Applications of neural networks in experimental physics

I. V. Kisel', V. N. Neskoromnyĭ, and G. A. Ososkov
Joint Institute for Nuclear Research, Dubna

Fiz. Elem. Chastits At. Yadra **24**, 1551–1595 (November–December 1993)

The review presents the theoretical foundations of various models of artificial neural networks and their application to topical problems of associative memory, optimization, and pattern recognition. The numerous applications of these networks in experimental physics, both as hardware realization of fast triggering systems for event selection and for the following software implementation of the trajectory data recognition, are also reviewed.

1. INTRODUCTION

The rapid development during the last 10–15 years of various theories of artificial neural networks (ANNs) was a reflection of and an attempt to overcome the gulf between the huge amount of factual material relating to the biological mechanisms of brain operation accumulated in neurophysiology at the end of the nineteenth and beginning of the twentieth century and the inadequate existing mathematical formalism and computational means of technical realization of the formalism. The principal advantages of the brain in fulfilling logical, recognition, and computational functions, using capabilities that are essentially parallel, nonlinear, and nonlocal, did not match the prevailing principle of sequential calculations with orientation of the mathematical formalism toward locality, linearity, and stationarity of the descriptions.

One could hardly claim that the gulf has been closed or is at least being successfully closed today. In this review we are not attempting a description of the theories of brain operation already developed by science. Nevertheless, some mathematical models of the functioning of neurons in their collective interaction that have been developed have proved to be extremely effective tools for solving very topical problems, if not in biology, at least in mathematics itself and in its numerous applications.

Included among these are problems whose solution is complicated precisely by nonlinearity, nonlocality, discreteness, and, often, nonstationarity of the situation. In particular, we have here problems of pattern recognition, the construction of associative memory (i.e., memory devices for rapid exchange based, not on address, but on the value of a fragment of a stored or extracted quantity) and optimization (i.e., search for the maximum of a functional in the presence of constraints on its parameters).

Thus, in this paper the word "neuron" should be not be understood literally, and, despite the biological origin, we deliberately emphasize in our terminology the artificial nature of the neural networks that we describe. Essentially, the theory of ANNs is a part of the general theory of dynamical systems in which particular attention is devoted to the investigation of the complicated collective behavior of a very large number of comparatively simple logical objects, which are called neurons.

It is important that although ordinary serial digital computers have up to now been used to solve this class of problems, the theory of neural networks has stimulated the development of not only entirely new algorithms for their solution but also appropriate specialized parallel computers designed for the best technical implementation of these algorithms, including fast analog neurocomputers based on the most recent achievements in optics.

The material of the paper is arranged as follows. In Sec. 2 we describe the biological principles that provide the basis of most models that use the algorithms and methods of neural networks. We construct a very simple neural network in the process of the symbolic "evolution" of which a certain functional, usually called the energy of the neural network, is minimized. The extrema of the functional correspond to a required configuration of the neural network, this being identical either to a recognized pattern or, depending on the problem to be solved, to the optimum fulfillment of certain *a priori* requirements. The analogy between binary neurons and the well-developed theoretical model of the Ising magnet leads to Glauber dynamics for the solution of the problem of finding the global minimum of the energy function and the avoidance of local minima of the function. For this, one uses mean-field theory with nonzero temperature, and this makes it possible to tunnel to the global minimum of the spin configuration.

In Sec. 3, we formulate and review solutions of some topical problems that use neural networks: the problems of associative memory and optimization and two problems of pattern recognition—with and without training. The last two problems are analyzed in more detail, since the result of their solution is particularly widely used in high-energy physics.

In Sec. 4 we describe different methods of realization of neural networks: virtual (in the form of an algorithm on a computer), electrical, and optical (in the form of a physical circuit). Depending on the class of problems to be solved, we also describe different architectural types of neural networks containing several coupled functional levels and manifesting certain rudiments of artificial intelligence.

In Sec. 5 we consider some possible applications of neural networks for the examples of problems that arise in the three main stages of analysis of experimental data in high-energy physics (on-line detection, off-line analysis to obtain the physical parameters, and subsequent interpretation with testing of hypotheses).

Particular attention is devoted to the formulation and

1063-7796/93/060657-20$10.00
© 1994 American Institute of Physics

methods of solution of problems involving the recognition of tracks and events and the testing of statistical hypotheses. Taking examples of studies carried on at the JINR, we analyze in detail the computational problems that arise both in the initial construction of the neural network and in the search for the global minimum of its energy functional. Various methods for accelerating the convergence of neural networks are described.

2. BIOLOGICAL FOUNDATIONS AND THE SIMPLEST MODELS

Neurobiological sources of neural networks

The first neuron cells of the cerebellum were discovered in 1836 by the neurophysiologist Purkinje. Seventy years later, the Nobel Prize for physiology was awarded to C. Golgi and S. Ramén y Cajal for investigations into the structure of neurons and the nervous system of man and vertebrates. The word "neuron" itself was introduced into scientific use by W. Waldeyer (1836–1921). The concept of "synapse" as converter of signals arriving at a neuron was introduced in 1897 by Sherrington.[1)]

There was thus formulated the *neural doctrine*, which encompassed as a single biological principle the nervous processes of all organisms from the very simplest to those possessing a central nervous system, including man. The doctrine regards the nervous system as a structure consisting of many elements, called neurons, that are coupled and work in parallel. An individual neuron is a processing unit, a cell, that consists of a "soma," the body of the neuron, to which there are connected, through the branchlike "dendrites," the numerous transmission lines from other neurons: "axons." The junction parts of a neuron, the "synapses," convert the input information into signals that are received by the neuron and put it into one of two stable states—excited or inhibited.

Theories of the biochemical molecular processes that occur in the nervous system, in the biological membranes, and implement the transmission of signals were developed much later.

The main fundamental proposition of the neural doctrine is that the neurons are regarded as inseparable elements of a highly parallel nervous system in which the processes of information analysis cannot be reduced to phenomena that occur in individual neurons.

This, in particular, is why the need arose for a multidisciplinary approach to the study and description of such processes. Theories of brain models could be developed only at the meeting point of many sciences: biology, physics, chemistry, mathematics, and microelectronics. Our further exposition in the framework of our chosen direction can also be regarded as an illustration of this fact. From the neurophysiological principles and descriptions we go over to simple models that are the analog of disordered magnetic systems, so-called spin glasses. Their theory is a part of statistical physics, which also suggested other approaches to the solution of the most important problems of ANNs, such as the use of mean-field theory and simulated annealing. Cellular automata, which are related to ANNs, also proved to be effective tools for solving the problem of the choice of the initial states of ANNs. Even greater interdisciplinary disciplines arise in the study of the problems of developing neurocomputers.

Completing our review of the biological foundations of ANNs, we mention also two important investigations that to a large degree determined the development of the theory of neural networks. The first was the work of McCulloch and Pitts,[2] who modeled a system of biological neurons by a set of binary, i.e., possessing only two states, objects coupled to each other and having a certain threshold level of excitation, on the attainment of which the neuron state changes. Thus, each *i*th neuron has two states, characterized by the fact that the neuron output V_i takes one of two values, V_i^0 or V_i^1, which are often simply equal to 0 or 1. In practice, the output of a real neuron is a frequency-modulated signal, so that different signal frequencies correspond to the different states. In what follows, we shall assume that at the neuron output a certain potential V_i is established, since in any case the real signal is subsequently subjected to frequency demodulation. The output signal of a neuron passes through its axon to a synapse and is transmitted through the synaptic junction (which is different for different pairs of neurons) and through the input branchlike dendrites to the input of neuron *j*. A global feedback loop is established, as a result of which the system of neurons possesses a nontrivial nonlinear behavior.

The input signal of each neuron consists of two components: a signal I_i that comes from outside the neural network and a signal that comes from the remaining neurons and is weighted with the synaptic weights T_{ij}. Thus, the total input signal has the form

$$H_i = \sum_{j \neq i}^{N} T_{ij} V_j + I_i. \tag{1}$$

The neural network evolves in accordance with the following rules. From time to time, each neuron estimates its input signal relative to the threshold value U_i, after which it either leaves its output signal unchanged or changes it in accordance with the stochastic algorithm

$$V_i \rightarrow \begin{cases} V_i^0, & \text{if } \sum\limits_{j \neq i} V_{ij} V_j + I_i < U_i \\ V_i^1, & \text{if } \sum\limits_{J \neq i} T_{ij} V_j + I_i > U_i. \end{cases} \tag{2}$$

Thus, the algorithm in the McCulloch–Pitts model is asynchronous. There are at least two important differences between a real neural network and the model. First, a real neuron must have a continuously varying output signal, which depends on the input signal. Second, the propagation of any signal through the network involves a time lag because of capacities, so that signal propagation must be described, not by a discrete algorithm, but by a differential equation with a noise source because of detection. Therefore, a more widely adopted form of description of the network evolution is a nonlinear function that has not a step but a "sigmoid" type:

$$V_i = g(H_i), \tag{3}$$

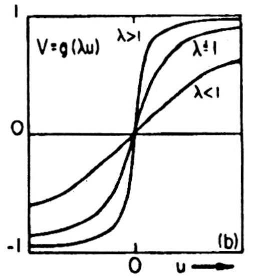

FIG. 1. General form of sigmoid function in its dependence on the parameter λ.

where $g(t)$ is shown in Fig. 1. a schematic representation of such a neuron is shown in Fig. 2.

In the second work that we wish to mention, the neuropsychologist Hebb[3] laid the foundations of the synaptic theory of memory and learning, i.e., the understanding of the properties of neural networks that create their diverse possibilities, including the occurrence of intelligence. Hebb formulated his learning rule as follows: "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." Hebb's rule of the change of the synaptic junctions makes possible the training of a network for operation as an associator (classifier) of images. Repeated appearance at the input of a stimulating image must prompt the network to generate another image (sign) associated with the first. Conversely, presentation of an image fragment to the network gives rise in accordance with Hebb's rule to generation of the complete image. Essentially, this is the basic idea of addressing by content and associative memory.[4]

Mathematical models of ANNs

Since Hebb, being a neuropsychologist, did not use mathematics and his book does not contain a single equation, the mathematical expression of his rule was proposed by Sutton:[5]

$$T_{ij}^{(m+1)}=T_{ij}^{(m)}+\eta V_j^{(m)}H_i^{(m)}, \tag{4}$$

where m is the discrete learning time, i.e., the number of the training iteration, η is a positive constant, which determines the rate of learning, and V_j is one of the input signals that gives H_i at the output.

In the book of Ref. 6, Hebb's rule (4) is regarded as a special case of a more general learning rule, in which the synaptic weight changes in proportion to an amplifying signal $r_{ij}^{(m)}$:

$$T_{ij}^{(m+1)}=T_{ij}^{(m)}+\eta r_{ij}^{(m)}. \tag{5}$$

For Hebb's rule, the amplifying signal has the form $r_{ij}^{(m)}\equiv V_j^{(m)}H_i^{(m)}$. A different form of amplifying signal was proposed by Widrow and Hoff[7] for so-called learning with a teacher (supervisor):

$$r_{ij}^{(m)}=[z_i^{(m)}-H_i^{(m)}]V_j^{(m)}, \tag{6}$$

where $z_i^{(m)}$ is a special teaching reference signal, $H_i^{(m)}=\Sigma_{j=1}^{n}T_{ij}^{(m)}V_j^{(m)}$. Usually, one writes

$$\Delta T_{ij}^{(m)}=T_{ij}^{(m+1)}-T_{ij}^{(m)}=\eta r_{ij}^{(m)}, \tag{7}$$

and the learning rule is called the delta rule. It ensures convergence of the weights during learning.

The concept of taught neural networks, called perceptrons, was introduced by Rosenblatt.[8,9] As is shown in Fig. 3, a perceptron is a uniflow ANN consisting of one or several layers of neurons possessing connections between the layers and also with inputs and outputs. The layers between the input and output layers are called hidden layers. Signals that reach the input of each of the neurons are transformed nonlinearly in accordance with (1), and also (2) or (3). In accordance with (1) and (3), a perceptron separates the complete N-dimensional space of input variables $\{x_i\}$ into K classes by means of hyperplanes defined by the equations

$$\sum_{j\neq i} T_{ij}x_j-I_i=0, \quad i=1,\ldots,K.$$

Not more than 2^K such classes can exist. The weights of the junctions and the thresholds change during learning and are then fixed for use of the perceptron as a classifier.

