General Introduction
James A. Anderson

Brain-Like Machines

Neural networks, connectionist models, or, using a more recent name, neuromorphic systems are systems that are deliberately constructed to make use of some of the organizational principles that are felt to be used in the human brain. This volume contains a number of the original papers that describe many of the important ideas and techniques that are used in these models. Although there are a great many variations between authors and systems, there are also great similarities, both in the problems they try to solve with networks and in the techniques they use. At the present time, in the infancy of the field, similiarities predominate; as evolution continues, more variety will appear.

Many, if not most, of the ideas contained here were originally proposed to explain observations in neurobiology or psychology. Understanding human behavior and brain construction are perennially interesting and important questions, and the purely scientific desire to understand one of the most complex systems in nature is still the main motivation for many of the people who work with neural networks.

However, if you really understand something, you can usually make a machine do it. If you want to make machines think, act, or move like humans, a good initial strategy for most scientists is to study how humans think, act, or move. This strategy is not a guaranteed winner: extensive studies of human limb motion would probably not have suggested the wheel as the most appropriate way to move across smooth surfaces.

Humans have always wanted to build intelligent machines. Stories and legends of automata, robots, and mechanical men have been common for thousands of years, and are even more so today when we are actually building them. Current computer technology potentially allows us to build systems of a complexity that approaches the number of elements and interconnections of the brain. At the same time, we do not know how to organize this complexity or what functions to compute with it, even if we did manage to build it. Nor do we know if building a complex, brain-like system would yield intelligent behavior without other important features, some of which may still be unknown.

Somehow the brain is capable of taking neurons—the brain's basic computing elements—which are five or six orders of magnitude slower than silicon logic gates, and organizing them so as to perform some computations many times faster than the fastest digital computer now in existence. One way the brain seems to have managed to do this is by massive hardware parallelism: that is, the computing elements are arranged so that very many of them are working on a problem at the same time. Since there are huge numbers of neurons, somehow the weak computing powers of these

many slow elements are combined together to form a powerful resultant. The speed of neurons has not increased much in evolution, once a few tricks like myelination were developed. The hardware and software cooperate, so the way to get more power seems to be to add more neurons, a strategy highly developed in our own massive cere'bral cortex.

In the fifty-year history of digital computers, a quite different evolutionary path was followed. Conceptually, and often in reality, there is only one computing engine, the Central Processing Unit (CPU), and this unit has gotten faster and faster. At first it was made of vacuum tubes; now it is made of silicon VLSI (Very Large-Scale Integrated) circuitry; and in the future even faster compounds such as gallium arsenide will be used. Increase in computing speed has come about largely through faster hardware. Only in the last decade has good use been made of the first steps toward parallelism, where several CPUs are working at the same time on the same problem. It has turned out to be extraordinarily difficult to coordinate and program multiple CPUs of a traditional type.

Part of the reason that attempts to coordinate multiple fast CPUs have been difficult is that users and manufacturers want to run the same kinds of software, which run well on traditional computers, on parallel machines—only faster. It is unlikely that it will be possible to run traditional software on neural network machines. The appropriate software may be *very* different from traditional software.

It is worth pointing out that neural models have a narrow biological base. They are essentially all models of the newer parts of the mammalian nervous system, usually the cerebral cortex. Sometimes the earlier stages of sensory processing will be considered, but usually as a way of preprocessing information for a cortex to 'look at.' This is not necessarily bad. Because it evolved recently and is anatomically rather homogeneous, with only a few cell types and standard connection patterns to deal with, cortex may be more easily understandable than older, more highly optimized parts of the nervous system. The older parts of the vertebrate nervous system may be quite different in organization from the cortex, and invertebrates may be very different indeed.

Attempts to model cerebral cortex have an important consequence. If we model cortex, we have a good idea of its function. Most of our complex cognitive functions seem to be carried on there: speech, language, perception. Ideally, when we look at cognitive psychology or cognitive science, we are looking at software that runs on a cortical computer. This means it is possible to get reasonably good data on some of the organizational details of the output of the system by studying how humans do it.

