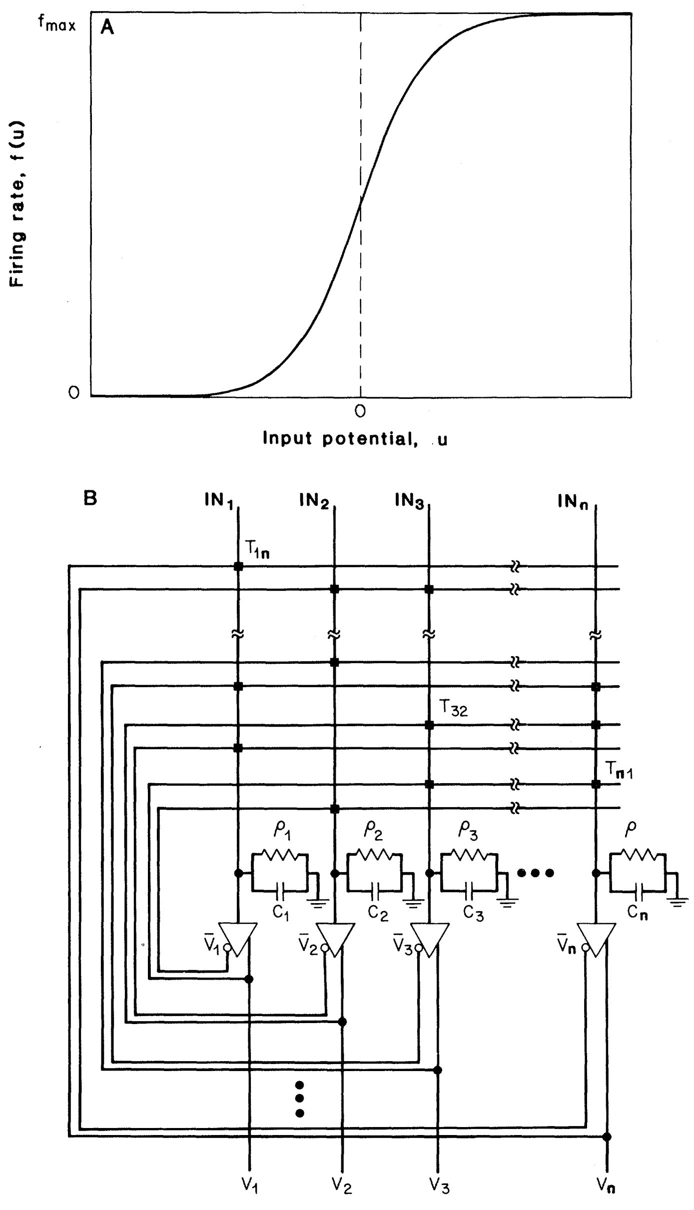

Fig. 3. (A) The sigmoid monotonic input-output relation used for the model neurons. (B) The model neural circuit in electrical components. The output of any neuron can potentially be connected to the input of any other neuron. Black squares at intersections represent resistive connections (with conductance T_{ij}) between outputs and inputs. Connections between inverted outputs (represented by the circles on the amplifiers) and inputs represent negative (inhibitory) connections.

In the model represented by Eq. 1 and Fig. 3, the properties of individual model neurons have been oversimplified, in comparison with biological neurons, to obtain a simple system and set of equations. However, essential features that have been retained include the idea of a neuron as transducer of input to output, with a smooth sigmoid response up to a maximum level of output; the integrative behavior of the cell membrane; large numbers of excitatory and inhibitory connections; the reentrant or fedback nature of the connections; and the ability to work with both graded-response neurons and neurons that produce action potentials. None of these features was the *result* of approximations. Their inclusion in a simplified model emphasizes features of the biological system we believe important for computation. The model retains the two important aspects for computation: dynamics and nonlinearity.

The model of Eq. 1 and Fig. 3 has immense computing power, achieved through organized synaptic interactions between the neurons. The model neurons lack many complex features that give biological neurons, taken individually, greater computational capabilities. It seems an appropriate model for the study of how the cooperative effects of neuronal interactions can achieve computational power.

A New Concept for Understanding the Dynamics of Neural Circuitry

A specific circuit of the general form described by Eq. 1 and Fig. 3 is defined by the values of the synapses (T_{ij}) and input currents (I_i). Given this architecture, the state of the system of neurons is defined by the values of the outputs V_i (or, equivalently, the inputs u_i) of each neuron. The circuit computes by changing this state with time. In a geometric space with a Cartesian axis for each neural output V_i, the instantaneous state of the system is represented by a point. A given circuit has dynamics that can be pictured as a time history or motion in this state space. For a circuit having arbitrarily chosen values for the synaptic connections, these motions can be very complex, and no simplifying description has been found. A broad class of simplified circuits, however, has a unifying principle of behavior while remaining capable of powerful computation. These circuits are literally or effectively symmetric.

A symmetric circuit is defined as having synaptic strength and sign (excitation or inhibition) of the connection from neuron i to j the same as from j to i. The two neurons need not, however, have the same input-output relation, threshold, or capacitance. Our model circuit (Fig. 3B) is symmetric if, for all i and j, T_{ij} is equal to T_{ji}. This symmetry refers only to connections between neurons in the circuit. It specifically excludes the input connections (represented in Fig. 3B as the input currents I_i) and any output connections from the circuit.

