Alternative Knowledge Acquisition: Developing A Pulse-Coded Neural Network†

W. B. Dress

Oak Ridge National Laboratory
Instrumentation and Controls Division
Oak Ridge, Tennessee 37831-6007

Abstract

After a Rip-van-Winkle nap of more that 20 years, the ideas of biologically motivated computing are re-emerging. Instrumental to this awakening have been the highly publicized contributions of John Hopfield and major advances in the neurosciences. In 1982, Hopfield showed how a system of maximally coupled neuron-like elements described by a Hamiltonian formalism (a linear, conservative system) could behave in a manner startlingly suggestive of the way humans might go about solving problems and retrieving memories. Continuing advances in the neurosciences are providing a coherent basis in suggesting how nature's neurons might function.

This paper describes a particular model for an artificial neural system designed to interact with (learn from and manipulate) a simulated (or real) environment. The model is based on early work at Sandia Laboratories by Iben Browning. The Browning model, designed to investigate computer-based intelligence, contains a particular simplification based on observations of frequency coding of information in the brain and information flow from receptors to the brain and back to effectors. The ability to act on and react to the environment was seen as an important principle, leading to self-organization of the system.

Introduction: A Perspective On Artificial Intelligence

The essence of Artificial Intelligence (AI) is an emulation of human intelligence and intelligent performance at a high functional level. Early on, any detailed model of natural intelligence was abandoned as either too complex or too restrictive. Creativity in AI centered around more tractable, but nonetheless, fascinating problems of mimicing the conceptual abilities of human intelligence (e.g., problem solving, theorem proving, knowledge manipulation) as well as devising algorithms and conceptual paradigms to make the life of computer scientists easier (multi-tasking; word processing; rule-based systems; semantic nets, frames and scripts; expert systems). A compelling reason for this choice was the restriction imposed by available hardware generally subsumed under the generic phrase, "The von Neumann Bottleneck." Brain-like functions (parallel and distributed processing, associative memory recall) were extraordinarily difficult to map directly onto existing computer hardware architectures; simulation of these functions, while interesting, proved to be impractical. There were a few researchers whose overriding interest was in the functioning of the human brain and mind and of means for simulating such behavior. These dedicated few were intent on seeing an artificial intelligence (ai) arise which overcame the semantic barrier of AI. Classical AI is primarily concerned with the manipulation of symbols, exploring syntactical relationships while ignoring meaning. A parallel would be the field of experimental psychology under B. F. Skinner's influence:

† Research performed at Oak Ridge National Laboratory, operated by Martin Marietta Energy Systems, Inc., for the U. S. Department of Energy under Contract No. DE-AC05-84OR21400.

enormous strides in a narrow direction were made by ignoring or even denying the existence of the mind. Similarly, the laws of syntax and symbol manipulation can be studied and applied (very profitably, too, as expert systems prove) without any reference to the underlying meaning (semantics). However, the era of great conceptual and computational advance is probably over—AI needs to become ai and consider meaning just as psychology needed to turn from a pure behaviorist paradigm and acknowledge the mind. Recent important advances in the cognitive sciences [AND85] are illustrative of the fruitfulness of this particular paradigm shift.

Background: What are Neural Nets?

If we tend to blame von Neumann for his infamous computational bottleneck, we must also remember that he pointed the way out: cellular automata [NEU66] made up the first neural nets and are directly responsible for the several connectionist architectures available in hardware form today [e.g., HIL85]. Probably any collection of coupled, similar, and even simple computing elements may be labeled as a "neural network," with apologies to those dedicated neuroscientists who have studied and wrestled with the complexities of real neurons and real neural networks over the past three quarters of a century.

The field of artificial neural networks is large, encompassing the simplest models with behavior described by linear non-stochastic summing networks [HOP86] to systems with stochastic connectivity and non-linear behavior modeled closely on our current understanding of nature's neural networks and their complex interactions with perception and learning [GRO81].

