NEURONS, DYNAMICS AND COMPUTATION

Brains have long been regarded as biological computers. But how do these collections of neurons perform computations?

John J. Hopfield

The question "How does it work?" is the motivation of many physicists. Condensed matter physics, chemical physics and nuclear physics can all be thought of as descriptions of the relation between structure and *properties*. The components of a biological system have *functional* properties that are particularly relevant to the operation of the system. Thus it is especially important in biology to understand the relation between structure and *function*. Such understanding can be sought at the level of the molecule, the cell, the organ, the organism or the social group.

The function of a nervous system is to do computations. Recognizing a friend, walking and understanding a spoken sentence all involve computations. The analysis of the nervous system presented here relates the biophysics of nerve cells, statistical physics and dynamical systems to the way a biological "machine" computes.

I use the word "compute" here only in the very fuzzy sense of performing a useful task of a kind that a digital computer can also perform. For example, one can program a digital machine to compare a present image with a set of images generated from a three-dimensional representation of the head of a friend, and thus in principle the problem of recognizing a friend can be solved by a computation. Similarly, the question of how to drive the actuators on a robot given the present posture of the robot and the desired state of dynamic balance is fundamentally a problem in classical mechanics, which can be solved on a digital computer. While we may not know how to write efficient algorithms for these tasks, such examples do illustrate that one may usefully describe what the nervous system does as computation. However, that one can use a digital computer to model the outcomes of experiments done on a nervous system does not *ipso facto* mean that the brain is a computer, since digital computers can be used to model most physical systems.

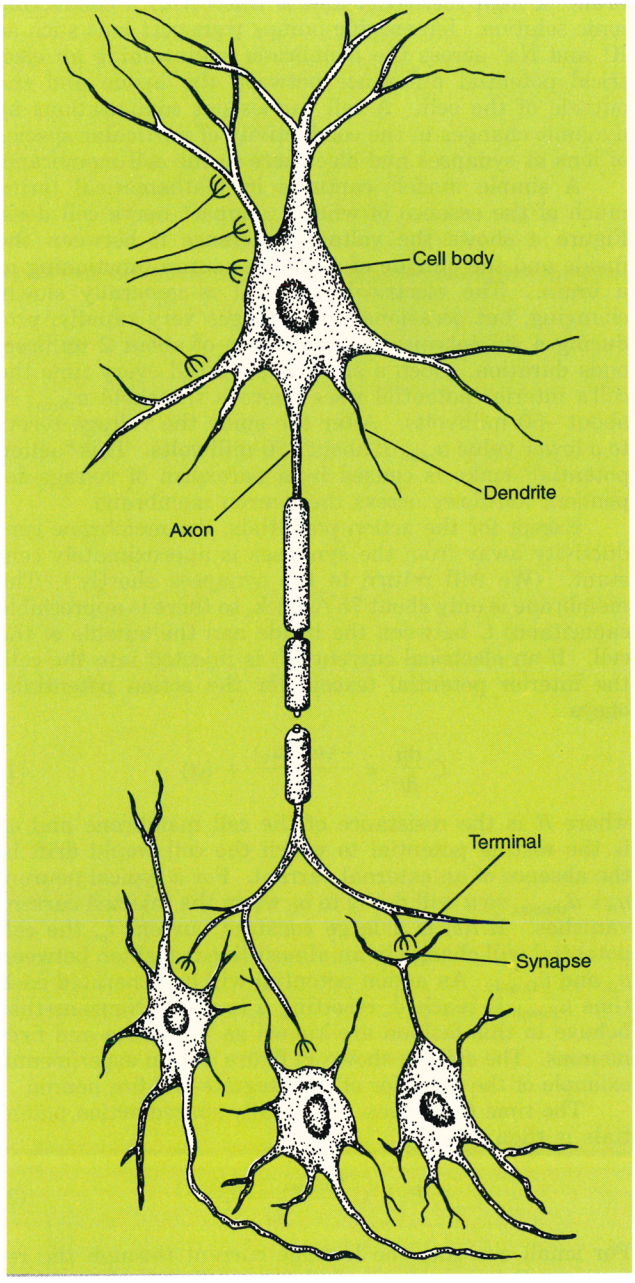
For the purposes of this article, we view a computer as an input–output device, with the input and output signals in the same general medium or format. Thus in a very simple digital computer, the input is a string of bits (in time), and the output is another string of bits. The computer produces a transformation on the inputs to generate the outputs. Within this view, the brain is a computer. For example, a million axons carry electrochemical pulses from the eyes to the brain. Similar signaling pulses drive the muscles of the vocal tract. When we enter a room, look around and say, "Hello, Jessica," our brain is producing a very complicated transformation from one parallel input pulse sequence coming from the eyes to another parallel output pulse sequence that results in sound waves being generated.