Symmetry of the connections results in a powerful theorem about the behavior of the system. The only additional conditions necessary are that the input-output relation of the model neurons be monotonic and bounded and that the external inputs I_i (if any) should change only slowly over the time of the computation. The theorem shows that a mathematical quantity E, which might be thought of as the "computational energy," decreases during the change in neural state with time described by Eq. 1. Started in any initial state, the system will move in a generally "downhill" direction of the E function, reach a state in which E is a local minimum, and stop changing with time. The system cannot oscillate. This concept can be illustrated graphically by a flow map in a state-space diagram. Each line corresponds to a possible time-history of the system, with

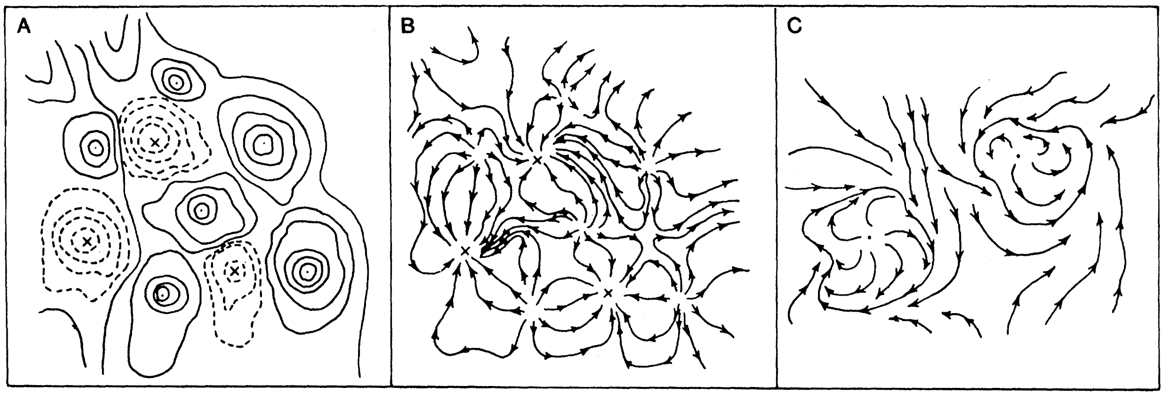

Fig. 4. (A) Energy-terrain contour map for the flow map shown in (B). (B) Typical flow map of neural dynamics for the circuit of Fig. 3 for symmetric connections ($T_{ij} = T_{ji}$). (C) More complicated dynamics that can occur for unrestricted (T_{ij}). Limit cycles are possible.

arrows showing the direction of motion. The structure imposed on the flow map for a circuit with symmetry is illustrated for a two-dimensional state space in Fig. 4. With symmetric connections, the flow map of the neural dynamics resembles Fig. 4B. Such a flow, in which each trajectory goes to stable points and stops, results from always going "downhill" on an "energy-terrain," coming to the bottom of a local valley, and stopping. The contour map of an E function that matches the flow in Fig. 4B is shown in Fig. 4A; it shows separated hills and valleys. The valleys are located where the trajectories in Fig. 4B stop. For a nonsymmetric circuit, the complications illustrated in the flow map in Fig. 4C can occur. This flow map has trajectories corresponding to complicated oscillatory behaviors. Such trajectories are undoubtedly important in neural computations, but as yet we lack the mathematical tools to manipulate and understand them at a computational level. The motion of a neural circuit comprising N neurons must be pictured in a space of N dimensions rather than the two dimensions of Fig. 4, but the qualitative picture of the effects of symmetric synaptic strengths is exactly the same.

The computational energy is a global quantity not felt by an individual neuron. The states of individual neurons simply obey the neural equations of motion (Eq. 1). The computational energy is our way of understanding why the system behaves as it does. A similar situation occurs in the concept of entropy in a simple gas. We understand that when a nonequilibrium state is set up with all the air molecules in one corner of the room, a uniform distribution will rapidly result. We explain that fact by the tendency of the entropy of isolated systems to increase whenever possible, but the individual molecules know nothing of entropy. They simply follow their Newtonian equations of motion.

Symmetric chemical synapses are observed in neural systems (*24*). Nonrectifying electrical synapses are intrinsically symmetric synapses of positive sign (*25*). Lateral inhibition in the visual system of *Limulus* is implemented with symmetric inhibitory synapses (*6*). An asymmetric network can also behave as though it were symmetric. In the olfactory bulb, the local circuit of mitral cell to granule cell to mitral cell provides an equivalent symmetric inhibitory connection between the pair of mitral cells (*26*). A similar situation occurs in the circuit shown in Fig. 1, where a direct equivalence between a neural circuit which is manifestly not symmetric and one which is effectively symmetric can be made if the inhibitory interneurons (IN_1, IN_2) are faster than other neurons.

The requirement of symmetry for this theorem can also be weakened. We have proven stability for a wide class of circuits having organized asymmetry between two sets of neurons with different time constants (*16*). (A neurobiological example would be the existence, in mammalian systems, of fast inhibitory interneurons that could provide effective symmetric inhibitory connections between neurons that are otherwise excitatory.) In one potentially useful example (*16*), stability could be guaranteed even though the sign of T_{ij} was always opposite that of T_{ji}. Also, there is a family of transformations by which a broader class of synaptic organizations can be made equivalent to symmetric ones (*27*). From an empirical viewpoint, moderate disorganized asymmetry (for example, having a random set of connections missing in an otherwise symmetric associative memory circuit) has little experimental effect on dynamic stability (*28*). Because the general features of symmetric circuits persist in circuits that are only equivalently symmetric, and real neural circuits can often be so viewed (except for inputs and outputs), the behavior of symmetric circuit models should be of direct use in trying to understand how neural computation is done in biology.