This suggests also that a good way to find out what kind of software runs well on neural networks will be to look at what kind of software runs well on *us*. This means that cognitive science, besides its intrinsic scientific interest, might be viewed as a technique for reverse engineering the software for a parallel computer. Having an insight into writing, and running, good software for neuromorphic computers might save a great deal of unhappiness and failed expectations. As one example, expecting massively parallel neural network machines to balance a checkbook, perform logic, or keep fine detail straight might be unwise, since these are functions that are notoriously difficult for humans. Asking the same machines to make good guesses, disambiguate, resolve conflicting information, or form concepts might be reasonable.

We are so used to the constructional peculiarities of traditional computers that we have a tendency to think of them as familiar, but somewhat quirky, old friends. This familiarity often blinds experienced users to the extreme unnaturalness of the mindset required in order to use traditional computers effectively. Neural network systems might turn out to be truly 'user friendly' since they work like us!

Theoretical Themes

Several themes constantly recur in construction of neural network models: network structure, learning algorithms, and knowledge representation. Let us first sketch the generic connectionist model.

The Generic Connectionist Model

There are very many neurons, or nerve cells, in the human brain, at least ten billion. Each neuron receives inputs from other cells, integrates the inputs, and generates an output, which it then sends to other neurons, or, in some cases, to effector organs such as muscles or glands. Neurons receive inputs from other neurons by way of specialized structures called *synapses* and send outputs to other neurons by way of output lines called *axons*. A single neuron can receive on the order of hundreds or thousands of input lines and may send its output to a similar number of other neurons. A neuron is a complex electrochemical device that contains a continuous internal potential called a *membrane potential*, and, when the membrane potential exceeds a threshold, the neuron can propagate an all-or-none *action potential* for long distances down its axon to other neurons. Synapses come in a number of different forms, but two basic varieties are of particular note: *excitatory* synapses, which make it more likely that the neuron receiving them will fire action potentials, and *inhibitory* synapses, which make the neuron receiving them less likely to fire action potentials (see figure 1).

Neuroscientists usually measure the *activity* of a neuron by its firing frequency, i.e., the number of action potentials per second or something closely related to firing frequency. Biological neurons are *not* binary, that is, having only an on or off state as their output. Outputs are continuous valued and the neuron acts something like a voltage to frequency converter, converting membrane potential into firing rate (see paper 23 on the *Limulus*). Many network models use elements that are continuous valued to some extent. However, a number of neural network models assume that the basic computing elements are binary, that is, can only be on or off. The resulting binary valued systems are valuable, often give useful insights into the behavior of complex networks of whatever type, and are often easier or more convenient to analyze than systems of more complex neurons (see figure 2).

Given our degree of ignorance of nervous system function and the early stages of our modeling efforts, it would be most unreasonable to dismiss one or another assumption made by a modeler as "unbiological" until we see how the resulting system works. And of course, biological plausibility is significant only if you want to model the brain. If, instead, we desire to construct a useful device, there is no reason whatsoever to be bound by the way the brain happens to do it.