The idea of *composition* is very important in this view of a computer. The output of one computer can be used as the input for another computer of the same general type, since both signals are in the same medium. Within this view, a digital chip is a computer, and large computers are built as composites of smaller ones. Similarly, each neuron (see figure 1) is a simple computer, and the brain is a large composite computer made of neurons.

Computers as dynamical systems

A real, physical digital computer is a dynamical system and computes by following a path in its space of physical states.[1] (See figure 2.) Its operation is most simply illustrated for batch-mode computation, in which all the inputs are supplied at the start of the computation (unlike

John Hopfield is the Roscoe G. Dickinson Professor of Chemistry and Biology at the California Institute of Technology, in Pasadena, California.



Neurons are simple computers. Each neuron receives inputs at synapses and computes an output that is transmitted along its axon to as many as 1000 other neurons. The brain may be regarded as a composite computer made up of a network of neurons. (Adapted from ref. 2.) **Figure 1**

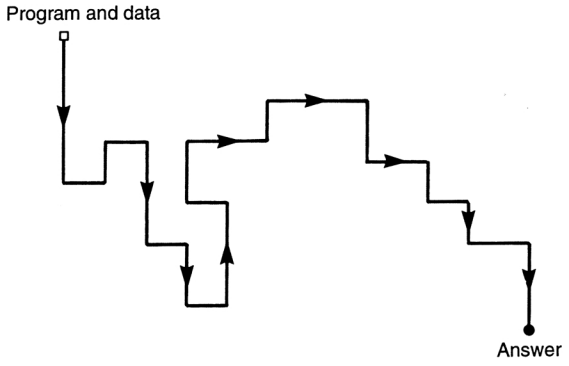
interactive computation, in which inputs may continue to arrive during the computation). The computer has N storage registers, each storing a single binary digit. A binary vector of N bits, such as 10010110000. . . , specifies the logical state of the machine at a particular time. This binary state changes into a new state each clock cycle. The transition map describing which state follows which is implicitly built into the machine by its design. Thus one can describe the machine as a dynamical system that changes its discrete state in discrete time.

The user of the machine has no control over the dynamics, which is determined by the state transition map. The user's program and data and a standard initialization procedure prescribe the starting state of the machine. The motion of the dynamical system carries out the computation. In a batch-mode computation, the answer is found when a stable point of the discrete dynamical system—a state from which there are no transitions—is reached. A particular subset of the state bits (for example, the contents of a particular machine register) will then describe the desired answer.

Batch-mode analog computation can be similarly described with continuous time and state-space variables. The idea of computation as a process carried out by a dynamical system in moving from an initial state to a final state is the same as in the discrete case. In the analog case, one can think of the motion in state space as describing a flow field, and computation is done by moving with this flow from start to finish. (See figure 3.) The final state is typically a point attractor—a location in the state space to which all nearby states will evolve. (Of course real "digital" machines contain only analog components. The digital description is only a compact representation in fewer variables that contains the essence of the continuous dynamics.)

One of the most important resources for intelligent behavior is powerful associative memory, in which partial and perhaps somewhat erroneous knowledge of a memory can nevertheless give access to the complete memory. A system whose dynamics in a high-dimensional state space is dominated by a substantial number of point attractors can be regarded as an associative memory: The location of a particular point attractor can be obtained from partial information (an inexact description of the attractor location) by merely initializing the system in accord with the partial information and allowing the dynamics to evolve the state to the nearest attractor. For such a system to be useful and biological it must be possible to insert memories (new attractors) into the system by a biologically plausible algorithm. We will consider this in greater detail below. Error-correction codes, used with data transmission, can also be construed as attractors.

Computations more complicated than the recovery of memories can also be directly formulated in terms of the fixed points of dynamical systems. For example, when



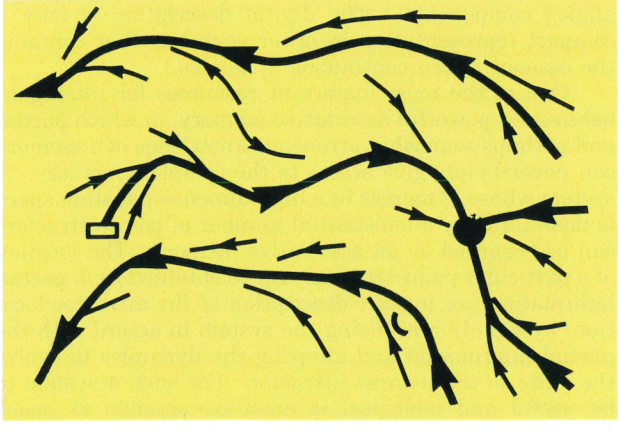
State of a digital computer follows a path through its space of discrete states. In batch-mode computation the path goes from an initial state representing the program and the data to a final stable state representing the answer. **Figure 2**

a problem can be posed as an optimization, its solution can also often be posed on a dynamical system for which the stable (fixed) points are minima of the desired variable. The location of the solution is found by following a trajectory of motion to its end.

