Journal of Statistical Physics, Vol. 51, Nos. 5/6, 1988

Neural Networks: A Biased Overview

Eytan Domany[1]

Received October 29, 1987

An overview of recent activity in the field of neural networks is presented. The long-range aim of this research is to understand how the brain works. First some of the problems are stated and terminology defined; then an attempt is made to explain why physicists are drawn to the field, and their main potential contribution. In particular, in recent years some interesting models have been introduced by physicists. A small subset of these models is described, with particular emphasis on those that are analytically soluble. Finally a brief review of the history and recent developments of single- and multilayer perceptrons is given, bringing the situation up to date regarding the central immediate problem of the field: search for a learning algorithm that has an associated convergence theorem.

KEY WORDS: Neural networks; perceptron; memory; learning.

1. INTRODUCTION

I present a brief overview of a field that is relatively new to physicists. I choose to describe a small subset of the problems and some models; my choices are only a reflection of my personal tastes. Moreover, my bias is set mainly by ignorance; physicists should realize that the field of neural networks has been active (under different names) for a few decades. In a recent meeting ten different scientific disciplines were represented, and the weight of physicists is probably less then 10% in terms of their contributions. This overview does not attempt to present the prevalent attitudes in the field, or to serve as a comprehensive introductory list of references.[2]

[1] Department of Electronics, Weizmann Institute of Science, Rehovot, Israel.

[2] For recent reviews in the physics literature see Refs. 1; for reviews and introductory texts from other fields see Refs. 2.

The ultimate goal of research in the field of neural networks is to gain understanding of how the brain works.[3] In my opinion *none* of the work on neural networks in the physics literature brings this goal any closer. I believe that the field is intriguing and exciting enough to justify research without making unfounded claims to the contrary. Of course, one should keep the ultimate goal in mind, and question whether certain assumptions made are or are not in conflict with neurophysiological observations. To this end I describe in Section 2 some of the problems one would like to understand and present a bare minimum of "neurophysiology for physicists" as well as a dictionary connecting biological concepts to physics terminology. Next, a brief description of memory is given; an attempt is be made to explain why physicists are interested in this aspect of neural modeling and to state the questions we can hope to resolve. Some physicists' models are briefly described in Section 3: the Hopfield model, the effect of anisotropic bonds, a model with anisotropic and highly diluted bonds, and a feedforward layered network. For the last two models one can obtain an analytic solution of the dynamics. Section 4 presents an incomplete history of learning in feedforward networks, starting with the perceptron as introduced by Rosenblatt, through the objections raised by Minsky and Papert, to the backpropagation algorithm, whose successes and limitations will be demonstrated.

2. STATEMENT OF THE PROBLEM

2.1. What Do Brains Do?

As stated above, the ultimate goal is to understand how the brain works. Of the vast number of perceptual, cognitive, and motor functions of the brain,[4,5] I list a few. First of all, it is an incredible memory device. Its storage capacity is enormous; it is able to recall information on the basis of partial or extremely noisy and even distorted input. Perception, pattern recognition, our ability to recognize and classify the objects perceived are most impressive. Some brains are capable of thought and are even quite competent at solving problems. Brains are responsible for making decisions and converting them into action. All these are fascinating aspects of the *operation* of brains. An even more fascinating and complex issue is that of *learning*: what are the mechanisms that enable brains to acquire the capacity to operate in the manner described above? Obviously, we learn while we function, and this separation into operation versus learning should not be taken too sharply. Learning is an adaptive self-organizing process; this, in turn, makes it the most intriguing, complex, and difficult issue to understand.

As stated in the Introduction, I do not see much real progress being made toward understanding the problems mentioned. However, quite a lot of interesting and useful work is being done, aimed at constructing "machines" that function in a manner that resembles some aspects of the operations of a brain. By "machine" I do not mean necessarily an assembly of clogs and wheels, but rather computer algorithms or models, and possibly, but not necessarily, their optical or electronic implementations. Most of the models or algorithms share a few common features that represent the attempt of the modelers to reflect biological reality, to which we now turn.

2.2. Neurobiology in a Nutshell

The human brain contains on the order of 10^{12} nerve cells or neurons.[5] A typical neuron is represented schematically in Fig. 1. Three distinct regions are identified: the cell body (soma), whose diameter is on the scale of tens of micrometers (up to 80 μm for pyramidal cells), the dendrites, and the axon. The dendrites form an intricate tree that serves as the

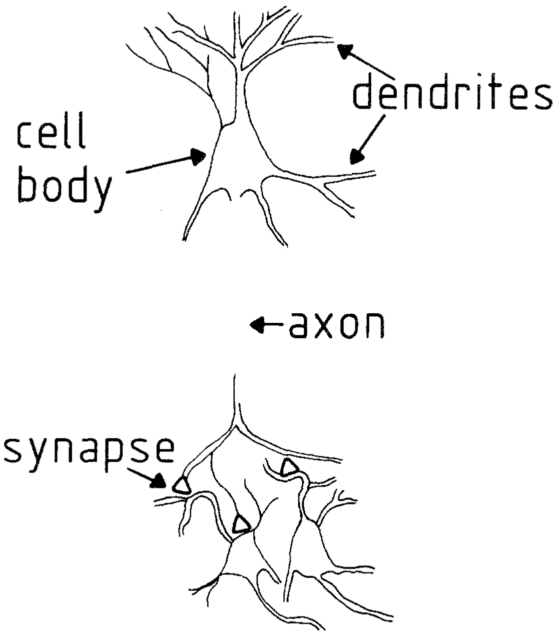

Fig. 1. The three parts of a neuron: the dendritic tree, cell body, and axon. The branches of the axon end at synapses (represented by triangles), by means of which cells communicate. (Adapted from Kandel and Schwartz.[5])

main input terminal of the cell. The single axon, whose length and diameter vary widely, serves as the output channel. Cells communicate by means of synapses, indicated as small triangles in Fig. 1. A pyramidal cell will have on the order of 10^4 synapses; for special cases, such as the Purkinje cell of the cerebellum, the number of contacts may be as high as 150,000! Synapses are contacts between terminals of the axon of one (presynaptic) cell and the dendrites (or cell body) of another (postsynaptic) cell. A schematic representation of a synapse is shown in Fig. 2. This is a chemical synapse; these are prevalent in cortex, and are believed to be "responsible" for the brain's most important attribute—the ability to learn, i.e., the capacity of plastic modification. Chemical synapses are strictly unidirectional. The means of communication is by release of chemical transmitters by the nerve terminal, which diffuse across the synaptic cleft to receptors embedded in the membrane of the postsynaptic cell. Release of the transmitters is triggered by the arrival of an electric nerve impulse at the nerve terminal. This impulse is the result of an action potential that travels down the axon, away from the cell body. The action potential is a sharp spike (temporal

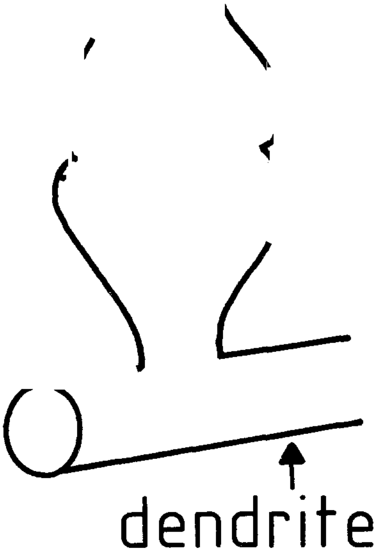

Fig. 2. Schematic representation of a synapse. Upon arrival of action potential, neurotransmitters are released from the presynaptic cell (upper part) and diffuse across the synaptic cleft to receptors on the postsynaptic cell (lower part). (Adapted from Kandel and Schwartz.[5])

extent of about 1 msec) of depolarization. At rest, there is a potential difference of about 65 mV across the cell membrane (negative inside). During passage of an action potential the polarization is briefly reversed (see Fig. 3); this is due to opening of ionic channels in the membrane of the axon. The action potential occurs when a cell "fires," and it is an "all-or-none" phenomenon: its height and duration are fixed. Variation of the firing of a cell takes the form of increased frequency of action potentials. The speed at which the action potential travels down the axon depends on its diameter; characteristically, it is on the scale of 1 m/sec. The typical time scale for neuronal activity is in the millisecond range. Whether a neuron fires or not is determined by the synaptic potentials that appear at its "input terminals" due to the arrival of chemical transmitters, which, in turn, were released as a result of the activity of the presynaptic cells. Hence, neuronal activity is the result of the weighted integration of the activities of other cells; the weights are determined by the synaptic efficacies. Synapses may be strong or weak, excitatory or inhibitory (in the latter case firing of the presynaptic cell inhibits activity of the postsynaptic one). Integration of all synaptic potentials takes place at the trigger zone (usually the part of the axon adjacent to the soma).

The only purpose of this telegraphic (and grossly simplified) description of neurobiology is to present a few facts so that the extent to which various models do or do not adhere to biological observations can be appreciated.

