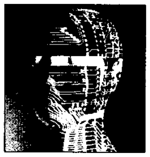


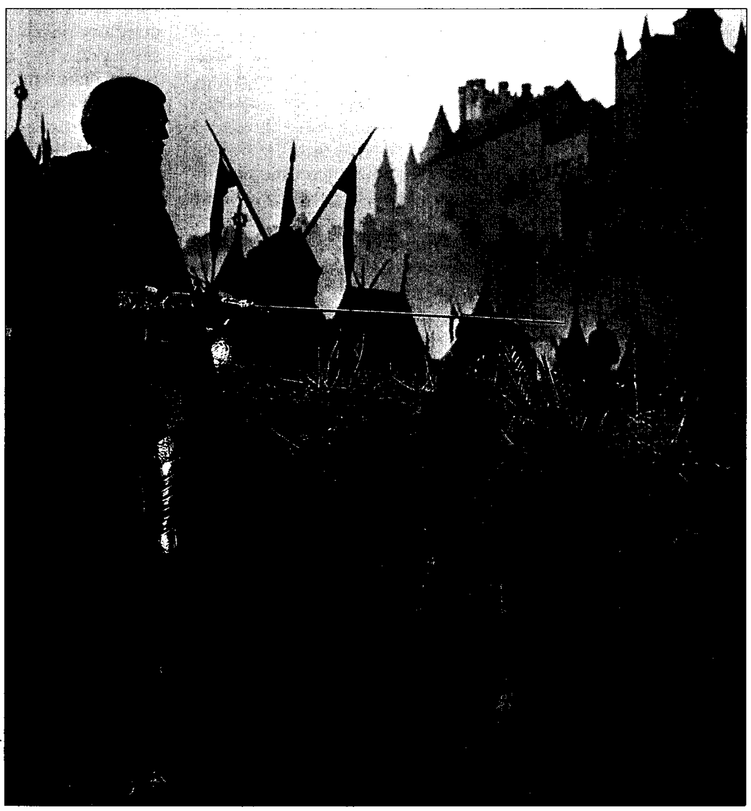
Early Neural Network Development History: *The Age of Camelot*

R.C. Eberhart and R.W. Dobbins
The Johns Hopkins University
Applied Physics Laboratory

The development history of neural networks can be divided into four segments, or "Ages." We have arbitrarily set the beginning of the first Age to the time of William James, about a century ago. We call this the Age of Camelot, and it ends in 1969, with the publication of the book by Minsky and Papert on perceptrons [1]. This article reviews these early days of neural network research. Following the Age of Camelot is the Dark Age (or Depression Age) running from 1969 until 1982, when Hopfield's landmark paper on neural networks and physical systems was published [2]. The third age, the Renaissance, begins with Hopfield's paper and ends with the publication of Parallel Distributed Processing, Volumes 1 and 2, by Rumelhart and McClelland in 1986 [3,4]. The fourth age, named the Age of Neoconnectionism after the review article by Cowan and Sharp on neural nets and artificial intelligence [5], runs from 1987 until the present. Much of the material in this article is excerpted from a book edited by the authors [6]. Also presented in the book is further discussion of the other three Ages of neural network development.

The history is reviewed here somewhat differently than in most other articles on neural networks, in that the focus is on people rather than just on theory or technology. We review the contributions of a number of individuals, and relate them to how neural network tools are being implemented today. A neural network tool is an analysis tool, modeled after the massively parallel structure of the human brain. This tool simulates a highly interconnected, parallel computational structure with many relatively simple individual processing elements, or neurodes.

