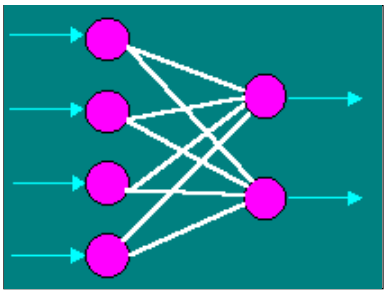
Perceptrons:
An Associative Learning Network

by

Michele D. Estebon
Virginia Tech
CS 3604, Spring 1997



In the early days of artificial intelligence research, Frank Rosenblatt devised a machine called the perceptron that operated much in the same way as the human mind. Albeit, it did not have what could be called a "mental capacity", it could "learn" - and that was the breakthrough needed to pioneer todayís current neural network technologies.

A perceptron is a connected network that simulates an associative memory. The most basic perceptron is composed of an input layer and output layer of nodes, each of which are fully connected to the other. Assigned to each connection is a weight which can be adjusted so that, given a set of inputs to the network, the associated connections will produce a desired output. The adjusting of weights to produce a particular output is called the "training" of the network which is the mechanism that allows the network to learn. Perceptrons are among the earliest and most basic models of artificial neural networks, yet they are at work in many of todayís complex neural net applications.

An Associative Network Model

In order to understand the operations of the human brain, many researchers have chosen to create models based on findings in biological neural research. Neurobiologists have shown that human nerve cells have no memory of their own, rather memory is made up of a series of connections, or associations, throughout the nervous system. A response in one cell triggers responses in others which triggers more in others until these responses eventually lead to a so-called ìrecognitionî of the input stimuli. This is how we recognize objects or events that we see.

A Bit of History

Frank Rosenblatt invented the perceptron in 1957 at the Cornell Aeronautical Laboratory in an attempt to understand human memory, learning, and cognitive processes. On 23 June 1960, he demonstrated the **Mark I Perceptron**, the first machine that could "learn" to recognize and identify optical patterns.

Rosenblattís work was a progression from the biological neural studies of noted neural researchers such as D.O.

Hebb and the works of Warren McCulloch and Walter Pitts. McCulloch and Pitts had been the first to describe the concept of neural networks. They developed the *MP neuron*, which was based on the point that a nerve will fire an impulse only if its threshold value is exceeded. This model was somewhat of a scanning device which read pre-defined input- output associations to determine its final output [7]. MP neurons had fixed thresholds and did not allow for learning. They were " hard-wired logic devices, [which] proved that networks of simple neuron-like elements could compute " [6].

Since the *MP neuron* did not have the mechanisms for learning, it was extremely limited in modeling the functions of the more flexible and adaptive human nervous system. D.O. Hebb suggested that "when an axom of cell A is near enough to excite 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 Aís efficiency as one of the cells firing B is increased" [7]. This implied a "learning" network model where not only could the network make associations, but it could also tailor its responses by adjusting the weight on its connections between neurons.

Rosenblatt took this into consideration, and in **The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain**, published in 1958, he summarizes the basis of his work on perceptron theory in the following assumptions:

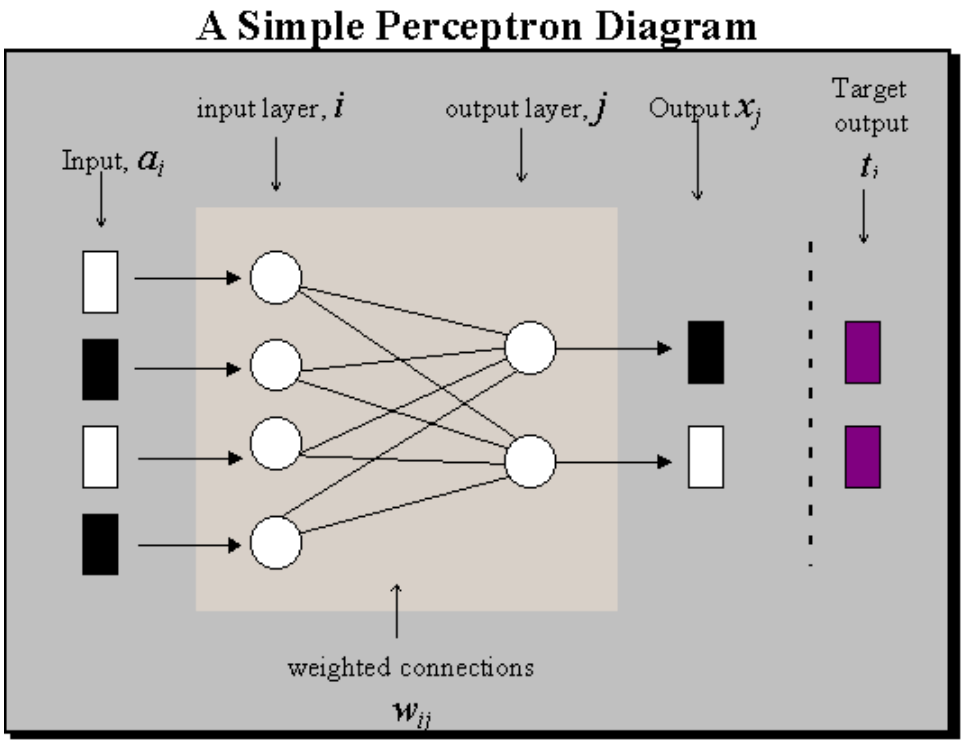
1. The physical connections of the nervous system which are involved in learning and recognition are not identical from one organism to another. At birth, the construction of the most important networks is largely random, subject to a minimum number of genetic constraints.
2. The original system of connected cells is capable of a certain amount of plasticity; after a period of neural activity, the probability that a stimulus applied to one set of cells will cause a response in some other set is likely to change, due to some relatively long-lasting changes in the neuron themselves.
3. Through exposure to a large sample of stimuli, those which are most "similar" (in some sense which must be defined in terms of the particular physical system) will tend to form pathways to the same sets of responding cells. Those which are markedly "dissimilar" will tend to develop connections to different sets of responding cells.
4. The application of positive and/or negative reinforcement (or stimuli which serve this function) may facilitate or hinder whatever formation of connections is currently in progress.
5. *Similarity*, in such a system, is represented at some level of the nervous system by a tendency of similar stimuli to activate the same sets of cells. Similarity ... depends on the physical organization of the perceiving system, an organization which evolves through interaction with a given environment. The structure of the system, as well as the ecology of the stimulus- environment, will affect, and will largely determine, the classes of ìthingsî into which the perceptual world is divided. [3]