2.3. The Elements of Modeling

The picture of neurobiology presented above is too complex for physicists to model. The main conclusion I can draw from this description, to be used as a "working definition" of the brain as the object of modeling,

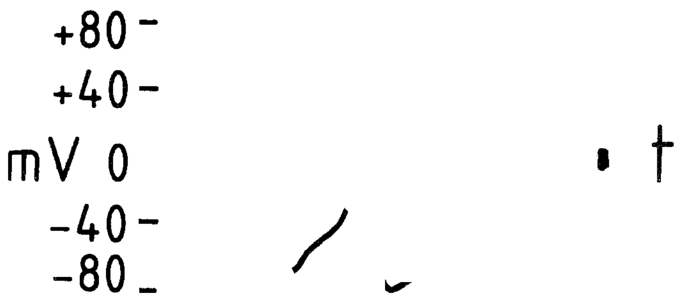

Fig. 3. An action potential, as measured at a point along the axon of a neuron. The temporal extent is on the order of milliseconds.

is as follows: *The brain is an assembly of a very large number of intricately coupled degrees of freedom, with complex dynamics.* I view our role as physicists as one of trying to study, classify, and understand such dynamic systems.

At this point I make a few comments about modeling in general, and in this field in particular. All models are supposed to be a simplified representation of some reality. The main reason for simplification is the hope that analytic statements can be made. This is indeed achieved sometimes, but in the process the model may have lost any resemblance to a real physical system. This does not necessarily mean that the model and its solution are useless or senseless. Many examples to the contrary spring to mind, especially in statistical mechanics. To mention only a few, modeling a magnet by a two-dimensional lattice of Ising spins with only nearest neighbor interactions must have appeared as senseless to the naive observer as modeling ice by arrows placed on the edges of a square lattice. Nevertheless, a number of basic scientific issues were resolved, and many more raised, by Onsager's solution of the Ising model and, more recently, by Baxter's work. Exact solutions quite often teach us something; sometimes we only learn to ask new questions from them, but even that is important. Turning now to the field of neural networks, one indeed sometimes has the feeling that "everything goes": everyone invents his or her favorite model, there are no rules to the game, and, what is most crucial, there is not much unambiguous empirical observation to guide the modeler. All this is true, and should be borne in mind, especially when claims about possible biological relevance are made. I view making unjustified claims as much more dangerous than refusing to draw biological comparisons and conclusions, even when temptation to do the latter is strong.(6)

Real biological systems are extremely complex and difficult to model. In the process we may simplify and approximate for our convenience; however, we should try to keep as many of the basic biological features in our model as possible. Since by and large we do not know which are the features of central importance, and in any case will not be able to do much about any system that faithfully represents biological reality, we should view our efforts with all modesty and humility.

With all this in mind, we turn now to a dictionary that relates biological concepts to physical ones. The state of cell i at time t is represented by a binary Ising spin S_i, which takes the value $+1$ if the cell fires and -1 when it is quiescent. This binary representation reflects the "all-or-none" feature of the action potential and is a fairly widely accepted approximation to neuronal activity.(7) The synaptic efficacies are represented by couplings or bonds of the Ising spins: $J_{ij} > 0$ represents an excitatory synapse that determines the influence of cell j on cell i, while $J_{ij} < 0$ corresponds to inhibition. In addition, local thresholds θ_i associated with each cell translate into local fields. Cell i "decides" to fire according to the value of V_i, its membrane potential at the trigger zone, which is compared to a threshold θ_i, with the probability of firing given by

$$\text{Prob}(S_i = +1) = f(V_i - \theta_i) \tag{1}$$

where f is any sigmoid function such as shown in Fig. 4. The width of the region in which f increases from near 0 to near 1 is a measure of the stochasticity of the process, and translates to "temperature." T. The deterministic limit ($T=0$) is the step function also shown. Throughout what follows we assume the deterministic $T=0$ limit, unless otherwise stated.

The dynamics of the system is now determined by the manner in which the membrane potential of cell i at time t is generated by the activities of all other cells. The widely accepted representation of the integration described above is one of a linear weighted sum, given by

$$V_i(t+\delta t) = \sum_j J_{ij} S_j(t) \tag{2}$$

Here a discrete time dynamics is assumed. When (2) is used in the $T=0$ limit of (1), the following deterministic dynamic rule results:

$$S_i(t+\delta t) = \text{sign}\left[\sum_j J_{ij} S_j(t) - \theta_i\right] \tag{3}$$

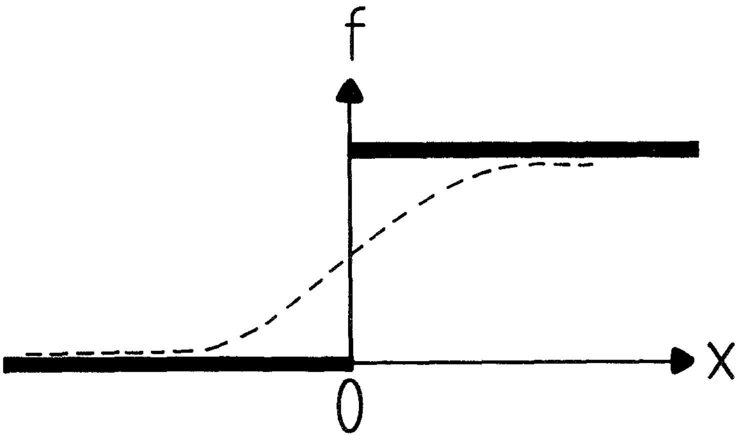

Fig. 4. The probability of firing as a function of the membrane potential. The width of the region in which the (dashed) curve rises from 0 to 1 is referred to as temperature T. The deterministic ($T=0$) limit of such a process is given by the heavy solid lines.

Table I. The "Dictionary" Used to Translate Terminology of Neural Science into Physics, and the Dynamic Rule Used

Cell i	Spin i
Fires/quiescent	$S_i = +1/-1$
Synaptic efficacies	Bonds J_{ij}
Excitatory	$J_{ij} > 0$
Inhibitory	$J_{ij} < 0$
Thresholds θ_i	Local fields θ_i

$$S_i(t+\delta t) = \text{sign}\left[\sum_j J_{ij} S_j(t) - \theta_i\right]$$

The dictionary and dynamic rule are summarized in Table I. In summary, neurons are represented by the models we consider as linear threshold elements. *A neural network is a connected assembly of such elements* (*spins*). I turn now to describe a model memory; this aspect has received most attention in the recent physics literature.

2.4. Model Memory

We would lke to construct a machine that can function as a memory device with the following properties (to be explained below):

1. Content-addressable, associative, noise-tolerant
2. Distributed, robust
3. Fast retrieval
4. Adaptive

I will explain briefly what is meant by these attributes, by means of a simple example. Think of the memory as a box that has an input window of N bits and an output window of N bits. It also has a switch that can be thrown either to Operation (O) or Learning (L) (see Figs. 5 and 6). In the learning stage a set of M key input patterns is presented to the box, and by whatever means they become associated with M (or fewer) key outputs. Denote the M input patterns by ξ_μ^{in} and the outputs by ξ_μ^{out}. During the learning stage a mapping is established between these M points of input space to the associated points in output space, as shown schematically in Fig. 5. For example, we have taught our machine to recognize the image of our friend Joe; whenever his image is presented as input, the machine responds by printing his name on the output screen. Now throw the switch to Operate. First of all, we expect the machine to recognize all the key patterns it was taught. Moreover, if a noisy, distorted, or partial key

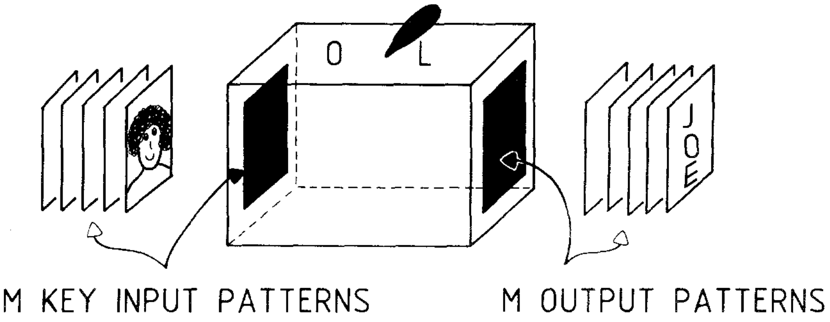

Fig. 5. The learning (L) stage of a memory. In this stage the memory is trained to associate M input patterns with M output patterns. In the example shown, presentation of a particular caricature elicits the printed characters JOE as its associated response.

pattern is presented, the machine still should respond with an output that is in some sense (to be defined below) "close" to the correct key output. These features, presented in Fig. 6, are referred to as the memory being noise-tolerant, associative, content-addressable, etc. This property results from the fact that the mapping described above contracts the (input) phase space; that is, a sizable domain of attraction in input space, around every key pattern, is mapped onto a much smaller domain in the vicinity of each key output pattern.

The next test to which we subject our machine is more severe: put it in the same room with my son, to whom we give a hammer. The result of this interaction is shown in Fig. 7, together with a plot of the quality of the output versus amount of damage, as obtained from a particular network. The

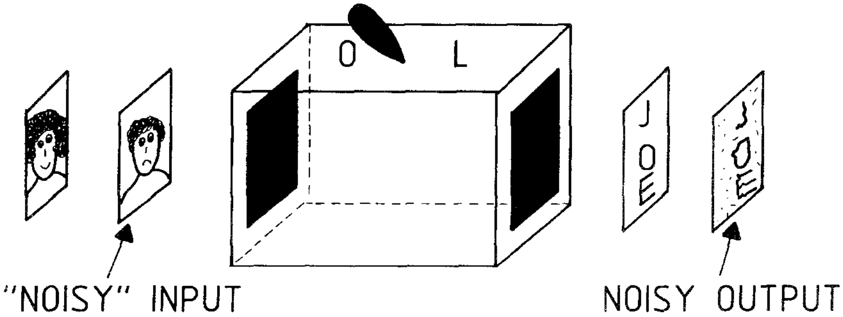

Fig. 6. Operational stage of the memory. The key pattern (the caricature of Joe) that the memory was trained to recognize is indeed perfectly recognized. Furthermore, when a different caricature of the same person is presented (shorter hair, frown vs. smile), the resulting output is still "close" to the key output. Similar response is expected for presentation of noisy (e.g., blurred) key input patterns.