The selection of individuals discussed in this article is somewhat arbitrary. The intent is to provide a broad sampling of people who contributed to current neural network technology, not an exhaustive list. Some well known neural networkers are mentioned only briefly, and others are omitted altogether. We discuss the selected people and their contributions roughly in a chronological order.



culver pictures

The Age of Camelot

William James

We begin our look at neural network history in the Age of Camelot with perhaps the greatest American psychologist who ever lived, William James. James also taught, and thoroughly understood, physiology. It has been almost exactly a century since James published his "Principles of Psychology," and its condensed version, "Psychology (Briefer Course)" [7]. James was the first to publish a number of facts related to brain structure and function. He first stated, for example, some of the basic principles of correlational learning and associative memory. In stating what he called his Elementary Principle, James wrote:

"Let us then assume as the basis of all our subsequent reasoning this law: *When two elementary brain processes have been active together or in immediate succession, one of them, on reoccurring, tends to propagate its excitement into the other.*"

This is closely related to the concepts of associative memory and correlational learning. James seemed to foretell the notion of a neuron's activity being a function of the sum of its inputs, with past correlation history contributing to the weight of interconnections, when he wrote:

"The amount of activity at any given point in the brain cortex is the sum of the tendencies of all other points to discharge into it, such tendencies being proportionate to the number of times the excitement of each other point may have accompanied that of the point in question; to the intensity of such excitements; and to the absence of any rival point functionally disconnected with the first point, into which the discharges might be diverted."

McCulloch and Pitts

More than half a century later, McCulloch and Pitts [8] published one of the most famous "neural network" papers, in which they derived theorems related to models of neuronal systems based on what was known about biological structures in the early 1940s. In coming to their conclusions, they stated five physical assumptions: The activity of the neuron is an 'all-or-none' process; a certain fixed number of synapses must be excited within the period of latent addition in order to excite a neuron at any time, and this number is independent of previous activity and position on the neuron; the only significant delay within the nervous system is synaptic delay; the activity of any inhibitory synapse absolutely prevents excitation of the neuron at that time; the structure of the net does not change with time.

The period of "latent addition" is the time during which the neuron is able to detect the values present on its inputs, the synapses. This time was described by McCulloch and Pitts as typically less than 0.25 ms. The "synaptic delay" is the time delay between sensing inputs and acting on them by transmitting an outgoing pulse, stated by McCulloch and Pitts to be on the order of 0.5 ms. The neuron described by these five assumptions is known as the "McCulloch-Pitts neuron." The theories they developed were important for a number of reasons, including the fact that any finite logical expression can be realized by networks of their neurons. They also appear to be the first authors since William James to describe a massively parallel neural model.

While the paper was very important, it was (and still is) very difficult to read. In particular, the theorem proofs presented by McCulloch and Pitts have stopped more than one engineer in their tracks! Furthermore, although the paper has proved to be an important milestone, not all of the concepts presented in it are being implemented in today's neural network tools. In this article, comparisons are not made between the theories and conclusions of McCulloch and Pitts (or anyone else), and current theories of neural biology. The focus here is strictly on the implementation (or non-implementation) of their ideas in neural network tools.

One example of something not generally being implemented is their all-or-none neuron. A binary, on or off, neurode is used in neural networks such as the Boltzmann Machine, but it is not generally used in most neural networks today. Much more common is a neurode whose output value can vary continuously over some range, such as from 0 to 1, or -1 to 1. Another example involves the signal required to "excite" a neurode. First of all, since the output of a neurode generally varies continuously with the input, there is no "threshold" at which an output appears. Some neural network tools use neurodes that activate at some threshold, but this is not commonly done.

For neurodes with either continuous outputs or thresholds, there is no "fixed number of connections" (synapses) that must be excited. The net input to a neurode is generally a function of the outputs of the neurodes connected to it upstream (presynaptically), and the connection strengths to those presynaptic neurodes.

