Neural Networks, Vol. 1, pp. 3–16. 1988
Printed in the USA. All rights reserved.

0893-6080/88 $3.00 + .00

ORIGINAL CONTRIBUTION

An Introduction to Neural Computing

TEUVO KOHONEN

Helsinki University of Technology

(*Received and accepted* 28 *September* 1987)

Abstract—*This article contains a brief survey of the motivations, fundamentals, and applications of artificial neural networks, as well as some detailed analytical expressions for their theory.*

Keywords—Neural computing, Neural networks, Adaptive systems, Learning machines, Pattern recognition.

1. HISTORICAL OVERVIEW

Theoretical explanations of the brain and thinking processes were first suggested by some ancient Greek philosophers, such as Plato (427–347 B.C.) and Aristotle (384–322 B.C.). Rather physical views of mental processes were held by Descartes (1596–1650) and the 18th century empiricist philosophers.

The class of so-called cybernetic machines to which the "neural computers" belong have a much longer history than generally believed: Heron the Alexandrian built hydraulic automata around 100 B.C. Among the numerous animal models, which have been built to demonstrate need-driven behavior in variable living conditions, one may mention the "Protozoon" of Lux from 1920, the "dogs" of Philips from 1920 to 1930, the "Homeostat" of Ashby from 1948, the "Machina Speculatrix" and "Machina Docilis" of Walter from 1950, the "ladybird" of Szeged from 1950, the "squirrel" of Squee from 1951, the "tortoise" of Eichler from 1956, and many versions of a "mouse in the labyrinth" (Nemes, 1969).

Abstract, conceptual information processing operations were performed by mechanical devices a couple of centuries ago, for example, the slide rule for the demonstration of syllogisms by Ch. Stanhope (1753–1816), and many mechanisms for set-theoretic and logic operations devised in the 19th century.

Analytical neural modeling has usually been pursued in connection with psychological theories and neurophysiological research. The first theorists to conceive the fundamentals of *neural computing* were W. S. McCulloch and W. A. Pitts (1943) from Chicago, who launched this research in the early 1940s. Models for adaptive stimulus-response relations in random networks were set up by Farley and Clark (1954). These theories were further elaborated by Rosenblatt (1958), Widrow and Hoff (1960), Caianiello (1961), and Steinbuch (1961). Many implementations of "neural computers" were realized in the 1960s.

It is difficult to describe in detail what happened in the subsequent 25 years or so in neural modeling. There were many scientists who made rigorous works on the biophysics of neural networks and psychophysical studies of the sensory systems. It has been estimated that the number of papers on vision alone is at least on the order of 10,000. The number of papers on general neural modeling amounts perhaps to a few thousand.

2. ON THE NATURE OF NEURAL COMPUTING

The primary purpose of all neural systems is *centralized control* of various biological functions. Some of them are responsible for energy supply; the corresponding neural systems are connected with metabolism, cardiovascular control, and respiration. There are neural mechanisms for biological rhythms, emergency functions, etc. The above functions are common to most animals, and in the biological neural structures it is even possible to discern various "sediments" from different phases of evolution. In the higher animals, on the other hand, the major capacity of the central nervous system is connected with *behavior*, that is, control of the state of the organism with respect to its environment, which encompasses many different tasks ranging from elementary actions to complicated social behavior. When talking of "neural computing," however, one usually only has in mind the *sensory and motor functions*, as well as some kind of "internal processing,"

Requests for reprints should be sent to Teuvo Kohonen, Helsinki University of Technology, Laboratory of Computer and Information Science, Rakentajanaukio 2C, SF-02150 Espoo, Finland.

loosely called *thinking*. All of these functions are mutually dependent in one way or another, but it may be possible to conceptualize some of them in idealized forms.

In the development of information technology there now seems to exist a new phase whereby the aim is to replicate many of these "neural" functions artificially. Nonetheless, it is not always clear which ones of the above aspects are meant. A central motive seems to be to develop new types of computers. For instance, one may aim at the implementation of *artificial sensory functions*, in order to make the machines "see" or "hear"; this is an extension of the more traditional instrumentation techniques. Analysis of, say, satellite data may in itself comprise a practical task which has to be automated in one way or another. Then there exist needs to develop "intelligent robots" whose *motor control and other actions* must be integrated with the sensory systems; some kind of internal "neural computer" is needed to coordinate these actions. It further seems that certain expectations are held of new types of "thinking machines," which have no sensors or actuators, but which are capable of answering complex queries and solving abstract problems. The science-fiction robots, of course, are doing all of this. Although the potential of the Artificial Intelligence methods by which these problems were earlier approached has been known for about 25 years, it is still hoped that new avenues to Artificial Intelligence could be opened when massive parallelism of the computing circuits, and new technologies (eventually, optical computing) are developed whereby the computing capacity is increased by orders of magnitude. Before that, however, it will be necessary to find out *what* to compute. There is at least one new dimension of computation visible which has been very difficult to reach by the digital computers, namely, to take into account all the high-order statistical relationships in stochastic data. Apparently the biological brain always has to deal with stochastic signals, whereas even in the best "intelligent" machines developed so far, all data occur in discrete form. In other words, the knowledge they handle is usually of the linguistic type, and the pieces of information inputted to the machines have always been prepared carefully by human beings. One intriguing objective would therefore be to leave such a preprocessing to the machines.

Since the concepts of (artificial) "neural networks" and "neural computers" appear widely in publicity nowadays, it may be necessary to specify their exact meaning. Briefly, in relation to the above motives, we may give the following definition:

"Artificial neural networks" are massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with the objects of the real world in the same way as biological nervous systems do.

It is necessary to note that such "neural computers" do not execute typical machine instructions of digital computers, unless they are made to *emulate* the behavior of physical neural networks. In principle, the basic processing operation performed by every processing element is an *analog operation*, or *transformation* of its input signals.

In biological neural networks, the neural cells (*neurons*), or tightly interacting groups of them correspond to the above processing elements. The interconnections are made by the outgoing branches, the *axons*, which make variable connections, *synapses*, with other neurons (or perhaps with other parts like muscles and glands). Neural networks are thus systems of simple, tightly interconnected processing elements. The elementary function of the latter is to act as selective, "tuned" filters to particular signal patterns. The complex operation of neural networks results from abundant feedback loops which, together with nonlinearities of the processing elements and adaptive changes of their parameters, can define even very complicated dynamic phenomena.

One peculiarity of biological neural networks is their size: in the whole human central nervous system there are on the order of 10^{11} neurons, but the number of their interconnections is still higher, probably up to the order of 10^{15}. It does not seem possible to *program* the function of such a system according to a prior plan, not even by genetic information; consider, too, that the size and structure of the network is changing radically during and after childhood, when it is already in use.

It is true that certain textural features of the network are inherited, and during ontogenesis, the neural projections (certain axon bundles) grow approximately to those places in which they are later needed. In other words, a rough allocation of the resources, and the main communication paths are formed according to a genetic plan, whereas the rest of the "programming," especially the content of *memory*, must be acquired postnatally. Programming of such a network can then only mean two things: (a) The *structures* of interconnections between the cells are altered; and (b) The *strengths*, or "signal transmittances" of these interconnections are changed. It seems that there exist rather clear strategies of how to change the strengths in the right direction, whereas changes in the interconnectivities are more difficult to define, because they usually have radical effects on the network behavior, especially concerning sequential operation and hierarchical functions.

It is indeed very difficult to imagine how such an enormous network could be "programmed" at all. One possibility, especially relating to the sensory subsystems could be that the system structure, or the dynamical process defined by it, in some way directly tends to *image* the sensory experiences or other occurrences. One does not thereby mean any photographic models, static representations of the environment, or metrically

faithful copies of signals; "imaging" must be understood in a more general and abstract sense, for example, that certain feature dimensions of the observations are imaged, or that there appear events in the behavior of the network, the temporal, or logic relationships of which can be put to correspond to similar relationships between some events of the exterior world or its history. This is the "internal representation" in the most general sense, as it apparently occurs in mental images, too. It may be understandable that some kind of images of experiences are formed in the primary sensory areas of the brain, and that memory, whatever its realization, stores copies of experiences and their relations. The latter are then logically faithful, although physically reduced images of the exterior events. The imaging property of memory is needed to complement, correct, and extrapolate (predict) the primary sensory information.