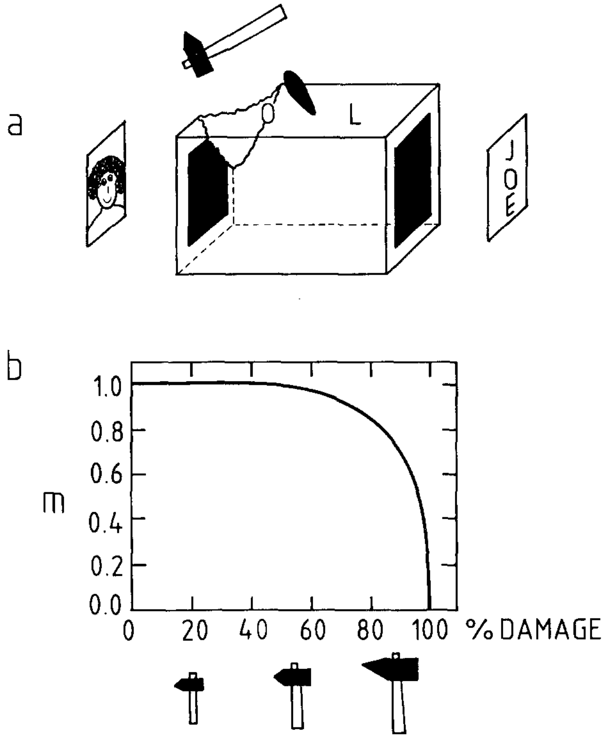

Fig. 7. (a) The memory, after interaction with a hammer-wielding opponent. (b) Performance (m is the overlap of the actual output, in response to a key input pattern, with the corresponding key output) as a function of damage. This curve was obtained for a particular network.

ability to function at a level close to perfection in spite of sizable damage goes by the name of "robustness"; it is possible to have this feature only if the memory is distributed, i.e., any particular pattern is spread over the entire network.

From the fact that fast retrieval is required, we only conclude that the network should operate in a massively parallel fashion.

The last aspect mentioned, that of adaptivity, has to do with the manner in which the network is expected to learn; discussion of the issue of learning is left to Section 4. I now discuss why physicists have been interested in modeling memory.

2.5. Why Physicists?

So, why did physicists become interested in neural networks in general, and memory in particular? The reason is that by making a few extra assumptions, the problem becomes closely related to the extensively studied problem of spin glasses.[8] To see this, note that if one uses the dynamic rule (3), but (a) flips one spin at a time, and (b) uses symmetric bonds, $J_{ij}=J_{ji}$, then we have precisely the equations that describe relaxational dynamics of Ising spins at $T=0$! That is, an energy function can be defined,

$$E=-\frac{1}{2}\sum_{ij} J_{ij} S_i S_j - \sum_i \theta_i S_i \tag{4}$$

and under the dynamic rule (3), E cannot increase; spins will flip until a stable state, i.e., a local minimum of the energy function, in which each spin is aligned with its internal field, is reached. To make contact with the concept of a memory, a further important assumption is made: (c) stable states are associated with the stored key patterns.

Since one is interested in a memory with high capacity, one would like to have an energy function that has many local minima, and *spin glasses* have precisely this property.

In fact, however, the problem of a neural network as a memory device is the *inverse* spin-glass problem. In a spin glass a typical question is phrased as follows: given a set of couplings J_{ij} [or their probability distribution $P(J_{ij})$], what is the "energy landscape"? In particular, where (in phase space) are the stable states (see Fig. 8)? On the other hand, the question relevant to memories is the opposite: given a set of points ξ_μ in phase space, can one find a set of couplings J_{ij} such that the resulting energy function will have stable states at (or near) ξ_μ? By the way, a

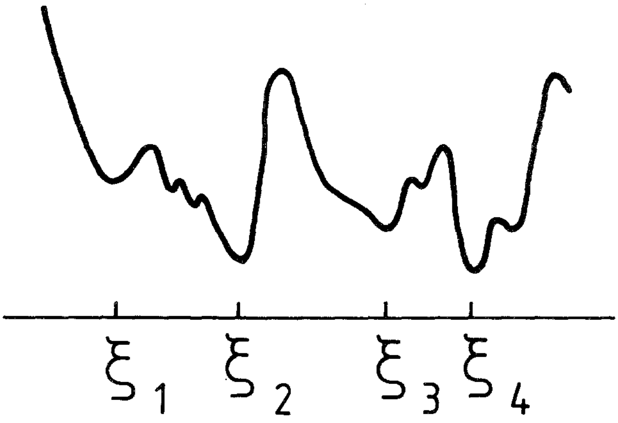

Fig. 8. Schematic representation of a potential energy landscape with many local minima ξ_ν.

dynamic procedure that finds such couplings constitutes learning in spin-glass networks. I now describe a few model neural networks studied by physicists in recent years.

3. SOME PHYSICISTS' MODELS

3.1. The Hopfield Model

The assumptions (a)–(c) mentioned above and the connection to spin glasses obtained by adopting them are due to Hopfield.[9] To define the Hopfield model completely, a choice has to be made for the bonds J_{ij} and the local fields θ_i. Hopfield chose $\theta_i = 0$, and for the couplings the "Hebbian perscription"[10]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{M} \xi_{i\mu} \xi_{j\mu} \tag{5}$$

for $i \neq j$, and $J_{ii} = 0$. To understand this choice on the simplest level, note that with these couplings the energy function can be expressed as

$$E = -\frac{N}{2} \sum_{\mu}^{M} (m_\mu)^2 + \frac{1}{2} M \tag{6}$$

where

$$m_\mu = \frac{1}{N} \sum_{i=1}^{N} S_i \xi_{i\mu} \tag{7}$$

is the overlap of the spin configuration S_i with they pattern μ. Up to an additive constant, E is now given as the negative sum of squares of the overlap of the spin configuration with the M key patterns. For a set of random key patterns a random spin configuration will typically have $m_\mu \simeq 1/\sqrt{N}$, so that (6) yields $E \simeq 0$; while for $S_i = \xi_{i\mu}$ we get $E \simeq -N/2$. So we expect that the (random) key patterns are approximate stable states of the energy function with the bonds given by (5). Hopfield studied the behavior of the network thus defined. A particular pattern v is "stored" by the network if an initial state $S_i(0)$ near ξ_v develops under the dynamic rule (3) to a stable state S_i^* whose overlap with ξ_v is large, i.e., $m_v^* \simeq 1$. The quality of the network was found to depend on the number of stored patterns; it performed well if the important parameter

$$\alpha = M/N \tag{8}$$

was not too large. To demonstrate what is meant by "good performance," I present simulations performed by Kinzel.[11] For a network of 400 spins, 30 key patterns were used, 29 of which were random, and the 30th had the geometrical shape of the letter A. (Note that since each spin is connected to every other spin, there is no unique meaning to arranging the spins in any geometrical shape; only correlations between the various patterns are meaningful.) Now the network was set in an initial state in which 30% of the spins were reversed with respect to the pattern A. In four sweeps through the network, the stable state A was reached, as shown in Fig. 9. However, it was also found that not all noisy patterns relax to the exact key pattern closest to them: sometimes the final state differs a little from the key pattern. Moreover, when the noise level is too high (i.e., the overlap of the initial state with a key pattern is too small), the network relaxes to spurious stable states that are not near the key pattern. This remarkable performance of the Hopfield network gives rise to a number of questions, such as, What is the limiting value of α for which the key patterns are stable? What is the effect of introducing stochasticity ($T>0$) to the system? What are the spurious stable states described above? How large are the domains of attraction of the various stable states under the dynamic rule (3)?How sharp are the boundaries of these domains of attraction? Some of these questions were answered in a remarkable series of papers of Amit *et al.*[12] These authors simply took the Hopfield Hamiltonian, and solved the *equilibrium statistical mechanics* of the model. This work revealed a most important property of the Hopfield model: it is exactly soluble! In the resulting phase diagram (for simplicity, I present it as obtained without replica symmetry breaking) there is a phase denoted by F in Fig. 10, at low temperatures and small number of stored patterns, in which the equilibrium state is characterized by large overlap with a single key pattern. In phase F these "ferromagnetic" memory states have the lowest free energy. In a different region (SG), a spin-glass phase, the

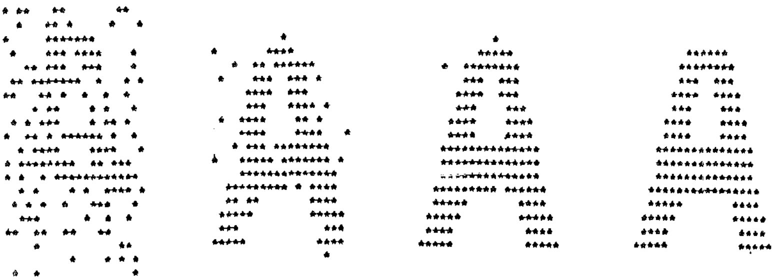

Fig. 9. Simulation of a Hopfield network with 400 spins and 30 key patterns, of which 29 are random, and one forms the letter A. An initial state is generated by flipping 30% of the spins of the state A and letting the network evolve; after four sweeps of the lattice the stable key pattern A is reached. (From Kinzel.[11])