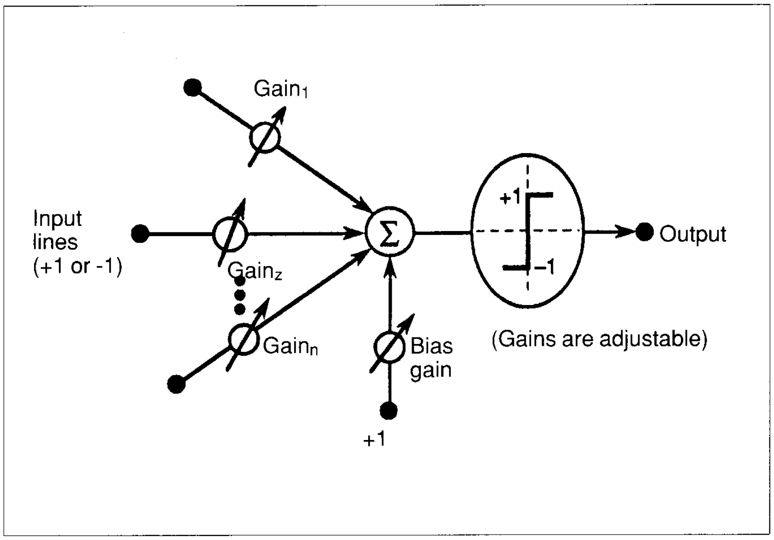
A third example is that there is generally no delay associated with the connection (synapse) in a neural network tool. Typically, the output states (activation levels) of the neurodes are updated synchronously, one slab (or layer) at a time. Sometimes, as in Boltzmann Machines, they are updated asynchronously, with the update order determined stochastically. There is almost never, however, a delay built into a connection from one neurode to another.

A fourth example is that the activation of a single inhibitory connection usually does not disable or deactivate the neuron to which it is connected. Any given inhibitory connection (a connection with a negative weight) has the same absolute magnitude effect, albeit subtractive, as the additive effect of a positive connection with the same absolute weight.

Referring to the fifth assumption of McCulloch and Pitts, it is true that the structure of a neural network tool usually does not change with time, with a couple of caveats. First, it is usual to "train" neural networks, such as backpropagation and self-organizing networks, prior to their use. During the training process, the structure doesn't usually change, but the interconnecting weights do. In addition, it is not uncommon, once training is complete, for neurodes that aren't contributing significantly to be removed. This certainly can be considered a change to the structure of the network.

But wait a minute! What are we left with from McCulloch and Pitts' five assumptions? If truth be told, when referring to today's neural network tools, we are in most cases left with perhaps one: the fifth. Then why do we make so much of their 1943 paper? First of all, because they proved that networks of their neurons could represent any finite logical expression. Second, because of their use of a massively parallel architecture. And third, McCulloch and Pitts provided the stepping stones for the development of network models and learning paradigms that followed.

Just because neural network tools don't currently always reflect their work doesn't imply in any way that their work was bad. Our neural network tools don't always reflect what we currently understand about biological neural networks, either. For instance, it appears that in many cases, a neuron acts somewhat like a voltage controlled oscillator (VCO), with the output frequency a function of the input level (input voltage). The higher the input, the



1. Adaline, an adjustable neuron, consists of a single neurode with an arbitrary number of input elements, each of which may assume a value of +1 or -1; and a bias element that is always at +1.

more pulses per second the neuron generates. Neural network tools usually work with basically steady state values of the neurode from one update to the next.

Donald Hebb

The next personality along our journey through the Age of Camelot is Donald O. Hebb. His 1949 book entitled "The Organization of Behavior" [9] was the first to define the method of updating synaptic weights that we now refer to as "Hebbian." He is also among the first to use the term "connectionism."

Hebb presented his method as a "neurophysiological postulate" in the chapter entitled "The First Stage of Perception: Growth of the Assembly." It is stated as follows:

"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B, is increased."

Hebb made four primary contributions to neural network theory. First, he stated that in a neural network, information is stored in the weight of the synapses (connections). Second, he postulated a connection weight learning rate that is proportional to the product of the activation values of the neurons. Note that his postulate assumed that the activation values are positive. Since he didn't provide for the weights to be decreased, they could theoretically go infinitely high.