The goals of neural network research are perhaps as diverse as the models used. Certainly, some neuroscientists wish to see a research tool for refining queries about real neurons and their possible range of behavior. Associative memories for rapid recall of large information sets and dynamic pattern recognition are worthy goals, as is the application of statistical mechanics models to the important problems of scheduling and routing. The promise of storing very large (terabytes) amounts of information in a single device is a driving force for some researchers. Should artificial neural networks exhibit emergent properties ("spontaneous production of new things" [POP77]), the psychological and philosophical implications would provide other compelling reasons for study. A particular application of the unique properties of the pulse-coded model may extend scheduling into the time domain. A real neural network can develop plans in response to a changing and complex environment, not just respond as a deterministic sequence of stable system states. Will an artificial one be as flexible?

Goals of Current Research

The goals set for this current research into artificial neural systems are both practical and theoretical. They are:

1) to develop a methodology for treating the difficult (impossible?) problems arising from complexity that is found in most advanced systems (e.g., power plant control, control of space-based weapons),
2) to demonstrate the control potential offered by a neural system in an intelligent autonomous robot, and
3) to demonstrate the properties of self-organization and emergence. A self-organizing system is one that modifies its own behavior in response to external influences. It is not programmed for adaptation to specific external inputs, but rather it is given a generic prescription for incorporating new information. Initially, such a system can do nothing; as it interacts with its environment, it will learn to cope successfully as it is guided by stimulus and response conditioning. Eventually, complex behavior (hopefully

appropriate) will emerge from such a system. We will have reached the goal of artificial intelligence, as intelligence may be defined as capacity for generating an appropriate response to unforeseen circumstances.

The Browning Model — A Frequency-Coded Network

A large part of the neuroscience literature is concerned with the nature of the "spike" or pulse activity of neural messages, its effects on cell membranes, its transmission properties, and its biochemical interactions. The computational properties of networks of neurons and the properties of information flow through the network are generally too speculative for the neuro-laboratory. However, it is evident that neuroscientists are fascinated with the informational aspects as well as with the problem of emergence of mind [POP77] and [PRI86]. The work of neuro-psychologist Karl Pribram at Stanford University is a notable exception to the rule of saving information-theoretical notions for the future.

Even as Pribram was developing his theory of junctional micropotentials, set forth in Pribram [PRI71], Browning was designing and experimenting with a computer model based on the idea of the brain as a frequency-response system that was capable of acting on its environment and receiving information in the form of pulse trains with instantaneous frequency-coded meaning. His efforts are summarized in a report [BRO64], wherein we learn that both frequency coding and environmental interaction were key concepts in his frequency-based model.

An example of the Browning network written in Basic is to be found in Winkless and Browning [WIN78]. Essentials of the network are described and an example given shows how the pulses flow through the network and that the node firing the fastest will soon dominate all network activity, inhibiting all outputs except one. Inhibition is manifest even though the model contains no specifically inhibitory mechanisms. The model developed in the present work is an extension of the one cited above. A personal computer with a mouse, menu system, windowing capabilities, and the dynamically modifiable and extensible Forth language created a development environment ideal for experimenting with neural networks. Experimenting in this area is essential because any viable network has numerous parameters of unknown value, such as network connectivity, node refractory period, node threshold, etc. There are two approaches to the determination of these variables: random trial and error (based on a genetic algorithm approach) and random trial and error (based on best guess observations).