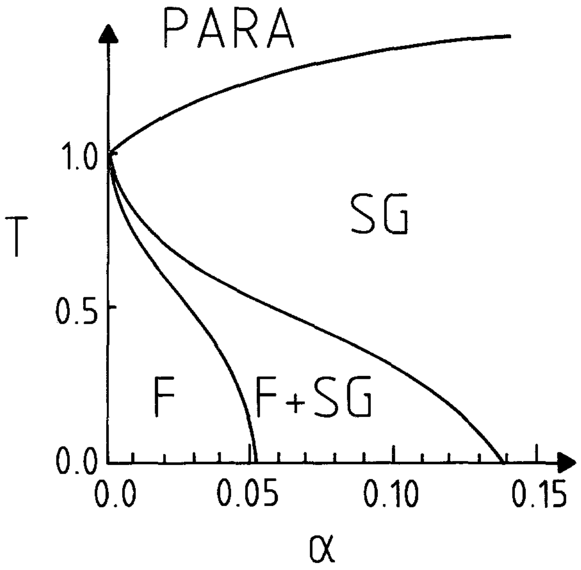

Fig. 10. Phase diagram of the Hopfield model, as obtained without replica symmetry breaking.[12] In the F phase the memory states (large overlap with one key pattern) have the lowest free energy. In the SG phase the memory states are unstable; the equilibrium "spin-glass" states have small overlaps with key patterns. In the F + SG phase the memory states are metastable; their free energy is higher than that of the spin-glass states.

ferromagnetic states are unstable, and only spin-glass states, whose overlap with key patterns is small, are stable. In the region between F and SG (denoted F + SG), both kinds of states are stable, but the spin-glass states have lower free energy. At high temperatures a paramagnetic phase is found. The problem of modeling a memory, as stated above, is a dynamic one: will a noisy key pattern flow to a stable state that is close to the (noiseless) key pattern? On the other hand, the phase diagram was derived for a static, equilibrium problem; properties are averaged over *all* states of the spin system, stable and unstable, each with its Boltzmann weight. Nevertheless, one expects and indeed finds (as confirmed by simulations) that many results of the dynamics discussed above agree with and can to a large extent be predicted by the equilibrium calculation. For example in the SG and Para phases the system does not relax to an embedded pattern. The existence of a limiting value of α has also been confirmed by simulations. For more results on the basic Hopfield network and its extensions the reader is referred to various reviews.[1,2] A number of basic properties of the Hopfield model and its solution are summarized as follows:

1. Symmetric couplings are assumed; cell i has the same effect on j as j on i.
2. The network is fully connected.
3. Functional uniformity is assumed; all cells are equivalent in receiving input, processing the information, and generating an output.
4. Input is associated with the initial state, output with the final (stable) state of a dynamic process. "Recognition" corresponds to a persistent firing pattern of cells.
5. The key patterns are random.
6. The dynamics has many spurious stable states.
7. No solution of the dynamics exists.

Even though these points should be borne in mind, especially when claims of relevance to biology are made, the Hopfield model and its solution are one of the most important contributions of physicists to the field of neural networks. A very similar model, with the same choice of couplings (5), was introduced much earlier by Little.(13) However, the Little model has parallel dynamics, there is no associated energy function, and no corresponding equilibrium statistical mechanics.

The fact that there is a Hopfield Hamiltonian allowed investigation of the attractors of the associated dynamics by first solving the equilibrium problem. Thereby(14) concepts such as the thermodynamic limit, phase diagram, and phase transitions were introduced to the field of neural networks. In addition, the fact that the model is exactly soluble has provided the field with a *standard*; various networks are compared with the Hopfield model in terms of their properties, phases, and storage capacity. Having exactly soluble models should also raise the standards concerning the quality of work done on neural networks outside the physics community.

I now describe two interesting extensions or modifications of the Hopfield model. The first addresses analytically the effect of asymmetric bonds; the second solves exactly the dynamics of an asymmetric and extremely diluted Hopfield network.

3.2. Hopfield-Type Models with Anisotropic Bonds

The assumption of symmetric bonds is probably one of the most biologically unrealistic features of the Hopfield model. Hence, an obvious direction for extension or modification is lifting this restriction. The first analytic attempt in this direction was made by Hertz *et al.*(14) These authors considered the dynamics of a network in which the discrete Ising-like spin variable is replaced by a continuous one, ϕ_i. The dynamic equations used are

$$\frac{d\varphi_i}{dt} = -r\varphi_i - u\varphi_i^3 + \sum_j w_{ij} J_{ij} \varphi_j + \theta_i + \zeta_i \tag{9}$$

for all sites $i = 1,..., N$. The variables ζ_i represent a Gaussian noise, the couplings J_{ij} are of the form (5) as used by Hopfield, but the new bond variables w_{ij} introduce asymmetry in the system. For *each direction* of each bond, w_{ij} and w_{ji} are independently chosen from the distribution

$$P(w) = p\delta(w-1) + (1-p)\,\delta(w) \tag{10}$$

That is, the bond that determines the effect of cell j on i is either zero or given by J_{ij}. Hertz *et al.* found that the spin-glass phase becomes unstable by the anisotropy, while the memory (i.e., ferromagnetic) states do not become degraded. However, it should be noted that as soon as anisotropic bonds are introduced, cycles of various periods may appear as stable states of the network. With asymmetric bonds there is no Hopfield Hamiltonian, no meaning of equilibrium statistical mechanics, and no exact solution is available.

At this point, however, Derrida *et al.*[16] noticed that if in addition to the asymmetry introduced by independently choosing w_{ij} and w_{ji} according to (10), one also takes the limit of extreme dilution, e.g., $p \to 0$ *as* $N \to \infty$, with

$$p \ll (\log N)/N \tag{11}$$

and the number of stored patterns is given by

$$M = \alpha p N \tag{12}$$

then the model can be solved exactly! This is quite interesting; the model is defined in terms of its dynamics, as given by Eq. (3), and an exact solution means that its *dynamics* is soluble. The "canonical" problem of dynamics can be stated as follows:

Given an initial state $S_i(0)$, such that its initial overlaps with the key patterns are given by

$$m_\mu(0) = 0 \qquad \text{for} \quad \mu \neq \nu; \qquad m_\nu(0) > 0 \tag{13}$$

and the spin configuration develops in time, what are the values of the overlaps $m_\mu(t)$?

With the initial condition (13), Derrida *et al.* find that only $m_y(t) = m(t) \neq 0$, and the solution $m(t)$ is given, for deterministic discrete time dynamics, in the form of a map, or recursion relation:

$$m(t+\delta t) = (2/\sqrt{\pi}) \int_0^{m(t)/(2\alpha)^{1/2}} dy \exp(-y^2) \tag{14}$$

That is, the overlap at time $t+\delta t$ is determined by its value at time t! Hence the long-time behavior of the model is governed by the stable fixed points of (14). The fixed point value m^* is shown as a function of α in Fig. 11. The network can function as a memory for $\alpha < \alpha_c = 2/\pi$; in this regime there is a stable fixed point with $m^* > 0$, while for $\alpha > \alpha_c$ the only fixed point has vanishing overlap, and hence no recall of the key pattern. The transition is continuous, i.e., the limiting overlap goes to zero as $\alpha \to \alpha_c$. It is interesting to note that the recursion relation (14) has appeared previously;[11,17] it gives exaxtly the first time step of a different model, but is only approximate for later times. This model is a layered feedforward neural network, whose dynamics is also exactly soluble, and to which we now turn.

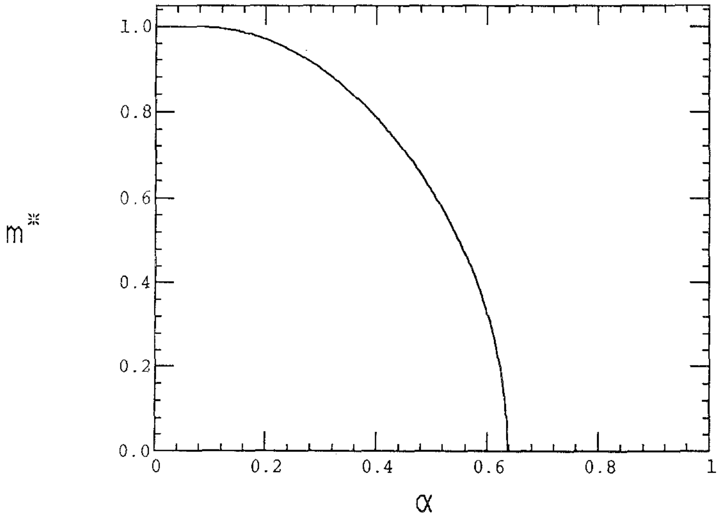

Fig. 11. Fixed point m^* of the recursion relations (14) versus α [defined in Eq. (12)]. For $\alpha > 2/\pi$ the only fixed point is $m^* = 0$.