Another important function of the nervous systems is to define *actions* which are part of behavior, and control the state of the organism relative to the environment or circumstances. While the internal representations on which thinking is based can be derived from inputs in a rather straightforward way, definition of the output actions must be based on completely different strategies. In fact, there hardly seems to exist any other possibility to program the actions than to apply the principle of reward and punishment in order to alter the mechanisms which are responsible for them; some kind of *backpropagation* of information is then needed. The meaning and "quality" of the actions, on the other hand, must be judged, not from the immediate movements, but from some performance criterion which takes into account the wanted result, sometimes rather indirectly. Often the actions are only correct if they are made in a certain sequence, whereby the motor mechanisms must contain circuits that define such sequences, and which are changed in relation to the learning results. It will be clear that programming the actions is a much more indirect process than programming the internal representations; random trials may not be avoided.

Programming input and output functions, however, only leads to a "behavioristic" operation in which the stimuli and the responses are considered most relevant. Certainly it is possible to implement rather complex automata and need-driven behavior that way. It seems, however, that some expectations are held about neural networks being able to act as *computers* for abstract problems, too, whereby this computation takes place in the internal state of the network. We shall revert to the problem of *searching from knowledge data bases* (memory) in Section 4.2. Instead, let it be mentioned briefly at this point that one category of problems which is sometimes believed to be amenable to "neural computing" consists of various *optimization tasks.* They can be very simple and concrete, such as optimal allocation of the organism's own resources, or more abstract like the well-known "traveling-salesman problem" (Hopfield & Tank, 1986). In setting up such problems, the neural network is directly regarded to constitute a static or dynamic *analogy* of the problem structure; for instance, if the purpose is to minimize a loss function where the total cost is built up of interdependent cost elements, then it may be possible to put these elements to correspond to the connection strengths in a network; the network structure itself then represents the different combinatorial alternatives of the problem "graphically." The activities in different parts of the network shall isomorphically stand for values of corresponding variables of the problem. When the network, by relaxation of its activity due to its feedback loops converges to certain optimal states, then this state, by isomorphism, is also assumed to define the solution to the original problem. The main problem seems to be who shall program such a network.

One could also interpret the interest in "neural computers" in a slightly different way. The new technologies, such as optical processing of information, high-density semiconductor networks, and eventually new materials like the "spin glasses" offer an unforeseen capacity for computation. On the other hand, their elementary operations may not be perfect. The question is then how it would be possible to utilize this vast, although slightly unreliable capacity. Obviously the brain has an answer to this, and so the new computer technology has been motivated by the biological brain research. Obviously, too, such networks must be highly *adaptable.* This consideration then leads to the next problem: *What component and system properties of the network underlie such a high degree of adaptation that the system, in a complex way, becomes able to exhibit the wanted behavior?* Apparently, if such a property exists, it must also be possible to dress it into analytical *training or learning strategies, and algorithms.* This is the crucial problem of the neural network theory.

3. WHAT THE BIOLOGICAL NEURAL NETWORKS ARE NOT

In order to understand the real potential of neural computing, it will be necessary to make a clear distinction between neural and digital computers. Below are some assertions followed by their argumentation.

The biological neural systems do not apply principles of digital or logic circuits.

A digital computing principle must be either asynchronous or synchronous. If it were asynchronous, the duration of the neural impulses would have to be variable in order to hold one of the binary values for indefinite periods of time, which is not the case. If the principle were synchronous, one would need a global clock to which the pulses are synchronized. This is not

the case either. The neurons cannot be threshold-logic circuits, because there are thousands of variable inputs at most neurons, and the "threshold" is time-variable, being affected by arousal, attention, etc.; the accuracy and stability of such circuits is not sufficient then, to define any Boolean functions. The collective processes which are centrally important in neural computing are not implementable by logic circuits. Accordingly, the brain must be an *analog* computer.

Neither the neurons, nor the synapses are bistable memory elements.

All physiological facts speak in favor of neurons acting as (nonlinear) analog integrators, and the efficacies of the synapses change gradually. At least they do not flip back-and-forth.

No machine instructions or control codes occur in neural computing.

Due to stability problems discussed above, the format of such codes cannot be maintained for any significant periods of time, in particular during the growth processes.

The brain circuits do not implement recursive computation, and are thus not algorithmic.

Due to the stability problems, the neural circuits are not stable enough for recursive definition of functions like in digital computing. An algorithm, by definition, defines a function recursively.

Even on the highest level, the nature of information processing is different in the brain and in digital computers.

In order to emulate each other, at least on some level of abstraction, the internal states of two computing systems must be equally accessible. Such an equivalence does not exist between the brain and the programming systems. Artificial computers can neither acquire nor interpret all the human experiences on which the assessment of values is based.

4. APPLICATIONS OF NEURAL COMPUTERS

This section describes more closely those application areas for which neural computers have been suggested, and the particular problems thereby encountered.

4.1. Pattern Recognition

The term "pattern recognition" was introduced in the early 1960s, and it originally meant detection of simple forms such as handwritten characters, weather maps, and speech spectra. A more ambitious objective, of course, has all the time been to implement *artificial perception*, that is, to imitate the functions of the biological sensory systems in their most complete forms. The first experiments around 1960 were indeed based on elementary "neural networks," known by names like Perceptron (Rosenblatt, 1958), Adaline (Widrow & Hoff, 1960), and Learning Matrix (Steinbuch, 1961), respectively. The first steps, as always, were easy, but it was soon realized that the performance of the biological sensory systems is very difficult to reach. Even high computing capacity, achievable by parallel computing circuits, did not solve the problems, and especially in image analysis there exist requirements which are very difficult to fulfill: (a) *invariance* of detection with respect to translation, rotation, scale, perspective, partial occlusion, and modest marring of the objects, especially under widely varying illumination conditions; and (b) relation of observations to various *contexts* at different levels of abstraction, in order to distinguish the events more selectively.

Notice that animals are capable of paying *attention* to individual objects in a scene, for each of which the invariance of perception must separately be valid. This ought to show that the easy tricks such as preprocessing of the complete scene by the Fourier, or Mellin transformations with subsequent application of "template matching" cannot constitute the whole solution.

What was obviously also ignored is that even the most developed biological sensory systems do not operate autonomously; sensory perception is always closely associated with global *cognitive processes*. For the replication of the sensory functions, it is then not enough to imitate the sensory system, but one has to replicate the whole brain with all its thinking capabilities, and refine the recognition accuracy by high-level learning.

It would, of course, be unreasonable to wait for the solution of all these problems before proceeding with possible applications. In engineering, the problems are usually simplified. Take for example locomotion: it is difficult to implement coordinated limb movements, but a more straightforward method is to apply wheels and to modify the terrain, building roads. Similarly there exist plenty of applications for which artificial, nonnatural solutions may even be more effective than the biological ones. The development of pattern recognition (especially computer vision) took this course in the mid-1960s. Notice that the spatial acuity of the mammalian vision varies by a factor of twenty when comparing the foveal and peripheral areas of the retina, and the eyeball is in continual saccadic or nystagmic motion. Nonetheless, by some delicate sampling and reconstruction of the visual information, a steady and clear visual perception becomes possible. Nothing comparable has been achieved or even tried in computer vision, where the image field, first of all, is dis-

cretized into a regular array of picture elements (pixels) which are then scanned and grouped into homogeneous areas (segmented), their contours are analyzed, and their geometric and topological relationships are described by "picture grammars." It is self-evident that it is possible to achieve certain types of invariance with respect to picture signals, if only topological relations of such parts are taken into account in recognition. Effective as these methods may be, however, they have very little in common with the principles of operation of the biological sensory functions, and thus leave the basic problem of "neural computing" in perception open.

If now artificial "neural network" methods are being developed for the same purpose, it will be necessary to redevelop the "pattern recognition" methods starting from scratch, thereby letting the circuits themselves learn the elementary features and functions, and to become self-organized without heuristic programming.

The most important *application areas* for "neural pattern recognition" could be the same as those for which conventional, heuristic methods have been developed during the past thirty years: (a) remote sensing, (b) medical image analysis, (c) industrial computer vision (especially for robotics), and (d) input devices for computers.

More concrete tasks for which special computer equipment has already been developed are: (a) segmentation and classification of regions from images, (b) recognition of handwritten characters and text, (c) recognition of speech, and (d) processing, especially restoration of noisy pictures.