When the network is functioning, many cells can be active simultaneously. To

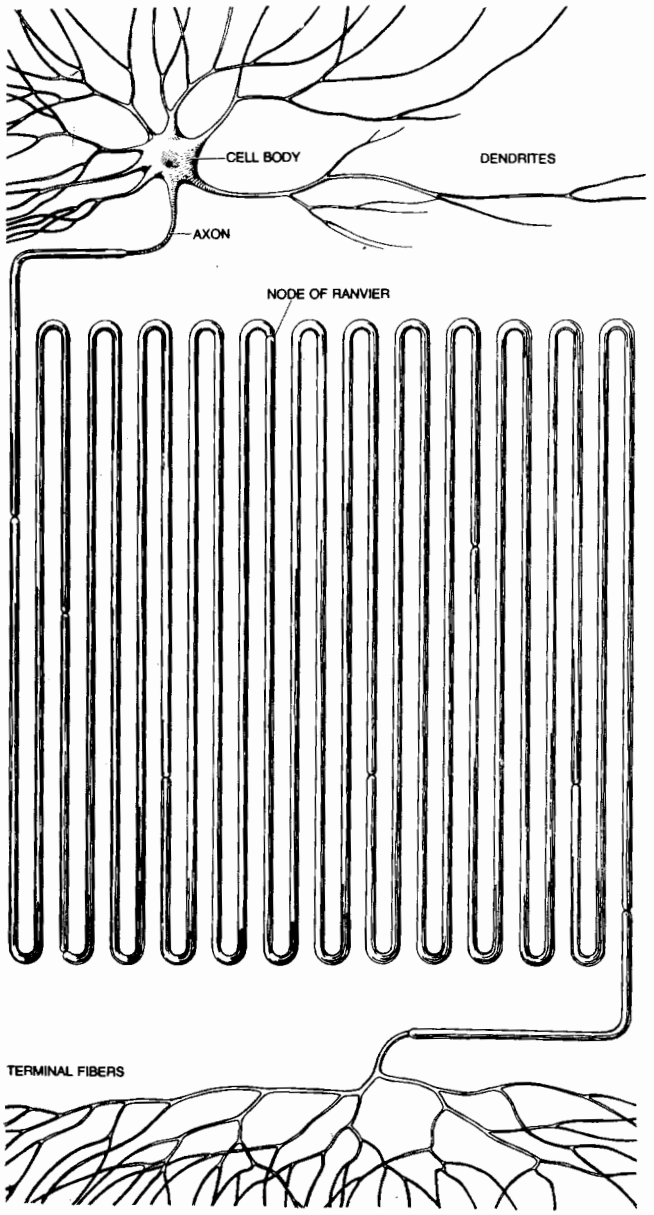
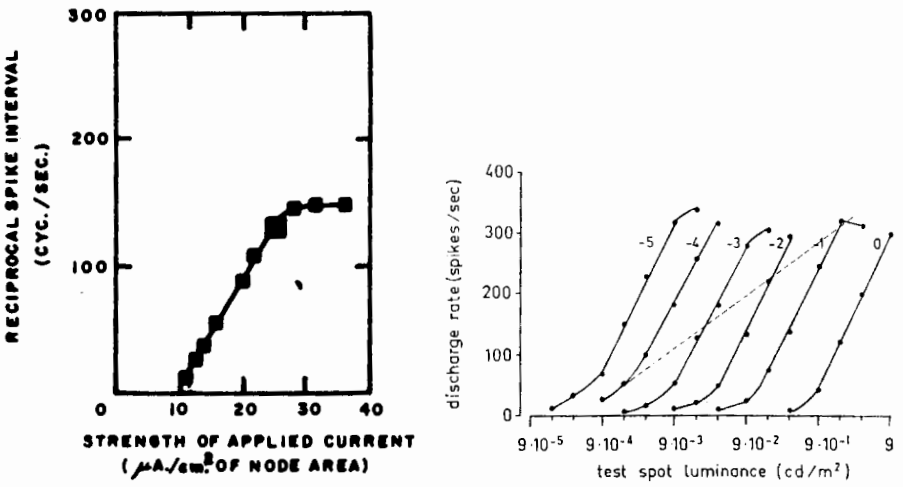


Figure 1
The typical neuron of a vertebrate animal can carry nerve impulses for a considerable distance. The neuron depicted here, with its various parts drawn to scale, is enlarged 250 times. The nerve impulses originate in the cell body, and are propagated along the axon, which may have one or more branches. This axon, which is folded for diagrammatic purposes, would be a centimeter long at actual size. Some axons are more than a meter long. The axon's terminal branches form synapses with as many as 1,000 other neurons. Most synapses join the axon terminals of one neuron with the dendrites forming a "tree" around the cell body of another neuron. Thus the dendrites surrounding the neuron in the diagram might receive incoming signals from tens, hundreds, or even thousands of other neurons. Many axons, such as this one, are insulated by a myelin sheath interrupted at intervals by the regions known as nodes of Ranvier. [Caption and figure from C. F. Stevens (1979), "The neuron," *The Brain*, Scientific American (Ed.), San Francisco: Freeman, p. 73.]