In general, systems having organized asymmetry can exhibit oscillation and chaos (*29*). In some neural systems like central pattern generators (*8*), coordinated oscillation *is* the desired computation of the circuit. Processing in the olfactory bulb also seems to make explicit use of oscillatory patterns (*30*). In such a case, proper combinations of symmetric synapses can enforce chosen phase relationships between different oscillators, an effect similar to those presented above.

Hard Problems Naturally Solved by Model Neural Circuits

In thinking about how difficult computational problems can be done on such networks, it is useful to recall the simple problem of associative memory, which these networks implement in a "natural" fashion (*10, 13*). This naturalness has two aspects. (i) The symmetry of the networks is natural because, in simple associations, if A is associated with B, B is symmetrically associated with A. (ii) If the desired memories can be made the stable states of a network, the desired computation (given partial information as input, find the memory that most resembles it) can be directly visualized as a motion toward the nearest stable state whose position is the recalled memory. Finally, the way the connection strengths must be chosen for a given set of memories can be easily implemented by learning rules (*13*) such as the one proposed by Hebb (*31*).

To what extent can more difficult computations—for example, those relevant to object recognition or speech perception—be carried out naturally on these model neural circuits? One of the characteristics of such computations seems to be a combinatorial explosion—the huge number of possible answers that must be considered. The desired computation (for example, matching a set of words to a sound pattern) can often be stated as an optimization. Although it is not yet known how to map most biological problems onto model circuits, it is now possible to design model circuits to solve nonbiological problems having combinatorial complexity.

Because well-defined problems have been used, the effectiveness of the neural circuit computation can be quantified. We will review two circuit examples.

The idea of most algorithms or procedures for optimization is to move in a space of possible configurations representing solutions, progress in a direction that tends to decrease the cost function being minimized, and hope that the space and method of moving are smooth enough that a good solution will ultimately be reached. Such ideas lie behind conventional computer optimization algorithms and the recent work in simulated annealing (*32*) and Bayesian image analysis (*33*). In our approach (*15–17*), the optimization problem is mapped onto a neural network in such a way that the network configurations correspond to possible solutions to the problem. An E function appropriate to the problem is then constructed. The form of the E function is chosen so that at configurations representing possible solutions, E is proportional to the cost function of the problem. Since, in general, E is minimized as the circuit computes, the dynamics produce a path through the space that tends to minimize the energy and therefore the cost function. Eventually, a stable-state configuration is reached that corresponds to a local minimum in the E function. The solution to the problem is then decoded from this configuration.

It is particularly easy to construct appropriate E functions when the sigmoid input-output relation is steep, because in this "high-gain" limit, each neuron will be either very near 0 output or very near its maximal output when the system is in a low E stable state (*14*). In the high-gain case, the energy function is

$$E = -\frac{1}{2}\sum_{i,j} T_{ij}\, V_i\, V_j - \sum_j I_j\, V_j \qquad (2)$$

When lower gain is considered, terms containing the function $g_i(u_i)$ must be included in E (*14*). The following two examples make use of this high-gain limit.

The simple seven-neuron circuit described in Fig. 1 was designed according to this conceptual framework to be a four-bit analog-to-binary (A-B) converter. Given an analog input to the circuit represented by the time-averaged impulse activity in the input axon Q, the neural circuit is organized to adjust the firing rates in the principal neurons so that they can be interpreted as the binary number numerically equal to the time-averaged input activity. Reorganization of the data in Fig. 2A will illustrate this computation. In each time epoch in Fig. 2A, assign the value 0 or 1 to the variable V_i representing the output of P_i; if P_i is firing strongly, $V_i = 1$; if it is quiescent, $V_i = 0$. Represent the activity in axon Q by a continuous variable X. The value of the binary word interpreted from the ordered list of numbers ($V_3V_2V_1V_0$) is plotted in Fig. 2B for each of the different values of input strength X. The data points (asterisks) lie on a staircase function (dotted line) characteristic of an A-B converter. (Although not shown, the outputs computed for any other input would also lie on this curve.)

Through the consideration of a specific energy function in the high-gain limit and the synaptic strengths and inputs listed in Table 1, the behavior of the neural circuit can be predicted and understood. We decide in advance that outputs $V_3V_2V_1V_0$ of P_3 through P_0 are interpreted as a computed binary word. The problem to be solved is stated as an optimization: Given analog input X, which binary word (set of outputs) best represents the numerical value of X? The solution is provided when the following E is minimized (*16*):

$$E = -\frac{1}{2}\,(X - \sum_{j=0}^{3} 2^j\, V_j)^2 + \sum_{j=0}^{3} (2^{2j-1})\,[V_j(1 - V_j)] \qquad (3)$$

The second term in E is minimized (and numerically equal to 0) when all V_j are either close to 0 or close to 1. Since E is minimized as the circuit converges, stable states having the correct "syntax" tend to develop. Since the first term in E is a minimum when the expression in the parentheses vanishes, this term biases the circuit towards the states closest, in the least-squares sense, to the analog value of X. The E in Eq. 3 is like that in Eq. 2, a quadratic in the V_i. Rearranging Eq. 3 and comparing it with this general form yields values for T_{ij} and I_i for a circuit of the form in Fig. 3B that can be deduced within a common scale factor as

$$T_{ij} = -2^{(i+j)};\ I_i = (-2^{(2i-1)} + 2^i X) \qquad (4)$$

The coefficient of X in I_i is the synaptic strength from the input axon Q to the principal neurons. These specific values are equal to the strengths of the "effective" synapses tabulated in Table 1. Knowledge that E is minimized as the circuit computes provides an *understanding* of how this synaptic organization both enforces the necessary syntax and biases the network to choose the optimum solution.