In the late 1950s there was no such choice, the genetic approach was taken of necessity. Actually, a genetic approach is more systematic and not subject to the experimenter's biases. Browning's work was done in Fortran on a batch machine (and we think we have it rough!). Without the interactive ability, the experimenter had to visualize the network's behavior from the output of a stream of numbers on a paper printout. Today, graphical representations can evolve and "behavior" be rewarded or punished as it occurs. The danger is in laziness, as it is easier to modify a piece of code and try it again in a matter of minutes than to really think about it (when you had days to weeks between computer runs, you "tend to think a lot," as Browning noted). The scheme for an artificial neural network exhibiting self-organizing properties includes an environment, a means to sense it, and a means to manipulate it. A brain needs to have sensory input; without sensory input the brain is in a pathological state called catatonia. Can a network operate without the ability to manipulate its environment? Of course, it operates very nicely, but it never learns anything. This is dramatically illustrated in both Pribram [PRI71] and Popper and Eccles [POP77], where the work of R. Held and A. Hein is described. A kitten deprived of the opportunity to interact with its environment other than by passive sensory means is a seriously learning-impaired animal.

Figure 1 illustrates how a neural network might interact with an environment. The sensory inputs (labeled afferents in the figure) are essential to learning and may be essential to the operation of the network if spontaneous cell firing is not allowed [PRI71]. The environment may be as simple as a set of muscles controlling the motion of a simulated finger on the computer's display screen. Feedback from the environment may be added by injecting a high frequency pulse train into one or more of the network's inputs (pain) when the finger approaches the screen's boundaries. The system quickly learns to avoid the boundaries and avoids the "painful" regions by allowing the finger to hover in the middle region of the screen.

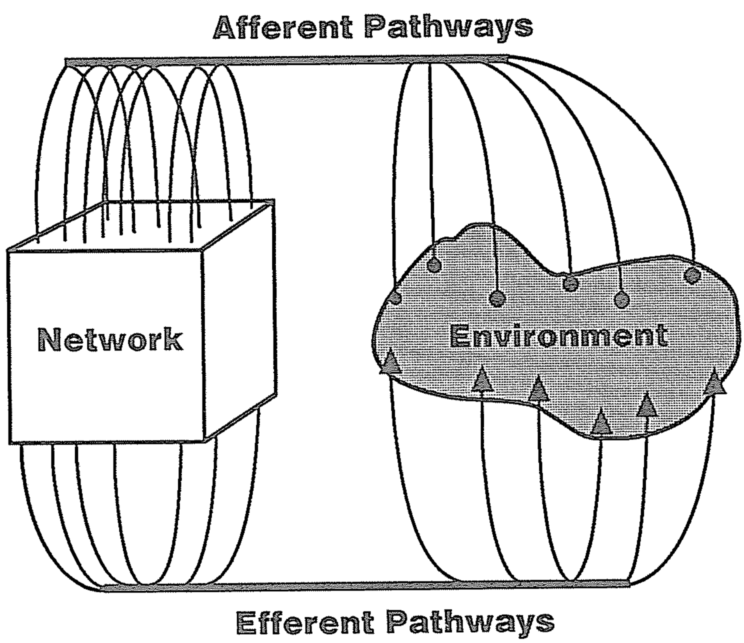

Figure 1. Interaction pathways between the network and an environment, showing net receptors (sensors) as black dots and effectors as arrows.

Note the similarity of this diagram to those of adaptive control theory [e.g., HAR81]. The network illustrated above reacts to and acts on the environment. As it does so, certain patterns of node firings stabilize and others disappear. When a change in inputs is made, certain learned pathways are selected, and the network is then responding to a supplied stimulus.

Description of the Model

The fundamental unit of the Browning network (and the modified version developed in the course of this work) is the node or neural cell. Each cell has a set of properties:

1. List of input synaptic junctions
2. List of output synaptic junctions
3. A refractory period when the cell cannot be made to fire
4. A threshold value that is compared to the sum of active inputs
5. A cell potential which slowly charges, allowing inputs to become more effective in overcoming the cell's threshold
6. Spontaneous firing when the cell's potential reaches a maximum
7. A "differential permeability" for modeling cell fatigue and certain pathologies

8. A cell status and an identification number.

The data structure schematic, shown in Figure 2, illustrates the relationships between the nodes and synapses. The node structure indicates those data needed to model a neuron, and is used for recording the state of each node as the system evolves.