3.4. A Feedforward Layered Neural Network

Feedforward neural networks have been studied for many years, and a brief history of the subject is given in Section 4. A layered feedforward neural network, also called multilayer perceptron, was recently introduced to the physics literature as a model associative memory.[17] Introduction of this model was motivated by a number of questions we hoped to resolve by studying it. We thought of feedforward networks as the antithesis of the point of view represented by the Hopfield model. We wanted to know whether the properties of the Hopfield model that made it an impressive memory depended at all on some of the assumptions made. In particular, we wanted to know whether having symmetric bonds, and therefore an underlying Hamiltonian, is an important ingredient of the design of a network. Another aspect is that of *feedback*; it is obviously of central importance for learning, but is it important for the operation of the network, as claimed? To our surprise we found that a simple feed-forward layered network has most of the attractive properties that characterize the performance of the Hopfield model. Moreover, our layered network, as discovered subsequently, is also analytically soluble;[18] an exact solution of its dynamics was obtained. We now turn to describe the architecture and operation of the layered network, summarize the comparison with the Hopfield model, and present a few results derived from the solution.

The basic units of our network are binary linear threshold elements of the kind described above. These elements change their state in discrete time steps, according to the dynamic rule (3) with $\theta_i = 0$, as in Hopfield's model. Moreover, the spins are connected by Hebbian bonds of the form (5). However, the architecture of the network is different: it has $l = 1,..., L$ layers, with $i = 1,..., N$ spins (cells) S_i^l per layer. Each cell is *connected to all* cells of the neighboring layers (see Fig. 12). The bonds are, however, *unidirectional*: the state of layer $l + 1$ is determined by the state (at the previous time step) of layer l. In the deterministic limit, the dynamic rule is given by the properly modified form of (3)

$$S_i^{l+1} = \text{sign}\left[\sum_{i}^{N} J_{ij}^l S_j^l\right] \tag{15}$$

Note that all spins in a given layer are updated simultaneously. The couplings or bonds J_{ij}^l are chosen by (5)

$$J_{ij}^l = (1/N) \sum_{\nu=1}^{\alpha N} \xi_{i,\nu}^{l+1} \xi_{j,\nu}^l \tag{16}$$

where $\nu = 1,..., \alpha N$ are the stored key patterns.

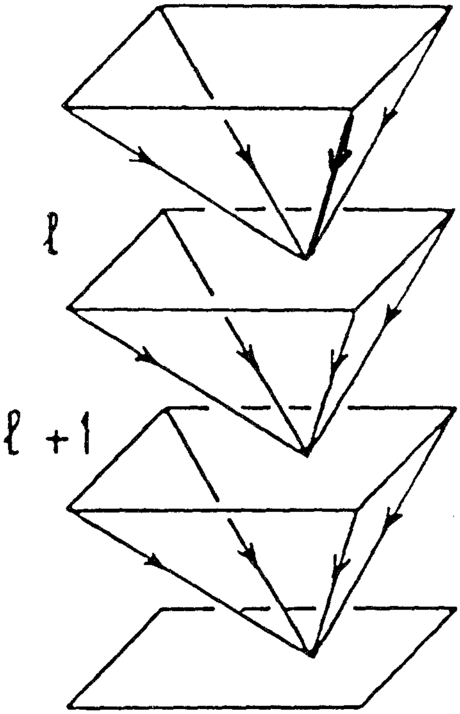

Fig. 12. A layered feedforward network. The state of a cell in layer $l+1$ is determined by the states of all cells of layer l. Input patterns are presented to the first (top) layer, and the output is read out from the last.

It should be noted that now each key pattern carries a layer index. This is a central feature that characterizes this class of model neural networks; it has of conceptual as well as technical significance. Conceptually, it represents the fact that only the first layer of a network is in direct contact with the "external world," and hence only on the first (input) layer are the representations of the key patterns externally dictated. On all subsequent layers the system is free to choose an internal representation of any key pattern. In particular, with proper self-organization of internal and output representation, obtained in an iterated learning procedure, we have shown[17] that the network is capable of perfect recall of key patterns and excellent recognition of noisy input patterns. No such iterated learning is allowed in the network reviewed here. We assume that the internal representations $\xi^l_{i,\nu}$ of the key patterns are randomly chosen; all $\xi^l_{i,\nu} = \pm 1$ with equal probability. It is precisely this fact of the independent choice of representations on different layers that technically allows analytic solution of our model. The solution yields information on the time development of the system: the first layer is set in an initial state, which determines the state of the next layer at the next (discrete) time step, and so on. Hence, obviously our model can also be viewed as one with a single layer of cells, but time-dependent couplings; i.e., a cellular automaton in which the dynamic rule is a (random) function of time.

By exact solution of our model we mean the same as described above; i.e., given an initial state with overlap m^1 with a key pattern, we have a recursive formula that yields the overlap on any subsequent layer/time step (averaged over all key patterns ξ). The recursions have the form

$$m^{l+1} = (2/\pi)^{1/2} \int_0^{a^l} \exp(-y^2/2)\, dy = \operatorname{erf}(m^l/(2\alpha q^l)^{1/2}) \tag{17}$$

where

$$q^{l+1} = 1 + (2/\alpha\pi) \exp[-(a^l)^2], \qquad a^l = m^l/(\alpha q^l)^{1/2}$$

with m^1 set by the initial state and $q^1 = 1$. The equation for m^2 was obtained previously[11,17] and is the same as Eq. (14) of Derrida *et al.*[16] Here, however, it is exact only for going from the first to the second layer; for the evolution on subsequent layers it is only approximate. This approximation is modified in (17) by the fact that the "width" parameter q^l also changes with l, as if the effective value of α were renormalized by increasing layer index/time.

The long-time, large-L behavior of the overlap is determined by the fixed points of (17), $m^l = m^*$ (the recursion for q is parasitic, dragged by m^l). The solution of the fixed point equation is plotted versus α in Fig. 13. The $m^* = 0$ fixed point is always stable; for $\alpha < \alpha_c = 0.269$, however, two additional solutions exist. The branch with higher values of m^* is stable and the lower branch unstable. For the relevant parameters of the problem, namely α and the initial overlap m^1, the dynamics governed by this fixed point structure gives rise to the phase diagram of Fig. 14. For $m^1 > m_c^1$ the limiting overlap $m^* \neq 0$, and its value is given by the upper branch of Fig. 13. As the boundary of this phase is crossed, m^* jumps *discontinuously* to zero; the transition is first order.

Such discontinuous change of the limiting overlap appears to be a fairly common feature of various neural network models.[19] In many instances it is not easy to see this in numerical simulations. One may find that for a finite-sized system the average m^* (averaged over the various patterns ξ, for example) is a smooth function of some variable. However, as the size is increased, the function may (slowly) become steeper. In such

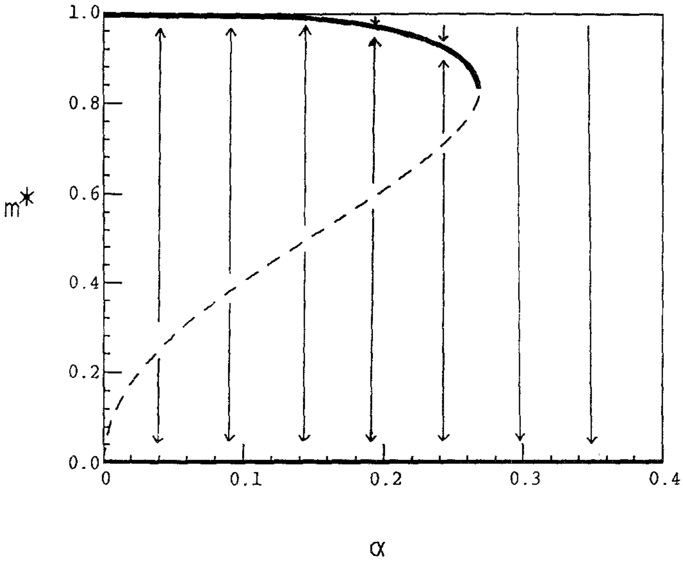

Fig. 13. Fixed points of the recursions (17) versus α. Heavy lines indicate stable fixed points that govern the asymptotic overlap m^*. For $\alpha > \alpha_c = 0.269$, only the $m^* = 0$ fixed point is stable. Fixed points indicated by the dashed line are unstable.