Our second example is a neural circuit that computes solutions to the traveling salesman problem (TSP) (*15*). In this frequently studied optimization problem (*34*), a salesman is required to visit in some sequence each of n cities; the problem is to determine the shortest closed tour in which every city is visited only once. Specific problems are defined by the distances (d_{ij}) between pairs of cities (i, j). Assigning letters to the cities in a TSP permits a solution to be specified by an ordered list of letters. For example, the list *CAFGB* is interpreted as "visit C, then A, then F, then G, then B, and finally return to C." For an n-city TSP, this list can be decoded from the outputs of $N = n^2$ neurons if we let a single neuron correspond to the hypothesis that one of the n cities is in a particular one of the n possible positions in the final tour list. This rule suggests the arrangement illustrated in Fig. 5 for displaying the neural output states. The output of a neuron (V_i) is graphically illustrated by shading; a filled square represents a neuron which is "on" and firing strongly. An empty square represents a neuron that is not firing. The output states of the n neurons in each row are interpreted as information about the location of a particular city in the tour solution. The output states of the n neurons in each column are interpreted as information about what cities are to be associated with a particular position in the tour. If the neuron from column 5 in row C is "on," the hypothesis that city C is in position 5 in the final tour solution is true.

Hypothetically, each of the n cities could indicate its position in any one of the n possible tour locations. Therefore, 2^N possible "neural states" could conceivably be represented by these outputs. However, only a subset of these actually correspond to valid solutions to the TSP (valid tours): a city must be in one, and only one, position in a valid tour, and any position must be filled by one and only one city. This constraint implies that only output states in which only one neuron is "on" in every row and in every column are of the correct "syntax" to represent valid solutions to the TSP. A TSP circuit that is to operate correctly must have synapses favoring this subset of states. Simple lateral inhibition between neurons within each row and column will provide this bias. For example, if $V_{B,2}$ (representing city B in position 2) is "on," all other neurons in row B and column 2 should be inhibited. This can be provided by the inhibitory connections from neuron $V_{B,2}$ drawn in Fig. 5 (red lines). Similar row and column inhibitory connections are drawn for neuron $V_{D,5}$. A complex "topology" of syntax-enforcing connections is generated. We can also think of these connections as contributing a term to the E function for the circuit. For example, a term $+A\ V_{X,i}\ V_{Y,i}$ in E makes a contribution $-A$ to the synaptic strength $T_{X,i;Y,i}$ and represents a mutual lateral inhibition between neurons (X,i) and (Y,i). The term is positive (higher E) when both of these neurons are "on," but contributes nothing if only one of the two is "on." The proper combination of similar terms in an E function can specify the synapses that coordinate correct syntax.

In a syntactically correct state representing a valid solution (tour), if neurons $V_{X,i}$ and $V_{Y,i+1}$ are both "on," the salesman travels from city X directly to city Y. Therefore, the distance $d_{X,Y}$ between these two cities is included in the total tour length for that solution. A term of the form $+d_{X,Y}\,V_{X,i}\,V_{Y,i+1}$ in the E function provides a "distance" contribution of d_{xy} to the value of E when these neurons are "on." Similar terms, properly summed, will add to E a value equal to the length of the tour. Since the circuit minimizes E, the final state will be biased toward those valid solutions representing short tours. Such inhibitory connections are drawn in Fig. 5 with blue lines for neurons $V_{B,2}$ and $V_{D,5}$. In TSP and in the earlier example, the rules of syntax are expressed in inhibitory connections. It seems easier to define what these systems should not do (by inhibitory connections), and to define what they should do by default, rather than to define what they should do by writing syntax in excitatory connections.

The inhibitory synapses define the computational connections for the TSP circuit. With a common sigmoid gain curve, R, and C for each neuron, the description of the circuit is complete. The gain curve is chosen so that with zero input, a neuron has a nonzero but modest output. This circuit can rapidly compute good solutions to a TSP problem (*15*). When started from an initial "noise" state favoring no particular tour, the network rapidly converges to a steady state describing a very short tour. The state of the circuit at several time points in a typical convergence is illustrated in Fig. 6. In a 30-city problem, there are about 10^{30} possible tours—the combinatorial problem has gotten completely out of hand. But the circuit of 900 neurons can find one of the best 10^7 solutions in a single convergence—a few time constants of the circuit. It selects good answers and rejects bad ones by a factor of 10^{23}.