A 1024-node system with an average of 10 synapses into and out of each node would require 24 Kbytes and perhaps 16 Kbytes in a streamlined version. A 1 Mbyte computer has capacity for perhaps a 50,000-node network as well as an operating system and the code for running the network (including code for controlling external sensors and effectors, and for graphics).

Technically, the network is operated in a synchronous fashion, but logically it supports asynchronous operations. A (simulated) system clock effectively freezes network activity of each cycle. During this time, the state of each cell is altered according to the following algorithm:

1. Check each of the network receptors and adjust the instantaneous firing rates of any affected network input cell,
2. Set each synapse on the output list of a node that fired the previous cycle ("pulse state") to its active state,
3. Add a small amount of "noise" by randomly selecting a few cells for spontaneous firing,
4. For each node in the network, check if it has waited for its refractory period (nominally a few system cycles),

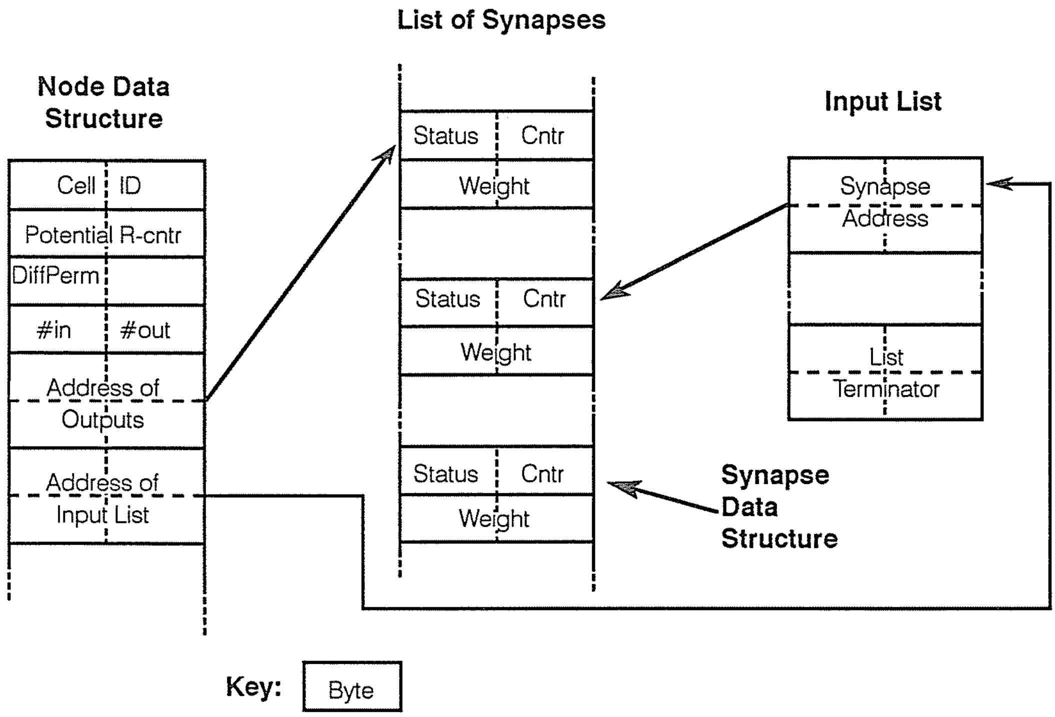

Figure 2. Data structures for representing the nodes and synapses in the model. "R-cntr" is the counter for the refractory period where a node may not fire; "DiffPerm" denotes the current differential permeability of the cell. The present model allows 16 bytes per cell (this could be reduced to 10 in a streamlined version by eliminating No. 8 and consolidating others). The synaptic junctions consist of a status, a weight, and a delay counter, which is not presently used. In addition, each synapse has an associated address that is used to identify a node's input, making a total of 8 bytes per synapse.