A simple dynamical model of neurobiology

Figure 1 depicts the anatomy of a "typical" neuron in a mammalian brain.[2] In gross terms, it has three regions: dendrites, a cell body and an axon. Each neuron is connected to approximately 1000 other neurons by structures called synapses. A nerve cell functions as an input–output device. Inputs to a cell are made at synapses on its dendrites or on the cell body. The cell produces outputs that drive other cells lying at synapses at the terminals of its axon. When considering a particular synapse, we call the cell producing the output the presynaptic cell while the one receiving the input is postsynaptic.

The interior of each cell is surrounded by a membrane of high resistivity and is filled with a conducting ionic solution. Ion-specific pumps transport ions such as K^+ and Na^+ across the membrane, maintaining an electrical potential difference between the inside and the outside of the cell. A cell carries out computations by dynamic changes in the conductivity of particular species of ions at synapses and elsewhere in the cell membrane.



An analog computational system has a continuous space of states. The state-space flow field must be focused onto paths to negate the effects of errors but is otherwise similar to that of a digital system (see figure 2). **Figure 3**

A simple model[3] captures in mathematical terms much of the essence of what a compact nerve cell does. Figure 4 shows the voltage difference u between the inside and the outside of a simple neuron functioning in a brain. The electrical potential is generally slowly changing, but occasionally it changes very rapidly, producing a stereotypical voltage spike of about 2 milliseconds duration. Such a spike is produced every time the cell's interior potential rises above a threshold u_{thresh} of about −50 millivolts. After the spike the voltage resets to a lower value u_{reset} of about −70 millivolts. This "action potential" spike is caused by a paroxysm of voltage-dependent ion flows across the neuron membrane.

Except for the action potentials, the membrane conductivity away from the synapses is approximately constant. (We will return to the synapses shortly.) The membrane is only about 75 Å thick, so there is appreciable capacitance C between the inside and the outside of the cell. If an electrical current $i(t)$ is injected into the cell, the interior potential (except for the action potentials) obeys

$$C\frac{du}{dt} = \frac{-(u-u_0)}{R} + i(t) \qquad (1)$$

where R is the resistance of the cell membrane and u_0 is the resting potential to which the cell would drift in the absence of an external current. For a typical neuron, $u_0 < u_{\text{thresh}}$, so u will decay to u_0 when the injected current vanishes. If $i(t)$ is a large constant current i_c, the cell potential will change in an almost linear fashion between u_0 and u_{thresh}. An action potential will be generated each time u_{thresh} is reached, resetting u to u_{reset}. Neurons that behave in this fashion are known as "integrate and fire" neurons. The spiking shown in figure 4 is an experimental example of the behavior of an integrate-and-fire neuron.

The time P between the equally spaced action potentials is then roughly

$$P = C\frac{u_{\text{thresh}} - u_{\text{reset}}}{i_c} \qquad (2)$$

For small currents the leakage current through the resistance R is important, and for small enough constant currents the leakage current will prevent the cell from firing at all. The black curve in figure 5 shows the firing rate $1/P$ as a function of current i_c for a realistic cell.

We will take action potentials to be delta functions, lasting a negligible time. They propagate at constant velocity along an axon. The transmission is nonlinear, but the shapes of the pulses are actively maintained. When an action potential arrives at a synaptic terminal of an axon, the terminal releases a neurotransmitter (such as acetylcholine or glutamate), which in turn activates specific ionic conductivity channels in the postsynaptic dendrite. For reasons including diffusion and chemical inactivation, this conductivity pulse $\sigma(t)$ is not a delta function but has nonzero duration. It can be modeled as

$$\sigma(t) = \begin{cases} 0 & t < t_0 \\ s\,e^{-(t-t_0)/\tau} & t > t_0 \end{cases} \qquad (3)$$

where s is the maximum conductivity of the postsynaptic membrane in response to the action potential, and τ is

the time constant of the pulse. Each synapse from cell j to cell k has its own particular maximum conductivity s_{kj}. The conductivity is ion specific, and the current that flows depends on the chemical potential difference V_{ion} between the inside and the outside for that ion. Thus for a synapse from cell j to cell k, an action potential arriving on cell j's axon at time t_0 causes a current

$$i(t) = \begin{cases} 0 & t < t_0 \\ S_{kj}\, e^{-(t-t_0)/\tau} & t > t_0 \end{cases} \tag{4}$$

to flow into cell k. The parameter $S_{kj} = V_{\text{ion}} s_{kj}$ can have either sign, depending on the sign of the free energy difference driving the selected ion type. If S_{kj} is positive, the synapse is "excitatory" because it tends to excite the neuron k to fire. Similarly, a negative S_{kj} corresponds to an "inhibitory" synapse.