In the case of learning in accordance with the Δ rule it is usual to employ, in a sample of M training cycles, an algorithm of backpropagation of errors, which is a generalization of the least-squares method to multilayer perceptrons, since it minimizes with respect to all values of the

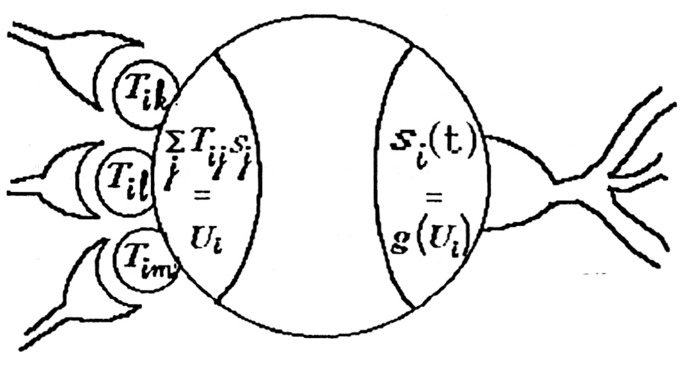

FIG. 2. Schematic representation of a neuron.

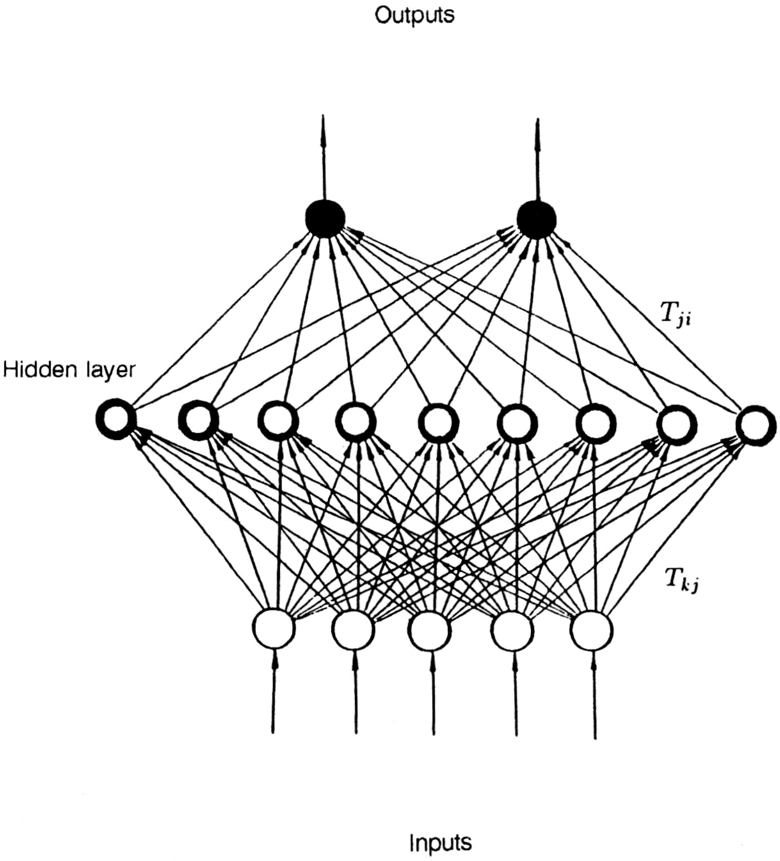

FIG. 3. Schematic arrangement of a three-layer perceptron.

weights a functional consisting of the sum of squares of the differences between the output signal $\{y_i^{(m)}\}$ of the perceptron in each of the cycles and a reference image $\{z_i^{(m)}\}$ presented in this cycle:

$$E=\sum_{m=1}^{M}\sum_{i}(y_i^{(m)}-z_i^{(m)})^2\rightarrow\min. \tag{8}$$

For example, for a three-layer perceptron there are N input signals $\mathbf{X}_N=(x_1,...,x_N)$, a hidden layer of n neurons $\mathbf{H}_n=(h_1,...,h_n)$, and l output signals. Denoting by T_{ij} the weights that connect the input signals to the hidden neurons, and by T_{jk} the connections of these to the output signals of the perceptron, we have, in accordance with (1) and (3),

$$\begin{aligned}h_j&=g(H_j), \quad H_j=\sum_k T_{jk}x_k;\\ y_i&=g(Y_i), \quad Y_i=\sum_j T_{ij}h_j.\end{aligned} \tag{9}$$

Differentiating (8) with respect to T_{ij} and T_{jk}, and equating to zero the derivatives, we obtain a system of linear equations, from which we find

$$\Delta T_{ij}^{(m+1)}=-\eta(y_i^{(m)}-z_i^{(m)})g'(y_i)h_j, \tag{10}$$

$$\Delta T_{ik}^{(m+1)}=-\eta\sum_i T_{ij}^{(m)}g'(y_i^{(m)})g'(h_j^{(m)})x_k, \tag{11}$$

and this actually corresponds to the Δ rule.

For the simple case $M=1$, when the inputs are classified into only two classes, Rosenblatt showed[9] that if the input vectors, which belong to different classes in accordance with their proximity to standards, are separated in the space of inputs by some hyperplane (compactness hypothesis), then the learning algorithm converges. Unfortunately, these simple three-layer perceptrons do not make it possible to construct more complicated separating surfaces in the space of inputs. The introduction of additional hidden layers makes it possible to form more complicated convex separating surfaces, but at the same time the learning is greatly complicated.

Hopfield's model

The publications on neural networks became a veritable explosion after the papers of Hopfield,[10-12] who solved these problems by constructing an "energy" functional for a neural network that for the dynamics of the original neural network is none other than a Lyapunov function.[13-16] The existence of a stationary stable point for the algorithm (2) was proved for an arbitrary symmetric matrix of synaptic weights T_{ij}. Indeed, it was found to be sufficient to construct the function

$$E=-\frac{1}{2}\sum_{i\neq j}^{N}\sum_{j=1}^{N}T_{ij}V_iV_j-\sum_{i=1}^{N}I_iV_i+\sum_{i=1}^{N}U_iV_i. \tag{12}$$

The change in E when the state of neuron i changes by ΔV_i is obviously

$$\Delta E=-\left(\sum_{j\neq i} T_{ij}V_j+I_i-U_i\right)\Delta V_i.$$

A positive ΔV_i corresponds to a positive bracket, and conversely, so that $\Delta E\leqslant 0$ during the process of evolution, and, since E is bounded, there exists a limiting stationary value.

In a real neural network, there is no reason at all why the matrix T_{ij} must be symmetric because of the topology of the connections between the neurons. Indeed, neuron i receives a signal from neuron j through one synapse, while neuron j receives a signal from neuron i through another. The dynamics of a neural network with a nonsymmetric matrix of synaptic weights was investigated in Refs. 17 and 18. Moreover, in the process of evolution the T_{ij} themselves could change. As a result, it was found on the basis of Ref. 19 that the condition of symmetry is in many cases unimportant, although when it does not hold it is difficult to interpret E as the energy of the neural network. In addition, for nonsymmetric weights limit cycles and oscillations can occur in the system, like the ones investigated in Refs. 20–22.

An interesting question is that of a limit point of a neural network consisting of neurons with the continuous response function (sigmoid function) shown in Fig. 1 as a function of the parameter that controls the degree of difference of this continuous function from a step function.[11] For such a neural network, one can also construct a function E that decreases monotonically on the solutions of the differential equation that determines the continuous dynamics of the neurons. It was shown that as the sigmoid function tends to the step function the stationary point of the system of continuous neurons with given matrix T_{ij} tends to the stationary point of the discrete neurons with the same matrix of connections. The reason for this is that for continuous V_i the first term in the expression (12) is decisive for the minimum (in the absence of I_i), and it is linear for every V_i, this giving a position of the minimum at the corners of the hypercube $-1\leqslant V_i\leqslant 1$. For an almost flat sigmoid function, in contrast, the only stable point is the trivial solution $V_i=0$.

We now consider how a definite image is "remembered" in Hopfield's model. In other words, suppose that we are given some fixed configuration of the neural network. It is then required to choose the synaptic weights T_{ij} in such a way that for any initial configuration situated sufficiently close to the required image the required configuration of neurons is obtained in the limit as a result of the dynamics (2). We note here that the dynamics (2) of the model with the function (12) is none other than the Glauber dynamics of the Ising magnet with long-range interaction at zero temperature. For this, it is necessary to replace the neurons by classical spins $S_i=\pm 1$; then the energy will have the form (here and in what follows, the excitation threshold is taken equal to zero)

$$E=-\frac{1}{2}\sum_{i,j=1}^{N} T_{ij}S_iS_j. \qquad (13)$$

Here, the summation is over all spins, and it is assumed that $T_{ii}=0$.

The problem of remembering a definite image was solved in Ref. 23. We consider it for the example of one image. Suppose that the required image is a sequence $\{\xi_i\}$ $(\xi_i=\pm 1)$. It is obvious that the quadratic form

$$E=-\frac{1}{N}\sum_{i,j=1}^{N}(\xi_iS_i)(\xi_jS_j)$$

will have a minimum precisely when $S_i=\xi_i$ (or when $S_i=-\xi_i$, which also solves the problem). Then for the matrix of synaptic weights we obtain the expression

$$T_{ij}=\frac{1}{N}\xi_i\xi_j,$$

and this solves the problem of remembering one image. The factor $1/N$ is introduced to ensure that for large N the energy is proportional to the number of spins, as is necessary for the thermodynamics. For several images $\{\xi_i^p\}$ $(i=1,...,N,\ p=1,...,M)$ at not too large M the solution will be the following Mattis magnet:

$$T_{ij}=\frac{1}{N}\sum_{p=1}^{M}\xi_i^p\xi_j^p. \qquad (14)$$

It is interesting that despite being constructed purely formally, solely on the basis of mathematical considerations, the expression (14) also has a deep biological basis. If during the storing of the image neuron i sends a pulse to neuron j, then the synaptic junction corresponding to this pair of neurons is strengthened, and signals pass between these neurons with greater weight, this leading to Hebb's rule[3] that we mentioned above.

It is well known that in the thermodynamic limit $N\to\infty$ the energy function of a magnet with the interaction (14) will have precisely $2M$ local minima, provided the images are not correlated, i.e., when

$$\frac{1}{N}\sum_{i=1}^{N}\xi_i^p\xi_i^q=O(N^{-1/2}) \quad \text{for } p\neq q.$$

It turns out that in the thermodynamic limit it is also possible to increase the number of images stored in the neural network, so that the number of images will be proportional to N:

$$M=\alpha N. \qquad (15)$$

Hopfield[10] investigated the largest possible value of the coefficient α for which most of the images can be recovered. It was found that for $0.14<\alpha<0.16$ the neural network rapidly saturates, spurious minima arise from the overlapping of different images, and some of the minima disappear. A numerical experiment was made with a neural network containing $N=100$ neurons. If $M=0.15N$ images were stored in the neural network, recovery of a required image occurred with probability 0.85. In 10% of the cases, the system arrived at a minimum that did not correspond to any image. With further increase in the number of images to $\alpha=1$, the model of the neural network becomes identical to the model of a spin glass.[24] Figure 4 gives an example of the recovery of an image representing the letter A stored in a neural network of $N=20\times 20$ spins together

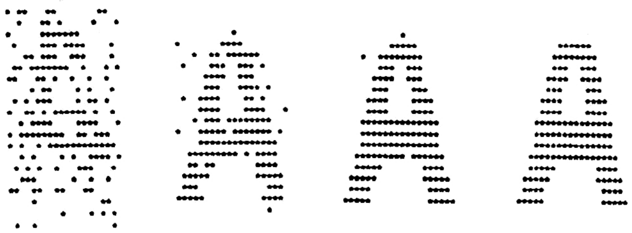

FIG. 4. Reconstruction of an image in the form of the letter A stored in a neural network of $N=20\times 20$ spins together with $M=30$ other random images.[24]

with $M=30$ other random images. The initial state of the neural network was chosen in the form of the same letter A, in which the spins were reversed with probability $p=0.3$. For the neural network four iterations were sufficient to "remember" the original image with practically no distortions.

This property of a neural network of reconstructing an image presented with a slight change (either with superimposed noise or containing only an important part of the original image) is called *associative memory*. By virtue of this property, one can create on the basis of a neural network a data base in which a search is made, not by means of a catalog, but in a process of calculations based on a presented fragment of the required image. All that we have said above was valid for uncorrelated images. However, if the images have common characteristic features (for example, if several letters of the alphabet are stored in the neural network), then the minima on the energy surface corresponding to similar letters merge, and the neural network gives a false result. Figure 5 shows how five letters were stored in a neural network (left-hand column) and what appeared after operation of the network (right-hand column). It can be seen that the neural network recovered only coincident fragments of the images.