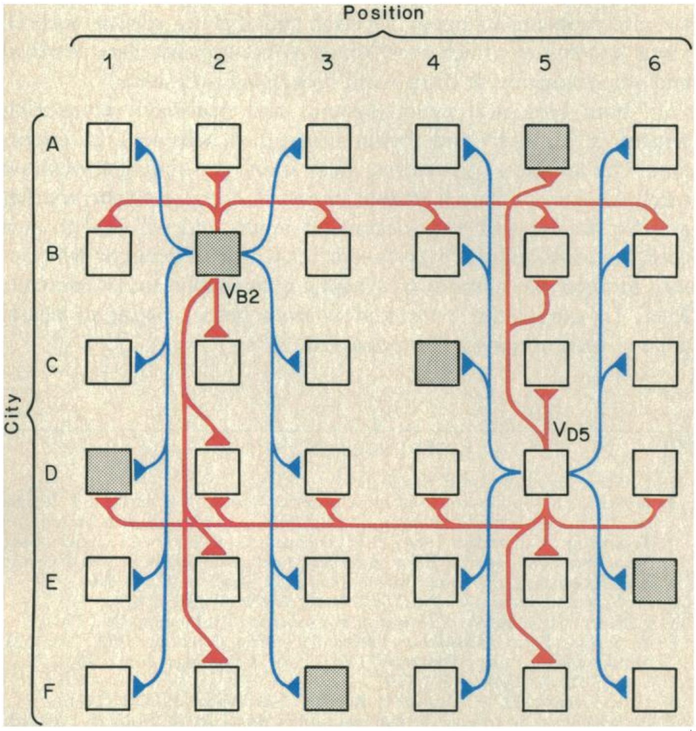

Fig. 5. A stylized picture of the syntax and connections of the TSP neural circuit. Each neuron is symbolically indicated by a square. The neurons are arranged in an n by n array. Each city is associated with n neurons in a row, and each position in the final tour is associated with n neurons in a column. A given neuron ($V_{X,j}$) represents the hypothesis that city X is in position j in the solution. The patterns of synaptic connection for two different neurons are indicated.

The continuous response characteristic of the analog neurons in the TSP circuit represents partial knowledge or belief. A value for $V_{X,j}$ between 0 and 1 represents the "strength" of the hypothesis that city X is in position j of the tour. During an analog convergence, several conflicting solutions or propositions can be simultaneously considered through the continuous variables. It is as though the logical operations of a calculation could be given continuous values between "true" and "false" and evolve toward certainty only near the end of the calculation. This is evident during the TSP convergence process (Fig. 6) and is important for finding good solutions to this problem (*15*). If the gain is greatly increased, the output of any given neuron will usually be either 1 or 0, and the potential analog character of the network will not be utilized. When operated in this mode, the paths found are little better than random. The analog nature of the neural response is in this problem essential to its computational effectiveness. This use of a continuous variable between true and false is similar to the theory of fuzzy sets (*35*) and to the use of evidence voting for the truth of competing propositions in Bayesian inference and connectionist modeling in cognitive psychology (*36*). Two-state neurons do not capture this computational feature.

Discussion

The work reviewed here has shown that a simple model of nonlinear neurons organized into networks with effectively symmetric connections has a "natural" capacity for solving optimization problems. The general behavior can be readily adapted to particular problems by appropriately selecting the synaptic connections. Optimization problems are ubiquitous where goals are attempted in the presence of conflicting constraints, and they arise in problems of perception (What three-dimensional shape "best" describes a given shading pattern in a two-dimensional image?), behavioral choice, and motor control (What is the optimum trajectory to move an appendage to minimize internal stresses?). Hence circuits consistent with this model could efficiently solve problems important in biological information processing.

Biologically relevant problems in vision have already been formulated in terms of optimization problems. Edge-detection, stereopsis, and motion detection can be described as "ill-posed" problems, and solutions can be found by minimizing appropriate quadratic functionals (*37*). The emphasis in these formulations has been simple convex problems with a single minimum in the energy. Networks solving these problems can be implemented by linear circuits having local connections. The nonlinear circuits described here can implement solutions to much more complex problems and have recently been used to solve the object-discontinuity problem in early vision (*18*).

The concept of an energy function and its use in circuit design provide an understanding of *how* model neural circuits rapidly compute solutions to optimization problems. The state of each neuron changes in time in a simple way determined by the state of neurons to which it is connected, but the organization of the synapses results in collective dynamics that minimize an E function relevant to the optimization problem. Knowledge of this E function helps us understand the collective dynamics. The two circuit examples reviewed here, the A-B converter and TSP circuit, were "forward-engineered." Given the optimization problem, a representation of hypothetical solutions to the problem as a particular set of neural states was constructed. Synaptic connections in the operating circuit move the neural state toward these solution states and bias