An equation of motion can be obtained as follows: For any neuron k, which fires action potentials at times t_n^k ($n = 1, 2, 3, \ldots$), define the instantaneous firing rate to be

$$f_k(t) = \sum_n \delta(t - t_n^k) \tag{5}$$

In classical electrical circuit theory, the current flowing into a capacitor as a function of time is a similar sum of delta functions because of the discreteness of electrons. The integral of $f_k(t)$ over a time interval yields the number of action potentials occurring within the time interval, and in this sense $f_k(t)$ is the instantaneous rate.

Differentiating equation 4 with respect to time yields

$$\frac{di}{dt} = -\frac{i}{\tau} + S_{kj}\,\delta(t - t_0) \tag{6}$$

Similarly, for the total current i_k into cell k one has

$$\frac{di_k}{dt} = -\frac{i_k}{\tau} + \sum_j S_{kj} f_j(t) + \text{sensory term} \tag{7}$$

where the "sensory term" is an additional term present only for sensory cells. This equation, though exact, is awkward to deal with because the times at which the action potentials occur are given only implicitly through equation 1.

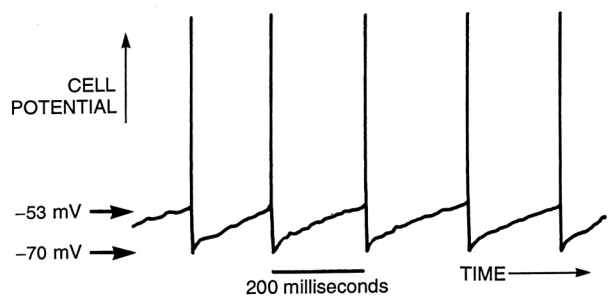
Synapse evolution algorithms

The synaptic strengths S_{kj} can also change with time, both during the development of an immature nervous system and as part of the learning and adaptation that go on in a mature one. While several such changes are seen in neurobiology, the most interesting variety is one in which the synapse strength S_{kj} changes as a result of the roughly simultaneous activity of cells k and j. This kind of change is needed if a nervous system is to "learn" the association between two events. A synapse whose change algorithm involves only the simultaneous activity of the pre- and postsynaptic neurons and no other detailed information (other than perhaps when to learn) is called a Hebbian synapse.[4]

A simple version of such dynamics might be written

$$\frac{dS_{kj}}{dt} = \alpha i_k f_j(t) - \text{decay terms} \tag{8}$$

where α is a positive parameter. The "decay terms," perhaps involving i_k and f_j, are essential if the system is to forget old information. A nonlinearity or control process is important to keep synapse strength from increasing without bound. Also, the parameter α might be varied by neuromodulator molecules that control the overall learning process. The details of the neurobiology are not yet thoroughly understood, and equation 8 is only a placeholder for a more adequate expression. Slightly more complex synapse change rules of a Hebbian type reproduce results of a variety of experiments on the development of eye dominance and orientation selectivity of cells in the visual cortex of the cat.[5]



Action potentials, spikes in the electrical potential of the inside of a neuron, are generated when the cell potential reaches a threshold (–53 mV in this example), discharging the cell. After discharging, the cell resets to about –70 mV. When a constant current is injected into the cell, action potentials are generated at a regular rate. (Adapted from data provided courtesy of James Schwaber, Du Pont Experimental Station, Wilmington, Delaware.) **Figure 4**

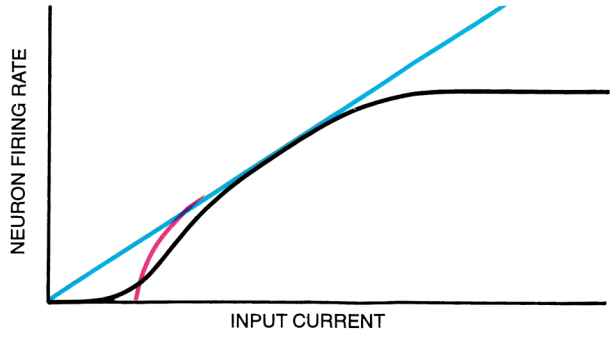
The synapse evolution algorithm, whatever its form, is one of the dynamical equations of the neural system. The tacit view is that learning and development involve synapse changes, whereas the dynamics of neural activity is what performs a computation. This need not be the case, however, and synapse modification should not be ignored as a means of doing some kinds of computation.