5. If it has, allow the cell potential to increase exponentially toward saturation,
6. Sum the cell's active inputs, add the cell's potential and compare to the system-wide threshold,
7. If the threshold is exceeded, fire the cell by,
   a. resetting its potential and refractory period to initial values,
   b. setting each synapse on the cell's output list to "pulse state,"
8. Reset all the cell's input synapses to the inactive state,
9. Check any network output nodes, and send messages to the appropriate effectors.

Pulse transit time is presently one system clock tick and the pulses persist for one system clock, making for a tightly coupled network (in the time domain) and imposing a severe restriction of simultaneity on a cell's inputs. This process could easily be made more realistic by using the synapse counter indicated above to simulate pulse delay as a function of distance between nodes. Persistence over a few system cycles could be easily added.

Memory, Learning, and Knowledge

Information in the model is distributed along various dynamic pathways. There is no idealized location for data (as in a von Neumann machine with its storage cells containing either definite data or prescriptions for controlling the machine). As pathways interact, information is modified, and all distinction between data and program disappears. Important questions for consideration are:

1) where and how is information stored? and
2) what is the information capacity of the network?

Only qualitative answers to these questions may be given at this time. Other network models have supplied definite answers to the capacity question [AMI85].

The network establishes certain pathways of node firings as it interacts with its environment. This behavior may be viewed as self-organization in response to environment or learning. Specific mechanisms of learning are being explored by modifying thresholds, cell permeability, and synaptic efficacy. A promising mechanism has been proposed [LYN84] describing structural changes taking place in response to certain types of stimuli. A more general theory of learning and memory is discussed in a recent paper [THO86]. Thompson examines forms of habituation and associative learning and recall from a neurobiological viewpoint. Another recent paper [KLO86] incorporates neurobiological findings into a computer model that can account for a wide spectrum of learning methods.

The Hardware Connection: Running Neural Nets

One of the problems of the early days is still with us: running a neural network on a serial machine can be painfully slow, too slow for interaction with the real world. There are several hardware fixes to be mentioned:

1) use a faster serial processor;
2) use some of the newer parallel machines such as hypercube architectures, connectionist architectures, distributed architectures; and
3) design specialized neural computing elements.

Each of these possibilities is being pursued at laboratories and businesses around the country.

We are pursuing all three possibilities at Oak Ridge National Laboratory. Transferring the code to the Novix NC4000 and later NC6000 development systems is a straight forward activity. The result should be 50 to 100 times faster (based on preliminary studies with the NC4000) than the present high-level Forth code. Access to a distributed processing system now under development at the Laboratory should show further improvement. This parallel machine has a Forth kernel running on each of its processors, so the port of the present network code should pose no problems. Either of these approaches would allow a modest network of a few thousand nodes to operate with a time constant of a few tens of milliseconds. This would be on the order of a human's response to certain situations and be quite acceptable for a certain class of real-time control problems (e.g., robotics).

The third approach, that of designing specific neuron-like cells in silicon, is being pursued at a number of places [MAR86]. At Oak Ridge, we are beginning a design based on the network described here.

Discussion

The collection of nodes and synapses may be viewed as a set of coupled oscillators. Certain momentary phase relationships become established and may persist or be disrupted as information flows through the network. If one set of pathways is momentarily persistent (even in the presence of noise), the affected outputs will continue to fire in a quasi-periodic manner. However, if one or more input cells change their firing rate, the established pattern of pathway firings may be disrupted and a new set will become stable. A formal description of this behavior can be made by identifying the persistent states with attractors in a dynamical system (defined on a phase space based on the nodes' properties). A change in pathways is then identified with an orbital perturbation. A rigorous treatment of these ideas depends upon a formal description of the network dynamics and may well be an intractable problem. It is often difficult to give formal meaning to such vague and intuitive ideas, especially ones describing the behavior of a complex system; however, Babloyantz has come remarkably close in Chapter 14 of her book [BAB86], wherein she develops a system of nonlinear equations along the lines of Nicolis and Prigogine [NIC77]. The resulting system of coupled neural-like elements (each having a membrane potential and delayed response times to input activity) undergoes bifurcation from stable to oscillatory states as excitatory membrane potential is increased. When the excitatory population parameter starts to oscillate, it closely resembles EEG traces of onset of epilepsy.

