the flow of activity in a neural network (*3*) can be described by a simple potential function (*4*). Some computational problems can be transformed into a more-or-less equivalent optimization problem by a so-called regularization procedure (*5*, *6*). Hopfield's potential function provides the link between this optimization problem and its solution in terms of neural network, since in the network the potential is automatically minimized (*1*); and by proper choice of the network parameters, notably the neuronal interconnections, one can produce any potential function which is a polynomial of a second order (*4*). A new result is that, approximately, any minimization problem can be broken down into two minimization problems for second-order functionals (*7*), and thus can be solved by an appropriately constructed neural network.

In particular for linear optimization problems this transformation into a second-order minimization problem (*8*) and the corresponding network solution appear to be very natural (*6*, *9*). In general it is, however, not at all clear under which conditions this transformation of the original problem into a network computation is reasonably economic (for example, in terms of the size of the network). In many cases a quite different, more economic network solution may present itself immediately (*10*).

From the point of view of economy it would be desirable to compare not only different network approaches with one another, but to compare network approaches to more conventional approaches to computation. Since these network approaches have a strong impact on analog computing and special-purpose machines—as opposed to their effect on the conventional von Neumann computer architecture—it is often difficult to compare the computational complexity of these new network procedures to that of more conventional computing approaches.

The long-standing problem of realizing an associative memory that can be addressed by arbitrary parts of a (stored) pattern—often referred to as the problem of pattern matching for content-addressable memories—has also been considered in the framework of these networks (*11–13*). In this particular case it seems possible to compare the network solution to more conventional pattern-matching approaches (*14*) in terms of their computational complexity, that is, the number of operations needed in the retrieval of a stored pattern from a (sufficiently large) part of it (*15–17*). Such a comparison may seem to favor the conventional approaches because of the common conviction that a network of n neuron-like elements (or nodes) can only store less than n patterns reliably (*2*, *16*). However, the immense possibilities provided by appropriate coding of the patterns to be stored have been widely overlooked. In fact, a proper coding of the patterns to be stored into sparse 0-1-sequences (that is, sequences containing mostly zeroes) makes it possible to store effectively (*18*) much more than n patterns in a network of n (neuron-like) elements (*19*). More exactly, if z patterns, each containing x bits of information, have to be stored, one can use a network of $n = 2\sqrt{x \cdot z}$ elements, provided that the patterns are coded into 0-1-sequences of length n that contain all the same low number ($k \approx \sqrt{x}$) of ones (*20*). In the retrieval one would conventionally need $b \cdot z$ comparisons ($1 < b \leq x$), whereas the network performs only less than $k \cdot n \approx 2x\sqrt{z}$ simple counting operations.

Therefore, if z (the number of patterns to be stored) is large compared to x (the information content of one pattern), a situation that is quite typical for the usual applications, then the network method is computationally simpler than conventional methods, and also the number of patterns that can be stored (z) is considerably larger than the number of network elements (n), which is in clear contradiction to Hopfield's estimate (*2*) $z \approx 0.15\ n$. But this associative memory network only works so well if the patterns are coded sparsely [the probability of a nonzero element in a codeword should be $k/n \approx 1/(2\sqrt{z})$, whereas usually, for instance in the spin-glass literature, it is assumed to be about 1/2]. Consequently the next step in the network approach to memory should be the investigation of sparse-coding techniques.

GUNTHER PALM
Max Planck Institut fur Biologische Kybernetik,
Spemannstrasse 38,
D-7400 Tubingen 1,
West Germany