2A 2B

Figure 2
(A) Curve of the reciprocal mean spike interval in response to a maintained depolarizing current of 1-sec duration and of increasing strength for a Class I crab axon. [Adapted with permission from Reginald A. Chapman (1966), "The repetitive responses of isolated axons from the crab, *Carcinus maenas*," *Journal of Experimental Biology* 45. Copyright © 1966 by the Company of Biologists.] (B) Intensity versus response characteristics of a retinal on-center ganglion cell at different adaptation luminances: ordinate, discharge during the first 50 msec of the response; abscissa, test spot luminance (size of test spot 20-min arc). The figures on the curve are the logarithm of the adaptation luminance at which the curves were obtained. [Abridged caption and figure from O. D. Creutzfeldt, (1972), "Transfer function of the retina," *EEG Journal*, Supplement No. 31: *Recent Contributions to Neurophysiology*, J.-P. Courdeaux and P. Gloor (Eds.), Amsterdam: Elsevier.]

Overall, both these examples of real neurons display a common picture of the response of a neuron to stimulation. In one case (A) the stimulus is electrical, and in the other (B) it is light intensity. Response is taken to be firing frequency. The response displays a threshold stimulus below which there is little or no response; then, as stimulus intensity increases, there is a somewhat linear regic ›; and finally, there is a region of saturation where there is little increase in response as stimulation increases. The response to light shown in (B) indicates that the form of the response remains almost unchanged as average illumination increases, due to a variety of adaptation phenomena. Neural network modelers sometimes approximate this response function as a sigmoid or as a linear function with clipping.

describe the system at a moment in time, we have to give the activities of all the cells in the system at that time. This set of simultaneous element activities is represented by a *state vector*, corresponding to the activities of many cells.

Neural networks have lots of computing elements connected to lots of other elements. This set of connections is often arranged in a *connection matrix*. The overall behavior of the system is determined by the structure and strengths of the connections. It is possible to change the connection strengths by various learning algorithms. It is also possible to build in various kinds of dynamics into the responses of the computing elements.

Learning

Detailed computations in neural networks are largely performed by the connection strengths—hence the name 'connectionist.' There is often a decoupling between the learning phase and the retrieval phase of operation of a network. In the *learning phase*,

the connection strengths in the network are modified. Sometimes, if the constructor of the network is very clever or if the problem structure is so well defined that it allows it, it is possible to specify the connection strengths a priori. Otherwise, it is necessary to modify strengths using one of a number of useful learning algorithms, many of which are described in detail in the papers following.

There are currently a large number of learning algorithms used to set connection strengths. We shall discuss them in more detail in the introductions to the papers and in the papers themselves. Learning algorithms have been the heart of neural network research for the past three decades.

Network Operation

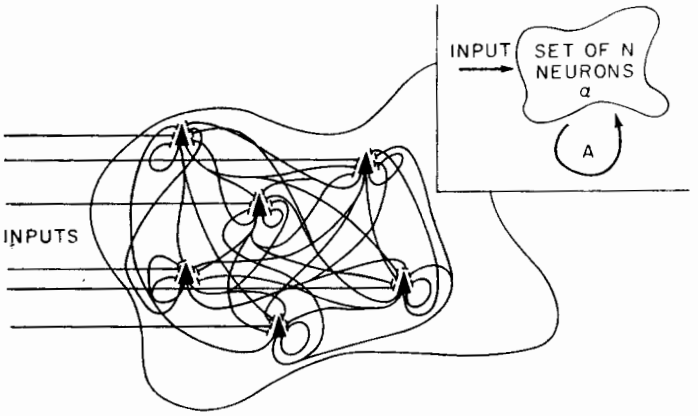
In the *retrieval phase*, some initial information, in the form of an initial state vector or activity pattern, is put into the system. In most connectionist networks, if no initial information is provided, nothing useful is retrieved, and the more information is provided, generally the more reliable is the output information. The initial input pattern passes through the connections to other elements, giving rise to an output pattern. If the system works properly, the output pattern contains the conclusions of the system. Given the complexity of the connections and the fact that many operations are going on simultaneously, it is often very hard to analyze exactly what is going on in the network. Although programs for traditional computers may be complicated, there is always faith that a program bug has a simple, usually localized, cause. This may not be true for a neural network, since both correct information and error may be spread out over many connections and many model neurons. This widespread distribution of computation also leads to a number of intrinsic error mechanisms such as interference between different events that are difficult, and probably impossible, to get rid of, given that many of the desirable features of networks (noise and damage tolerance, generalization) arise from the same cause.