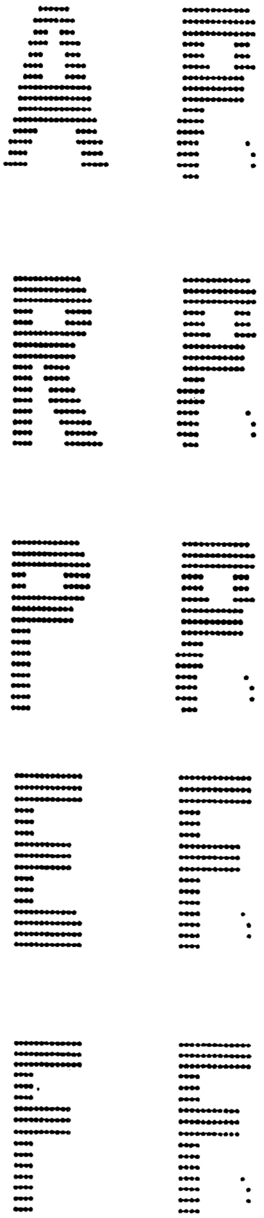

FIG. 5. Recognition of correlated images.[24]

The resolution of the problem was rather unexpected. For this, we recall that the information in a neural network is stored, not in its elements (neurons), but in the connections between them, i.e., in the matrix of synaptic weights. The total number of different elements in a symmetric matrix is $N(N-1)/2$. In other words, even if $M=0.15N$, a neural network contains much excess information on the images, and this increases the reliability of the neural network even when a large number of connections between the neurons is eliminated. In the case of correlated images, it was found to be sufficient to eliminate "spoiled" connections. These are defined as follows. In accordance with (14), all the synaptic weights T_{ij} are calculated. If for at least one image the sign of the corresponding term in the sum is not identical to the sign of the complete sum, then this connection between the neurons is eliminated: $T_{ij}=0$. Figure 6 demonstrates the result of operation of such a modernized neural network containing the images of five letters of the Latin alphabet. Whereas before the elimination of the undesirable elements from the matrix of synaptic weights the neural network refused to recognize even the original "pure" images of the letters, after the elimination procedure, when the number of remaining connections was only 16% of the original number, the neural network reconstructed reasonably well (right-hand column of Fig. 6) the images of letters to which a noise level of even 20% had been added (left-hand column).

The problem of increasing the capacity of a neural network is very important. Naturally, Hopfield's result is valid only for the simplest single-layer neural network. The significant increase in the capacity of a neural network by the introduction of a hidden layer was demonstrated in Ref. 25. In Ref. 26, as in the case with the remembering of letters, the characteristics of the neural network was also improved by simplifying it. The capacity of the neural network was increased by restricting the range of possible values of T_{ij}. Whereas for Hopfield $T_{ij}\in[-M,M]$, in this case the weights took only the three possible values -1, 0, and $+1$. This also gives additional advantages, in economy

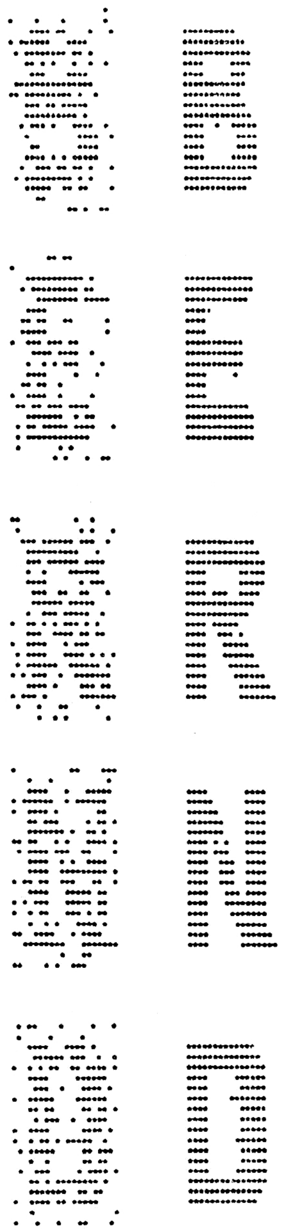

FIG. 6. Recognition of correlated images with allowance for elimination of spoiled connections.[24]

of space, when the neural network is implemented in the form of an electronic chip.

Even in a standard neural network the number of stored images can be increased. Indeed, suppose that we need to store, for example, 1 million uncorrelated images. Then the simplest neural network must contain $N \approx 6.7$ million neurons. In such a case, each image may contain more than 6 Mbits of information. However, there is often much less information in an image, i.e., the image contains only X bits of information, most of which, moreover, consists of zeros, there being only $K \sim \sqrt{X}$ ones in the image. Then to store M such images one requires a neural network of only $N=2(MX)^{1/2}$ elements. Compared with the organization of an ordinary catalog, there is a significant gain in time in a content-address search in a neural network—one requires only $KN \approx 2^X \sqrt{M}$ operations instead of $\sim M$ for ordinary image selection.[27]

Monotonic decrease of the energy function of a neural network was originally obtained for serial dynamics of the neural network, i.e., when at each time only one neuron (in turn or randomly selected) evaluated its state and then changed it. A very important result was obtained in Ref. 28, where it was asserted that the differences between serial dynamics and parallel dynamics (when all neurons evolve simultaneously) are unimportant from the point of view of obtaining a stationary state. This opened the way to wide use of parallel computer systems for modeling neural networks.

In this section, the main attention was devoted to the problem of associative memory and the problem of pattern recognition. As we showed, in such problems there is first "training" of the junctions, i.e., teaching of the ANN, during which the matrix of synaptic weights is modified. This operation requires fairly considerable computer resources, and therefore the finished matrix of synaptic weights may become a subject of author's rights together with standard data bases. After this, the algorithms of the neural networks make it possible to find quite economically the local minimum of the energy function closest to the initial configuration.

However, there exists an extensive class of optimization problems in which it is necessary to find, not a local, but a global minimum for a specially chosen functional that has the form of the energy function (12). In these problems, it is first necessary to choose the form of the matrix T_{ij} in accordance with required conditions.

3. OPTIMIZATION PROBLEMS

Mean-field theory in Hopfield's model

Thus, let us consider the problem of finding the global minimum of a system of classical spins whose energy is determined by the expression

$$E=-\frac{1}{2}\sum_{i,j=1} T_{ij}S_iS_j .$$

To find the global minimum by the methods of neural networks, the system must be able to pass from one minimum to another, lower minimum. This can be done if one introduces a "temperature" into the model. After this, there are two possibilities. First, as in the Monte Carlo method, one can allow in the system the presence of energy fluctuations to permit the system to escape from a local minimum. Second, one can allow the system to have "below-barrier" configurations to enable the system to "tunnel" to a neighboring, lower minimum. The choice between these alternatives—above- and below-barrier transitions—depends on the ratio of the height of the barrier and its width. For a neural network with strongly irregular constants T_{ij}, the energy surface is similar to the energy surface of the spin-glass model, and for this it is known that in the model there exist very high but rather narrow potential barriers between the local minima.[29]

Thus, we conclude that it is better to tunnel through the barriers. Mathematically, this can be realized by using mean-field theory. The essence of it is the following calculation:

$$0\approx-\frac{1}{2}\sum_{i,j=1}^{N}T_{ij}(S_i-\langle S_i\rangle_T)(S_j-\langle S_j\rangle_T)$$

$$=-\frac{1}{2}\sum_{i,j=1}^{N}T_{ij}S_iS_j+\sum_{i,j=1}^{N}T_{ij}S_i\langle S_j\rangle_T$$

$$-\frac{1}{2}\sum_{i,j=1}^{N}T_{ij}\langle S_i\rangle_T\langle S_j\rangle_T,$$

i.e., the energy can be expressed approximately as a linear expression plus a constant:

$$E=-\frac{1}{2}\sum_{i,j=1}^{N}T_{ij}S_iS_j\approx-\sum_{i,j=1}^{N}T_{ij}S_i\langle S_j\rangle_T+\text{const},\tag{16}$$

where $\langle S_i\rangle_T$ denote the thermal mean values of the spins S_i.

There is an extensive literature on the mathematically rigorous justification of this procedure. We shall not dwell on it in detail, especially in application to Hopfield's model; we merely mention the classical study of Ref. 30. As a rule, mean-field theory becomes exact in the thermodynamic limit $N\to\infty$ for models with vanishingly weak interaction of infinitely long range. Therefore, to apply mean-field theory correctly in a specific case, it is necessary to ensure that the number of interacting spins be very large and the intensity of the interaction of each pair of spins become small. We note that the presence of the factor $1/N$ in the matrix T_{ij} [see the expression (14)] and the double summation over all the spins makes it possible to use the results of mean-field theory with considerable confidence for large but finite values of N.

We introduce notation for the mean field produced by the remaining spins at site i:

$$U_i=\langle H_i\rangle_T=\sum_{j=1}^{N}T_{ij}\langle S_j\rangle_T.\tag{17}$$

Then in accordance with the simplest problem of the mean value of a spin in an external field, we obtain an expression of the form

$$V_i=\langle S_i\rangle_T=\tanh\left(\sum_{j=1}^{N}T_{ij}V_j/T\right).\tag{18}$$

In mean-field theory, Eq. (18) is called the self-consistency equation. In it, the unknown values of V_i are expressed in terms of all the remaining V_j. For the spatially homogeneous case, when V does not depend on the site number j, we have one equation; for the general case of a neural network, we have a system of equations. It is practically impossible to solve this system exactly, but the neural network can solve the problem by itself (!) by successive iterations.

As in the preceding class of associative-memory problems, the neural network in each step of the evolution calculates the local field (17) for each neuron [i.e., it calculates the argument of the hyperbolic tangent in the expression (18)], after which it determines new mean values of the spin V_i. The difference between the mean-field dynamics with temperature and the dynamics (2) is that the response function is a continuous sigmoid function, and not a step function. As a result of this, the ±1 spin is replaced by its mean value V_i, which takes any value in the interval $(-1,1)$. For this dynamics, there exists a critical "temperature" T_c, above which only the trivial final state $V_i=0$ is possible when the time increases without limit.[11] Thus, to find the global minimum of the functional (16), it is necessary to take the initial temperature T sufficiently high, but below the critical value, and after the neighborhood of the global minimum has been found, the temperature should be decreased to zero in order to improve the accuracy of the solution of the problem.

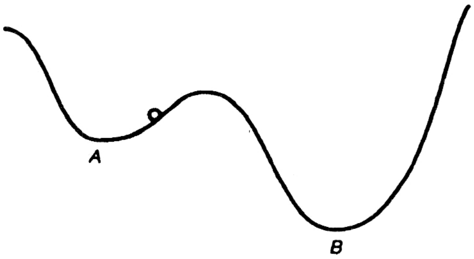

FIG. 7. The mechanism of simulated annealing.

Below, in Sec. 5, Hopfield's model is used as the main tool in problems involving the recognition of the tracks of charged particles on the basis of the fact that *a priori* information on the magnetic field and detector geometry makes it possible to determine the matrix of synaptic weights in advance.

The procedure of "simulated annealing"

In the minimization procedure for the ANN energy function, there is a real danger of reaching one of the local minima instead of the global minimum. A possible way of avoiding this is best explained by the mechanical analogy with a heavy ball rolling on the surface of the energy function (see Fig. 7). If there is moderate shaking of the system, the ball will jump with high probability to a deeper level and remain there. However, if the shaking is strong, then with equal probability it may reach any of the levels. Therefore, the best strategy will be to begin by shaking strongly, but then shake more and more weakly. One can see here a similarity with the procedure of annealing metals, in which a low energy of the bonds in the metal is achieved by an initial heating to a very high temperature, which gives rise to a phase transition through the point of breaking of the crystal bonds, with a subsequent slow cooling. Therefore, the corresponding procedure for finding the global minimum of the ANN energy surface was called simulated annealing.

We now demonstrate the algorithm of simulated annealing for the Hopfield network.[31] We denote the temper-

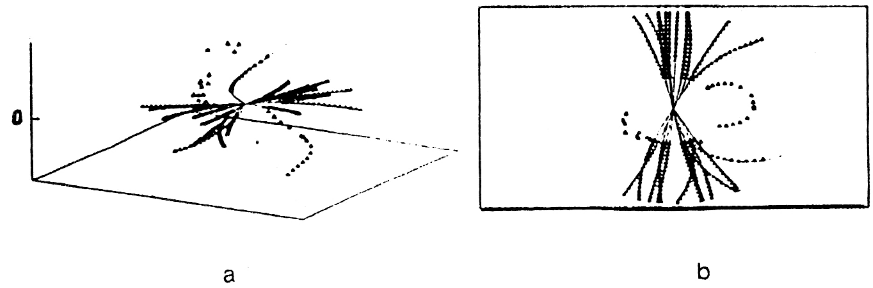

FIG. 8. Result of operation of the method of deformable templates: a) three-dimensional image, b) projection onto the xy plane.

ature by T, the time of operation of the network at each value of the temperature by t, and the number of input sites by N.