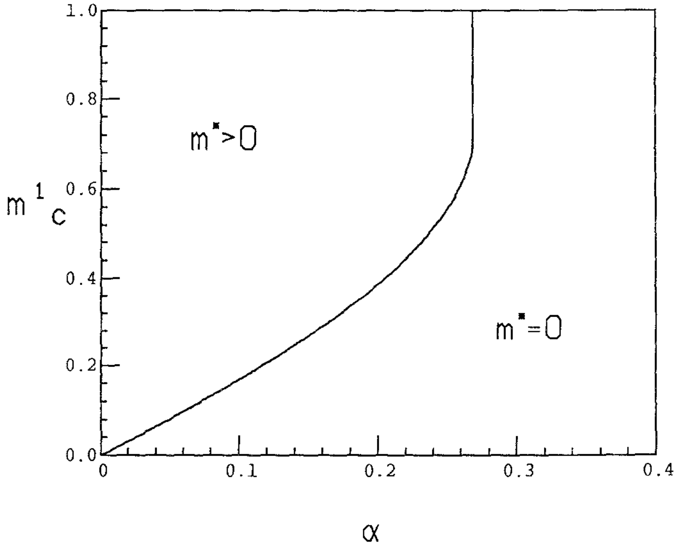

Fig. 14. Two phases, one with high limiting overlap and one with $m^* = 0$, are separated by a first-order line. For initial overlap $m^1 < m_c^1$ the network flows to $m^* = 0$. In the memory phase the limiting overlap is given by the upper branch of Fig. 13.

cases it is more revealing to consider the *histograms* of m^*, as demonstrated in Fig. 15. As can be clearly seen, especially for $N = 100$, even though the average m^* is a smooth function of m^1, the histograms reveal a pronounced bimodal distribution, with the weights of the two peaks varying relatively slowly with m^*. This, however, is a finite-size effect; as the system size increases, the "jump" from the distribution centered on

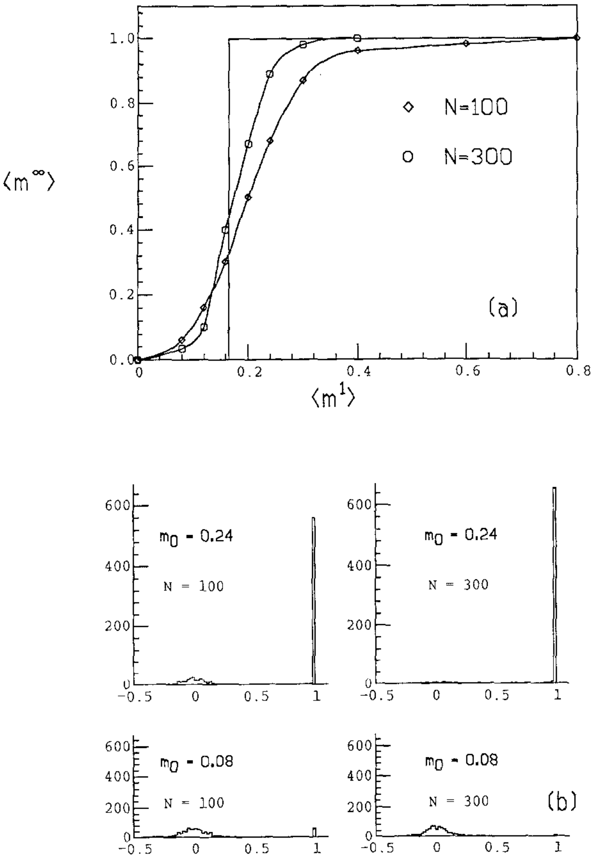

Fig. 15. (a) Asymptotic versus initial overlap for $\alpha = 0.10$. The step function is obtained from the exact solution (17); averages over simulations yield the smooth curves plotted. Note that as system size increases, the simulations approach the analytic result. (b) Histograms of the simulations obtained from ensembles of networks with different key patterns and initial states (all with fixed initial overlap). The histograms present the fraction of cases that reach a limiting overlap (of the value given on the horizontal axis). For an initial overlap of 0.24 [larger than the critical m_c^1 $(\alpha = 0.1)$ of Fig. 14] most members of the ensemble flow to final overlap near 1. This fraction increases with system size N. For initial overlap of $0.08 < m_c^1$ the "false" peak at $m \sim 1$ shrinks as the system size increases.

$m^* = 0$ to that near 1 becomes sharper. Finite-size effects are relatively unimportant *away* from the transition region; in this regime (see Fig. 16), excellent agreement with the exact solution (valid for $N \to \infty$) is obtained, even for N as low as 200. As discussed above, finite-size effects become important as the phase boundary is approached. This can be seen to some extent in Fig. 16; while the upper two curves (corresponding to initial overlaps m^1 well within the $m^* \neq 0$ phase) exhibit perfect agreement with simulations, development from $m^1 = 0.2$ does deviate slightly from the exact solution. This is due to the fact that near m_c^1 some members of the simulated ensemble flow to the "wrong" phase. However, as N increases, the relative weight of these "errors" decreases. The lower curve of Fig. 16 shows another interesting effect. Even though the final overlap is zero, initially the overlap increases. Similar increase was found for the first time steps of the Little[20] and Hopfield[21] models.

It is interesting to note that the model exhibits "critical slowing down." Relaxation to the limiting value of m^* is exponential; $m^l - m^* \sim$

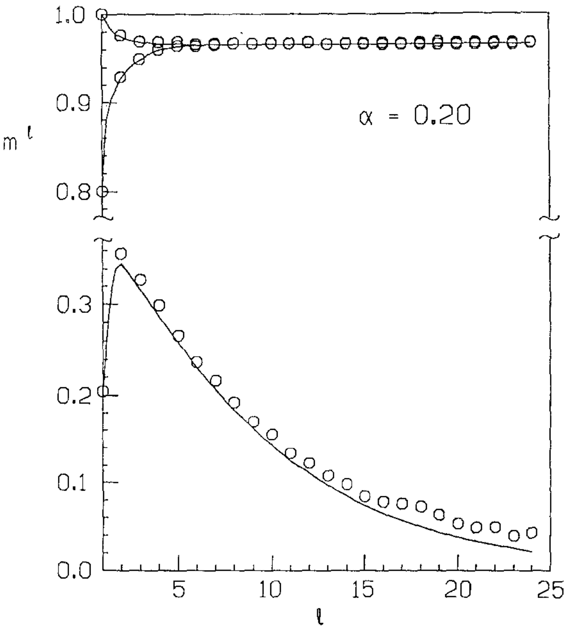

Fig. 16. Overlap m^l as function of layer index l (or time). For the two upper curves the initial overlap is higher than m_c^1. For $N = 200$ the averages obtained from simulations (circles) agree perfectly with the analytic curves. The lower curve starts at an initial overlap of 0.2, less than m_c^1 ($\alpha = 0.20$). Deviations of the circles obtained from simulation from the analytic curve are due to finite-size effects.

$\exp(-l/\tau)$. The relaxation rate τ is determined by the recursion relations (17), linearized near m^*. Since as $\alpha \to \alpha_c$ two branches merge, one stable and one unstable, the fixed point at α_c must be marginally stable, and hence τ must diverge. Indeed, we find[18] that $\tau \sim (\alpha_c - \alpha)^{-1/2}$.

To summarize our discussion of this feedforward layered network, we list some of its properties as compared to the Hopfield model.

1. All couplings are unidirectional.
2. Only cells in adjacent layers are connected; full connectivity between neighboring layers.
3. Functional asymmetry: the first (last) layer serves as the input (output) terminal of the network.
4. Input sets the first layer in an initial state: the state of the last layer is read out as output.
5. Random key patterns are used as input, output, and internal representation. The "image" of a key input pattern on subsequent layers can be determined[17] by a self-organizing learning procedure.
6. There is no meaning to "stable states"; however, every final state of the last layer is read out, whether it is near a key output pattern or not. This may pose the same difficulties as the spurious states of the Hopfield model.
7. The dynamics of the network is exactly soluble.

The layered feedforward network described above as a model melory is also called a *multilayer perceptron.* The perceptron has a long, interesting, and instructive scientific history, which is the topic of the next section.

4. PERCEPTRONS

4.1. Definition

Imagine a machine that is able to recognize our friend Joe from Section 2. That is, as the "eye" of the machine is pointed in various directions, a warning light goes on whenever Joe appears in its field of vision. For any other image this light is off. This can be viewed as a classification task: the machine classifies all possible inputs to either Joe or non-Joe. Furthermore, the machine acquires this skill in a training session with its master, in an adaptive fashion! And we still have not mentioned the most amazing aspect of this machine: there is a convergence theorem associated with its learning algorithm!

The perceptron,[22] invented and studied by F. Rosenblatt in the 1950s and 1960s, was claimed to be precisely such a machine. It is represented in Fig. 17.[23] The image is projected onto a screen, which is divided into square cells. A cell will be either on or off, depending on, say, the amount of integrated light on it. This set of 0's and 1's is viewed by a single layer of binary decision elements, represented as small, cubic boxes in Fig. 17. Every box "sees" a preassigned *finite* field of vision, i.e., set of cells of the screen. As a simple example (not drawn), imagine that each box sees a 2 × 2 square of cells of the screen, and each such square has a single box associated with it. Now these binary decision elements respond to the pattern that appears in their field of vision by generating an output of either 0 or 1. Denote the output of decision element i as S_i. We get $S_i = 1$ if and only if the pattern seen by box i belongs to its "truth class." The truth class of each box is hard-wired and does not change during the learning process (described below). For example, in the case of the 2 × 2 field of vision we may choose the set of patterns shown in Fig. 18 as the truth class: occurrence of one of these patterns in the field of element i gives rise to

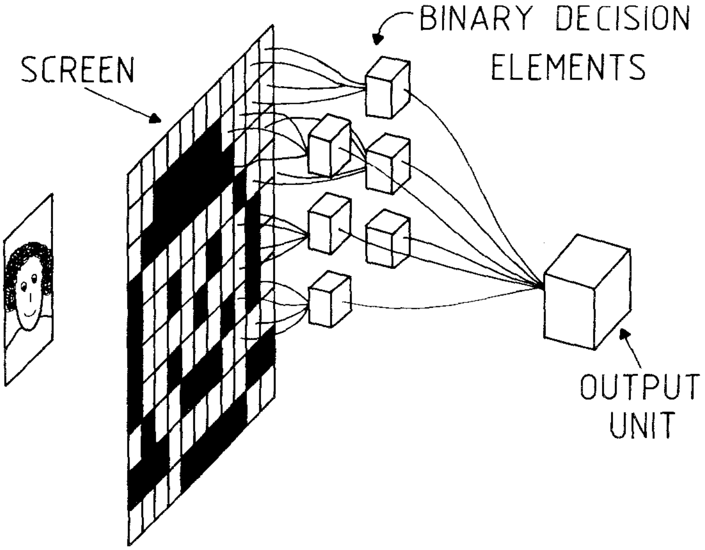

Fig. 17. The Perceptron (adapted from Ref. 23). The image presented is projected onto a screen. Information is fed into local decision elements, whose binary (0, 1) response is determined by the pattern in their field of vision. The weighted sum of these responses determines the state (0, 1) of the single output unit.