On a more ambitious level, one may wish to achieve the capabilities of: (a) image analysis (referring to different thematic levels of abstraction, such as monitoring of land use on the basis of satellite pictures), (b) image understanding (interpretation of scenes), and (c) speech understanding (parsing and interpretation of spoken sentences).

To implement these tasks, certain basic problems still call for better understanding, for instance, those concerning the *intrinsic properties* (features) of input information, such as: (a) the most natural pattern primitives (lines, their curvatures and end points, edges, statistics of point groups), (b) visual information which describes the surface curvature and cusps, (c) texture, and (d) phonological invariants in speech.

On the other hand, integration of these functions into a high-level cognitive system is an objective which is orders of magnitude more elaborate than generally believed. There thus seems to exist plenty of potential applications but also unsolved problems. It does not seem reasonable to continue the development of sophisticated heuristic methods: it is estimated that over 30,000 papers and a couple hundred textbooks have already been written on technical pattern recognition and computer vision during the past 30 years (cf. Young & Calvert, 1974), and if the solutions would have been achievable in that way, they were already visible.

4.2. Knowledge Data Bases for Stochastic Information

A large memory capacity, and readiness to recall relevant items from it are often held as signs of intelligence. A mental storage for a large number of relations is similarly imagined to form the knowledge base for thinking and problem solving. The notion that human memory operates according to associative principles is in fact very old: Aristotle published some theoretical treatises on memory and reminiscence where these laws were qualitatively expounded.

Although it may be clear that most objects of the exterior world are distinct and discrete, and their occurrences can be described by logic relations, the primary information obtained from them in the form of sensory signals is stochastic, fuzzy, and very seldom expressible in terms of distinctive features. If the description of the exterior world can be made *verbally*, these relations, of course, can readily be expressed in a concise and discrete way, but such an operation is not possible without *a thinking and understanding subject*. For those who are used to defining problems by formal logic or other discrete formalisms like Artificial Intelligence techniques, it seems difficult to realize how much is actually required from an artificial system before it is able to automatically form concepts, distinct attributes, and other discrete abstract representations from the fuzzy, nondistinctive, and stochastic sensory signals. However, this phase is indispensable before one can talk of *genuine* "neural networks" or "neural computers."

If, nonetheless, one could *tentatively* assume that the preanalysis and selection of semantic items and their relations had already been made, their storage is then most effectively made in *spatially separate memory locations*. (A node in a semantic network also corresponds to a location.) To find an item quickly on the basis of its content, there exist classical solutions, based on both software (*hash-coding*) and hardware (*content-addressable memory, CAM*) which have been known since 1955 (for a review, see Kohonen, 1987).

Let us consider the logic of searching in somewhat more detail. *Semantic information* usually consists of data items and their links (relations, "associations") to subsets of other items. The *knowledge* acquired in such a data base can be managed through long chains of such links, which are realized when the associations are made to overlap. It is perhaps illustrative to compare a data base and the searching process with a system of mathematical equations and their solution. When we present a query, we in fact set up one or more "equations" which contain unknown variables: For instance, we may specify a number of partial relations in which

some of the members are unknown, and the system has to find from memory all the relations that match with the "equations" in their specified parts, whereby a number of solutions for the unknown variables become known.

The above discussed the so-called *relational data bases* which are widely used for business data. Searching of information from them calls for the following elementary operations: (a) parallel or otherwise very fast matching of a number of search arguments (such as known members in the above relations) with all the elementary items stored in memory, and their provisional marking, and (b) analysis of the markings and sequential readout of the results that satisfy all conditions.

In order to find a solution to a searching task which is defined in terms of several simultaneous incomplete queries, as the case in formal problem solving tasks usually is, it is thus not sufficient to implement a content-addressable (or autoassociative) memory, but the partial searching results must somehow be buffered and studied sequentially. This, of course, is completely expedient for a digital computer, which can store the candidates as lists and study them by a program code; in a neural network, however, tasks like holding a number of candidates in temporary storage, and investigation of their "markings" would be very cumbersome.

In artificial neural networks, the searching arguments are usually imposed as initial conditions to the network, and solution for the "answers" results when the activity state of the network relaxes to some kind of "energetic minimum." One has to note the following facts that are characteristic of these devices: (a) Their network elements are *analog devices*, whereby representation of numerical variables, and their matching can only be defined with relatively low accuracy. This, however, may be sufficient for prescreening purposes which is most time-consuming; (b) A vast number of relations in memory which only approximately match with the search argument can be activated. On the other hand, since the "conflicts" then cannot be totally resolved but only minimized, the state of the neural network to which it converges in the process represents some kind of *optimal* answer (usually, however, only in the sense of Euclidean metric); and (c) The "answer," or the asymptotic state which represents the searching result has *no alternatives*. Accordingly, it is not possible, except in some rather weird constructs, to find the complete set of solutions, or even a number of the best candidates for them. It is not sure that the system will converge to the *global* optimum; it is more usual that the answer corresponds to one of the local optima which, however, may be an acceptable solution in practice.

4.3. Optimization Computations

In general, the objective in optimization is to allocate a limited amount of resources to a set of certain partial tasks such that some *objective* or *cost function* is minimized (or maximized). A great number of variables usually enter the problem, and to evaluate and minimize the objective function, a combinatorial problem has to be solved. In large-scale problems such as optimization of economic or business operations, the systems of equations are usually static, although nonlinear, and if conventional computers are used, the solutions must be found in a great many iterative steps.

Another category of complex optimization tasks is met in systems and control problems which deal with physical variables and space- and time-continuous processes. Their interrelations (the restricting conditions) are usually expressed as systems of partial differential equations, whereas the objective function is usually an integral-type functional. Mathematically these problems often call for methods of variational calculus. Although the number of variables then may be orders of magnitude smaller than in the first category of problems, exact mathematical treatment of the functionals again creates the need of rather large computing power.

It may come as a surprise that "massively parallel" computers for both of the above categories of problems existed in the 1950s. The *differential analyzers*, based on either analog or digital computing principles and components, were set up as direct analogies for the systems to be studied whereby plenty of interconnections (feedbacks) were involved. For details of these systems and the many problems already solved by them, see Korn and Korn (1964), Aoki (1967), and Tsypkin (1968).

It may then also be obvious that if the "massively parallel computers" such as the "neural networks" are intended to solve optimization problems, they must, in principle at least, operate as analog devices; the dynamics of their processing elements must be definable with sufficient accuracy and individually for each element, and the interconnectivities must be specifically configurable.

4.4. Robot Control

There are two main categories of robots: *trajectory-programmed* ones, and so-called *intelligent robots*. To program the former, a human operator first controls their movements and actions in the desired way, whereby the sequences of coordinates and commands are stored in memory. During subsequent use, identical trajectories and commands are defined by the memorized information. The "intelligent" robots are supposed to plan their own actions; typical applications for them are assembly tasks whereby the components have to be located and fetched from random places, or the robots may be moving freely in the natural environment, at the same time performing non-programmed tasks.

The "intelligence" exhibited by the robots has so far been implemented by AI programs, which means

that the strategies have to be invented and programmed heuristically by a human being. It is often desirable to have a higher degree of learning in such robots, which then calls for "neural computers." For instance, learning of locomotion in an unknown environment is a task which hardly can be formalized by logic programming, and coordination of complex sensory functions with the motor ones cannot be solved in analytical form. Some computer simulations have already been performed which have demonstrated such autonomous learning capabilities (e.g., Barto, Sutton, & Anderson, 1983).

4.5. Decision Making

A more abstract and complex version of behavior which nonetheless belongs to the same category as the robot operation is the non-rule-based decision making, eventually connected with playing games. In the conventional AI implementations, the conditions and actions entering the problem are described as a decision tree, the evaluation of which is a combinatorial problem, and for the solution of which the branches have to be studied up to a certain depth. This, however, is not exactly the way in which a natural object thinks. He may make similar formal analyses in order to avoid bad decisions, but when it comes to the final strategy, then other reasons, based on hunches and intuitive insight into the situation become more important. These capabilities, however, may also result in sufficiently large artificial learning systems which operate according to "neural computing" principles. The performance criterion thereby applied is more complex, although implicit, and it will be learned automatically from examples as some kind of high-order statistical description. It may then be said that such strategies are also stored in the form of rules, whereas these rules are established automatically, and they only exist in implicit form, as the collective states of the adaptive interconnections.