1. Initial state $T_0=1.5$, $t_0=N$.

2. Slow heating: $T \Leftarrow T/0.8$, $T=N$ until the specific energy fluctuation $(\langle E^2\rangle - \langle E\rangle^2)/T$ is less than 0.05.

3. Gradual cooling: $T \Leftarrow T\cdot 0.95$, $t=N$, until the number of neurons that have changed their state exceeds 50%.

4. Slow cooling: $T \Leftarrow T\cdot 0.95$, $t=16N$, until a stable state of the network is established.

Method of deformable templates

In the method of deformable templates in the theory of ANNs the neural network must, for a given set of standard images defined by a series of parameters, calculate the optimal values of these parameters. We demonstrate the operation of this model by one of the typical problems of determining the parameters of a prescribed number of tracks having the shape of a helix.[32]

We assume that we are dealing with a set of M deformable spiral templates specified by the angle of emission θ of the track in the XY plane, the curvature κ, and the parameter γ, which determines the inclination of the track to the Z axis (along the magnetic field): $(\theta_a,\kappa_a,\gamma_a)$, $a=1,...,M$. The problem is to approximate by them experimental points (x_i,y_i,z_i). The measure of the quality of the approximation is

$$E[S_{ia};\theta_a,\kappa_a,\gamma_a] = \sum_{i,a} S_{ia}M_{ia} + \lambda \sum_i \left\{ \sum_a S_{ia} - 1 \right\}^2, \quad (19)$$

where the double neuron S_{ia} determines whether the ith experimental point belongs to the ath track: $S_{ia}=1$ if it does, and $S_{ia}=0$ otherwise. We wish to minimize $E[S_{ia};\theta_a,\kappa_a,\gamma_a]$ with respect to S_{ia}, θ_a, κ_a, and γ_a under the condition that each point can either lie only on one track or not on any track (be noise).

The first term of Eq. (19) describes an interaction of the point with the track, and M_{ia} characterizes the strength of this interaction. Usually, M_{ia} is taken equal to the square of the distance from the point to the track, and in this case the equation is a generalization of the least-squares method in terms of the neural network.

The second term in Eq. (19) determines the critical distance up to which a point acts on a track: If $M_{ia} > \lambda$, then it is energetically advantageous to take $S_{ia}=0$. If M_{ia} is determined as the square of the distance from the track to the neuron, the critical distance is $\sqrt{\lambda}$. Such an approach corresponds to robust reconstruction of track parameters.[33]

The problem of finding the global minimum of the energy function (19) is solved as with a standard neural network. For a sequence of decreasing temperatures, the method of gradient descent gives

$$\Delta\alpha_a = -\eta \sum_i \hat{S}_{ia} \frac{\partial M_{ia}}{\partial \alpha_a} \quad (20)$$

with $\alpha_a = (\theta_a,\kappa_a,\gamma_a)$. In this equation, we have separated the factor

$$\hat{S}_{ia} = \frac{e^{-M_{ia}/T}}{e^{-\lambda/T} + \Sigma_b e^{-M_{ib}/T}}. \quad (21)$$

The method of deformable templates was tested on simulated data for the CERN DELPHI TPC detector. The result is shown in Fig. 8. The algorithm worked fairly well even in the case of intersecting or closely spaced tracks. This reliability of the method is largely determined by the factor (21).

Essentially, the method of deformable templates is an interpretation of the so-called flexible-arm method or elastic-network method,[34,35] which are very promising for track recognition under conditions of high multiplicity and noise (a comparison with ordinary methods is made in Ref. 35).

Self-teaching ANNs

Among the serious shortcomings of ANNs of the type of a classifying perceptron with teaching by a sample are above all the need to know in advance the number of classified signs and also the ability to choose the most representative and least correlated from them. Therefore, besides ANNs taught by a sample, self-teaching ANNs are of considerable interest.[36] They do not require prior knowledge of the required signs but develop them themselves in accordance with the structure of the analyzed data, using a clustering algorithm.

For the neurons h_j of a hidden layer, one chooses a "victor" h_m, for example, taking the maximum of the mean field,

$$h_m = \max_j (h_j), \quad (22)$$

which in the simplest self-teaching ANN with "winner-takes-all" organization becomes more "sensitive" to the presented data x_k through recalculation of the weight coefficients T_{jk}. In a self-teaching ANN with "competitive" learning, the mean field H_j of the hidden neuron h_j is calculated not as in (17) but as the scalar product of the weight vector $\mathbf{T}_j=(T_{j1},T_{j2},...,T_{jn})$ and the input vector $\mathbf{X}_N=(x_1,x_2,...,x_N)$:

$$H_j=|\mathbf{T}_j|\,|\mathbf{X}_N|\cos\theta_j. \qquad (23)$$

Thus, the determination of the victor h_m in (22) leads to the determination of the $\mathbf{T}_j$ that is closest to the input vector $\mathbf{X}_N$.

For this, as in (8), one minimizes the mean-square error of the difference

$$E=\frac{1}{2}\sum_{\mathbf{X}\in M}(\mathbf{X}-\mathbf{T}_j)^2\to\min_k,$$

where M is the subset of input sites (cluster) that give h_m as the maximum. The subset M can change during the process of self-teaching.

The dynamics of the renewal of the weights is obtained by using gradient descent:

$$\Delta T_{jk}=-\eta\frac{\partial E}{\partial T_{jk}}=\eta\sum_{\mathbf{X}\in M}(x_k-T_{jk}),$$

where, as before, η is the parameter of the learning rate. In practice, however, the weight vector $\mathbf{T}_j$ is renewed in each comparison with the image $\mathbf{X}\in M$ in accordance with the Δ rule

$$\Delta T_{jk}=\eta(x_k-T_{jk})_{\mathbf{X}\in M}$$

with normalization

$$T_{jk}\to\frac{T_{jk}}{\sqrt{\Sigma_{k'}T^2_{jk'}}}.$$

The determination of the subset M can be based on proximity to the victor, so that the Δ rule is changed as follows:

$$\Delta\mathbf{T}_j=\eta\Lambda(j,m)(\mathbf{X}-\mathbf{T}_j),$$

where $\Lambda(j,m)$ is a proximity function introduced to suppress (eliminate) lateral connections. Usually, $\Lambda(j,m)$ has the form of a "Mexican hat" or, as in the widely used packet JETNET 2.0,[36] is the characteristic function of the set $\|\mathbf{h}_j-\mathbf{h}_m\|\leqslant\lambda$ of sites $\mathbf{h}_j$ separated from the "victor" $\mathbf{h}_m$ by not more than λ:

$$\Lambda(j,m)=\begin{cases}1, & \text{if } \|\mathbf{h}_j-\mathbf{h}_m\|\leqslant\lambda;\\ 0, & \text{for the remaining } j.\end{cases}$$

The theoretical foundations of self-teaching ANNs are expounded in Ref. 15. The results of the use of such ANNs for recognition of string clusters in hadronic experiments, in which it was necessary to separate data with different quark species, are given in Ref. 36.

4. REALIZATION OF NEURAL NETWORKS

We have, in fact, already become acquainted with the simplest method of realizing a neural network. It is a virtual neural network in the form of a program for a computer. Admittedly, this should preferably be a fairly high-power parallel computer. We mention here characteristic features of a neural network that permit other possibilities of its realization:

- A neuron is a very simple logical device.
- The system consists of a very large number of identical neurons, and the result of operation of the neural network is weakly sensitive to the characteristics of a particular neuron.
- Each neuron is connected to a very large number of other neurons.
- The weights of the connections are different and, depending on the problem to be solved, can be changed.
- The analysis of information in the system is essentially parallel analysis.[37]

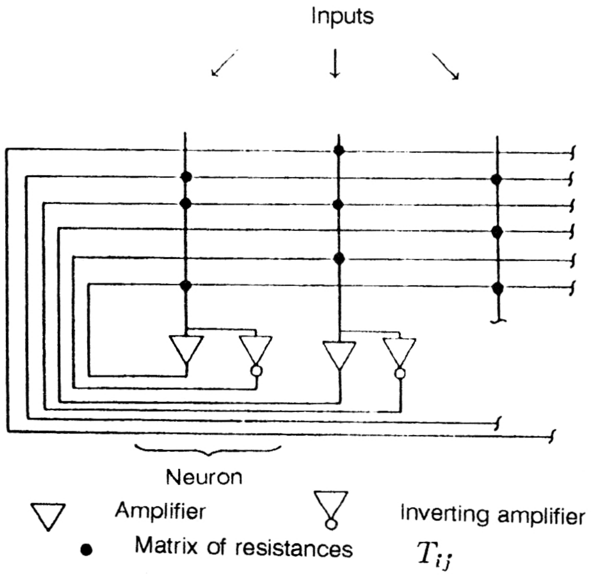

FIG. 9. Realization of an ANN in the form of an electrical circuit.[11]

There exist numerous firms specialized in this direction that have produced packages of programs that can be successfully operated on IBM personal computers to analyze the results of experiments. One of these programs, JETNET 2.0,[36] is especially designed to analyze data in high-energy physics and, in particular, to look for jets in multiparticle production. The original text of the program, written in FORTRAN, is distributed through the program library of the journal Computer Physics Communications.

The advances in technology have made it possible to create a specialized analog computer in which neuron functions are executed by a very simple processor, which either has a discrete output or is a small amplifier with sigmoid output characteristic. A matrix of ohmic resistances situated at the intersections of the outputs and inputs of these amplifiers implements the functions of the matrix of synaptic weights, and it is possible to realize a planar circuit in which all elements are connected to each other (Fig. 9). The differential equations for the voltage are given in Ref. 11, which analyzes the influence of the continuous output characteristic of the neurons on the limit state that the

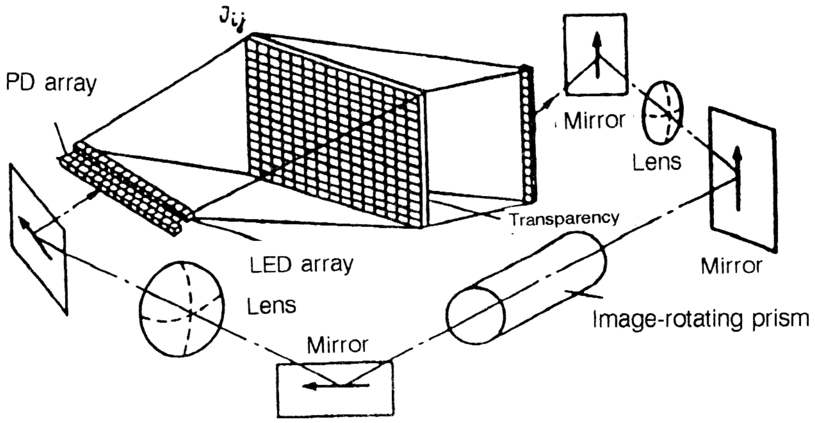

FIG. 10. Optical realization of a neural network.

neural network reaches. A similar working realization of a neural network is described in Ref. 38.

In his interview in the newspaper Poisk (No. 13, 1993), A. Galushkin, the director of the Scientific Center of Neurocomputers of the Russian Academy of Sciences, defines a neurocomputer as a computational system in which the algorithms for problem solving are represented in the logical basis of an ANN. Besides the realization of a neurocomputer in an ordinary research computer, there already exist various forms of parallel neurocomputers, including even a powerful supercomputer in which many neurons are realized in a single crystal. The aim of the creation of these computers is to perform calculations of optimization problems in a time shorter by 2–3 orders of magnitude than in existing supercomputers at the same cost and energy requirement. There is already talk of reaching teraflops, i.e., 10^{12} floating-point operations per second.

It is interesting that work began in Russia on the construction of the first neurocomputers at the same time as in the United States—at the end of the sixties and beginning of the seventies. This work led to the development of a three-layer perceptron with weights calculated in advance through teaching on ordinary computers. They were used for problems that do not require high rates of variation of the parameters such as sonar problems and medical diagnostics. The subsequent "stagnation" in the development of neurocomputers in Russia was due to the general lag in technology, which is only now being overcome.