A synapse change algorithm underlies the most widespread application of artificial neural networks. A feedforward network (one without closed-loop pathways) is a trivial dynamical system, but such a network, with an appropriate set of connections, can solve nontrivial pattern classification problems.[6] The computational power in this case is embodied in finding the correct set of connections, a process most often done by a highly artificial synapse change dynamics that is quite unrelated to biology. Even so, it is the dynamical method of determining the connection strengths that results in a useful feedforward network.

Classical neurodynamics

Neural network dynamics can be described in two extreme limits. In one description, the "classical approximation," individual action potentials have little effect and their precise timing is unimportant for coding information. We can then adopt the point of view that deals only with large numbers of action potentials in a statistical fashion. This paradigm has been used in much of neurobiology and neuromathematics. In an alternative paradigm the precise relative timing of action potentials arriving along different axons is very important, and the detailed time intervals between action potentials on a single axon are used to code significant information. I will describe this paradigm later.

The classical approximation makes use of the fact that there will be many contributions to the sum on the right-hand side of equation 7 during a reasonable time interval as a result of the high connectivity. (The sum



Average firing rate of a neuron depends on the input current. The approximation of "integrate and fire" without leakage or saturation would give the blue straight line. Leakage due to the cell's resistance results in the red curve, while noise fluctuations smooth out the sharp break in the curve at zero firing rate, producing the black curve. Saturation effects in the mechanism that generates action potentials set an upper limit on the rate. **Figure 5**

over j typically includes thousands of cells that may make nonzero contributions.) In that case, it should be permissible to ignore the spiky nature of $f_j(t)$ and replace it with a convolution of $f_j(t)$ and a smoothing function. In addition, f_j is presumed to be a function of i_c, denoted $V(i_c)$, when i_c is slowly varying in time. What results is like equation 7, but with $f_j(t)$ now a smooth function of time, $f_j(t) = V(i_j(t))$. Thus[7,8]

$$\frac{di_k}{dt} = -\frac{i_k}{\tau} + \sum_j S_{kj} V(i_j(t)) + \text{sensory term} \qquad (9)$$

The main effect of the approximation is to neglect fluctuation noise and to assume there are no strong correlations between spike events. (A similar description could be given in electrical circuits of the passage from considering discrete electrons to regarding charge as a continuous variable.)

In many regions of the brain, the dominant connectivity is quite short range, and signal propagation delays are negligible. While propagation delays can easily be introduced into equation 9, the mathematics of equations with delays is rather more complicated.

Computation with fixed connections

With one set of parameters, equation 9 describes a Vax. With another, it can mimic the electronics in a television receiver. The set of all equations represented by equation 9 is far too general to have simple universal properties. If computation is to be done by a convergence to point attractors in the space of analog variables, one way of achieving that end is to consider a restricted set of networks that can be shown to converge to fixed points. We will therefore examine networks whose motion can be understood as the state's moving generally downhill on an "energy function" (or Lyapunov function) that may have a complicated landscape with many minima.

The simplest case that is sufficiently flexible to be of interest is a symmetric network,[8] defined by $S_{ij} = S_{ji}$. The function

$$E = -\sum_{k,j} S_{kj} V_k V_j + \frac{1}{\tau} \sum_k \int_0^{f_k} V^{-1}(f)\, df - \sum_k I_k V_k \qquad (10)$$

where V^{-1} is the inverse of the function V, $V_j = V(i_j)$ and the current I_k in the third term on the right comes from external or sensory inputs, can be shown to always decrease under the equations of motion unless all variables have stopped changing. Since E is bounded below, this implies that the state will converge to the location of one of the minima of this function. The dynamics is understood through the minimum-seeking nature of equation 9.

While symmetric connections are not the usual case in neurobiology (for example, an excitatory neuron can receive inputs of either sign but can make only positive connections to other neurons), there are various circumstances in which more complex and biological networks are equivalent to symmetric networks. Even feedforward networks can, by an appropriate transformation, be made equivalent to symmetrically coupled networks.