5. THE FORMAL NEURON

The logic circuits are building blocks of digital computers. For historical reasons, because the first computers were conceptualized simultaneously with the first steps made in neural computing, it is still often believed that the operation of neural networks could also be described in terms of *threshold-logic units.* Assume a processing element depicted in Figure 1 which is then imagined to describe a biological neuron.

Each of the continuous-valued input signals $\xi_j, j = 1, 2, \ldots, n$ shall represent the electrical *activity* on the corresponding input line, or alternatively, the momentary frequency of neural impulses delivered by another neuron to this input. In the simplest formal model, the output frequency η is often approximated by a function

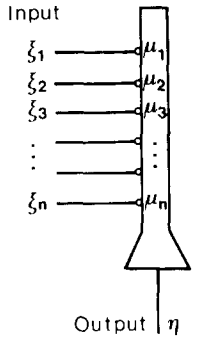

FIGURE 1. The formal neuron.

$$\eta = \text{const. } \mathbf{1}(\sum_{j=1}^{n} \mu_j \xi_j - \theta) \tag{1}$$

where $\mathbf{1}(\cdot)$ is the *Heaviside function* (=1 for positive argument, =0 for negative and zero argument), the μ_j are called *synaptic efficacies* or *weights,* and θ is a *threshold.* Thus η is here restricted to binary values (zero and high activity, respectively). If the ξ_j similarly assume only binary values, then, after scaling all signals to $\{0, 1\}$, it can be shown that an arbitrary Boolean function is implementable by a suitable selection of (positive and negative) weights μ_j and the threshold.

In reality, however, the neural cell acts as some kind of nonlinear "leaky integrator" of input. If all the biophysical and chemical phenomena taking place at the cell membrane were taken into account, we would have to describe the triggering cycle of the neuron using a couple of dozen state variables. For the basic processing element in neural computing, however, we need a much simpler, although realistic mathematical operator. This author has recently pointed out (Kohonen, 1988) that a rather influential modeling law for a formal neuron is the following which qualitatively complies with the real biophysical process. Let the output η be a continuous-valued nonnegative activity variable. In accordance with the operation of the biological neuron, η is described by a dynamic system equation which is written

$$d\eta/dt = I - \gamma(\eta). \tag{2}$$

Here I is the integrated effect of all input currents to the cell membrane, and thus triggering frequency, whereas $\gamma(\eta)$ is a nonlinear loss term which opposes this effect.

It is still unclear how accurately the input term I can be approximated by a linear function of the input signals; there is some evidence for nonlinear interactions between the synapses (cf. Rall & Segev, in press; Shepherd, in press). For artificial neural networks, however, we are free to assume that I can be expressed like in equation (1),

$$I = \sum_{j=1}^{n} \mu_j \xi_j. \tag{3}$$

If now the input signals are held steady or are changing slowly, η approaches the asymptotic equilibrium whereby $d\eta/dt = 0$, and then η can be solved from equations (2) and (3) yielding

$$\eta = \gamma^{-1}(I), \tag{4}$$

where γ^{-1} is the inverse function of γ. In reality, saturation effects are always encountered at high activity which means that the loss term $\gamma(\eta)$ must be a progressively increasing function of activity η. On the other hand, remembering that η cannot become negative, it is possible to deduce that $\eta = \eta(I)$ then indeed coarsely resembles the Heaviside function.

6. ADAPTIVE FORMATION OF FEATURE-SENSITIVE UNITS

Although the threshold-logic operations would already define many information processing functions, the discussion becomes still more interesting if the μ_j are time-variable. In most neural models, the assumption is made that *the synaptic efficacies, or the weights μ_j change in proportion to the product of input and output signals;* this is the famous hypothesis of Hebb (1949) dressed in analytical form. However, since all activities must be nonnegative, the changes would always be unidirectional, which is not possible in the long run. Therefore, some kind of *forgetting* or other similar effect must be introduced which makes the changes reversible. This author is of the opinion that the "forgetting" effect must be proportional to activity η, that is, that the forgetting is "active," which is a biologically motivated assumption. Then the *law of adaptation* would read

$$d\mu_j/dt = \alpha\eta\xi_j - \beta(\eta)\cdot\mu_j \tag{5}$$

where α is a "plasticity parameter," and $\beta(\eta)$ is some positive function of η. In order to implement the interesting phenomena discussed later, it seems necessary that in the Taylor expansion of $\beta(\eta)$ the constant term is zero. This assumption sounds natural, too, because at zero activity, no changes may be made in the cell structure. Then, for a rather general selection of the functional law, the following rule seems to be valid (cf. Sec. 4.3 of Kohonen, 1988). Let us first denote $\mathbf{x} = [\xi_1, \xi_2, \ldots, \xi_n]^T$ and $\mathbf{m} = [\mu_1, \mu_2, \ldots, \mu_n]^T$ where we have used matrix notations, and T means the transpose. Let $\mathbf{C}_{xx}$ be the correlation matrix of $\mathbf{x}$.

Proposition: If the ξ_j are stochastic variables with stationary statistical properties, then the μ_j, according to equation (5), converge to asymptotic values such that $\mathbf{m}$ then represents the largest-eigenvalue eigenvector of $\mathbf{C}_{xx}$.

In other words, the neuron then becomes most sensitive to a particular type of statistical input "feature" which describes certain fundamental properties of input $\mathbf{x}$. This result actually needs a mathematically rigorous proof, which for several case examples has been presented in Kohonen (1988).

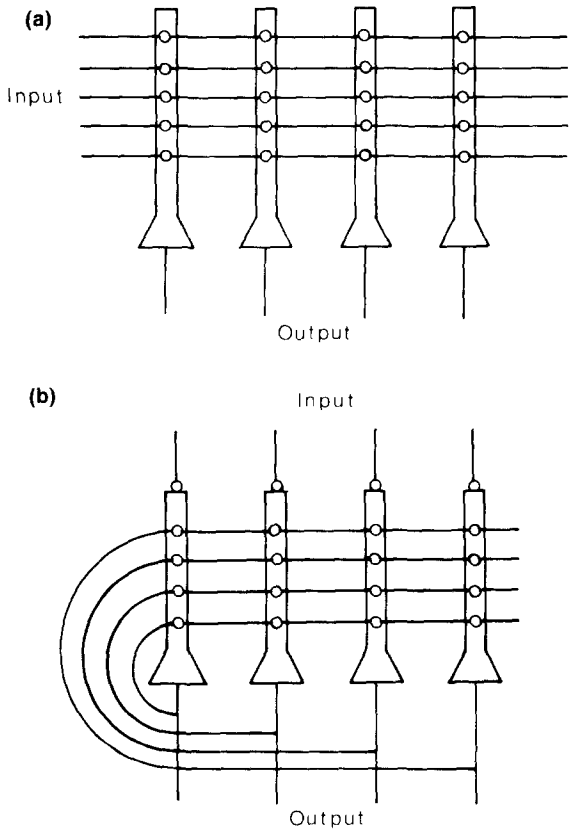

FIGURE 2. Basic network structures.

7. NETWORK STRUCTURES

The basic structure in most neural networks formed of elementary processing elements is the *crossbar switch* (Figure 2a). The next step towards more complex functions results from inclusion of *lateral feedback* (Figure 2b). This kind of connectivity has for a long time been known in neuroanatomy; we applied it to modeling in the early 1970s (cf. a rather late publication on neural associative memory Kohonen, Lehtiö, & Rovamo, 1974, and another series of publications on the "novelty filter" based on it, e.g., Kohonen, 1976).

In an attempt to devise complete *operational modules* for neural systems, it is useful to define the basic network structure containing adaptive crossbars, for input as well as for feedback (Figure 3).

The most natural topology of the network is two-dimensional (Figures 2 and 3 only show a one-dimensional section of it for simplicity), and the distribution of lateral feedbacks within this subsystem, in the first approximation, could be the same around every neuron; however, the distribution of the interconnections could strongly depend on the distance between two points in the network. Within an artificial module, all units ("neurons") could receive the same set of input

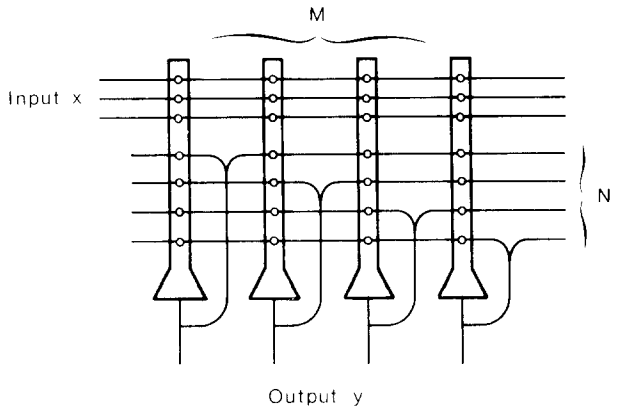

FIGURE 3. Operational module for neural systems.

signals in parallel. None of these assumptions is unnatural, even when relating to biological structures.