Toward Formal Description

The network is constructed on a stochastic model of connectivity with both feedforward and feedback allowed, making any rigorous mathematical description both complex and difficult. Even though the individual network elements and their transfer functions can be taken as linear, the interaction of the net as a whole with the environment is reminiscent of a system of transformations (reactions) wherein information (chemical species) undergoes changes. The resulting system can exhibit self-organization if interactions are at least of cubic order [NIC77].

If the computational field of the network is abstracted to the set of connections describing the network's connectivity, the nodes (network neurons) may be taken as simple pulse-summing junctions with a two-state (binary) transfer function. The information processing (aquisition, storage, modification) is contained in the synaptic field and its self-modification is mediated by sequential patterns of firing of the nodes. Two synapses (a and b) are said to be connected if synapse a terminates on a node that can send output pulses to synapse b. In Figure 3, synapses a, b, and c are connected to synapses a', b', and c'. This connectivity is neither associative nor reflexive, thus a' is not necessarily connected to a if a is connected to a'; a is not necessarily connected to itself. This type of connectivity is denoted as $a \rightarrow na'$, where n is the length of the connection (number of nodes traversed from a to a'). Note that n is not necessarily unique as there may exist several paths from one node to any other node, and not all need be of the same length. A modified form of transitivity holds across nodes: If $a \rightarrow nx$ and $x \rightarrow mb$, then $a \rightarrow kb$, where $k = n + m$ is the (non-unique) composite length.

We may now make a few definitions:

1. If $a \rightarrow a'$ then a is said to be causally connected to a'.
2. A causal path π is an ordered set of causally connected elements, each adjacent pair obeying transitivity with length 1. Thus $\pi = \{a, b, c, ..., z\}$ is a causal path iff $a \rightarrow 1b$, $b \rightarrow 1c$, $c \rightarrow 1 \ldots, \rightarrow 1z$.
3. A firing sequence or a pulse sequence is an ordered set of node firings, each separated by one system clock period.

Theorems may be developed describing how each causal path has a unique length, and a causal path of length >1 may be broken into subpaths of length minimally 1. Causal paths have loops and branches pruned off. A branch is a unique and different causal path, connected to the original one, while a loop is a closed causal path. The formalism describing loops, subpaths, and branches belongs to a branch of graph theory and will not be further discussed.

Note that the time dimension has not yet appeared explicitly. Time is the basis for the functioning of this model network, and obviously it is related to the notion of a causal path. The key is to note that path length is a measure of the time required for a pulse to traverse the path:

Path Length times System Clock Period = Duration of Pulse Sequence

A chain of node firings traversing a causal path does not necessarily have to traverse the path completely. If each node in the path is past its refractory period and the sum of active inputs (synapses) is greater than the node threshold, only then will the firing continue down the causal path. Of more than passing interest is a theorem which says that if a pulse sequence is on a causal