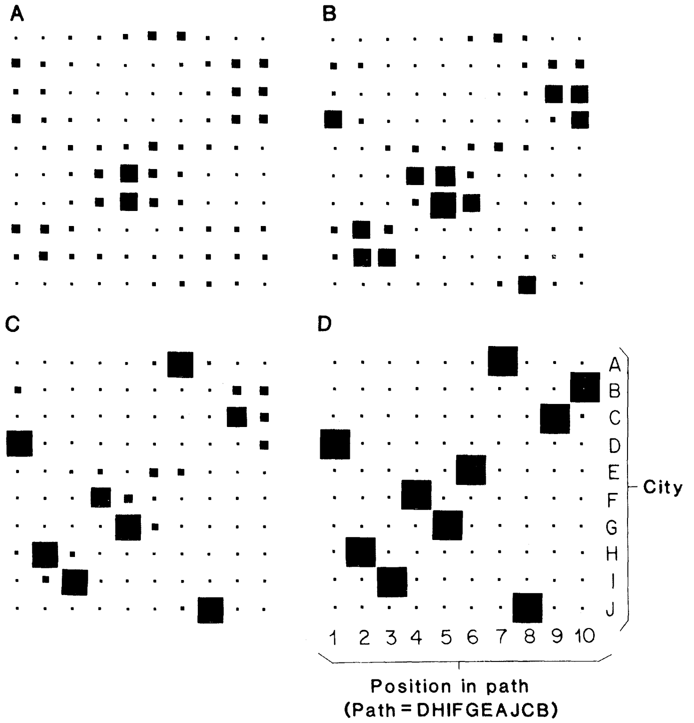

Fig. 6. The convergence of a ten-city analog circuit to a tour. The linear dimension of each square is proportional to the value of $V_{X,i}$. (A to C) Intermediate times. (D) The final state. Indices illustrate how the final state is decoded into a tour (solution of the TSP).

this motion toward the best solution. The values of these synaptic strengths were summarized in the single algebraic statement of the E function. [The two problems illustrate different ways in which "data" modulate the circuit parameters: as input currents in the A-B converter or as changes in the connection strengths in the TSP circuit (*17*).] Forward-engineered examples of model neural circuits add to the known repertoire of computational circuits that seem neurobiologically plausible. The general problem of neurobiology is "reverse engineering"—to understand the operation of a complex biological circuit with unknown design principles and internal representations. In general, the set of neural circuits whose functioning is understood provides an information base for hypothesizing function in biological neural circuits in the same way that the study of understood electrical circuits aids the attempt to understand or reverse engineer an unfamiliar electrical circuit diagram.

When a problem falls naturally onto a neural circuit, its convergence to a collective analog decision in a few time constants represents immense computation for the amount of hardware involved. For example, the 30-city TSP can be done on a network of 900 neurons. When that kind of combinatorial problem occurs in perception and pattern recognition, the input to the system will occur in parallel and take little time. A biological neural network of this structure would converge to an answer in a few neural time constants, thus in about 0.1 second. An electronic circuit of the same structure would converge in about 1 μsec. A comparably good solution to this problem, with conventional algorithms used for the TSP, can be found in about 0.1 second on a typical microcomputer having 10^4 times as many devices. The effectiveness of the neural system on the basis of computations per device per time constant is great in comparison with the usual general-purpose digital machine. The ability of the model networks to compute effectively is based on large connectivity, analog response, and reciprocal or reentrant connections. The computations are qualitatively different from those performed by Boolean logic.

Other specific circuit designs have been studied. Many problems in signal processing can be described as the attempt to detect the presence or absence of a waveform having a known stereotyped shape in the presence of other waveforms and noise. (The recognition of phonemes in a stream of speech is conceptually similar, but fraught with large problems of variability from the stereotype form.) We have described the general organization of neural circuits that could solve this task (*16*). Energy functions have been described for other combinatorial optimization problems, including graph coloring (*17*), the Euclidean-match problem (*17*), and the transposition code problem (*15*). Circuits that relax the restriction on a symmetric connection matrix (as biology does) have also been studied. A circuit designed to provide solutions to linear programming problems (*16*) functions without oscillation when the characteristic times of these elements are properly specified, even though its computing elements have antisymmetric connection strengths. The associative memory originally discussed (*10*) and used in a model of learning in a simple invertebrate (*38*) can be described as an optimization problem (*15*). The same conceptual framework can seemingly be applied to a large number of different problems.

Because the basic idea of the model neural circuit can be expressed as an electrical circuit, there have been efforts to build such hardware. Associative memories of 32 neurons (amplifiers) have been built in conventional electrical circuit technology (*39*). A 22-neuron circuit has been successfully microfabricated on a single silicon chip (*40*). Shrinking this kind of network to a compact size seems possible (*41*). The most compact and useful form of such a device would involve an electrically writable resistance change in a two-terminal device, which would function approximately as a Hebbian (*31*) synapse. Examples of such material fabrications exist (*42*). A 32-neuron system has been fabricated that uses optics to implement connections (*43*). Technological questions have so far focused chiefly on associative memory. Similar circuits could be used to solve problems in signal detection and analysis, such as artificial visual systems, in which there tends to be immense data overload and where concurrent distributed processing is desired.

In both biological neural systems and man-made computing structures, hierarchy and rhythmic or timed behaviors are important. The addition of rhythms, adaptation, and timing provides a mechanism for moving from one aspect of a computation to another and for dealing with time-dependent inputs and will lead to new computational abilities even in small networks. Hierarchy is necessary to keep the number of synaptic connections to a reasonable level. To extend the present ideas from neural circuit to neural system, such notions will be essential.