The most promising realization of a neural network, which opens up a route to sixth-generation computer architecture, is an optical neurocomputer. In it, extensive parallelism can be achieved together with a computing rate 2–3 orders of magnitude higher than in the existing supercomputers under otherwise equal conditions. A great advantage is that as the light rays propagate they do not influence each other and can intersect many times. The matrix of synaptic junctions, realized in the form of a three-dimensional hologram of size 1 cm^3, can contain more than a trillion connections for the optical images stored in it.[39] An optical realization of Hopfield's model is described in Ref. 40, and the optical realization of a model of associative memory in Ref. 41. (see Fig. 10). Moreover, nothing prevents further miniaturization of the elements of the neural network, and this will make it possible to use successfully the advances of nanotechnology.

There is an impressive brief list of results of the use at the FNAL tevatron[44] of specially constructed multilayer perceptrons, which were realized in a single crystal and were used both for fast (a few microseconds) on-line recognition of muon tracks with an accuracy inferior by only a factor 2 compared with complicated off-line methods and for jet recognition in $p\bar{p}$ collisions, and also triggers realized instrumentally in a commercial ANN (INTEL 8170NX Electrical Trainable Analog Neural Network, Intel Corp., Santa Clare, California).

With the development of technology, when transputer systems crossed the frontier of 1.5-μm microminiaturization, the possibility arose of developing even "neurochips." Unfortunately, this technology is not yet available to us because of the strict KOKOM rules. Nevertheless, attempts to develop neurochips using less sensitive domestic technology may lead to a significant growth of results with respect to productivity. The domestic production of neurocomputers based on the standard microprocessor basis is actually not inferior to the American production. In particular, as Galushkin asserts in the interview that we quoted, the domestic neurocomputer Gerkules (Hercules) is as effective as the well-known neurocomputer AWZA.

At the present time, the greatest difficulty arises from the problems of translating the many topical problems associated with pattern recognition in the analysis of signals and images and with the control of the nonlinear dynamical systems that arise in the automation of mechanical, physical, and chemical processes into algorithms suitable for introducing them into neurocomputers. This difficulty has stimulated the development of a new branch of computational mathematics—"neuromathematics."

5. APPLICATIONS OF ANNs IN ELEMENTARY-PARTICLE PHYSICS

Three stages in the analysis of experimental data

Among the ever growing number of applications of ANNs, an important place is occupied by problems involving the analysis of experimental data in elementary-particle physics. They can be nominally divided into three groups in accordance with the three main stages in the analysis of data (on-line detection, off-line analysis to obtain physical parameters, and subsequent interpretation with testing of hypotheses). This analysis itself can be regarded as a rigid selection aimed at a radical reduction of redundancy in the experimental information, since in modern high-statistics experiments a useful event may be contained among 10^{10}–10^{12} background events. However, the methods must not be so stringent that they lose this one useful signal. Examples of experimental facilities and the formulation of the associated mathematical problems of analysis in all three stages can be found in the reviews of Refs. 50 and 51.

The first step of the selection is made already in the process of the experiment itself by means of a so-called trigger system, i.e., by means of ultrafast electronics, to separate the possible useful information from the gigantic flood of experimental data (more than 10^7 events per second). This gives a considerable (by 4–5 orders of magnitude) increase in the probability of finding interesting events and a corresponding reduction in the volume of detected data that must be analyzed in the subsequent, second stage. Applications of ANNs in the first stage of analysis of experimental information are illustrated by examples in the most recent publications,[42,52,53] which give descriptions of the realization of second-level triggers in various calorimeters and fast-tracking systems.

The use of ANNs in a second-level trigger to detect jets and centers of showers in calorimeters was investigated in Refs. 45 and 46, in the first of which a commercial ANN was used. In both cases, the use of a three-layer network with one output and appropriate training on several hundred events showed that it was entirely suitable for a second-level trigger. The question of the number of neurons in the hidden layer and the length of the training sample requires additional investigation. Investigations into the use of ANNs of the type of three-layer perceptrons for the design of a calorimetric trigger to select events with *B* mesons were made at the JINR.[52,53]

In the second stage, the data are filtered from the background, trajectories of charged particles and energy dissipation of neutral particles are recognized, and the events are reconstructed in space in order to determine the required physical parameters.

Investigations on the use of ANNs and cellular automata in the second stage of analysis to filter track information and reconstruct the trajectories of charged particles will be considered in more detail later for the example of experimental data obtained using the cylindrical spectrometer ARES (JINR). There will be a detailed presentation of the methods of accelerating the calculations and overcoming the difficulties that arise with increasing multiplicity of the events and increasing noise in the data.

The data obtained in the second stage are used in the final, third stage to select the most probable physical hypothesis by means of statistical criteria. In the third stage, ANNs are used in problems involving classification of quarks, identification of decays of short-lived particles, and estimates of two empirical distributions.

The problem of determining in a track a deflection through a small angle was solved by means of a three-layer perceptron,[43] at the input of which there were fed in deflections in two planes of measured three-dimensional coordinates from a fitted helicoid and its curvature. The output neuron took the values ±1, depending on the presence or absence of a deflection. In the case of a deflection, the same data were passed to the input of another ANN, which produced the radial coordinate of the point of deflection. The network was trained and tested for robustness with respect to the loss of counts during measurement using data that simulated the ALEPH experiment. Comparison of the results with the most powerful statistical method using a Kalman filter showed that, for comparable accuracy and reliability, the ANN method worked several tens of times faster.

In Ref. 47, a multilayer perceptron was used after training on 27 thousand events corresponding to classes with *b, c,* and light quarks for subsequent classification of 50 thousand events into the same three classes. Much attention was devoted to reducing the number of input variables (to 15) by using the methods of discriminant analysis, and this ensured more rapid learning and subsequent operation of the network.

A similar approach was proposed in Ref. 48 for the classification of decays of Z^0 bosons on the basis of data from the DELPHI experiment and in Ref. 49 for the identification of the possible τ-decay channels. In Ref. 48 there are detailed explanations of the choice of the number of neurons in the hidden layer and the change of the training parameters; also introduced are parameters that characterize the quality of operation of the network after training, for example, the signal efficiency ε_s (the number of events correctly put in a definite class divided by the total number of events in this class) and the frequency p (the number of events correctly put in a definite class divided by the total number of events classified as belonging to this class).

Cellular automata

A cellular automaton can be regarded as a simplified local form of a neural network. Simple and transparent, it enables one to understand the basic features of a parallel algorithm.

Cellar automata[54] are dynamical systems that evolve in discrete, usually two-dimensional, spaces consisting of cells. Each cell can take several values; in the simplest case one has a single-bit cell: 0 and 1. The laws of evolution are local, i.e., the dynamics of the system is determined by an unchanged set of rules, for example, a table, in accordance with which the new state of a cell is calculated on the basis of the states of the nearest neighbors surrounding it. It is important that this change of states is made simultaneously and in parallel, and the time proceeds discretely.

Thus, cellar automata make it possible to simulate many phenomena in the natural sciences, and also to create general models of parallel computing processes (like a Turing machine, which makes it possible to simulate the most general serial calculations).

Cellular automata became particularly popular in the seventies through the publications of M. Gardner in Scientific American (later translated into Russian) devoted to

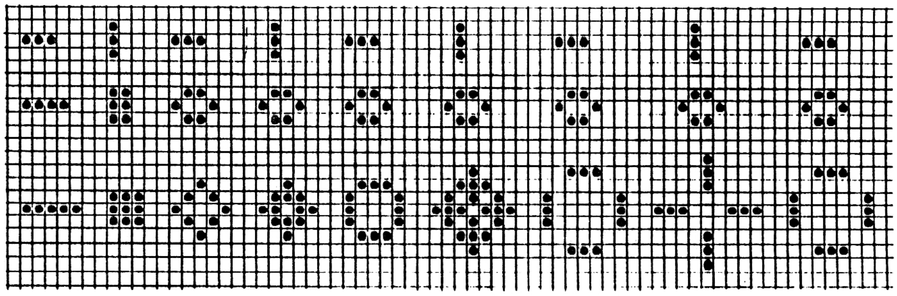

FIG. 11. Examples of the development of colonies in the game Life.

Conway's game Life.[55] The rules of the simple cellular automaton in this game imitate the development of a colony of formal organisms—cells that are born and die, i.e., take the values 1 and 0, depending on the number of neighbors that surround each of these cells. A measure of the flow of time is provided by the succession of generations of the colony, which occurs in accordance with the following rules:

1. All live cells in the eight cells next to a given cell along the horizontal, vertical, and diagonal are regarded as the neighbors of that cell.

2. If a cell has less than two neighbors, it dies from loneliness. If a cell has more than three neighbors, it dies from overcrowding.

3. If next to an empty cell there are precisely three live neighboring cells, then in this cell a live cell is born.

4. The cells die and are born at the changing of the generations. Thus, a cell that has died can give birth to a new cell, but a cell that is born cannot resurrect one that has died, and the death of one cell, reducing the local density of a population, cannot prevent the death of another.

Such a colony can grow all the time, continuously changing its position, shape, and number of cells. However, most often the colony ultimately becomes stationary or repeats cyclically the same finite set of states. The length of the cycle is called the period of the colony. An example of the development of a colony is shown in Fig. 11. The number of the generation increases to the right. The upper series represents an oscillating triplet, a so-called blinker. A configuration that becomes stable in the third step is shown in the central row. The bottom row shows a more complicated configuration, which initially grows to the seventh step but then decays into four blinkers.

On the basis of this example, one can formulate general rules for constructing cellular automata:

1. The state of a cell is discrete (usually 0 and 1, although one can have automata with a larger number of states).

2. There are a restricted number of cells that are neighbors; usually they are the nearest cells.

3. The rules that determine the dynamical development of the cellular automaton usually have a simple functional form and depend on the problem to be solved.

4. A cellular automaton is a tractable system, i.e., the changes of the states of the cells take place simultaneously.

On the basis of the rules for constructing cellular automata, we formulate specific features of an automaton for filtering tracks in proportional chambers.

First, we define a live cell as a cluster, i.e., a continuous group of wires that have fired, and we call an empty cell, not containing a count, a dead cell. To keep broken (due to lack of response of chambers) tracks alive, it is also necessary to introduce phantom cells, which would correspond to clusters if the chambers had correctly responded. Thus, in our case a cell has four states.

Second, to specify the rule for determining neighbors, we consider the characteristic features of a discrete detector of the type of a multiwire proportional chamber.[56]

In the general case, a multiwire proportional chamber (Fig. 12) consists of a planar grid of equally spaced anode wires that lies between two parallel planes of grounded cathodes. When a charged particle passes through the grid, an electron avalanche develops in the gas that fills the chamber, and this leads to a pulse on the nearest wire. The equipotential lines in the proportional chamber make it possible to describe between the wires imagined equidistant planes perpendicular to the plane of the wires. In addition, by virtue of the presence of electronegative additives in the chamber gas, the electron-collection distribution function is a step function, i.e., on both sides of the plane of wires one can describe two further imagined planes such that all electrons formed within them will be collected on the wire.

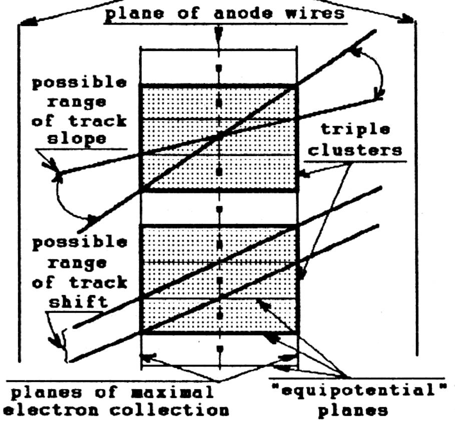
cathode planes
plane of anode wires
possible range of track slope
triple clusters
possible range of track shift
planes of maximal electron collection
"equipotential" planes

FIG. 12. Schematic representation of the operation of a proportional chamber.

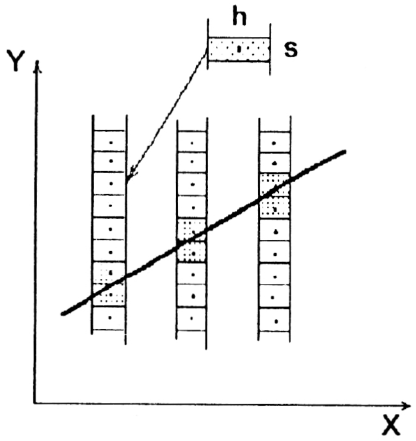

FIG. 13. Model of a detector.