Network Structure

There are a number of ways of organizing the computing elements in neural networks. Typically the elements are arranged in groups or layers. A single layer of neurons that connects to itself is referred to as an *autoassociative* system. Single-layer and two-layer systems, with only an input and an output layer, are easy to analyze and quite powerful, and are used extensively in many of the papers we present. More recently, learning algorithms have become available that can force the network to develop appropriate connection strengths in multilayer networks. There is currently a great deal of effort devoted to understanding multilayer systems. Such systems are potentially much more powerful than one- and two-layer systems, but they are also more complex and harder to analyze. For example, sometimes there are elements in the middle layers that are neither input nor output neurons, and are referred to as *hidden units* (see figure 3).

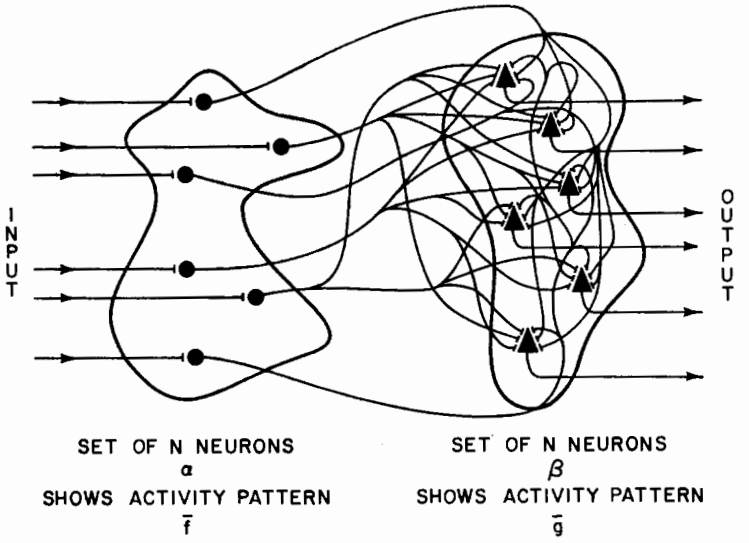
Representation

In the earliest days of neural networks, major research attention focused on the network structure, assumptions about the properties of the model neurons, and the learning algorithms used to change connection strengths.



1. SET OF N NEURONS, α
2. EVERY NEURON IN α IS CONNECTED TO EVERY OTHER NEURON IN α THROUGH LEARNING MATRIX OF SYNAPTIC CONNECTIVITIES A

3A



3B

Figure 3
(A) One-layer network. A group of neurons feeds back on itself. [Caption and figure from James A. Anderson, Jack W. Silverstein, Stephen A. Ritz, and Randall Jones, "Distinctive features, categorical perception, and probability learning: some applications of a neural model," *Psychological Review* 84, figure 3.] (B) Two-layer network with an input set and an output set. This is a strictly feedforward network, where the input set projects to the output set, but not vice versa. [Figure from James A. Anderson, Jack W. Silverstein, Stephen A. Ritz, and Randall Jones, "Distinctive features, categorical perception, and probability learning: some applications of a neural model," *Psychological Review* 84, figure 1.] (C) Three-layer network. Schematic drawing of the network architecture. Input units are shown on the bottom of the pyramid. ... Each hidden unit in the intermediate layer receives inputs from all of the input units on the bottom layer, and in turn sends its output to all 26 units in the output layer. [Abridged caption and figure from Terrence J. Sejnowski and Charles R. Rosenberg, "NETtalk: a parallel network that learns to read aloud," The Johns Hopkins University Electrical Engineering and Computer Science Technical Report JHU/EECS-86/01.]