Others since Hebb have labeled learning that involves neurons with negative activation values as "Hebbian". This is not included in Hebb's original formulation, but is a logical extension of it.

Third, he assumed that weights are symmetric. That is, the weight of a connection from neuron A to neuron B is the same as that from B to A. While this may or may not be true in biological neural networks, it is generally applied to implementations in neural network tools for computers.

Fourth, he postulated a "cell assembly theory" which states that as learning occurs, strengths and patterns of synapse connections (weights) change, and assemblies of cells are created by these changes. Stated another way, if simultaneous activation of a group of weakly connected cells occurs repeatedly, these cells try to coalesce into a more strongly connected assembly. All four of Hebb's contributions are generally implemented in today's neural network tools, at least to some degree. We often refer to learning schemes implemented in some networks as Hebbian.

Frank Rosenblatt

In 1958, a landmark paper by Frank Rosenblatt [10] defined a neural network structure called the perceptron. The perceptron was probably the first honest-to-goodness "neural network tool" because it was simulated in detail on an IBM 704 computer at the Cornell Aeronautical Laboratory. The computer-oriented paper caught the imagination of engineers and physicists, despite the fact that its mathematical proofs, analyses and descriptions contained tortuous twists and turns. If you can wade through the variety of systems and modes of organization in the paper, you'll see that the perceptron is capable of learning to classify certain pattern sets as similar or distinct by modifying its connections. It can therefore be described as a "learning machine."

Rosenblatt used biological vision as his network model. Input node groups consisted of random sets of cells in a region of the retina, each group being connected to a single Association Unit (AU) in the next higher layer. AU's were connected bidirectionally to Response Units (RU's) in the third (highest) layer. The perceptron's objective was to activate the correct RU for each particular input pattern class. Each RU typically had a large number of connections to AU's.

Rosenblatt devised two ways to implement the feedback from RUs to AUs. In the first, activation of an RU would tend to excite the AUs that sent the RU excitation (positive feedback). In the second, inhibitory connections existed between the RU and the complement of the set of AUs that excited it (negative feedback), therefore inhibiting activity in AUs which did not transmit to it. Rosenblatt used the second option for most of his systems. In addition, for both options, he assumed that all RUs were interconnected with inhibitory connections.

Rosenblatt used his perceptron model to address two questions. First, in what form is information stored, or remembered? Second, how does stored information influence recognition and behavior? His answers were as follows:

"...the information is contained in connections or associations rather than topographic representations...since the stored information takes the form of new connections, or transmission channels in the nervous system (or the creation of conditions which are functionally equivalent to new connections), it follows that the new stimuli will make use of these new pathways which have been created, automatically activating the appropriate response without requiring any separate process for their recognition or identification."

The primary perceptron learning mechanism is "self organizing" or "self associative" in that the response that happens to become dominant is initially random. But Rosenblatt also described systems where training, or "forced responses" occurred.

This paper laid the groundwork for both supervised and unsupervised training algorithms as seen today in backpropagation and Kohonen networks, respectively. The basic structures set forth by Rosenblatt are therefore alive and well.

Widrow and Hoff

Our last stop in the Age of Camelot is with Bernard Widrow and Marcian Hoff. In 1960, they published a paper entitled "Adaptive Switching Circuits" that, particularly from an engineering standpoint, has become one of the most important papers on neural network technology [11]. The paper is important from several perspectives. We'll briefly mention a few, and go into more detail about a few more.

Widrow and Hoff are the first engineers we've talked about in our article. Not only did they design neural network tools that they simulated on computers, they implemented their designs in hardware. And at least one of the lunchbox-sized machines they built "way back then" is still in working order! Widrow and Hoff introduced a device called an "Adaline" (fig. 1). Adaline stands for Adaptive Linear. It consists of a single neurode with an arbitrary number of input elements that can take on values of plus or minus one, and a bias element that is always plus one. Before being summed by the neurode summer, each input, including the bias, is modified by a unique weight that Widrow and Hoff call a "gain". This reflects their engineering background, since the term gain refers to the amplification factor that an electronic signal undergoes when processed by an amplifier. The term "gain" may be more descriptive of the function performed than the commonly used term weight.