If more complex systems have to be defined, an arbitrary new organization can then be implemented by interconnecting such operational modules, each one perhaps with somewhat different internal structure and/or parameters. This would then correspond to the "compartmental" approximation frequently made in the description of other physiological functions.

We may now write the system equations for one module in the following general way:

$$d\mathbf{y}/dt = f(\mathbf{x}, \mathbf{y}, \mathbf{M}, \mathbf{N}), \tag{6a}$$

$$d\mathbf{M}/dt = g(\mathbf{x}, \mathbf{y}, \mathbf{M}), \tag{6b}$$

$$d\mathbf{N}/dt = h(\mathbf{y}, \mathbf{N}). \tag{6c}$$

Here $\mathbf{x}$ is the vector of all external inputs to the module, $\mathbf{y}$ the vector of all output activities, and $\mathbf{M}$ and $\mathbf{N}$ two (adaptive) connectivity matrices, respectively. In biological systems, equation (6a) might be called the *relaxation equation,* and its time constants are small, of the order of ten milliseconds; $\mathbf{y}$ describes electrical activities, and among other things involves the effect of diffusion of light ions. Equations (6b) and (6c) have much larger time constants, say, weeks, because $\mathbf{M}$ and $\mathbf{N}$ describe changes in proteins and anatomical structures. Both of these equations are *adaptation equations:* equation (6c) further seems to describe the basic function of *associative* or *content-addressable memory.*

8. PATTERN RECOGNITION BY LEARNED RESPONSES

Fundamentally, pattern recognition is a *decision process* for which the only and ultimate criterion is minimization of the average number of misclassifications. Accordingly, the theory of the *average conditional loss in decision making* which stems from the Bayes theory of probability (Kohonen, 1988) constitutes the best theoretical setting. It comes perhaps as a surprise that there already exists a neural network which has a similar performance (cf. Section 9); nonetheless we shall commence with certain other methods which enjoy a doctrinal status nowadays. For certain tasks such as adaptive control, they may even be optimal.

In a simple approach to pattern recognition, the neural network operates as a "black box" which receives a set of observation signals constituting the input vector $\mathbf{x}$, and produces a response η_i on one of its output ports i, each port being assigned to a different class of observed items. A straightforward idea followed in neural network theory has been to stipulate that the output must be $\eta_i = 1$ if $\mathbf{x}$ belongs to class i, and $\eta_j = 0$ for $j \neq i$. This may work in certain very nonlinear discriminator systems which are designed by heuristic methods. A physical network, however, usually has a continuous transfer function such that

$$\eta_i = f_i(\mathbf{x}, \mathbf{m}) \tag{7}$$

where $\mathbf{m}$ is the vector of *internal parameters* of the network. If the system is stimulated by input $\mathbf{x}$, it is then thought that $\mathbf{x}$ belongs to class i if for all $j \neq i$, $\eta_i > \eta_j$. The theoretical problem is to devise an adaptive process such that given a set of pairs of input and output $(\mathbf{x}^{(k)}, \eta_i^{(k)})$ one finally hopes to have

$$\forall(i, k),\ \eta_i^{(k)} = f_i(\mathbf{x}^{(k)}, \mathbf{m}). \tag{8}$$

In all practical applications the statistical density functions of the different classes overlap whereby only optimal approximate solutions to equation (8) can be found. In a traditional statistical approach one defines the least-square functional J,

$$J = \sum_i E\{(\eta_i - f_i(\mathbf{x}, \mathbf{m}))^2\} \tag{9}$$

where $E\{\cdot\}$ is the mathematical expectation. For the determination of the parameter vector $\mathbf{m}$ one must then minimize J requiring that

$$\text{grad}_{\mathbf{m}}(J) = 0, \tag{10}$$

where the left side means gradient of J in the parameter space. This is the general formulation of the problem; it is then even not necessary that the neural network consists of a single layer of processing elements, or is layered at all. The network may even contain internal, delayed feedback paths, in which case it may be assumed that the $\eta_i = f_i(\mathbf{x}, \mathbf{m})$ represent asymptotic equilibrium values for a steady input $\mathbf{x}$, after the transient phenomena have died out, like in the feedback structures of Figures 2 and 3.

Important special cases are the Perceptron and the Adaline which have one layer of adaptive units between input and output and in which there is a separate parametric vector $\mathbf{m}_i$ associated with each output. In particular, in the Adaline, there are *linear outputs* $\eta_i = \mathbf{m}_i^T\mathbf{x} - \theta_i$ which may be converted into *digital outputs* by discriminator elements. If then the input and output samples $(\mathbf{x}^{(k)}, \eta_i^{(k)})$ are applied sequentially, $k = 1, 2, \ldots$, an iterative solution of equation (10), in which all

terms in the sum over i can be minimized independently, can be dressed into the form of the familiar *Widrow-Hoff equation* (Widrow & Hoff, 1960). If, for analytical simplicity, θ_i is regarded as the $(n + 1)$th component of the $\mathbf{m}_i$ vector, whereby the $(n + 1)$th component of $\mathbf{x}$ is -1, respectively, we can write (using these so-called augmented vectors) $\eta_i = \mathbf{m}_i^T\mathbf{x}$, whereby

$$\mathbf{m}_i^{(k+1)} = \mathbf{m}_i^{(k)} + \alpha^{(k)}[\eta_i^{(k)} - \mathbf{m}_i^{(k)^T}\mathbf{x}^{(k)}]\mathbf{x}^{(k)}, \qquad (11)$$

with $\{\alpha^{(k)}\}$ another decreasing sequence of "gain parameters"; a possible choice is $\alpha^{(k)} = \text{const.}/k$. After a great many steps, the $\mathbf{m}_i^{(k)}$ then converge to the optimal values. This process is identical with the mathematical method called *stochastic approximation*, which was earlier developed by Robbins and Monro (1951), as well as Kiefer and Wolfowitz (1952) for regression problems.

For a random network configuration, it is difficult to find a similar theoretically justified optimal recursive expression; however, it is always possible to optimize the internal parameters directly from equation (10). For instance, randomly selected internal parameters (components of the $\mathbf{m}$ vector) can be adjusted iteratively in such a direction that J always decreases. A similar idea is applied in the so-called "Boltzmann machines" (Hinton & Sejnowski, 1984). This process, however, is computationally formidably slow, as can be imagined. Therefore there exists considerable interest in methods by which faster training of, say, multilevel networks becomes possible. The solution, however, usually remains suboptimal. In the principle of *backpropagation of errors*, originally invented by Werbos (1975, 1982), the differences $\eta_i^{(k)} - f_i(\mathbf{x}^{(k)}, \mathbf{m})$ are converted into hypothetical signals which are then propagated through the network in the opposite direction, thereby adjusting the internal parameters in proportion to them, like in the Widrow-Hoff scheme. The correction signals thus pass several layers, not only the last one like in the classical constructs, and all the corrections can be made in parallel by special hardware.

While it is possible to demonstrate many interesting learned stimulus-response relationships in networks of the above kinds, like conversion of pieces of text into phonetic transcriptions (which can then be synthesized into speech by standard methods) (Sejnowski & Rosenberg, 1986), one has to emphasize that in difficult pattern recognition tasks such as recognition of natural speech, none of these principles works particularly well; for instance, the recognition of phonemes from continuous speech by the optimal "learned response" method defined above has only succeeded with an accuracy of 65%, whereas the theoretically optimal Bayes classifier has an accuracy exceeding 80%. (Using auxiliary syntactic methods which take into account the coarticulation effects (Kohonen, 1986a), we have in fact transcribed continuous speech into text with an accuracy of 90 to 95%.) As stated earlier, and explained in the next section, *an accuracy comparable to that of the Bayes classifier is achievable by another simple network which is based on decision-controlled learning.* The main reason for the failure of the learned-response method in difficult recognition tasks seems to be that the "learning" criterion always tries to define a correct *amplitude* to each response, independent of the fact whether the sample is inside or at the border of the class, which is an unnecessarily strong condition, whereas this principle does not pay any attention to the exact location of the *decision surfaces* between the classes. In decision-theoretic classifiers, however, the decision surfaces (which define misclassifications) are directly optimized. Further it seems to be advantageous, as in the following method, to have several outputs for each class.