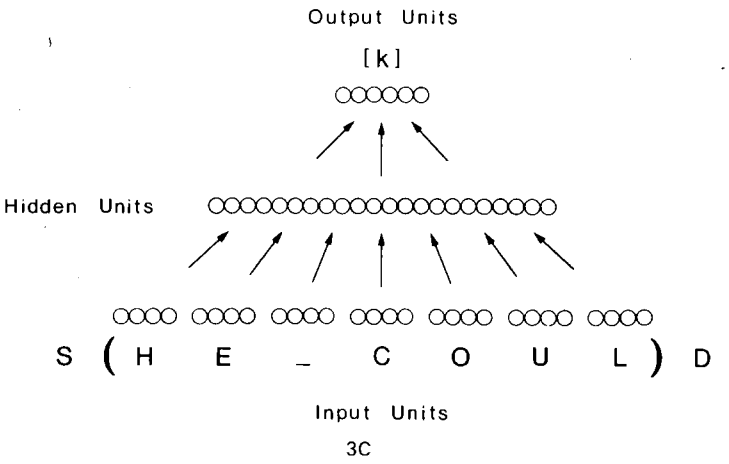


Figure 3 (continued)

Now that we have some idea of what various learning algorithms and network structures can do, there is a growing interest in the problems of representation of information in the network. The mammalian brain often follows very simple rules for representation, for example, arranging visual information into topographic maps on the cortex or mapping the surface of the body onto a cortical map that looks like a distorted map of the body. Presumably the way the brain represents information is useful for the kinds of things the brain does with it.

There is also interest in studying systems that can organize themselves, that is, develop the optimal way of arranging the information they must represent. Modelers can either make use of representations given to us by nature or improve on them.

Our feeling is that problems of representation may prove to be the major area for research in neural networks in the next few years. This is because *all* practical applications of networks are critically dependent on the way the problem is represented. With the proper description of the inputs, many learning algorithms will probably work adequately. Without the proper description, none may work. One of the major criticisms of neural networks has been the necessity up to this point to 'handcraft' the representations in order to make the systems work. There are some obvious commonsense general rules for representations that also seem to be brain-like. First, similar inputs *usually* should give rise to similar representations. Second, things to be separated should be given widely different representations. Third, if something—a sensory property or feature of the input—is important, lots of elements should be used to represent it. Fourth, do as much lower-level preprocessing as possible, so the learning and adaptive parts of the network need do as little work as possible. Build 'invariances' into the hardware and do not require the system to learn them.

The Future

We have provided an Afterword after the last paper, to summarize some of our feelings about the way the field of neural networks has developed and where it now stands.

These comments are meant to be terse. The field of neural networks has had disasters in the past, and the science as a whole has passed from great enthusiasm to rejection to, now, modest excitement. Current understanding of what networks can and cannot do is much more realistic than it was twenty years ago, and our technical resources are much greater, so it is easy to contemplate special-purpose hardware and very large systems. Perhaps in this cycle of reincarnation neural networks will take their place as a useful complement to and extension of traditional computer hardware. We hope so.

Bibliographic Note

We have collected a number of papers that we feel are particularly useful for understanding neural networks. Sometimes these are not the first published versions of the ideas, though often they are. We have usually tried to find the clearest exposition of a point of view, or the most telling example or application of it. We shall point out some of the earlier literature when we think it is appropriate or make historical comments in the introductions to the papers. We have tried to make these introductions to the individual papers reasonably self-contained. Our own experience has been that introductions to individual papers are read much more often than general introductions such as this.

The field of brain modeling and neural networks has attracted over the years a number of scientists with exceptionally strong egos. Protracted and bitter battles over priorities have been waged, remarkable for the field at a time when there were so few participants and so little respect, and still remarkable now, in better times.

Our own feeling is that all ideas have roots. It is virtually impossible to find an 'original' idea that was not presented earlier in the scientific literature, if one is willing to take the effort to search the historical record. When the need for an idea is in the air, the idea is usually discovered by several groups simultaneously. A recent example of this from the neural network literature is the simultaneous discovery of the back propagation learning algorithm in three places in 1985.

We should simply point out that science is a cooperative effort and that no scientist is an island. All are subject to the ideas of the time. To use Newton's metaphor, if one scientist happens to pick up a shinier or more valuable pebble than another, perhaps he only needed enough wisdom to be standing at the point on the beach where those pebbles were likely to be. Or, he had slightly sharper eyes. Or, he spent a lot of time at the shore. Or, perhaps, he only stubbed his toe on something and happened to pick it up through dumb luck. Now, with many scientists and others searching the neural net beaches, lots of new pebbles may be discovered. Science is unlike art and literature in that if one scientist does not find a particular idea, then another one will, and soon.