The behavior of a symmetric system is easiest to understand in the limit of high gain, where the sigmoid response of the neurons (figure 5) approximates a step. Then, under most circumstances, every stable state must have each neuron either at maximal activity or zero activity, and each such state lies at one of the corners of a 2^N-dimensional hypercube, where N is the number of neurons. The stable-state problem is then isomorphic with an Ising spin problem, but with each spin having the possible values 0 and 1 (instead of the more usual –1 and 1) and with the S_{kj} serving as the exchange interactions in the Ising Hamiltonian. This connection with spin systems, and in a limiting case to a spin glass, has permitted extensive analysis of the stable states.[9] (See the article by Haim Sompolinsky in PHYSICS TODAY, December 1988, page 70.)

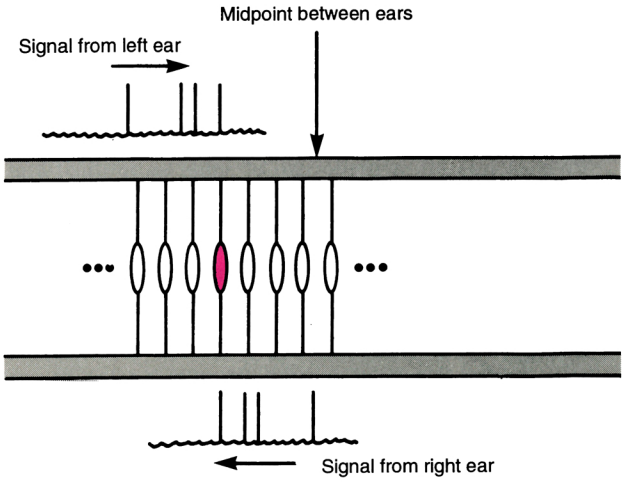
An associative memory can be constructed as follows. If a "1" is defined to be a neuron firing at maximal rate, and a "0" as a neuron not firing, in the high-gain limit a memory is simply a state vector such as $\mathbf{V}_{\text{mem}} = 1,0,0,1,0,0,0,0,1, \ldots$. The synapse change

$$\Delta S_{kj} \propto (2V^k_{\text{mem}} - 1)(2V^j_{\text{mem}} - 1) \qquad (11)$$

will make $\mathbf{V}_{\text{mem}}$ a new fixed point of the dynamics, that is, a new memory. This synapse change is of the Hebbian type. Networks of a more biological flavor, having fixed patterns of inhibitory connections and low mean activity rates and carrying memory information only in excitatory connections, also function as associative memories.

One can design symmetric networks to find solutions to many complex tasks that can be posed as minimizations.[10] For example, such networks have found solutions to the classical traveling salesman problem (in which a salesman wishes to visit each member of a set of cities once with the minimum amount of traveling) and to some practical problems of circuit-board layout. Researchers study such synthetic problems in part to learn about the computational power that can be obtained from a single convergence to a fixed point of a symmetric network. We expect the highly fed-back neural circuitry of all complex nervous systems to exhibit this computational power. We might think of the single convergence to a fixed point as the fundamental computing step of a brain, just as the single clock cycle is the fundamental computing step of a conventional digital machine. However, the high connectivity of the neural system and the analog nature of its convergence allow the performance of quite complex tasks in a single step.

If we abandon the restriction of symmetry, the next simplest system is the excitatory–inhibitory network.



Binaural localization of sound by an owl depends on the timing of individual action potentials. Sequences of spikes from the two ears, representing the sound detected, travel along antiparallel axons. The two streams of spikes will tend to coincide and excite neurons at a location (red) whose position corresponds to the time delay between the sounds detected at each ear. **Figure 6**

The neurons in such a network fall into two classes: One has only excitatory outputs; the other has only inhibitory outputs. Such networks can oscillate or even exhibit chaos[11] (in computer modeling), and it has been difficult to gain enough mathematical control of them to see how to use them for powerful computation. Oscillatory systems are now being intensely investigated, however, because oscillatory behaviors are commonly found in the brain, for example, in the olfactory bulb and the neocortex.

Action potential synchronization

The foregoing analysis was based on the idea that individual action potentials are insignificant and that the precise information about when particular action potentials occur is generally not relevant. Information is implicitly encoded in the short-time average of the number of action potentials generated by each cell. This paradigm lies behind most of the studies of the response of individual neurons in mammals. For example, a neuron is called strongly responsive to a stimulus generated by a moving bar of light if the neuron generates action potentials at a high rate only during the presentation of that stimulus. Detailed information about the relative timing of different action potentials is ignored.