9. THE LEARNING VECTOR QUANTIZATION (LVQ)

This is a new method (Kohonen, 1986b, 1988) which, nonetheless, has already found its way to practical speech recognition (Kohonen, Torkkola, Shozakai, Kangas, & Ventä, 1987). If it is described in an algorithmic form, then it is not even necessary to define any network topology for it. A more general form of it, called "topology-preserving maps" (Kohonen, 1988, 1982a, 1982b; Kohonen, Mäkisara, & Saramäki, 1984) cannot be discussed here.

Assume a predetermined number of processing elements, each one provided with a parametric vector $\mathbf{m}_i$. Their number may be a multiple (say, ten times) of the number of classes considered. The vectors $\mathbf{m}_i$ are allocated to different classes, as explained below, and labeled correspondingly. The same input vector $\mathbf{x}$ shall then be broadcast to all elements, and it is assumed that each element is somehow able to evaluate the *similarity* of $\mathbf{x}$ and its $\mathbf{m}_i$. The most similar $\mathbf{m}_i$ then defines the classification of $\mathbf{x}$. This method, explained in more detail below, works with almost arbitrary similarity measures, some of which may be biologically more correct than the others. For simplicity, it is now assumed that the norms of the differences $\|\mathbf{x} - \mathbf{m}_i\|$ can be evaluated by the processing elements. (The "neural" justification of this approximation has been presented in Kohonen, 1988.) Further it will be necessary to assume that these results can be compared mutually. This comparison seems to be implementable by certain collective phenomena in a laterally interconnected "neural network" (Kohonen, 1988), although we shall not consider physical implementations here. Let the best-matching processing element have index c and be called the "winner."

The following simple algorithm is a supervised method, like learning in stimulus-response networks usually is. Consider a sequence of *training inputs* $\{\mathbf{x}^{(k)}\}$, *each one with a known classification.* Let the $\mathbf{m}_i$, $i = 1$,

. . . , K be *initialized* in the following way. First of all we have to know the a priori probability of occurrences of various classes among the input samples, and a corresponding fraction of the available processing elements is then allocated to each class and *labeled* in accordance with that class. For the initial values of the various $\mathbf{m}_i$, values of the first samples of $\mathbf{x}^{(k)}$ with known classification can often be taken; the processing elements are labeled correspondingly. (It is sometimes safer to initialize the $\mathbf{m}_i$ by values which are averages of their due classes, with a small random noise component superimposed on them.)

New inputs $\mathbf{x}^{(k)}$ are thereafter compared with all the $\mathbf{m}_i$ in parallel, and the "winner," indexed by c, is found at each step. The $\mathbf{m}_i$ are then updated recursively according to the following rule, whereby $\mathbf{m}_i^{(k)}$ means the value of $\mathbf{m}_i$ at step k:

$$\mathbf{m}_c^{(k+1)} = \mathbf{m}_c^{(k)} + \alpha^{(k)}(\mathbf{x}^{(k)} - \mathbf{m}_c^{(k)}),$$

if the classes of $\mathbf{x}^{(k)}$ and $\mathbf{m}_c^{(k)}$ agree

$$\mathbf{m}_c^{(k+1)} = \mathbf{m}_c^{(k)} - \alpha^{(k)}(\mathbf{x}^{(k)} - \mathbf{m}_c^{(k)}),$$

if the classes of $\mathbf{x}^{(k)}$ and $\mathbf{m}_c^{(k)}$ disagree

$$\mathbf{m}_i^{(k+1)} = \mathbf{m}_i^{(k)} \quad \text{for } i \neq c. \tag{12}$$

Here $\{\alpha^{(k)}\}$ is a similar decreasing sequence of "gain parameters" as in stochastic approximation, equation (11); however, it has turned out that over a finite training interval, $\alpha^{(k)}$ could decrease to zero linearly with k, starting with a small value, say, .1 or .2. The recommended number of training steps could be, for example, 500 to 5000 times the number of processing elements. This may sound large, but since the algorithm is computationally extremely light, the convergence is reasonably fast. Special parallel (or array processor) hardware is readily developed for this method (Kohonen *et al.*, 1987).

Naturally one may not be able to collect such a large number of samples in a real application. A smaller training set may then be recycled, preferably with random permutation of its members.

After training, the $\mathbf{m}_i$ will have acquired such values that classification by the "nearest neighbor" principle, by comparison of $\mathbf{x}$ with the $\mathbf{m}_i$, very closely complies with that of the Bayes classifier. Figure 4 represents an illustrative example whereby $\mathbf{x}$ was two-dimensional, and its probability density functions overlapped seriously. The decision surface defined by this "*learning vector quantization*" (LVQ) classifier seems to be near-optimal, although piecewise linear, and the classification accuracy even in this rather difficult example is very close the same as with the Bayes classifier, within a fraction of a percent.

10. LEARNED RESPONSES FOR MOTOR CONTROL AND OTHER ACTIONS

It may have become clear from the previous section that pattern recognition represents a typical decision process which has to be optimized according to misclassifications. A problem of completely different nature is to define proper output actions in response to a particular input, such as the motor reactions usually are; then each response is equally important, because no classes for the inputs are defined at all, and the optimization of the internal parameters by any of the methods discussed in the previous section is completely expedient; the J functional represents the only applicable criterion.

One aspect, as pointed out by Pellionisz (1986), is that for the input–output relation in systems which

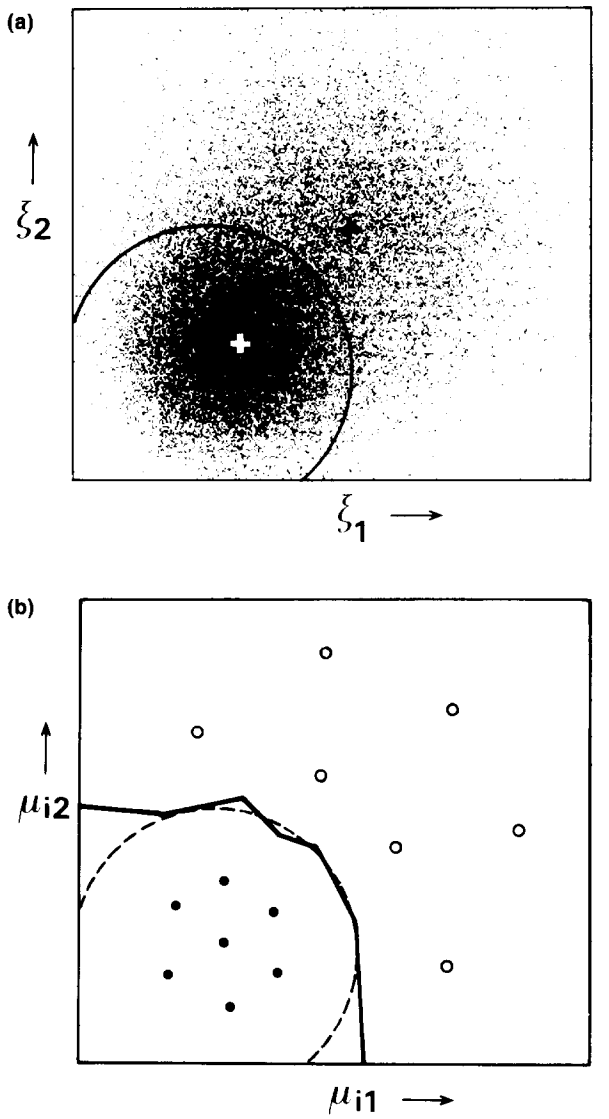

FIGURE 4. (a) The probability density function of $x = [\xi_1, \xi_2]^T$ is represented here by its samples, the small dots. The superposition of two symmetric Gaussian density functions corresponding to two different classes C_1 and C_2, with their centroids shown by the white and the dark cross, respectively, is shown. Solid curve: the theoretically optimal Bayes decision surface. (b) Large black dots: reference vectors of class C_1. Open circles: reference vectors of class C_2. Solid curve: decision surface in the learning vector quantization. Broken curve: Bayes decision surface.

contain interdependent mechanisms, transformation of signals in curvilinear coordinates must be considered carefully.