However, electrons that drift out of the region bounded by these two planes will be absorbed by the gas.

Both of these features enable us to regard the chamber as a chain of imagined rectangles, each of which surrounds a signal wire (Fig. 13). Two parallel sides of each rectangle are defined by the "equipotential" planes, and the other two are formed by the planes of maximal electron collection. This simple model is a fairly good approximate description of the operation of a proportional chamber.

When a charged particle crosses a rectangle, the wire inside it responds. If the track crosses several neighboring rectangles in the chamber, they all respond, forming a cluster. In the last case, the track crossed the left-hand side of the lower rectangle and the right-hand side of the upper rectangle (or vice versa, depending on the direction in which the track is traversed). The most important feature of this model of a discrete detector is the possibility of approximate solution of the inverse problem of reconstructing the track, namely, knowing the cluster structure, one can draw a conclusion about the range of possible angles at which the track passed through the cluster. In addition, when the slope of the track is fixed, the cluster structure determines the possible region of points of intersection of the track with the plane of signal wires.

Thus, we specify a rule for defining neighbors on the basis of the characteristic features of a track in the discrete detector, namely, neighbors are the clusters that lie in adjacent chambers through which one can describe a physically reasonable (admissible) track. We also define the region of possible neighbors as the region swept out in the neighboring chambers by such admissible tracks.

Third, we determine rules for the evolution of the automaton that will remove noise but keep the tracks. In the first place, we must restore clusters that have not responded in the chambers. We assume that if in a given chamber at the position of intersection of regions of possible neighbors of adjacent chambers there are no neighbors, then there was a failure in the operation of the chamber or the electronics. In this case, it is necessary to create a phantom cell, which is a neighbor of the cells that indicate it. It is then necessary to annihilate the noise points, i.e., the points for which there are either too few neighbors or too many. For the case of three-track events, we annihilate cells having less than two or more than four neighbors. To prevent the dying of tracks at the ends, we introduce nominal 0 and $(N+1)$ chambers, which are filled with neighbors that sustain all the cells of the outermost chambers.

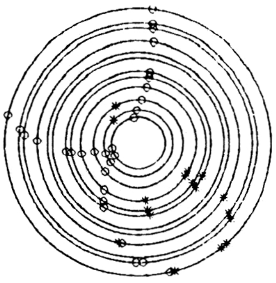

FIG. 14. Example of operation of a cellular automaton.

Fourth, we separate the production and death of cells into different tacts. In the first tact, we create cells, and in the second we let them die. This will be done in each step. This measure ensures the survival of broken tracks.

To register the transition of the automaton to a stable state of cyclic repetition, for each generation a check sum CRC is calculated, and if the check sums are the same, the iterations are stopped.

In the remaining details, the track filter is identical to a standard cellular automaton.

The cellular automaton was tested on track data obtained in an experiment[57] to look for the decay $\mu^+ \to e^+e^+e^-$ and to investigate the decay $\pi^+ \to e^+\nu_e e^+e^-$ (Ref. 58). In this experiment, there were ten chambers, i.e., on the average there were ten experimental points (clusters) per track.

An example of the operation of the cellular automaton is shown in Fig. 14. The crosses denote clusters rejected by the automaton as noise. In the upper part of chambers 2 and 3 (beginning from the center) one can see two rejected noise clusters. The track in the lower part of the figure returned to the detector after a collision with the outer wall, and this part was also rejected by the automaton. In chambers 9, 11, and 12 there are clusters from delta electrons close to the track. The cluster length is not shown in the figure, but two of these clusters were rejected precisely on the basis of an analysis of the cluster length. However, the delta-electron cluster in chamber 11 had the same length as the cluster lying on the track. Therefore, the automaton retained both of these clusters as admissible. Note the absence of a response of chamber 10 in this track. In this chamber, the cellular automaton created a phantom cell, and this saved this broken track from annihilation. This phantom cell is not shown in the figure, since it was used temporarily only during the stage of the evolution process of the cellular automaton. Chambers 7 and 8 were not used during the experiment.

Figure 15 shows the distribution of the number of events with respect to the number of clusters in an event (the thinner curve is the distribution after the operation of

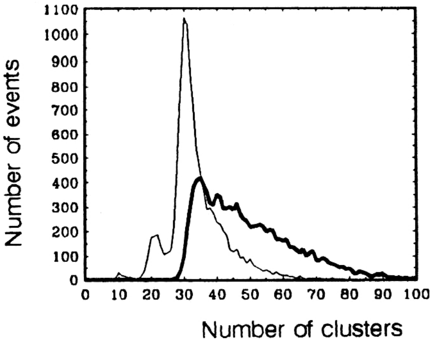

FIG. 15. Distribution of the number of events with respect to the number of clusters in an event before and after (the thinner curve) operation of the cellular automaton.

the automaton). In the original distribution it can be seen that as a result of the operation of the preliminary-selection programs (on-line analysis) there are almost no events with the number of clusters less than 30, and one can also see a long noise tail extending to 100 clusters per event. In the distribution obtained as a result of the operation of the automaton, there are clear peaks in the region of single- and two-track events, which remained as a result of spurious interpretation in the previous stage of the analysis, and three-track events (10, 20, and 30 clusters, respectively). It can be seen that the cellular automaton correctly eliminates the noise (on the average, 65–70%), after which we can make a natural determination of the number of tracks using the number of clusters.

The advantages of the automaton are its simplicity and speed. The program that realizes the cellular automaton on a personal computer evaluates approximately 25 events per second. Insertion in the computer of a specialization permitting *parallel* operation increases the analysis rate up to 1500 events per second or more. Such a rate makes it possible to use the cellular automaton in the on-line regime.

However, the result of operation of the cellular automaton most important for what follows is the grouping of the cells in accordance with the principle of possible membership of a track. It is obvious that the local mode of operation of the cellular automaton (allowance for only nearest neighbors) does not allow it to separate closely spaced or intersecting tracks. This problem must be dealt with in the next stage of the analysis.

The efficiency of the cellular automaton was estimated by examining several samples and was found to be 98%.

Use of ANNs to detect tracks of charged particles

Segment model. The first attempt to use neural networks to recognize tracks was made by Peterson[59] and Denby.[60] Their approach (the so-called segment method) is briefly described as follows.

There is a set of N experimental points in a plane. It is required to describe continuous smooth curves (tracks) through these N points. It is assumed that the tracks do not have kinks and bifurcations.

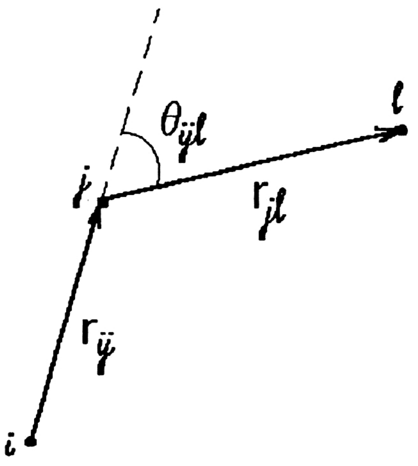

FIG. 16. The segment model.

We introduce binary neurons s_{ij} which establish whether points i and j are connected ($s_{ij}=1$) or not ($s_{ij}=0$), i.e., whether a given directed ($s_{ij}\neq s_{ji}$) segment belongs to a track or not. The energy function is constructed in the form

$$E=E_{\text{cost}}+E_{\text{constr}}. \tag{24}$$

The first (cost) term is chosen in such a way that it encourages short adjacent segments with a small angle between them (see Fig. 16):

$$E_{\text{cost}}=-\frac{1}{2}\sum_{ijkl}\delta_{jk}\frac{\cos^m\theta_{ijl}}{r_{ij}r_{jl}}, \tag{25}$$

where m is an odd integer exponent.

The second (penalty) term consists of two parts:

$$E_{\text{constr}}=\frac{\alpha}{2}\left[\sum_{ik}s_{ik}s_{kl}+\sum s_{ij}s_{jl}\right]+\frac{\beta}{2}\left[\sum_{ij}s_{ij}-N\right]^2. \tag{26}$$

The first part takes into account the ban on bifuractions, while the second establishes a balance between the number of active neurons and the number of experimental points. The parameters α and β are Lagrangian multipliers.[2)]

Application of the method of gradient descent to the energy function (24) leads to an evolution equation for neuron s_{ij} in the form of a step function:

$$s_{ij}=\frac{1}{2}\left(1+\text{sign}\left(-\frac{\Delta E}{\Delta s_{ij}}\right)\right). \tag{27}$$

However, such a procedure generally leads to some local minimum of the energy function, and this is not a satisfactory solution of the problem. The standard way of overcoming this problem is to introduce statistical noise into the system by means of mean-field theory; this leads to replacement of the step function (27) by a function of sigmoid form:

$$v_{ij}=\frac{1}{2}\left(1+\tanh\left(-\frac{\partial E}{\partial v_{ij}}\frac{1}{T}\right)\right). \tag{28}$$

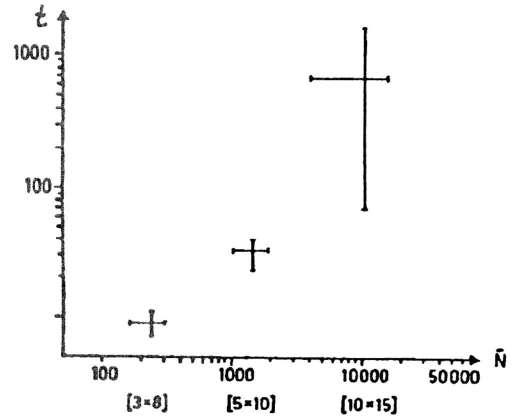

FIG. 17. Convergence time of the segment model as a function of the number of active neurons.

Here, as before, $v_{ij}=\langle s_{ij}\rangle_T$ are the values of the discrete neurons, averaged over the thermal ensemble, so that they are now continuous neurons with range of variation [0, 1]. Equations (28) are solved iteratively until a stable state is reached. The parameters α, β, T are adjustable.

In the segment model, N experimental points lead to $N(N-1)$ neurons and equations. In principle, N^3 operations must be made in each iteration. However, because most problems in track searches are local in nature, this number can be significantly reduced. There is a very low probability that two widely separated experimental points will be directly connected. Therefore, one can introduce a cutoff radius R_{cut}, which characterizes the interaction range of the neurons. This leads to a decrease in the number of active neurons by 2–3 times. If the average number of active partners within the interaction range R_{cut} is $\tilde{N}$, then $O(N\tilde{N}^2)$ calculations are needed. The results are not particularly sensitive to R_{cut}.

At the end of the evolution, neurons with $v_{ij}>0.5$ are taken to be active. The final state is then "cleaned," since the network may leave a certain number of bifurcations.

The segment method leads to good recognition of an event with a small number of tracks in the absence of experimental noise points. Figure 18 shows a typical evolution of the state of the neural network in the process of iterative solution at different stages for a five-track event.

A study was made of the operation of the segment method with increasing multiplicity $N_{\text{track}}\times N_{\text{signals/track}}=3\times 8$, 5×10, and 10×15. The convergence time was found to depend linearly on the number $\tilde{N}$ of active neurons (see Fig. 17).

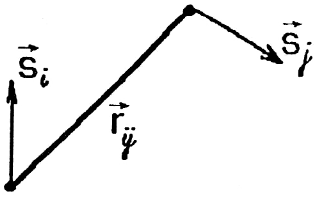

FIG. 19. Rotor model.

Such a network was found to be sensitive to the presence of noise counts. To overcome this shortcoming, it was proposed to increase the weight of the neuron segments along which additional "confirmatory" experimental points are found.[35]

Rotor model. The rotor model[59,62] can be regarded as a further development of the segment model.

In this model (Fig. 19), the neurons are represented in the form of rotors (unit vectors) $\mathbf{s}_i$. Thus, the neurons are characterized by a coordinate (they are placed at the experimental points) and a direction (which reflects the tangent to the track at the given point). The angle is a dynamical variable. These rotors interact with each other (in accordance with some proximity principle) and with the vector $\mathbf{r}_{ij}$ that connects them. For alignment of the rotors in tracks, the following energy function is proposed:

$$E=-\frac{1}{2}\sum_{ij}\frac{\mathbf{s}_i\mathbf{s}_j}{|\mathbf{r}_{ij}|^m}-\frac{\alpha}{2}\sum_{ij}\frac{(\mathbf{s}_i\mathbf{r}_{ij})^2}{|\mathbf{r}_{ij}|^m}, \tag{29}$$

where the first term aligns the rotors parallel to each other, and the second rotates them along the track. The factor $|\mathbf{r}_{ij}|^{-m}$ establishes locality of the interaction. The balance of these two influences is established by the Lagrangian multiplier α.