REFERENCES AND NOTES

1. D. Marr, *Vision* (Freeman, San Francisco, 1982).
2. Implementations different from those reviewed here are discussed in: T. Poggio and C. Koch, *Artificial Intelligence Lab. Memo 773* (Massachusetts Institute of Technology, Cambridge, 1984); S. E. Fahlman, G. E. Hinton, T. J. Sejnowski, *Proceedings of National Conference on Artificial Intelligence* (1983), p. 109; S. Geman and D. Geman, *IEEE Trans. Pattern Anal. Mech. Intell.* **6**, 721 (1984).
3. W. S. McCulloch and W. Pitts, *Bull. Math. Biophys.* **5**, 115 (1943).
4. F. Rosenblatt, *Principles of Neurodynamics* (Spartan, Washington, DC, 1961).
5. W. S. McCulloch, *Embodiments of Mind* (MIT Press, Cambridge, MA, 1965); M. Minsky and S. Papert, *Perceptrons* (MIT Press, Cambridge, MA, 1969); A. C. Scott, *J. Math. Psychol.* **15**, 1 (1977).
6. H. K. Hartline, H. G. Wagner, F. Ratliff, *J. Gen. Physiol.* **39**, 651 (1956).
7. For a summary of appropriate mathematical methods, see D. Noble, J. J. B. Jack, and R. Tsien [*Electric Current Flow in Excitable Cells* (Clarendon, Oxford, 1974)].
8. D. K. Hartline, *Biol. Cybern.* **33**, 223 (1979); D. H. Perkel and B. Mulloney, *Science* **185**, 181 (1974); A. I. Selverston, *Behav. Brain Sci.* **3**, 535 (1980); W. O. Freisen and G. S. Stent, *Biol. Cybern.* **28**, 27 (1977).
9. See, for example: R. D. Traub and R. K. S. Wong, *Science* **216**, 745 (1982).
10. J. J. Hopfield, *Proc. Natl. Acad. Sci. U.S.A.* **79**, 2554 (1982).
11. While space does not permit the review of modern developments using two-state neurons, the following references provide an introduction to this literature: W. A. Little, *Math. Biosci.* **19**, 101 (1974); ——— and G. L. Shaw, *ibid.* **39**, 281 (1978);

G. L. Shaw, D. J. Silverman, J. C. Pearson, *Proc. Natl. Acad. Sci. U.S.A.* **82**, 3364 (1985); M. Y. Choi and B. A. Huberman, *Phys. Rev. A* **28**, 1204 (1983); T. Hogg and B. A. Huberman, *Proc. Natl. Acad. Sci. U.S.A.* **81**, 6871 (1984); M. A. Cohen and S. Grossberg, *IEEE Trans. Syst. Man Cybern.* **SMC-13**, 815 (1983); K. Nakano, *ibid.* **SMC-2** (no. 3) (1972). For recent work on stochastic models applied to two states, see G. E. Hinton, T. J. Sejnowski, and D. H. Ackley [*Technical Report CMU-CS-84-119* (Carnegie-Mellon University, Pittsburgh, 1984)].
12. We refer to the dynamics as "classical" because we are ignoring propagation time delays and the quantal nature of action potentials, in analogy to classical mechanics. Similar equations have been described: T. J. Sejnowski, in *Parallel Models of Associative Memory*, G. E. Hinton and J. A. Anderson, Eds. (Erlbaum, Hillsdale, NJ, 1981), p. 189.
13. T. Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag, Berlin, 1984); T. Kohonen *et al.*, *Neuroscience* **2**, 1065 (1977); T. Kohonen, *Biol. Cybern.* **43**, 59 (1982); L. N. Cooper, F. Liberman, E. Oja, *ibid.* **33**, 9 (1979); D. Ackley, G. E. Hinton, T. J. Sejnowski, *Cognit. Sci.* **9**, 147 (1985); D. d'Humieres and B. A. Huberman, *J. Stat. Phys.* **34**, 361 (1984); K. Fukushima, *Biol. Cybern.* **36**, 193 (1980); A. G. Barto, R. S. Sutton, P. S. Brouwer, *ibid.* **40**, 201 (1981); *Parallel Models of Associative Memory*, G. E. Hinton and J. A. Anderson, Eds. (Erlbaum, Hillsdale, NJ, 1981). The capacity of associative memories constructed from networks of two-state neurons is discussed by P. Peretto [*Biol. Cybern.* **50**, 51 (1984)], D. J. Amit, H. Gutfreund, H. Sompolinsky [*Phys. Rev. Lett.* **55**, 1530 (1985)], R. J. McEliece and E. C. Posner [*JPL Telecommunications and Data Acquisition Progress Report 42-83* (1985), p. 209], Y.-S. Abu-Mostafa and J. St. Jacques [*IEEE Trans. Inf. Theory* **IT-31**, 461 (1985)], and L. Personnaz, I. Guyon, and G. Dreyfus [*J. Physique Lett.* **46**, L-359 (1985)].
14. J. J. Hopfield, *Proc. Natl. Acad. Sci. U.S.A.* **81**, 3088 (1984).
15. ——— and D. W. Tank, *Biol. Cybern.* **52**, 141 (1985).
16. D. W. Tank and J. J. Hopfield, *IEEE Circuits Syst.* **CAS-33**, 533 (1986).
17. J. J. Hopfield and D. W. Tank, in *Disordered Systems and Biological Organization* (Springer-Verlag, Berlin, 1986).
18. C. Koch, J. Marroquin, A. Yuille, *Proc. Natl. Acad. Sci. U.S.A.* **83**, 4263 (1986).
19. E. Mjolsness, *CalTech Computer Science Dept. Publication 5153:DF* (1984).
20. T. J. Sejnowski and G. E. Hinton, in *Vision, Brain, and Cooperative Computation*, M. Arbib and A. R. Hanson, Eds. (MIT Press, Cambridge, MA, 1985).
21. G. M. Shepherd, *The Synaptic Organization of the Brain* (Oxford Univ. Press, New York, 1979), p. 3.
22. For a review of observed input-output relations and their ionic basis, see W. E. Crill and P. C. Schwindt [*Trends NeuroSci* **6**, 236 (1983)].
23. K. G. Pearson, in *Simpler Networks and Behavior*, J. C. Fentress, Ed. (Sinauer, Sunderland, MA, 1976).
24. J. McCarragher and R. Chase, *J. Neurosci* **16**, 69 (1985).
25. M. V. L. Bennett and D. A. Goodenough, *Neurosci. Res. Program Bull.* **16**, 371 (1978).
26. W. Rall and G. M. Shepherd, *J. Neurophysiol.* **31**, 884 (1968).
27. C. Koch and A. Yuille, personal communication.
28. J. Platt, personal communication.
29. E. Harth, *IEEE Trans. Syst. Man Cybern.* **SMC-13**, 782 (1983).
30. W. J. Freeman, *Mass Action in the Nervous System* (Academic Press, New York, 1975).
31. D. O. Hebb, *The Organization of Behavior* (Wiley, New York, 1949).
32. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, *Science* **220**, 671 (1983).
33. S. Geeman and D. Geeman, *IEEE Trans. Pattern Anal.* **6**, 721 (1984).
34. M. R. Garey and D. S. Johnson, *Computers and Intractability* (Freeman, New York, 1979).
35. L. A. Zadeh, *IEEE Trans. Syst. Man Cybern.* **SMC-3**, 38 (1974).
36. D. H. Ballard, *Pattern Recognition* **13**, 111 (1981); J. A. Feldman and D. H. Ballard, *Cognit. Sci.* **6**, 205 (1982); D. L. Waltz and J. B. Pollack, *ibid.* **9**, 51 (1985); J. B. Pollack and D. L. Pollack, *Byte Magazine* **11** (no. 2), 189 (1986).
37. T. Poggio, V. Torre and C. Koch, *Nature (London)* **317**, 314 (1985).
38. A. E. Gelperin, J. J. Hopfield, D. W. Tank, in *Model Neural. Networks and Behavior*, A. Selverston, Ed. (Plenum, New York, 1985).
39. J. Lambe, A. Moopenn, A. P. Thakoor, *Proceedings of the AIAA: Fifth Conference on Computers in Space*, in press; J. Lambe, A. Moopenn, A. P. Thakoor, *Jet Propulsion Lab Publication* **85-69** (Jet Propulsion Laboratory, Pasadena, CA, 1985).
40. M. Sivilotti, M. Emmerling, C. Mead, *1985 Conference on Very Large Scale Integration*, H. Fuchs, Ed. (Computer Science, Rockville, MD, 1985), p. 329.
41. L. D. Jackel, R. E. Howard, H. P. Graf, B. Straughn, J. S. Denker, *J. Vac. Sci. Tech.*, in press.
42. A. E. Owen, P. G. Le Comber, G. Sarrabayrouse, W. E. Spear, *IEE Proc.* **129** (part 1, no. 2), 51 (1982).
43. D. Psaltis and N. Farhat, *Optics Lett.* **10**, 98 (1985).
44. Supported in part by NSF grant PCM-8406049.