11. THE LATERALLY INTERCONNECTED NETWORKS AS MEMORIES

Certain widely known neural network structures whose processing elements are even simpler than those discussed in Sections 5 and 6 may only briefly be delineated here. Consider, for instance, the network of Figure 3 whereby we first neglect the input crossbar, and have a unit delay in the feedback. Anderson, Silverstein, Ritz, and Jones (1977) have studied system equations of the type

$$\mathbf{y}^{(t+1)} = S[\mathbf{y}^{(t)} + \mathbf{N}\mathbf{y}^{(t)}], \tag{13}$$

where $\mathbf{N}$ is the feedback matrix (cf. Figure 3), $t = 0, 1, 2 \ldots$, the notation $S[\cdot]$ means a "linearized sigmoidal function" on all vector components, that is, for a scalar x, $S(x) = x$ for $|x| \leq F$, $S(x) = +F$ for $x \geq F$, and $S(x) = -F$ for $x \leq -F$, and F is thereby the saturation limit. Anderson *et al.* have shown that if $\mathbf{N}$ can somehow be defined as the correlation matrix $\mathbf{C}_{xx}$ of input $\mathbf{x}$, then $\mathbf{y}$ will converge to "eigenvectors," which often closely approximate the eigenvectors of $\mathbf{C}_{xx}$, at least correlating with the large and small components of it. In particular, if $\mathbf{C}_{xx}$ is formed of vectors $\mathbf{x}^{(k)}$ which are fewer in number than the rank of $\mathbf{C}_{xx}$, then, starting with the initial state of $\mathbf{y}^{(0)}$ which is an incomplete version of a binary $\mathbf{x}^{(k)}$, the output state $\mathbf{y}$ will converge to the latter. This is in effect an *autoassociative memory* operation. Hopfield (1982) has later demonstrated a similar result.

Another interesting property of the feedback network has been pointed out by the present author (Kohonen, 1988, 1974). Assume for simplicity that the $\mathbf{M}$ matrix of Figure 3 is an identity matrix, that is, the inputs are like in Figure 2 b, and a sequence of inputs $\{\mathbf{x}^{(k)}\}$ is applied such that each input vector $\mathbf{x}^{(k)}$ is held steady, until it is switched into a new one. Assume that equation (6c) can be written

$$d\mathbf{N}/dt = -\alpha\mathbf{y}\mathbf{y}^T, \quad \text{where}$$
$$\mathbf{y} = \mathbf{x} + \mathbf{N}\mathbf{y}, \tag{14}$$

and one should notice the minus sign in front of the right side of the upper equation. Then, with continued, iterative application of the sequence $\{\mathbf{x}^{(k)}\}$, the system is relaxed to an asymptotic state such that the overall input–output transfer operator of the network converges to an *orthogonal projection operator*. In other words, if later an arbitrary input vector $\mathbf{x}$ is applied, then the output $\mathbf{y}$ in response to $\mathbf{x}$ only contains that contribution in $\mathbf{x}$ which is "maximally new," that is, which is not expandable in terms of a linear combination of the $\mathbf{x}^{(k)}$. Then $\mathbf{y}$ is orthogonal to the space spanned by the $\mathbf{x}^{(k)}$. This output $\mathbf{y}$ is called the "novelty component," and in general it suppresses nonimportant features of the input, and enhances the relevant ones. This "novelty filter" can also act as an autoassociative memory; if the input $\mathbf{x}$ is an incomplete version of a memorized pattern $\mathbf{x}^{(k)}$, the latter, superposed on $\mathbf{x}$, appears at the output as the negative $-\mathbf{x}^{(k)}$.

Notice that the adaptive "novelty filter" has here been described as a genuine continuous-time adaptive physical system, not involving any precomputed memories like in many other "neural" models. On the other hand, a word of caution may be necessary: the elements of the $\mathbf{N}$ matrix may become very large (in principle, infinite). Therefore, a forgetting effect like in equation (5), or a "sigmoidal function" like in equation (13) must be employed. The transfer operator of the system is then no longer an ideal orthogonal projection operator, although with high dimensionality of the variables, it may be a rather good approximation of it.

12. IS THERE ANY PLACE FOR MENTAL IMAGES IN NEURAL COMPUTING?

There is nothing as primary and concrete in the psychic experiences as mental images. They occur as operands in thinking, and it is apparent that they are caused by memory effects, although their stereotype forms have resulted from dropping unessential details, and averaging over the more common characteristics. However, it would not be sufficient to say that the mental images are structured recollections from an (auto)associative memory. The main problem is that there exists no "projection screen" in the brain; there is no place where, say, recollections of retinal projection images are reconstructed. It has sometimes been said that the brain is "blind," it has no secondary "mental eye." (Notice that if it had it, we would also need a "mental mind," "mental mind's eye," and so on to infinity.) This paradox is familiar, but it has usually been dodged in scientific discussion. It would neither be correct to identify the mental images or mental items with *logic concepts*. The latter are actually only defined as the set of their characteristics or attributes, and the logic concepts, to be exact, were originally introduced only to define *languages*. The mental images, on the other hand, are direct copies, although possibly simplified and statistically averaged ones, of real experiences; in philosophy they are called *psychic concepts* for distinction.

Most "connectionist" neural networks, especially those intended for the handling of linguistic or other abstract information structures, define a concept either as a hypothetical singular representation (node or point), or as a set of attributes. A neural computer which deals with such discrete variables is then not dissimilar from any digital computer in principle. In the field of Artificial Intelligence it is hardly ever considered that our thinking in fact proceeds in a totally different way than logic reasoning. Therefore, if we talk

of the theory of "neural networks," at least then the essence of the mental items should somehow be made clear. It seems that the only reasonable way to explain them is to resort to a concept which is called *virtual image* in physics. Consider a mirror or lens, or perhaps a hologram which transmits optic wavefronts to the eye. The virtual image is that imaginary source of wavefronts which is sensed real, although it does not exist at all. This concept is now generalized.

For the present discussion, it is even not necessary to know how the neural signals are interpreted mentally; for instance, nobody thinks of such a thing when discussing optical imaging. On the other hand, in the reductionistic view (sometimes also called the psychophysical parallelism theory of Spinoza or W. James) it is assumed that a certain perception corresponds to a certain neurophysiological state. Re-establishing the latter would then evoke the former. Notice that the optical virtual images are in fact explained by the same principle: for the creation of an illusion, it will suffice to reconstruct the optic wavefronts, which are only approximate replicas of the original ones.

Consider now Figure 5 which depicts a network, consisting of two divisions, A and B. It is not necessary that one of the networks belongs to a lower hierarchical level and the other to the higher one; the division line can be abstract, imaginary, and different from case to case. The two divisions can even be intermingled. What is necessary to assume is that in this particular case, the result of perception is mainly formed by components of subsystem B, based on its input signal vector $\mathbf{y} = f(\mathbf{x}, \mathbf{M})$ where $\mathbf{x}$ is the signal vector which stimulates system A, and $\mathbf{M}$ is a parametric state of system A (equivalent to its transmittance). System B cannot know what there is inside A; so it assumes that A is always the same. Let the *unperturbed* state of system A ("empty memory") be denoted by $\mathbf{M}_0$. Then, if the signals transmitted by system A leave "memory traces" in it, the *perturbed* system state is denoted by $\mathbf{M} = \mathbf{M}_0 + \Delta\mathbf{M}$. In Figure 5b the same signal vector $\mathbf{y}$ is also assumed to be explainable by an "*empty memory*" and *effective input excitation* which contains an extra component, the *virtual image* $\Delta\mathbf{x}$. In other words, $\mathbf{y}$ shall be expressible in two alternative ways:

$$\mathbf{y} = f(\mathbf{x}, \mathbf{M}_0 + \Delta\mathbf{M}) = f(\mathbf{x} + \Delta\mathbf{x}, \mathbf{M}_0). \qquad (15)$$

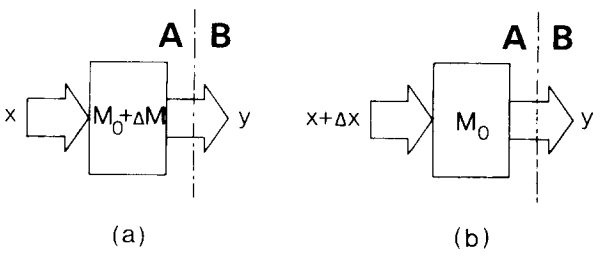

FIGURE 5. Illustration of the principle of virtual images.