Mean-field theory gives the dynamical equations

$$\mathbf{v}_i=\frac{-\frac{\partial E}{\partial \mathbf{v}_i}I_1\left(\left|-\frac{\partial E}{\partial \mathbf{v}_i}\right|/T\right)}{\left|-\frac{\partial E}{\partial \mathbf{v}_i}\right|I_0\left(\left|-\frac{\partial E}{\partial \mathbf{v}_i}\right|/T\right)}, \tag{30}$$

where $\mathbf{v}_i=\langle \mathbf{s}_i\rangle_T$ are the mean-field variables, and I_1 and I_0 are Bessel functions. Note that $\mathbf{v}_i$ are not unit vectors but take values in the range $0\leqslant|\mathbf{v}_i|\leqslant 1$. The length of the neuron $\mathbf{v}_i$ expresses the probability that it belongs to a track.

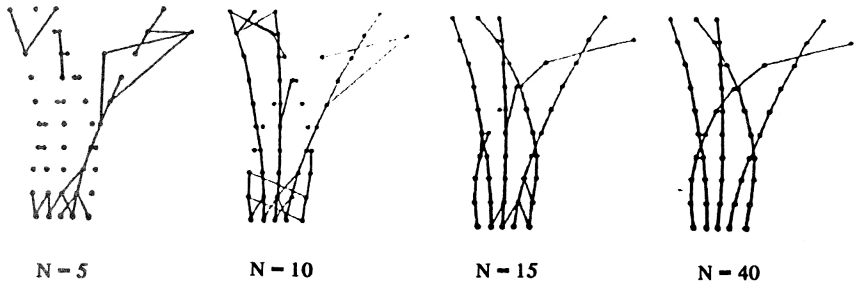

FIG. 18. Evolution of neural network in the segment model.[62] Only neurons for which $v_{ij}>0.5$ are shown; N is the number of iterations.

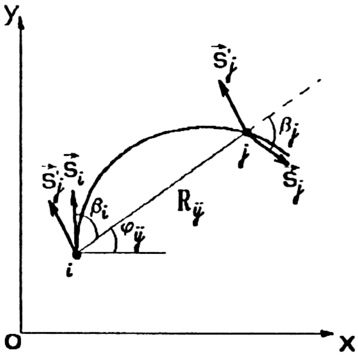

FIG. 20. Modified rotor model.

The reason for this is that in mean-field theory with nonvanishing temperature there is a dependence of the length of the vector $\mathbf{v}_i$ of the mean spin on the strength of the local field $-\partial E/\partial \mathbf{v}_i$ at site i, and this is a maximum for points situated on a track and decreases strongly with increasing distance from it.

The main advantage of the rotor model is the reduction in the number of dynamical variables to $2N$.

Modified rotor model. The modified rotor model of a neural network was originally developed to reconstruct tracks in proportional chambers.[63,64]

A natural consequence of the model of the operation of a proportional chamber given above (see Fig. 12) is the rotor model of a neural network, in which a neuron is characterized by a value, a coordinate, and an inclination. The following modification of the rotor model can be proposed. As an initial premise, we assume that the signals from a particle lie rather well on a certain circle. In the process of operation, the neural network must align the vectors $\mathbf{v}_i = \langle \mathbf{s}_i \rangle_T$ along the tangent to this circle and strongly decrease $\mathbf{v}_j$ for the j that do not lie on any true particle trajectory.

We first write down a two-particle interaction energy. Suppose that a circle passes through two points i and j (see Fig. 20); then the tangent vectors at them are related by

$$\beta_i = -\beta_j, \tag{31}$$

where β_i and β_j are the angles between them and the chord R_{ij}. Thus, if we take an energy function of the form

$$E_{ij} = -\mathbf{s}_i\mathbf{s}'_j, \tag{32}$$

where $\mathbf{s}'_j$ is the vector $\mathbf{s}_j$ reflected with respect to the chord R_{ij}, it will have a minimum for points that lie on one track. Note that the use of the scalar product makes the fields from different neurons additive.

The vector $\mathbf{s}'_j$ is most readily calculated using the following geometrical considerations. To reflect $\mathbf{s}_j$ with respect to R_{ij}, it is sufficient to rotate R_{ij} through the angle $-\varphi_{ij}$, so that it coincides with the OX axis; one must then make a reflection with respect to OX, and then reverse the angle to $+\varphi_{ij}$. In matrix form, this transformation is written as

$$T_{ij} = \begin{pmatrix} \cos(2\varphi_{ij}) & \sin(2\varphi_{ij}) \\ \sin(2\varphi_{ij}) & -\cos(2\varphi_{ij}) \end{pmatrix}. \tag{33}$$

Finally, the energy function takes a form that is much simpler than (26) and (29):

$$E = -\frac{1}{2} \sum_{i,j} s_i T_{ij} s_j. \tag{34}$$

For what follows, it is convenient to introduce the field $\mathbf{h}_i^j$ produced at the point of neuron i by neuron j. Obviously, the field produced by all the neurons at this point is equal to the sum of the fields from each neuron:

$$\mathbf{H}_i = \sum_j \mathbf{h}_i^j = \sum_j T_{ij} s_j. \tag{35}$$

In the new notation, our problem is formulated as follows:

$$E = -\frac{1}{2} \sum \mathbf{H}_i \mathbf{s}_i \to \min. \tag{36}$$

We now turn to the formulation of the dynamical equations. In the first place, we describe the evolution of the neuron angle of inclination. It is the same as in the rotor model:

$$\mathbf{v}_i^{(m+1)} = \frac{\mathbf{H}_i^m}{|\mathbf{H}_i^m|} \frac{I_1(|\mathbf{H}_i^m|/T)}{I_0(|\mathbf{H}_i^m|/T)}. \tag{37}$$

In the determination of the field here allowance is made for only connections between neurons that are admissible from the point of the view of the discrete structure.

Besides the change in the neuron angle of inclination, the discrete structure of the detector enables us to move the neuron position within the width of the cell relative to the cluster center (see Fig. 12). In our case, the neuron position can be changed at zero temperature, i.e., the neuron can be displaced at once to the point of maximum field or to the furthermost allowed point if the field maximum lies outside the allowed range of neuron positions.

A program that realizes the modified rotor model of a neural network was tested on real track events obtained using the ARES spectrometer during experiments to look for the forbidden decay $\mu^+ \to e^+e^+e^-$ (Ref. 57) and to investigate the rare decay $\pi^+ \to e^+\nu_e e^+e^-$ (Ref. 58). In this experiment, only 10 chambers were used. This means that each track contains on the average 10 experimental points (clusters).

Figure 21 shows the initial state of the neural network and the result of its operation. A "good" initial state could be chosen through the use, during the initial stage, of analysis of a cellular automaton that eliminated noise points (shown by crosses in the figure) and grouped the experimental points in accordance with the possibilities that they had to belong to different tracks. The result of operation of the neural network also shows that it works well in the case of closely spaced tracks.

On the average, the neural network solves the problem in four or five iterations. This high rate of convergence was made possible by the preliminary use of the cellular automaton, as we have already mentioned, and by allowance for

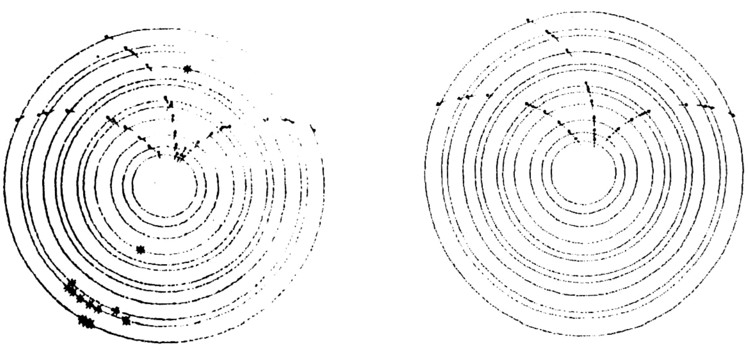

FIG. 21. Example of operation of a neural network: the initial and final states.

the discrete properties of the detector. The efficiency of the neural network was estimated by examination of several samples and was found to be 98%.

Analysis of the operation of the neural network shows that the computing time needed to find the global minimum of the energy function depends on two main factors: the ability of the ANN algorithm to avoid "falling" into one of the local minima; the number of degrees of freedom of the ANN, which for an ANN with N input signals is proportional to N^2 in standard search algorithms.

Among the methods that make it possible to overcome the first factor, the best known is the procedure of simulated annealing, which was described earlier and requires an appreciable amount of computing time. In this connection, a different approach was used in Ref. 65, which was to specify the initial configuration of the ANN rotors in such a way as to be near the global minimum. One way of doing this has already been mentioned—the use of a cellular automaton, both to reduce the number of input data by filtering them and to partly join individual counts into tracks on the basis of proximity. At the same time, the rotor model described above was too strongly oriented to the specific features of proportional chambers and undoubtedly needed further development to take into account effects associated with advance into the range of primary-beam energies of the order of several tera-electron-volts, which is associated with high multiplicity and intersections of tracks at small angles. In addition, there were indications of a higher sensitivity of such ANN models to noise.[35]

In this connection, it was necessary to investigate the following problems associated with the extraction of track information by means of ANNs:

●The creation of an optimum initial configuration of the ANN by means of an algorithm sufficiently general for applications, both in the presence and in the absence of a magnetic field.

●Robustness with respect to input data containing a lot of noise (signal-to-noise ratio up to 100%).

●Stability with respect to increasing multiplicity of the events.

A similar and, apparently, more general approach was considered in Ref. 32, which proposed a combination of a local Hough transform and the method of deformable templates. However, in Ref. 65, new suggestions were made for almost all stages of the ANN logarithm:

1. In the stage of initial formation of the rotors, it was proposed to determine their initial directions at each experimental point, using the peak of a special angle histogram, and the rotor length, using the relative number of points found at this peak.

In principle, this could give a good possibility for creating a deformable template (track candidates) and estimating their total number, but the authors left the ANN itself to develop this initial information to determine the most important connections.

2. As a device to keep out extraneous and noise neurons, it was proposed to multiply the weights T_{ij} by special robust factors taken from the theory of robust M statistics[66] of the type

$$w(t)=\begin{cases}\left(1-\left(\dfrac{t}{C_T}\right)^2\right)^2 & |t|\leqslant C_T;\\ 0 & |t|>C_T,\end{cases}$$

where $t = \widehat{v_i v'_j}$, $\mathbf{v}'_j = T_{ij}\mathbf{v}_j$ [see (33), (35), and (37)], $C_T=2°$.

3. The procedure of simulated annealing was replaced by the ANN dynamics (36)–(37) with optimally chosen temperature $T=1.5$.

This approach was applied to both straight and arc tracks, and it was tested in detail in Ref. 65 on the model already used in Ref. 32 with straight tracks emanating radially from a target. The model was made more complicated by having each track emanate from its own "vertex" in the target, this resulting in intersection of tracks at very small angles. To achieve greater resemblance to real events, the authors of Ref. 65 used the parameters of the cylindrical spectrometer ARES. The multiplicity was varied from 10 to 50 tracks (at high multiplicity, the tracks began to merge because of the continuous exposure of the chambers). After discretization, noise points were added to the data in a number equal to the number of "useful" counts (100% signal-to-noise ratio).

Thus, the neural network was used as a filter of noise counts. An example of such a 30-track event is shown on the left in Fig. 22. In this event, there are 300 points corresponding to tracks and 300 noise points (10 points per track). The result of applying the neural network is shown on the right. Here, the neurons rejected as noise are not shown, and the active neurons are represented by vectors with length proportional to their activation level. The outermost lines show the modeled straight tracks. In this case, 251 noise points were rejected by the network, and the

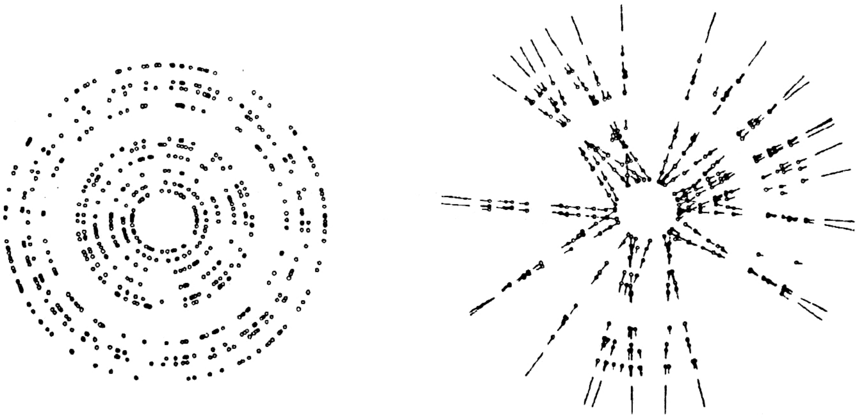

FIG. 22. Filtering by the neural network of a 30-track event with 100% noise.

remaining 49 noise points were added to track groups (on the average, 1.6 points per track).