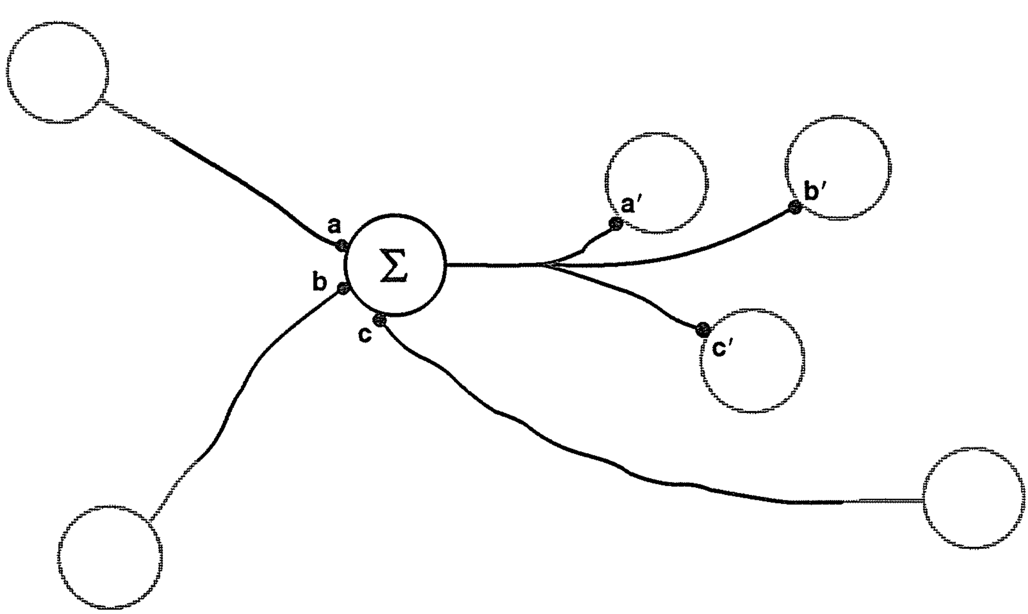

Figure 3. Network connectivity showing nodes as open circles and synapses as black dots. The complete network is made up of many such nodes or cells interconnected via the synapses. Efferent synapses of cell Σ are labeled **a**$'$, **b**$'$, and **c**$'$; and Σ's afferent synapses are labeled **a**, **b**, and **c**.

path, each member (synapse) of the path must become active in a sequential fashion, activity being separated by one system clock period. Proof follows directly from definition 3. This theorem leads to several important consequences about path interactions and path effectiveness.

A theorem of Browning's concerns path length: the shortest active path between nodes carries the information. Any paths that are longer have no effect and may be ignored. The proof considers the causal path lengths compared with the time it takes for the end node to recover to its firing potential. Of course, if the shorter causal path is too weak to pass the pulse, then it can't support a pulse sequence, and no information is transmitted.

A theorem describing typical network behavior states that if pulse sequences traverse two intersecting causal paths, the first sequence to reach the intersection will branch at the junction and prevent the second sequence from propagating past the junction if the lengths of the paths are less that the system recovery time. Thus certain paths (and hence information flows) dominate.

Continuation of this formal development will be deferred to a subsequent paper. Interested readers wishing to pursue similar ideas on connectivity and inheritance (of properties, not necessarily pulses or information) are referred to Touretzky [TOU86]. Another approach toward formalization may be followed using the language of directed graphs. The dynamics of the network (messages) would be reflected in the dynamics of the connections on the graph.

Summary and Future Directions

Neural nets are starting to show great promise in doing real-world computations, acquiring knowledge, storing information, and generating complex behavior in response to external conditions. The set of complex problems awaiting computer science in the 1990s is so vast that neural nets will continue to be studied until their success is unequivocally demonstrated or failure becomes evident. The simple model described above has already demonstrated learning and self-organization in response to a rather simple environment. The demonstration of emergence will probably involve a more sophisticated environment and more than one modality of learning [POP77].

Addition a real voice input based on the ideas of Gabor [GAB46] and Browning [BRO86], is planned for the near future. The network will then be given simulated visual input to correlate with real-time voice input. The speculation is that the environment will become rich enough to elicit emergent behavior. From a practical viewpoint, this may well lead to a form of computer voice recognition which goes a step beyond simple pattern matching to a speaker-independent system that exhibits fault tolerance and makes the usual perceptual errors.