At the output of the summer is a quantizer that is at +1 if the summer output, including the bias, is greater than zero. The quantizer's output is -1 for summer outputs less than or equal to zero.

The learning algorithm of the Adaline is particularly ingenious. One of the main problems with perceptrons was the length of time it took them to learn to classify patterns correctly. The Widrow-Hoff algorithm yields learning that is faster and more accurate. The algorithm is a form of supervised learning that adjusts the weights (gains) according to the size of the error on the output of the summer. Widrow and Hoff showed that the way they adjust the weights will minimize the sum squared error over all patterns in the training set. For that reason, the Widrow-Hoff method is also known as the Least Mean Squares (LMS) algorithm. The error is the difference between what the output of the Adaline should be and the output of the summer. The sum-squared error is obtained by measuring the error for each pattern presented to Adaline, squaring each value, and then summing all of the squared values. Minimizing the sum squared error involves an error reduction method called gradient descent, or steepest descent. Mathematically, it involves the partial derivatives of the error with respect to the weights.

But Widrow and Hoff showed that you don't have to apply the derivatives. They are proportional to the error (and its sign), and to the sign of the input. They further showed that, for n inputs, reducing the measured error of the summer by 1/n for each input will do a good job of implementing gradient descent. You adjust each weight until the error is reduced by 1/n of the total error you had to start with. For example, if there are 12 input nodes, each weight is adjusted to remove 1/12 of the total error. This method provides for weight adjustment (learning) even when the output of the classifier is correct. Consider the case where the output of the summer is 0.5, and the classifier output is 1.0. If the correct output is 1.0, there is still an error signal of 0.5 that is used to further train the weights. This is a significant improvement over the perceptron, which only adjusts weights when the classifier output is incorrect, and is one reason the learning is faster and more accurate. Widrow and Hoff's paper was prophetic, too. They suggested several practical implementations of their Adaline, stating: "If a computer were built of adaptive neurons, details of structure could be imparted by the designer by training (showing it examples of what he would like to do) rather than by direct designing."

An extension of the Widrow-Hoff learning algorithm is used today in backpropagation networks, and their work in hardware implementation of neural network tools heralded today's cutting edge work in VLSI by people including Carver Mead and his colleagues at Cal Tech [12]. Dr. Widrow is the earliest significant contributor to neural network hardware system development still working in the area of neural networks. He and his students also did the earliest work known to the authors in biomedical applications of neural network tools. One of his Ph.D. students, Donald F. Specht, used an extension of the Adaline, called an Adaptive Polynomial Threshold Element, to implement a vectorcardiographic diagnostic tool that used the polynomial discriminant method [13,14]. Widrow and his colleagues later did pioneering work using the LMS adaptive algorithm for analyzing adult and fetal electrocardiogram signals [15].

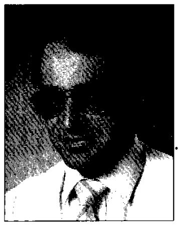
The Fall of Camelot

As the 1960s drew to a close, optimism was the order of the day. Many people were working in Artificial Intelligence (AI), both in the area exemplified by expert systems, and in neural networks. Although many areas remained to be explored, and many problems were unsolved, the general feeling was that the sky was the limit. Little did most folks know that, for neural networks, the sky was about to fall. In 1969, Marvin Minsky and Seymour Papert dropped a bombshell on the neural network community in the form of the aforementioned book called "Perceptrons." The book, which contained an otherwise generally accurate analysis of simple perceptrons, concluded that "...our intuitive judgement [is] that the extension [to multilayer perceptrons with hidden layers] is sterile." At the least, this statement has proven to be a serious mistake. Nevertheless, nearly all funding for neural networks dried up after the book was published. It was the beginning of the Dark Age.