Arctic Research in the National Interest

A. L. WASHBURN AND GUNTER WELLER

The Arctic Research and Policy Act of 1984 was designed to advance arctic research in the national interest. Some of the research fields that require attention are weather and climate; national defense; renewable and nonrenewable resources; transportation; communications and space-disturbance effects; environmental protection; health, culture, and socioeconomics; and international cooperation. A research framework recommended by the Arctic Research Commission includes, in order of priority, integrated investigations to understand: (i) the Arctic Ocean (including the marginal seas, sea ice, and seabed) and how the ocean and atmosphere operate as coupled components of the arctic system; (ii) the coupled atmosphere and land components and how their interaction governs the terrestrial environment; and (iii) the high-latitude upper atmosphere and its extension into the magnetosphere with emphasis on predicting and mitigating effects on communications and defense systems. A separate recommendation is for high priority research to resolve the major health, behavioral, and cultural problems related to the arctic environment. Recommendations are also made concerning support services and management.

THE ARCTIC IS IMPORTANT FOR MANY REASONS—DEFENSE, economic, political, and scientific (*1–4*). The Arctic Research and Policy Act of 1984 has now put some of these interests into sharper focus. Its stated purposes are "to establish national policy, priorities, and goals and to provide a Federal program plan for basic and applied scientific research with respect to the Arctic, including natural resources and materials, physical, biological and health sciences, and social and behavioral sciences" [5, Section 102(b)(1)]. The act established two cooperating groups to carry out its intent: (i) an advisory Arctic Research Commission consisting of five presidential appointees and the director of the National Science Foundation, who serves as an ex officio, nonvoting member, and (ii) an executive Interagency Arctic Research Policy Committee, consisting of a representative from ten named federal agencies and possibly others, which is chaired by the National Science Foundation representative.

Passage of the act reflected an increasing awareness in Alaska, in Washington, and among scientists and others that U.S. arctic

---

A. L. Washburn is a professor emeritus in the Department of Geological Sciences and the Quaternary Research Center, University of Washington, Seattle, WA 98195, and is a member of the U.S. Arctic Research Commission. G. Weller is a professor in the Geophysical Institute, University of Alaska, Fairbanks, AK 99701, and is chairman of the Polar Research Board of the National Research Council.