REFERENCES AND NOTES

1. J. J. Hopfield and D. W. Tank, *Science* **233**, 625 (1986).
2. J. J. Hopfield, *Proc. Natl. Acad. Sci. U.S.A.* **79**, 2554 (1982).
3. Some authors consider the same equations as a so-called spin-glass model. The relation between neural networks and physical spin-glass models is explained, for example, by W. Kinzel [*Z. Phys. B* **60**, 205 (1985)]. Spin-glass models were first related to optimization problems by S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi [*Science* **220**, 671 (1983)].
4. The function is given by $H\ (x_1, \ldots, x_n) = -(1/2)\sum_{ij} c_{ij}x_ix_j + \sum_i x_i\theta_i$, where $(x_1, \ldots, x_n)$ is the vector of neural activities, $c_{ij} = c_{ji}$ is the symmetric matrix of connection-strengths between neurons, and $(\theta_1, \ldots, \theta_n)$ are the neural thresholds. Note that $H(x_1, \ldots, x_n)$ is the general second-order polynomial functional, although the connectivity matrix c_{ij} has to be symmetric for the potential-function formalism.
5. The idea probably goes back to A. N. Tikhonov [*Soviet Math. Dokl.* **4**, 1035 (1963)].
6. T. Poggio, V. Torre, C. Koch, *Nature (London)* **317**, 314 (1985).
7. I have shown [G. Palm, *Biol. Cybern.* **34**, 49 (1979)] that any functional $S(x_1, \ldots, x_n)$ can be approximated as $S = \sum_i a_1\exp(L_ix)$, where the L_i are linear functionals of the vector $x = (x_1, \ldots, x_n)$. The problem $S^2(x)$ = min! can be reduced to the two problems (i) $S^2(y) = \sum_{ij} a_ia_jy_iy_j$ = min! and $\ln y_i = L_ix$ ($i = 1, 2, \ldots$), solve for x! Both problems can be solved by network methods [compare methods in (*4*), (*6*), and (*9*)].
8. That is, an optimization problem where the functional to be minimized is a second-order polynomial, such as the function H in (*4*).
9. D. W. Tank and J. J. Hopfield, *IEEE Trans. Circuits Syst.* **CAS-33**, 533 (1986). If the problem is simply to solve a system of linear equations as in the second problem in (*7*), then the regularization leads to the classical pseudo-inverse of a matrix [compare also (*11*)].
10. For example, in the traveling salesman problem, compare J. J. Hopfield and D. W. Tank, *Biol. Cybern.* **52**, 141 (1985).
11. T. Kohonen, *Associative Memory* (Springer Verlag, Berlin, 1977).
12. G. Palm, *Biol. Cybern.* **36** 19 (1980).
13. S.-I. Amari, *IEEE Trans. Syst. Man. Cybern.* **13**, 741 (1983); J. J. Hopfield, D. Feinstein, R. G. Palmer, *Nature (London)* **304**, 158 (1983); P. Peretto, *Biol. Cybern.* **50**, 51 (1984).
14. R. O. Duda and P. E. Harth, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
15. G. Palm and T. Bonhoeffer, *Biol. Cybern.* **51**, 201 (1984).
16. Y. S. Abu-Mostafa and J.-M. St. Jacques, *IEEE Trans. Inf. Theory* **31**, 461 (1985).
17. A. Lansner and O. Ekeberg, *IEEE Trans. Pattern Anal. Mach. Intell.* **7**, 490 (1985).
18. That is, with very low error probability in the recall.
19. This is already obvious from (*12*), but see also (*15*) and (*17*).
20. G. Palm, in *Brain Theory*, G. Palm and A. Aertsen, Eds. (Springer Verlag, Berlin, 1986), pp. 211–228.

12 November 1986; accepted 5 December 1986

Response: The intellectual thrust of our article (*1*) was to show the neurobiologist how a conceptual framework and methodology could be used to understand how a class of model neural circuits solve specific computational problems. In addition, we used examples of model circuit computation to illustrate the general problems neurobiologists face in understanding computation in biological neural circuits.

The use of Liapunov functions in stability analysis is widespread in physics and engineering. An energy function (Liapunov function) applicable to the task of mapping specific problems onto neural networks with symmetrical connections was first proposed by Hopfield in 1982 (*2*). The importance of this energy function is that it can be used with a theory and methodology which together provide an understanding of how specific problems could be solved by these networks. This is a central idea in the work we reviewed (*1*), and it distinguishes this work from that cited by Carpenter, Cohen, and Grossberg. As discussed in (*1*), the methodology and conceptual framework together with this function were first used to understand the mapping of associative memory onto a model neural circuit having discrete, stochastic (*2*), or continuous (*3*) response. The approach was determined to be of general use when it was used to map more conventional combinatorial optimization problems onto similar neural networks

(4). The model system employed in this work is described by a very general set of dynamical equations (equation 1 in the technical comment by Carpenter *et al.*). These equations have been used in a variety of other studies of neural network function. However, it is not these equations per se that define the theory or body of work we reviewed.

The relation between the stable points and dynamics of a system with continuous variables and a system with discrete variables provides a direct means of understanding how particular problems can be solved by such networks. This relation provides an additional benefit because of the rich statistical physics of the spin glass.

As we stated in our article (*1*), we agree with the assertion of Carpenter *et al.* that symmetry of connections only simplifies the classes of behavior that are exhibited by model networks having feedback. The relation between symmetry or "equivalent symmetry" and stability is indeed subtle. The work we reviewed focused on networks with symmetric connections because an understanding of system trajectories is a necessary prerequisite to understanding the computational capabilities of a model network. Systems with stable limit cycles can exist for reentrant nonsymmetric networks. Except for a few cases of networks with special forms of nonsymmetry that we and others have studied, we do not in general understand how to use these reentrant networks in a significant computational fashion. We would like general principles for understanding how reentrant networks with limit cycles (and without equivalent symmetry) can be organized for specific computational tasks, but useful answers have not yet been found. Thus our review considered only the case of equivalently symmetric connections and simple limit points.