Conclusion

Since 1987 we have been experiencing the Age of Neoconnectionism, so named by Cowan and Sharp. The field of neural networks and the development of neural network tools for personal computers have expanded almost unbelievably in the past several years. It is no longer feasible to assemble "all there is to know" about the current state of neural networks in one volume, or one set of volumes, as the PDP Research Group attempted to do in 1986--1988 [3,4]. The list of applications has grown from one highlighting biological and psychological uses to ones as diverse as biomedical waveform classification, music composition, and prediction of the commodity futures market. And another shift is occurring that is even more important. That is the shift to personal computers for neural network tool implementation. Not that this is the only important trend in neural network research and development today. Significant work is also occurring in areas ranging from the prediction of protein folding using supercomputers to formulation of new network learning algorithms and neurode transfer functions. This article has provided a summary of how it all started. As we stated at the outset, the work of only a few of the neural network researchers and developers was described. Many who contributed significantly to the field were omitted. The intent was to give you a flavor of how the basis for current neural network tools evolved.

Russell C. Eberhart, a Senior Member of the IEEE, received his Ph.D. in Electrical Engineering from Kansas State University in 1972. His is Program Manager in the Biomedical Programs Office of the Johns Hopkins University Applied Science Laboratory. He is Vice President of the IEEE Neural Networks Council, and Vice Chairman of the Baltimore Chapter of the EMBS.



Roy W. Dobbins has a Bachelor's degree in Electrical Engineering from the University of Natal and a Master's degree in Computer Science from the University of South Africa. Since 1987 he has been a consultant at the Johns Hopkins University Applied Physics Laboratory. Current interests include Neural Networks, CASE tools and real time programming.

The authors can be reached at the Johns Hopkins University, Applied Physics Laboratory, Johns Hopkins Rd., Laurel, MD 20707.

References

1. **Minsky M, Papert S:** *"Perceptrons."* MIT Press, Cambridge, MA, 1969.
2. **Hopfield, JJ:** Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci.*, 79:2554-2558 (1982).
3. **Rumelhart DE, McClelland JL:** "Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol I: Foundations." MIT Press, Cambridge, Mass., 1986.
4. **Rumelhart DE, McClelland JL:** "Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol. 2: Psychological and Biological Models." MIT Press, Cambridge, Mass., 1986.
5. **Cowan JD, Sharp DH:**, Neural nets and artificial intelligence. *Daedalus*, 117(1):85-121 (1988).
6. **Eberhart RC, Dobbins RW, eds:** "Neural Network PC Tools: A Practical Guide." Academic Press, New York, 1990.
7. **James, W:** "Psychology (Briefer Course)." Holt, New York, 1890.
8. **McCulloch WC, Pitts W:** A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115-133 (1943).
9. **Hebb, DO:** "The Organization of Behavior." John Wiley, New York, 1949.
10. **Rosenblatt, F:** The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386-408 (1958).
11. **Widrow B, Hoff ME:** Adaptive switching circuits. *1960 IRE Convention Record: Part 4—Computers: Man-Machine Systems, Los Angeles, 96-104 (1960).*
12. **Mead C:** "Analog VLSI and Neural Systems." Addison Wesley, Reading, MA, 1989.
13. **Specht, DF:** Vectorcardiographic diagnosis using the polynomial discriminant method of pattern recognition. *IEEE Trans. on Biomedical Engineering*, BME14(2):90-95 (1967).
14. **Specht, DF:** Generation of polynomial discriminant functions for pattern recognition. *IEEE Trans. on Electronic Computers*, EC-16(3):308-319 (1967).
15. **Widrow B, Glover JR Jr., McCool JM, Kaunitz J, Williams CS et al:** Adaptive noise cancelling: principles and applications. *Proc. IEEE*, 63(12):1692-1716 (1975).