Note added in proof:

This paper was written in early 1987. In the following year, the ideas presented above evolved into a neural network with added sensors (for vision, taste, and touch). The ensemble was called an "Adaptive Synthetic Insect." This insect lived on a computer screen and learned about its environment via self-organization of information received from the sensors due to its interactions with the environment. A genetic algorithm allowed the bug to follow evolutionary pathways in a genetic state space, thus optimizing its behavior over many generations. The latest bug has solved a problem imposed by its environment in a surprising way. The insect is penalized for touching the edges of the screen (walls) with its feelers, and rewarded for finding and consuming food in its environment. Instead of becoming sensitized to the walls, as I fully expected, the bug evolved the capability to walk backwards at a fast pace, and turn around occasionally to eat when food was located. If a property of intelligence is the ability to solve a problem in a surprising fashion, then this insect (computer program) has demonstrated both unsupervised discovery and a modicum of intelligence. Future evolution may contain more surprises!

References

[AMI85] Daniel J. Amit and Hanoch Gutfreund and H. Sompolinsky, "Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks," *Phys. Rev. Lett*, Vol. 55, No. 14, pp. 1530–1533, 1985.

[AND85] John R. Anderson, *Cognitive Psychology and Its Implications*, 2nd edition, W. H. Freeman, New York, 1985.

[BAB86] A. Babloyantz, *Molecules, Dynamics, and Life*, John Wiley & Sons, New York, 1986.

[BRO64] Iben Browning, "A Self-Organizing System Called 'Mickey Mouse'," A report published by Panoramic Research, Inc., Palo Alto, California, April 1964.

[BRO86] Iben Browning, private communication, 1986.

[GAB46] Denis Gabor, "Theory of Communication," *Journal of the I.R.E.*, Vol. 94, No. 26, pp. 429–457, 1946.

[HAR81] C. J. Harris and S. A. Billings, *Self-Tuning and Adaptive Control: Theory and Applications*, IEE Control Engineering Series 15, Peter Peregrinus Ltd., London, 1981.

[HIL85] W. D. Hillis, *The Connection Machine*, MIT Press, Cambridge, MA, 1985.

[GRO81] Stephen Grossberg, "Adaptive Resonance in Development, Perception and Cognition," *SIAM-AMS Proceedings*, Vol. 13, pp. 107–156, 1981.

[HOP86] John J. Hopfield and David W. Tank, "Computing with Neural Circuits," *Science*, pp. 625–633, 8 August, 1986.

[KLO86] A. Harry Klopf, "A Drive-Reinforcement Model of Single Neuron Function: An Alternative to the Hebbian Neuronal Model," from the Neural Networks for Computing Conference, Snowbird, Utah, 13–16 April, 1986.

[LYN84] Gary Lynch and Michel Baudry, "The Biochemistry of Memory: A New and Specific Hypothesis," *Science*, pp. 1057–1063, 8 June, 1984.

[MAR87] John Markoff, "The second wave in artificial intelligence," *San Francisco Examiner*, Sunday, January 4, 1987.

[NIC77] G. Nicolis and I. Prigogine, *Self-Organization in Nonequilibrium Systems*, John Wiley & Sons, New York, 1977.

[POP77] Karl R. Popper and John C. Eccles, *The Self and Its Brain*, Springer-Verlag, Berlin, 1977.

[PRI86] Karl H. Pribram, "The Cognitive Revolution and Mind/Brain Issues," *American Psychologist*, pp. 507–520, May 1986.

[PRI71] Karl H. Pribram, "Languages of the Brain," Prentice-Hall, New York, 1971.

[THO86] Richard F. Thompson, "The Neurobiology of Learning and Memory," *Science*, pp. 941–941, 29 August, 1986.

[TOU86] David S. Touretzky, *The Mathematics of Inheritance Systems*, Morgan Kaufmann, Los Altos, California, 1986.

[NEU66] John von Neumann, *Theory of Self-Reproducing Automata*, edited by A. W. Burks, University of Illinois Press, Urbana, Illinois, 1966.

[WIN78] Nels Winkless and Iben Browning, *Robots on Your Doorstep*, Robotics Press, Portland, Oregon, 1978.