In some situations, however, the timing of action potentials is very important to biological function. For example, the arrival of a synchronized set of action potentials on the many axons of the vagus nerve triggers the contraction of the heart. The timing of this pulse is essential to heart rhythm and blood pressure control. Action potential timing also matters in the binaural localization of a sound source by an owl. Each ear hears a sound of the same form, but one sound is delayed with respect to the other, and that delay determines the azimuthal location of a sound source. The brain measures the time delay with an array of neurons that detect coincidences between action potentials propagating with finite velocity along antiparallel axons from the two ears. (See figure 6.)

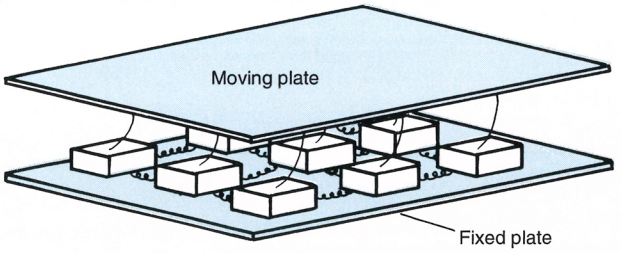
In addition, visual object perception involves a set of specific computational problems that must be somehow solved in the brain. One of these is related to how we piece together the different parts of particular objects in the visual field, so that an object is seen separated from the background of other objects. The idea that what moves together is a single object is one of the important criteria that the brain seems to use. (If you have ever wondered why the door edge behind the subject's head is so plain in the snapshot yet was so invisible in the viewfinder when you were taking the picture, the parallax caused by your minor motions is a key element of the answer.) Coherent oscillations or synchronized action potentials may occur in neurons responsive to separated but comoving edge elements of a single object. Ideas[12] and experiments[13] in this direction give impetus to theoretical work on the synchronization of action potentials in integrate-and-fire neurons.

Most of the theoretical work has been done to try to understand the range of phenomena occurring in simple systems and has not yet focused on how such systems can do useful computation. In the simplest case a group of excitatory integrate-and-fire neurons are all connected to one another ("all-to-all coupling"), and each coupling is excitatory and of the same strength. When the external or intrinsic current into each cell is the same (which would result in equal firing rates of all the cells if there were no connections between them) and the synaptic current due to an action potential at t_0 has the form

$$I(t) = \begin{cases} 0 & t < t_0 \\ I_0 e^{-(t-t_0)/\tau} & t > t_0 \end{cases} \qquad (12)$$

then the coupling synchronizes the action potentials of all the cells.[14] If the system begins in an arbitrary state of activity, it will evolve to a state of synchronized firing, a cyclic attractor for the system. If the neurons have a range of firing rates in the absence of connections, or if the connections themselves are not uniform, a broader class of behaviors occurs, including phase transitions to the synchronous state and the breakup of the cells into two classes, one group synchronized and one not.

In neurobiology, chemical events and molecular configuration changes take place between the occurrence of a presynaptic action potential and the ultimate current injection into the postsynaptic cell. Therefore the measured synaptic currents do not rise as a step but increase smoothly from zero. For this more realistic synaptic model, in the presence of noise, but with equivalent firing rates and equal all-to-all connections, there is a synchronization–desynchronization phase transition as a function of noise amplitude.[15] The less realistic model of equation 12 retains a synchronized phase for all noise amplitudes.



Spring–block model of earthquakes is somewhat analogous to models of sheets of neurons that account for individual neuron spikes. Springs connecting the blocks correspond to synaptic connections between neurons, while the slipping of a block corresponds to the firing of a neuron. Consequently the types of complex behavior seen in the earthquake models can be expected to recur in neurobiology. (Adapted from ref. 17.) **Figure 7**

In an elementary short-time view of neural computation, the computation is performed by a convergence to a point attractor of a dynamical system. The synchronization of spiking neurons is a special case of a more general dynamical system whose motion converges to a cyclic attractor. Richer dynamics with more complex attractors should allow attractor networks to solve more complicated problems than can be solved with simple point attractors.

The phenomena displayed by coupled integrate-and-fire neurons will be richer when the synaptic connection patterns are more complex. Even the replacement of the equal all-to-all coupling by a fixed near-neighbor synaptic coupling in two dimensions (to represent aspects of a sheet of cells such as occurs in the neocortex) greatly changes the kinds of behavior that are found. This problem, which does not seem to have been studied in neurobiology, has in a limiting case a very close parallel with the Burridge–Knopoff model[16] of earthquake generation at a junction between tectonic plates. (This point was jointly understood in discussions last spring between Andreas Herz, John Rundle and me.) In the Burridge–Knopoff model, the junction is represented by a set of slider "blocks" that are connected to a moving upper plate by springs and are dragged along the lower plate. (See figure 7.) The motion of each slider is described by stick–slip friction. Each slider is also connected by other springs to its nearest neighbors. An earthquake is initiated when one slider slips and triggers the motion of other blocks. A "slip" event corresponds to an action potential, the spring from plate to slider corresponds to an external current from elsewhere into each cell, and the springs between sliders correspond to synaptic connections. The slipping is "self-organized"[17,18] and produces a power-law distribution of earthquake magnitudes. While there is no exact correspondence with real neurobiology in this limiting case, it does extend our ideas of the kind of phenomena that can emerge from retaining action potential timing in neurodynamical equations.