The "computational approach" exemplified by the work of Marr (*5*), which we and many others have followed, distinguishes between the practical proof of the computational power of a neural circuit theory by example, and abstract mathematical theories. Good examples of this approach have been provided in the context of other neural network models (*6*), such as the demonstration by Sejnowski and Rosenberg of a layered neural network for text-to-speech conversion (*7*) that uses the back propagation algorithm (*8*).

With respect to the comment by Kohonen and Oja, in our article (*1*), the traveling salesman problem (TSP) was chosen to illustrate how the class of neural network models we discussed could compute solutions to optimization problems. It was chosen because it is easily stated, hard to solve, and has been the most widely studied optimization problem. How humans attempt to solve the TSP is not relevant to a discussion of our article. That neural activities in the TSP circuit are a "high level semantic representation" is only the result of the circuit design principles being applied to this particular problem. For example, at early stages of processing in the visual system, the activities in single neurons represent simple propositions, like the information that a spot of light is moving across the visual field in a particular direction. Neural circuit models based on the design principles we reviewed (*1*) but applied to computational problems in early vision (*9*) describe neural activities that represent only such simple propositions.

The work we reviewed focused on understanding principles of operation of given neural circuits, irrespective of how they are assembled. Nervous system development in higher animals involves genetics, self-organization, and learning. However, knowledge of the developmental rules and even a detailed anatomical description do not explain how a given circuit functions to produce an observed animal behavior. Assembly instructions and a detailed wiring diagram for a radio do not explain the principles of tuned oscillators and their use in radio communication. The distinction between understanding how a neural circuit functions and its anatomical description was the main point of the first section of our article (*1*). [Although not the topic of our review, Kohonen incorrectly states that no learning or adaptive effects have been studied in the nonlinear associative memories. See, for example (*10*).]

Decisions are an essential part of the computational behavior of nervous systems. The circuits we described compute decisions because of their nonlinear dynamics. Linear systems like those described by Kohonen and Oja are not capable of computing decisions. For example, movement detection, a decision computed by the fly's visual system, cannot be performed by a linear system (*11*).

JOHN J. HOPFIELD
Divisions of Chemistry and *Biology, California Institute of Technology, Pasadena, CA 91125*, and
Molecular Biophysics Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974
DAVID W. TANK
Molecular Biophysics Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974

REFERENCES AND NOTES

1. J. J. Hopfield and D. W. Tank, *Science* **233**, 625 (1986).
2. J. J. Hopfield, *Proc. Natl. Acad. Sci. U.S.A.* **79**, 2554 (1982).
3. ———, *ibid.* **81**, 3088 (1984).
4. J. J. Hopfield and D. W. Tank, *Biol. Cybern.* **52**, 141 (1985).
5. D. Marr, *Vision* (Freeman, San Francisco, 1982).
6. No comprehensive review of all of this work exists. For an introduction to the literature, see J. A. Feldman and D. H. Ballard, *Cog. Sci.* **36**, 205 (1982); W. P. Keirstead and B. A. Huberman, *Phys. Rev. Lett.* **56**, 1094 (1986); T. Poggio, V. Torre, C. Koch, *Nature (London)* **317**, 314 (1985); A. Barto, R. S. Sutton, C. W. Anderson, *IEEE Trans. Syst. Man Cybern.* **SMC-13**, No. 5 (1983); G. E. Hinton, T. J. Sejnowski, D. H. Ackley, *Technical Report CMU-CS-84-119* (Carnegie Mellon University, Pittsburgh, PA, 1984); S. Geman and D. Geman, *IEEE Trans. Pattern-Anal. Mach. Intell.* **6**, 721 (1984).
7. T. J. Sejnowski and C. R. Rosenberg, *Johns Hopkins University Electrical Engineering and Computer Science Technical Report JHU/EECS-86/01* (Johns Hopkins Univ., Baltimore, MD, 1986).
8. D. E. Rumelhart, G. E. Hinton, R. J. Williams, in *Parallel, Distributed Processing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. (MIT Press, Cambridge, MA, in press).
9. C. Koch, J. Marroquin, A. Yuille, *Proc. Natl. Acad. Sci. U.S.A.* **83**, 4263 (1986).
10. J. P. Nadal, G. Toulouse, J. P. Changeux, S. Dehaene, *Europhys. Lett.* **1**, 535 (1986); M. Mezard, J. P. Nadal, G. Toulouse, *J. Phys.* **47**, 1457 (1986).
11. T. Poggio and W. E. Reichardt, in *Theoretical Approaches in Neurobiology*, W. E. Reichardt and T. Poggio, Eds. (MIT Press, Cambridge, MA, 1981), p. 65.

25 November 1986; accepted 5 December 1986