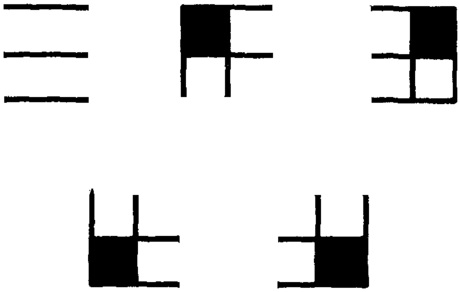

Fig. 18. A particular "truth class" of the local decision elements. Whenever one of the patterns shown appears in the 2 × 2 field of vision of decision element i, it responds by setting its output to $S_i = 1$.

$S_i = 1$, and $S_i = 0$ otherwise. Hence, every image projected on the screen draws an array S_i from the single layer of binary decision elements. This array is fed into the single output unit that represents the bulb that should light up in response to Joe. The output unit is a linear threshold element, which takes the value $S^{\text{out}} = 0, 1$ according to the rule (Θ is the Heaviside function)

$$S^{\text{out}} = \Theta\left(\sum_i J_i S_i - \theta\right) \tag{18}$$

There is quite a lot that such a simple perceptron can do. For example, using the truth class of Fig. 18 with $J_i = 1$ and $\theta = N - 1$, where N is the number of decision elements, the output unit will have the value $S^{\text{out}} = 1$ only in response to an input in which no two adjacent black squares are allowed. For many other examples see Ref. 23.

The next question concerns one's ability to determine a set of couplings J_i that will ensure performance of an assigned task. This is accomplished in the course of a *learning session.*

4.2. The Perceptron Learning Rule and Convergence Theorem

To start the learning session, the perceptron is initialized; all bonds are assigned some random value. Now two stacks of input images are prepared; one contains images (of Joe) that are supposed to draw $S^{\text{out}} = 1$, and the other images (non-Joe) that should elicit the opposite response. Now randomly pick an image from one of the stacks, and present it to the perceptron. The layer of decision units translates the image to an array of $S_{\text{out}} = 0, 1$, which determines via Eq. (18) the corresponding response of the output unit. Depending on this response, we now allow modification of the bonds, according to the *perceptron learning rule*, which can be stated as follows[24]:

1. Correct response generates no change.

2. Miss (image of Joe draws $S^{\text{out}}=0$) causes modification according to

$$J_i \rightarrow J_i + \eta S_i \tag{19a}$$

3. False alarm (response of $S^{\text{out}}=1$ to non-Joe input) gives rise to

$$J_i \rightarrow J_i - \eta S_i \tag{19b}$$

with $0<\eta<1$. This learning rule is easily understood; a miss means that the sum $\sum J_i S_i$ is too small. To increase it, those J_i that can raise the value of the sum (for the pattern just presented) are increased. The opposite is achieved in the case of a false alarm. This extremely simple learning rule has been reinvented and renamed many times since. The most interesting aspect of this rule is the existence of the associated *perceptron convergence theorem*. It states the following[24]:

If there exists a solution J_i^*, then the perceptron learning rule will converge to some solution in a finite number of steps for any initial choice of the couplings.

Having a simple, transparent, local learning rule and an associated convergence theorem is quite impressive. In fact, apparently Rosenblatt was so impressed by the perceptron that on various occasions he made very strong statements about its power and potential. These statements have sufficiently disturbed and upset a number of Rosenblatt's colleagues that they spent lots of time and effort investigating the properties of perceptrons. These endeavors culminated in a book by Minsky and Paper[24] in which they demonstrated that there is a large class of tasks that perceptrons are *unable to perform*. The convergence theorem starts with an "if"; when there is no solution, obviously no learning algorithm will converge to one! And, alas, the class of unsolvable problems is quite large. The source of this class of unsolvable problems, as well a possible way to overcome it, is discussed next.

4.3. Unsoluble Problems and Their Solution

To understand the source of the above-mentioned unsolvability, one should first understand how the perceptron solves a classification problem. Each input pattern is translated by the N binary decision elements into an array $S_i = 0, 1$. Hence, the two groups of inputs (Joe and non-Joe) can be represented as points in an N-dimensional space. The perceptron tries to find a hyperplane in this space that separates the two groups. Obviously, not every two groups of points is separable by a linear manifold, hence the class of problems that cannot be solved by the perceptron. The most notable example of such a problem is that of identifying connected versus disconnected figures.(24) Here I choose to present the simplest example, that of an XOR gate.(25)

Consider the perceptron of Fig. 19a; it has two decision elements and one output element. Suppose one is interested in a solution of the XOR problem: that is, whenever one and only one of the inputs is 1, the output has to be 1, and 0 otherwise, as shown in Fig. 19b. A solution means that there is a set J_1, J_2, and θ such that $S^{\text{out}} = 1$ if and only if $J_1 S_1 + J_2 S_2 > \theta$. To see that such a solution cannot be found, note Fig. 19c: input space consists of four points, and the two solid circles at (1, 0) and (0, 1) cannot be separated by a straight line from the two open circles at (0, 0) and (1, 1). This is the prototypical failure of perceptrons that has sharply reduced interest in them since Minsky and Papert published their book. The natural question to ask is, How can this problem be overcome? One simple solution is provided by introducing *hidden units*, thereby extending the network to *multilayer* perceptrons. Indeed, Fig. 20a demonstrates how inserting a hidden layer of three cells between input and output, with the couplings and thresholds as indicated, produces a solution of the XOR problem. To see how this was accomplished, note Fig. 20b: the four input points are now embedded in a three-dimensional space defined by the eight

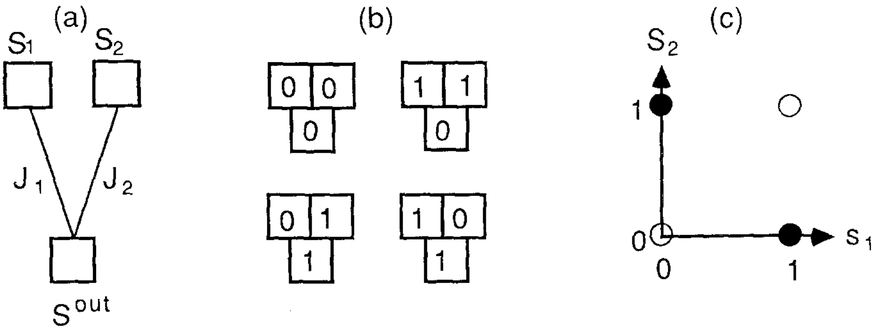

Fig. 19. (a) A Perceptron with two input units and a single output attempts to solve the XOR problem. (b) The association of inputs and required in order to solve this problem. Linear threshold elements that operate according to Eq. (18) are unable to solve the problem. (c) No straight line can separate the four points of in the manner indicated (solid versus open circles).

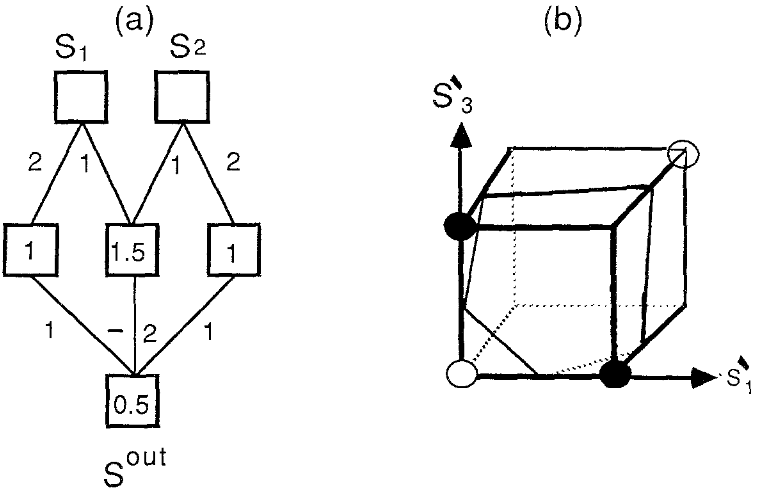

Fig. 20. (a) The Perceptron of Fig. 19 with a hidden layer of three cells S_i'. With the indicated values of the weights J_i (next to the connecting lines) and the thresholds (in the boxes) this perceptron solves the XOR problem. (b) This is accomplished by mapping the four points of Fig. 19(c) onto the four points indicated here; clearly, separation (by a linear manifold) into the required groups is now possible.

states of the three hidden units. These four points are now easily separated by a linear manifold (plane) into two groups, as desired. This simple example demonstrates that adding hidden units increases the class of problems that are soluble by feedforward, perceptronlike networks. However, by this generalization of the basic architecture we have also incurred a serious loss; there is no longer a learning rule.