The statistics of 1000 model events showed that the mean number of iterations increased from 5 to 20 when the multiplicity increased from 10 to 50 (although the spread of this number was fairly large, depending on random concentrations of tracks).

The reliability of such an ANN as a filter was found to be 100%, i.e., the proposed criterion for determining the global minimum,

$$\max|\mathbf{v}_i^{(m+1)}-\mathbf{v}_i^{(m)}|<0.05,$$

where m is the number of the iteration, makes it possible to include all model tracks in the set of track candidates and eliminate the overwhelming majority of the noise points (up to 85% for multiplicity 30). The small number of fictitious tracks formed from fragments of closely spaced tracks with the addition of noise points was readily eliminated by a subsequent straightening of the obtained tracks, together with one or two extraneous points that had become "stuck" to the real tracks.

6. CONCLUSIONS

In preparing this review, we mainly consulted, besides the few reviews in the domestic literature,[37,67,68] foreign sources, among which we may mention the well-known introductory reviews in the first issue of the journal Neural Networks of Kohonen, Grossberg, and Lippman,[16,69,70] and also Khanna's fundamental monograph.[6] Of course, we can make no claim to have included in this rather short review all work on artificial neural networks. Broadly speaking, the multidisciplinary nature of the approach to ANNs that we noted in the Introduction is determined by three directions of investigation: mathematical, physical, and technological. To these there correspond three groups of publications, of which the most extensive is the group with technological direction. Besides the information given in Sec. 4, we may also mention the detailed Russian-language review of hardware and software developments of ANNs in 15 North American cooperations given in Shirokov's paper in the collection of Ref. 68. Information on hardware parallel realizations of cellular automata can be found in the books of Refs. 54 and 71.

It may be that in this review we have not devoted sufficient attention to the literature on the physical interpretation of ANNs as large thermodynamic systems, although the fruitfulness of using mean-field theory, which gives an effective solution to the problem of finding the global minimum in Hopfield's model, has been rather strongly emphasized in Sec. 3. In the same group of studies on the physical interpretation of ANNs, we should probably also include studies that use the method of simulated annealing.[31,72]

The main attention in this review has been devoted to the literature on mathematical models, possibly because they have provided the most numerous successful applications of ANNs in the analysis of the data of physics experiments, in particular, in high-energy physics. Besides the rapidly progressing possibilities for using neurochips with previously developed logic in various trigger systems,[44–49] promising tendencies in the physical applications of the mathematical models are reflected in Refs. 32, 35, and 73, which develop the method of deformable templates and the elastic-arm method, and also the "dynamic" perceptron,[74] in which static determination of the characteristic functions of the sets that determine the range of action of the neurons is replaced by a dynamic determination. Such a modified topology makes it possible to use successfully this dynamic perceptron for rapid recognition of track information. Some helpful recommendations and standard programs for neurocalculations are given in Ref. 75. Recently, an ANN program classifier has also appeared in the CPC Program Library of Computer Physics Communications.[76]

From the point of view of the physics applications of ANNs, we were strongly influenced by studies made by several authors including C. Peterson,[32,59] whom we thank for his interest, for helpful discussions, and for sending us new reprints of his papers. We also thank Professor L. Zanello, who sent us copies of many papers that we used in this review.

[1)] More details about the history of neurobiology can be found in the paper by D. Hubel in Ref. 1.

[2)] Usually, the solutions are stable with respect to these parameters.[61]

[1] Sci. Am. **241**, No. 3, 45 (1979) (published as a book in Russian: *The Brain*, Mir, Moscow, 1984).

[2] W. S. McCulloch and W. H. Pitts, Bull. Math. Biophys. **5**, 115 (1943).

[3] D. O. Hebb, *The Organization of Behavior* (Wiley, New York, 1949). p. 62.

[4] T. K. Kohonen, *Content-Addressable Memories* (Springer-Verlag, Berlin, 1980) [Russ. transl., Mir, Moscow, 1982].

[5] R. Sutton and A. Barto, Psychol. Rev. **88**, 135 (1981).

[6] T. Khanna, *Foundation of Neural Networks* (Addison-Wesley, New York, 1989).

[7] B. Widrow and M. E. Hoff, Adaptive Switching Circuits. IRE WESTON Convention Record **4**, 96 (1960).

[8] F. Rosenblatt, Psychol. Rev. **65**, 386 (1958).

[9] F. Rosenblatt, *Principles of Neurodynamics* (Spartan, Washington, D.C., 1962) [Russ. transl., Mir, Moscow, 1965].

[10] J. J. Hopfield, Proc. Nat. Acad. Sci. USA **79**, 2554 (1982).

[11] J. J. Hopfield, Proc. Nat. Acad. Sci. USA **81**, 3088 (1984).

[12] J. J. Hopfield, Proc. Nat. Acad. Sci. USA **84**, 8429 (1987).

[13] S. Amari, IEEE Trans. Syst. Man Cybern. **13**, 741 (1983).

[14] J. A. Anderson, IEEE Trans. Syst. Man Cybern. **13**, 799 (1983).

[15] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. (Springer-Verlag, Berlin, 1990).

[16] T. Kohonen, Neural Networks **1**, 3 (1988).

[17] S. Grossberg, Math. Biosci. **4**, 210 (1969).

[18] S. Grossberg, J. Stat. Phys. **3**, 95 (1971).

[19] S. Grossberg, J. Stat. Phys. **1**, 319 (1969).

[20] G. A. Carpenter, J. Diff. Eqns. **23**, 335 (1977).

[21] G. A. Carpenter, J. Math. Anal. Appl. **58**, 152 (1977).

[22] S. Grossberg and A. Elias, Biol. Cybern. **20**, 69 (1975).

[23] J. J. Hopfield and D. W. Tank, Biol. Cybern. **52**, 141 (1985); Science **233**, 625 (1986).

[24] W. Kinzel, Z. Phys. B **60**, 205 (1985).

[25] R. J. Marks II, S. Oh, and L. E. Atlas, IEEE Trans. Circuits Syst. **36**, 846 (1989).

[26] M. Verleysen *et al.*, IEEE Trans. Circuits Syst. **36**, 762 (1989).

[27] G. Palm, in *Brain Theory*, edited by G. Palm and A. Aertsen (Springer-Verlag, Berlin, 1986), p. 211; G. Palm, Science **235**, 1227 (1987).

[28] A. D. Bruce, E. J. Gardner, and D. J. Wallace, J. Phys. A **20**, 2909 (1987).

[29] S. Kirkpatrik and D. Sherrington, Phys. Rev. B **17**, 4384 (1978).

[30] N. N. Bogolubov, Jr., *A Method for Studying Model Hamiltonians* (Pergamon Press, Oxford, 1972) [Russ. original, 2nd ed., Nauka, Moscow, 1974].

[31] C. Peterson, Neural Computation **2**, 261 (1990).

[32] M. Ohisson, C. Peterson, and A. L. Yuille, Comput. Phys. Commun. **71**, 77 (1992).

[33] P. J. Huber, *Robust Statistics* (Wiley, New York, 1981).

[34] R. Durbin and D. Willshaw, Nature **326**, 689 (1987).

[35] M. Gyulassy and H. Harlander, Comput. Phys. Commun. **66**, 31 (1991).

[36] L. Lonnblad *et al.*, Preprint LU TP 91-4, Lund (1991).

[37] S. M. Achasova, Programmirovanie No. 2, 40 (1992).

[38] A. P. Thakoor *et al.*, Appl. Opt. **26**, 5085 (1987).

[39] Y. Abu-Mostafa and D. Psaltis, Sci. Am. **256**, No. 3, 66 (1987).

[40] N. Farhal *et al.*, Appl. Opt. **24**, 1469 (1985).

[41] D. Psaltis and N. Farhat, Opt. Lett. **10**, 98 (1985).

[42] *Proc. of the Intern. Conf. on Computing in High Energy Physics*, CERN 92-07 (Geneva, 1992).

[43] G. Stimpfl-Abele, in *Proc. of the Intern. Conf. on Computing in High Energy Physics*, CERN 92-07 (Geneva, 1992), p. 642.

[44] B. Denby *et al.*, in *Proc. of the Intern. Conf. on Computing in High Energy Physics*, CERN 92-07 (Geneva, 1992), p. 674.

[45] T. Handler and E. Neis, in *Proc. of the Intern. Conf. on Computing in High Energy Physics*, CERN 92-07 (Geneva, 1992), p. 650.

[46] H. Andree, in *Proc. of the Intern. Conf. on Computing in High Energy Physics*, CERN 92-07 (Geneva, 1992), p. 654.

[47] J. Propriol *et al.*, in *Proc. of the Intern. Conf. on Computing in High Energy Physics*, CERN 92-07 (Geneva, 1992), p. 652.

[48] G. Cosmo *et al.*, in *Proc. of the Intern. Conf. on Computing in High Energy Physics*, CERN 92-07 (Geneva, 1992), p. 665.

[49] V. Innocente *et al.*, in *Proc. of the Intern. Conf. on Computing in High Energy Physics*, CERN 92-07 (Geneva, 1992), p. 669.

[50] G. A. Ososkov, in *Proc. of the Intern. Conf. on Probability and Mathematical Statistics PROBASTAT '91* (University Press, Bratislava, 1991) p. 353.

[51] M. Grote, Preprint CERN DD/87/3 (1987).

[52] V. V. Ivanov *et al.*, Preprint R10-92-156 [in Russian], JINR, Dubna (1992).

[53] Yu. A. Budagov *et al.*, Communication R10-93-140 [in Russian], JINR, Dubna (1993).

[54] *Theory and Applications of Cellular Automata*, edited by S. Wolfram (World Scientific, 1986).

[55] M. Gardner, "Mathematical games," Sci. Am. **223**, 4 (1970).

[56] A. A. Glazov, I. V. Kisel, E. V. Konotopskaya, and G. A. Ososkov, Communication E10-91-507, JINR, Dubna (1991).

[57] V. A. Baranov *et al.*, J. Phys. G **17**, Suppl. 57 (1991).

[58] V. A. Baranov *et al.*, Yad. Fiz. **55**, 2940 (1992) [Sov. J. Nucl. Phys. **55**, 1644 (1992)].

[59] C. Peterson, Nucl. Instrum. Methods **A279**, 537 (1986).

[60] B. Denby, Comput. Phys. Commun. **49**, 429 (1988).

[61] C. Peterson and B. Soderberg, Int. J. Neural Syst. **1**, 3 (1989).

[62] C. Peterson, Lund Preprint LU TP 90-6 (1990).

[63] A. A. Glazov, I. V. Kisel, E. V. Konotopskaya *et al.*, Communication E10-92-352, JINR, Dubna (1992).

[64] I. V. Kisel and G. A. Ososkov, in *CHEP92—Conf. on Computing in High Energy Physics* (CERN, Annecy, 1992), p. 646.

[65] S. Baginyan, I. Kisel, E. Konotopskaya, and G. Ososkov, Communication E10-93-86, JINR, Dubna (1993).

[66] G. Ososkov, in *Proc. of the Second Intern. Tampere Conf. in Statistics* (Tampere University Press, 1987), p. 615.

[67] *Automata* [in Russian] (Mir, Moscow, 1956).

[68] *Reviews of Science and Technology*, Vols. 1 and 2 [in Russian] (VINITI, Moscow, 1990).

[69] S. Grossberg, Neural Networks **1**, 17 (1988).

[70] R. Lippman, IEEE ASSP Mag. **4**, 4 (1987).

[71] T. Toffoli and N. Margolus, *Cellular Automata Machines: A New Environment for Modeling* (MIT Press, Cambridge, Mass., 1987).

[72] S. Kirkpatrik, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).

[73] A. L. Yuille, Neural Computation **2**, 1 (1990).

[74] A. Perrone *et al.*, SPIE Proc. Series Washington **1711**, 470 (1992).

[75] R. Necht-Nielsen, *Neurocomputing* (Addison-Wesley, New York, 1990).

[76] A. Cherubin, Comput. Phys. Commun. **72**, 249 (1992).

Translated by Julian B. Barbour