Simplicity and complexity

Digital machines and brains both carry out computation by being dynamical systems. A very simple representation of highly complicated neurobiology leads to a description in terms of coupled nonlinear differential equations. The equations presented here are a drastic simplification of real biological neurons. Many features could be added, including propagation delays, a position-dependent intracellular potential and the use of intracellular Ca^{2+} concentrations as dynamical variables. How important are such omitted features?

Physics has often made use of huge simplifications to get to the heart of issues. For example, conventional models of magnetism usually omit many details of multispin interactions, magnetoelastic coupling and longer-range interactions, and yet they capture much of the essence of magnetic phenomena. Physicists are therefore accustomed to ignoring inconvenient details.

Biology, being an evolutionary science, is different. If some quirky detail of neurobiology is useful in an important but special computation, that detail can be selected for and improved by evolution. As a result, in specific parts of the brain, particular details that are generally negligible elsewhere can be of utmost importance. The highly simplified model dynamics described in this article is thus far too impoverished to describe how a brain operates.

Nevertheless, *network* computation with high connectivity between analog elements is the means by which large brains gain an intelligence lacking in small nervous systems. The attractor behavior of equation 9 has proved to be robust to noise and to changes such as the addition of delays and action potentials. Even though the elements we have used are oversimplified abstractions, this robustness gives us reason to believe that we are making progress in understanding how networks of neurons and synapses can carry out complex computations.

References

1. J. J. Hopfield, in *Evolutionary Trends in the Physical Sciences*, M. Suzuki, R. Kubo, eds., Springer-Verlag, New York (1991), p. 295.
2. E. R. Kandel, J. H. Schwartz, *Principles of Neural Science,* 3rd ed., Appleton & Lange, Norwalk, Conn. (1991), p. 19.
3. D. Junge, *Nerve and Muscle Excitation*, Sinauer, Sunderland, Mass. (1981), p. 115, discusses some of the complications of the biophysics of this process.
4. J. Hertz, A. Krogh, R. G. Palmer, *Introduction to the Theory of Neural Computation,* Addison Wesley, Redwood City, Calif. (1991), p. 16.
5. M. F. Bear, L. N. Cooper, F. F. Ebner, Science **237**, 42 (1987).
6. See, for example, G. Dreyfus, in *Applications of Neural Networks*, H. G. Schuster, ed., VCH, New York (1992), p. 35.
7. While these equations are in common use, they have evolved somewhat, and do not have a sharp original reference. See, however, H. R. Wilson, J. D. Cowan, Biophys. J. **12**, 1 (1972); and W. Gerstner, J. L. van Hemmen, Network **3**, 139 (1992).
8. J. J. Hopfield, Proc. Natl. Acad. Sci. USA **79**, 2554 (1982); **81**, 3088 (1984).
9. E. Domany, J. L. van Hemmen, K. Schulten, eds., *Models of Neural Networks,* Springer-Verlag, New York (1991).
10. J. J. Hopfield, D. G. Tank, Biol. Cybern. **52**, 141 (1985). Y. Takefuji, *Neural Network Parallel Computing,* Kluwer, Boston (1992).
11. W. J. Freeman, Sci. Am., February 1991, p. 78.
12. R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk, H. J. Reitboeck, Biol. Cybern. **60**, 121 (1988).
13. C. M. Gray, W. Singer, Proc. Natl. Acad. Sci. USA **86**, 1698 (1989).
14. Y. Kuramoto, Physica D **50**, 15 (1991). R. E. Mirollo, S. H. Strogatz, SIAM J. Appl. Math **50**, 1645 (1990). L. F. Abbott, C. Vanvreeswijk, Phys. Rev. E **48**, 1483 (1993).
15. M. Tsodyks, I. Mitkov, H. Sompolinsky, Phys. Rev. Lett. **71**, 1280 (1993).
16. R. Burridge, L. Knopoff, Bull. Seismol. Soc. Am. **57**, 341 (1967).
17. Z. Olami, H. J. S. Feder, K. Christensen, Phys. Rev. Lett. **68**, 1244 (1992).
18. P. Bak, C. Tang, J. Geophys. Res. **94**, 15 635 (1989). ■