4.4. Learning in Multilayer Perceptrons

The canonical problem of learning in layered feedforward networks can be stated as follows. A learning rule is an algorithm that can be used to locate a set of bonds and thresholds that will map a given set of points in input space (i.e., on the first layer) to another given set of points in output space. Solution of this problem was simple in the case of the single-layer perceptron; there it was easy to identify the "culprit," i.e., the bond that was too strong or weak and thereby helped produce a wrong answer. For multilayer networks it is not clear which of the bonds that connect input to output is responsible for mistakes and successes. This is called the "credit assignment problem." It appeared to be solved by the recent introduction of the *back propagation* algorithm by Rumelhart *et al.*(25,26) This algorithm has appeared before under other names,(27) but these authors have eloquently demonstrated its power by showing that it can be used to locate solutions for a number of problems. One of the most elegant of these is the solution of the *parity problem*. One wants a classification scheme that differentiates inputs with an even number of 1's from those with an odd number. This problem is a fairly difficult one, since changing any single input unit throws the output from one class to the other. The XOR problem is a parity problem with two input units.

Rumelhart *et al.*[25] studied this problem with up to eight input units. A solution found by the backpropagation algorithm is shown in Fig. 21. In this solution the network found internal representations that serve as the column of mercury in a thermometer; all 1's are bunched to the left of the intermediate layer. That is, the state of the intermediate layer is determined by the *number* of 1's in the input, irrespective of where they occur. The alternating signs of the bonds from the second layer to the output cell ensure that the weighted sum of the second layer's activities will be either 0 or 1, depending on the parity of the number of 1's. Such a solution was found by the network, for four input cells, after about 3000 presentations of each (of the 16) input patterns.

Convergence of the learning algorithm to a solution that represents a certain logic is quite fascinating. In particular, the networks ability to

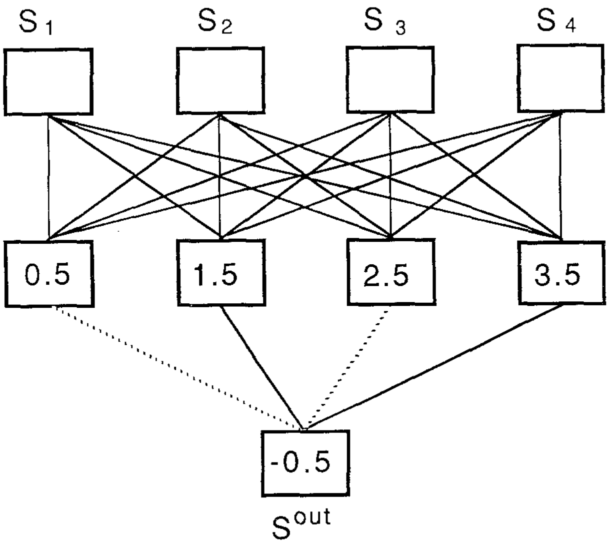

Fig. 21. A solution of the parity problem found by the backpropagation algorithm (as reported in Ref. 25). Solid connecting lines represent weights $J_i = 1$; dotted lines correspond to $J_i = -1$; numbers *in* the boxes stand for thresholds.

generate meaningful internal representations on the hidden layer is an indication that more progress along such lines may be expected.

To summarize: so far we have seen that introducing internal layers enlarges the class of problems soluble by feedforward networks; the backpropagation learning algorithm seems to resolve the credit assignment problem, and was shown to find "logical" solutions of nontrivial tasks.

However, unfortunately there is bad news following the good news. Something important is still missing: *there is no convergence theorem.* In fact, no theorem of the kind proved for the single-layer perceptron exists for backpropagation. Backpropagation is essentially a minimization procedure; it minimizes a cost function that measures the deviation of actual outputs of the network from the desired ones, over the space of weights and thresholds. As with most minimization procedures, it may get stuck in local minima, and there is no way to predict *a priori* either its rate of success, i.e., when an absolute minimum is found, or the time needed to attain it.

Finding a learning algorithm for multilayer systems for which a convergence theorem holds, or, at least, one for which the probability for success (and the time it takes) can be estimated, appears to be the most important immediate challenge of the field of neural networks.

5. SUMMARY

In this brief overview I tried to present a few aspects of research in the field of neural networks. Naturally I emphasized the contribution and possible role of physics in posing questions and getting answers to them. However, I also made an attempt to review a particular limited direction of research conducted by nonphysicists. As to the contribution of the physics community, I consider introduction of the Hopfield model, with its Hamiltonian and associated statistical mechanics, to be of central importance. This model has definitely enriched the field; new concepts were introduced and the notion of soluble models, in the sense of statistical mechanics, will probably play a role in future developments. I described how the extent to which attractive properties of the Hopfield model depend on a number of its underlying assumptions was investigated by solving *the dynamics* of a layered feedforward network. I believe that the most relevant area of potential contribution by physicists to neural networks concerns their dynamics. Enumeration and classification of attractors and studying means of controlling their size and position in phase space constitute only a partial list of tasks for which a physics background is useful. It is hoped that research along such lines will help address what appears at the moment as the central question of the field, i.e., that of *learning*. In particular, it would be gratifying if physicists could assist in the quest for learning algorithms for which convergence theorems can be proven, applicable to architectures that can solve wide classes of problems.

ACKNOWLEDGMENTS

The material presented in this article was assembled while I was visiting the Institute for Theoretical Physics at the University of California, Santa Barbara; I wish to thank the staff of the ITP for making my stay there most pleasant. This work was supported in part by the National Science Foundation under grant PHY82-17853, supplemented by funds from the National Aeronautics and Space Administration, by the U.S.–Israel Binational Science Foundation (BSF), and Minerva.

REFERENCES

1. J. L. van Hemmen and I. Morgenstern, eds., *Heidelberg Colloquium on Glassy Dynamics* (Springer-Verlag, 1986); E. Bienenstock, F. Fogelman Soulie, and G. Weisbuch, eds., *Disordered Systems and Biological Organization* (NATO ASI Series, Springer-Verlag, 1986); J. W. Clark, J. Rafaelski, and J. V. Winston, *Phys. Rep.* **123**:215 (1985); D. Farmer, A. Lapedes, N. Packard, and B. Wendroff, eds., Evolution, Games and Learning, *Physica* **22D** (1986); T. Hogg and B. A. Huberman, Artificial intelligence and large scale computation: A physics perspective, *Phys. Rep.* (1987), to appear.
2. T. Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag, 1984); G. E. Hinton and J. A. Anderson, eds., *Parallel Models of Associative Memory* (Erlbaum, Hillsdale, New Jersey, 1981); D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (MIT Press, 1986); R. Lippmann, An introduction to computing with neural nets, *IEEE ASSP Mag.* (April 1987).
3. A. I. Selverston, ed., *Model Neural Networks and Behavior* (Plenum Press, New York, 1985).
4. J.-P. Changeux, *Neuronal Man* (Pantheon Books, New York, 1985); J. S. Albus, *Brains, Behavior, Robotics* (Byte Books, Peterborough, 1981).
5. E. R. Kandel and J. H. Schwartz, *Principles of Neuroscience* (Elsevier, New York, 1985).
6. J. Maddox, *Nature* **328**:571 (1987).
7. W. S. McCulloch and W. Pitts, *Bull. Math. Biophys.* **5**:115 (1943).
8. K. Binder and A. P. Young, *Rev. Mod. Phys.* **58**:801 (1986).
9. J. J. Hopfield, *Proc. Natl. Acad. Sci. USA* **79**:2554 (1982).
10. D. C. Hebb, *The Organization of Behavior: A Neurophysiological Theory* (Wiley, New York, 1957); L. N. Cooper, F. Liberman, and E. Oja, *Biol. Cybernet.* **33**:9 (1979).
11. W. Kinzel, *Z. Phys. B* **60**:205 (1985).
12. D. J. Amit, H. Gutfreund, and H. Sompolinsky, *Phys. Rev. Lett.* **55**:1530 (1985); *Phys. Rev. A* **32**:1007 (1985); *Phys. Rev. A* **35**:2293 (1987); *Ann. Phys.* **173**:30 (1987).
13. W. A. Little, *Math. Biosci.* **19**:101 (1975).
14. G. Toulouse, *Nature* **327**:662 (1987).
15. J. Hertz, G. Grinstein, and S. Solla, in L. Van hemmen and I. Morgenstern, eds., *Glassy Dynamics* (Springer-Verlag, Berlin, 1987), p. 538.

16. B. Derrida, E. Gardner, and A. Zippelius, *Europhys. Lett.* **4**:167–173 (1987).
17. E. Domany, R. Meir, and W. Kinzel, *Europhys. Lett.* **2**:175 (1986).
18. R. Meir and E. Domany, *Phys. Rev. Lett.* **59**:359 (1987); *Europhys. Lett.* 4:645 (1987); *Phys. Rev. A* **37**:608 (1988).
19. I. Kanter and H. Sompolinsky, *Phys. Rev. A* **35**:380 (1987).
20. E. Gardner, B. Derrida, and P. Mottishaw, *J. Phys. (Paris)* **48**:741 (1987).
21. J. L. van Hemmen, in J. L. van Hemmen and I. Morgenstern, eds., *Heidelberg Colloquium on Glassy Dynamics* (Springer-Verlag, 1986), p. 547.
22. F. Rosenblatt, *Principles of Neurodynamics* (Spartan, Washington, D.C., 1961).
23. A. K. Dewdney, *Sci. Am.* **1984** (September).
24. M. Minsky and S. Papert, *Perceptrons* (MIT Press, 1969).
25. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, in D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (MIT Press, 1986), Vol. 1, p. 318.
26. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Nature* **323**:533–536 (1986).
27. D. B. Parker, Learning Logic, Invention Report S81-64, Stanford University (1982).