If the law by which $\Delta\mathbf{M}$ is formed has certain properties (like in holography, or if the adaptive changes depend on signals as discussed in Section 6), and the transfer function f has a simple analytical form, then $\Delta\mathbf{x}$ can be solved (accurately or approximately, i.e., in the sense of least squares). Computer simulations of this principle in neural networks with partly random structure have been reported in Kohonen (1971) and Kohonen (1973).

Explanations for many intriguing phenomena discussed in experimental psychology become now readily available: for example, hallucinations, so-called phantom effects in amputated limbs, and geometric illusions. Without the principle of virtual images, on the other hand, the neural networks or computers are only "behavioristic" machines, and at least it can be said that they then do not reflect the most important phenomenological features of thinking.

13. VISTAS TO NEUROCOMPUTER SYSTEMS

More complicated architectures for neurocomputers can be implemented by interconnecting several modules of the above types. For instance, Grossberg (1982) and Carpenter and Grossberg (in press) have suggested circuits for the interaction of different subsystems in the brain.

In engineering systems, the neurocomputers ought to be understood as special-purpose models or coprocessors which are operating under the control of a more conventional host computer. The latter defines a program code on the application level and schedules the application operations, while the "neurocomputer" implements various bulk tasks, such as preprocessing natural input information like images or speech, or acts as a "neural expert system," answering queries in a statistically optimal fashion.

A more complete view to neurocomputer architectures can be obtained from the books edited by McClelland and Rumelhard (1986), from the SPIE (The Society of Photo-Optical Instrumentation Engineers) Advanced Institutes proceedings (1987), as well as special issues published by Applied Optics (1986, 1987).

REFERENCES

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, **84**, 413–451.

Aoki, M. (1967). *Optimization of stochastic systems—Topics in discrete-time systems*. New York: Academic Press.

Applied Optics (1986, September 15). 25(18).

Applied Optics (1987, May 15). 26(10).

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *Institute of Electrical and Electronics Engineers Transactions*, **SMC-13**, 834–846.

Caianiello, E. R. (1961). Outline of a theory of thought-processes and thinking machines. *Journal of Theoretical Biology,* **2,** 204–235.

Carpenter, G. A., & Grossberg, S. (in press). A massively parallel architecture for a self-organizing neural pattern recognition machine. In *Computer vision, graphics, and image processing.*

Farley, B. G., & Clark, W. A. (1954). Simulation of self-organizing systems by digital computer. *Institute of Radio Engineers—Transactions of Professional Group of Information Theory,* **PGIT-4,** 76–84.

Grossberg, S. (1982). *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control.* Amsterdam: Reidel Press.

Hebb, D. (1949). *Organization of behavior.* New York: Wiley.

Hinton, G. E., Sejnowski, T. J., & Ackley, D. H. (1984). *Boltzmann machines: Constraint satisfaction networks that learn.* (Tech. Rep. No. CMU-CS-84-119). Pittsburgh, PA: Carnegie Mellon University.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America, **79,** 2554–2558.

Hopfield, J. J., & Tank, D. W. (1986). Computing with neural circuits: A model. *Science.* **233,** 625–633.

Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics,* **23,** 462–466.

Kohonen, T. (1971). Introduction of the principle of virtual images in associative memories. *Acta Polytechnica Scandinavica,* **El. 29.**

Kohonen, T. (1973). A new model for randomly organized associative memory. *International Journal of Neuroscience,* **5,** 27–29.

Kohonen, T. (1982a) A simple paradigm for the self-organized formation of structured feature maps. In S. Amari & M. A. Arbib (Eds.), *Competition and cooperation in neural nets,* Lecture Notes in Biomathematics (Vol. 45, pp. 248–266). Berlin: Springer-Verlag.

Kohonen, T. (1982b). Clustering, taxonomy, and topological maps of patterns. In *Proceedings of the Sixth International Conference on Pattern Recognition* (pp. 114–128). Silver Spring, MD: IEEE Computer Society.

Kohonen, T. (1986a). Dynamically expanding context, with application to the correction of symbol strings in the recognition of continuous speech. In *Proceedings of the Eighth International Conference on Pattern Recognition* (pp. 1148–1151). Washington, DC: IEEE Computer Society.

Kohonen, T. (1986b). *Learning vector quantization for pattern recognition.* (Tech. Rep. No. TKK-F-A601). Finland: Helsinki University of Technology.

Kohonen, T. (1987a). *Content-addressable memories* (2nd ed.). Berlin: Springer-Verlag.

Kohonen, T. (1988). *Self-organization and associative memory* (2nd ed.). Berlin: Springer-Verlag.

Kohonen, T., Lehtiö, P., & Rovamo, J. (1974). *Annales Academiae Scientarum, Fennicae, Series A. V Medica,* **167.**

Kohonen, T., Mäkisara, K., & Saramäki, T. (1984). Phonotopic maps—Insightful representation of phonological features for speech recognition. In *Proceedings of the Seventh International Conference on Pattern Recognition* (pp. 182–185). Silver Spring, MD: IEEE Computer Society.

Kohonen, T., & Oja, E. (1976). Fast adaptive formation of orthogonalizing filters and associative memory in recurrent networks of neuron-like elements. *Biological Cybernetics,* **21,** 85–95.

Kohonen, T., Torkkola, K., Shozakai, M., Kangas, J., & Ventä, O. (1987). Microprocessor implementation of a large vocabulary speech recognizer and phonetic typewriter for Finnish and Japanese. In J. A. Laver & M. A. Jack (Eds.), *Proceedings of European Conference on Speech Technology* (pp. 377–380). Edinburgh: CEP Consultants Ltd.

Korn, G. A., & Korn, T. M. (1964). *Electronic analog and hybrid computers.* New York: McGraw-Hill.

McClelland, J. L., Rumelhard, D. E., & PDP Research Group. (1986). *Parallel distributed processing.* Cambridge, MA: MIT Press.

McCulloch, W. S., & Pitts, W. A. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematics and Biophysics,* **5,** 115–133.

Nemes, T. N. (1969). *Kibernetikai Gepek (Cybernetic Machines).* Budapest: Akademiai Kiado.

Optical and Hybrid Computing. (1987). H. H. Szu, R. F. Potter (Eds.), *SPIE* (Vol. 634). Washington: SPIE.

Pellionisz, A. J. (1986). Tensor network theory of the central nervous system and sensorimotor modeling. In G. Palm & A. Aertsen (Eds.), *Brain theory* (pp. 121–145). Berlin: Springer-Verlag.

Rall, W., & Segev, I. (in press). Excitable dendritic spine clusters: Nonlinear synaptic processing. In R. Cotterill (Ed.), *Proceedings of Conference on Computer Simulation in Brain Science.* Cambridge University Press.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics,* **22,** 400–407.

Rosenblatt, F. (1958). The perceptron: A probabilisitic model for information storage and organization in the brain. *Psychoanalytic Review,* **65,** 386–408.

Sejnowski, T. J., & Rosenberg, C. R. (1986). *NETtalk: A parallel network that learns to read aloud.* (Tech. Rep. No. JHU/EECS-86/01). Baltimore, MD: Johns Hopkins University.

Shepherd, G. M. (in press). The significance of axon collaterals and distal dendrites in brain circuits. In R. Cotterill (Ed.), *Proceedings of Conference on Computer Simulation in Brain Science.* Cambridge University Press.

Steinbuch, K. (1961). Die Lernmatrix. *Kybernetik,* **1,** 36–45.

Tsypkin, Y. A. (1968). *Adaptation and learning in cybernetic systems.* Moscow: Nauka.

Werbos, P. (1975). *Beyond regression: New tools for prediction and analysis in behavioral sciences.* (Doctoral thesis and published report). Cambridge, MA: Harvard University.

Werbos, P. (1982). Applications of advances in nonlinear sensitivity analysis. In R. Drenick & F. Kozin (Eds.), *Systems modeling and optimization: Proceedings of the International Federation for Information Processing.* (pp. 762–770). New York: Springer-Verlag.

Widrow, B., & Hoff, M. E. (1960). Adaptive Switching Circuits. 1960 WESCON Convention, Record Part IV, pp. 96–104.

Young, T. Y., & Calvert, T. W. (1974). *Classification, estimation, and pattern recognition.* New York: Elsevier.