With these basic assumptions, Rosenblatt was able to treat the workings of a learning device as "an intricate switching network, where retention takes the form of new connections" [3]. At this, he embarked on his remarkable invention of the perceptron.

How does a perceptron work?

Its Architecture

Using the simple two-layer perceptron as an example, there is one layer of input nodes (layer 1) and one layer of output nodes (layer 2). Each layer is fully connected between the other, but no connections exist between nodes in the same layer (See figure below). When layer 1 sends a signal to layer 2, the associated weights on the connections are applied and each receiving node on layer 2 sums up the incoming values. If the sum exceeds a given threshold, that node in turn fires an output signal.



The outputs are summed across all the inputs (a[i]) received by a node (j) in the output layer. The output of each node is determined as such:

Outputs are determined by:

$$S_j = \sum_{i=0}^{n} a_i w_{ij}$$

If $S_j > \Theta$ then $x_j = 1$
If $S_j \leq \Theta$ then $x_j = 0$

where Θ is a predetermined threshold value

Training

By adjusting the weights on the connections between layers, the perceptronís output could be "trained" to match a desired output. Training is accomplished by sending a given set of inputs through the network and comparing the results with a set of target outputs. If there is a difference between the actual and the target outputs, the weights are adjusted on the adaptive layer to produce a set of outputs closer to the target values.

New weights are determined by adding an error correction value to the old weight. The amount of the correction is determined by multiplying the difference between the actual output (x[j]) and target (t[j]) values by a learning rate constant (C). If the input nodeís output (a[i]) is a 1, that connection weight is adjusted, and if it sends 0, it has no bearing on the output and subsequently, there is no need for adjustment. Thus, the process can be summed as follows:

Error Correction Formula:

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} + C(t_j - x_j)a_i$$

$$C = \text{learning rate}$$

This training procedure is repeated until the networkís performance no longer improves. The network is then said to have "converged". At this point, it has either successfully learned the training set or it has failed to learn all of the answers correctly [1]. If it is successful, it can then be given new sets of input and generally produce correct results on its own.

The functional limitation of a two-layer perceptron, however, is that it can only recognize linearly separable patterns due to only having one adaptive layer [1]. A linearly separable pattern is one that can be separated into two distinct classes by drawing a single line.

However, this limitation fell to the wayside after the introduction of the *back error propagation* paradigm, or *backprop*, as it is more commonly known. Backprop extends the perceptron by implementing a multiple, hidden layer network which is also referred to as a multiple layer perceptron (MLP).

> When two or more layers of weights are adjusted, the network has middle - or hidden - layers of processing units. Each hidden layer acts as a layer of 'feature detectors' - units that respond to specific features in the input pattern. These feature detectors organize as learning takes place, and are developed in such a way that they accomplish the specific learning task presented in the network [1].

Backprop first processes the inputs, checks for errors in the output set, and then proceeds to *back propagate* (hence its name) the errors such that the hidden layers can adjust for the errors as well. Backprop is the most popular paradigm applied in neural nets today.

The Net Effect

The introduction of the perceptron sparked a wave in neural network and artificial intelligence research. However, in 1969, Marvin Minsky and Seymour Papert published a book called **Perceptrons: An Introduction to Computational Geometry**, which emphasized the limitations of the perceptron and criticized claims on its usefulness. In effect, this killed funding for artificial intelligence and neural network research for 12-15 years.

Then in 1982, John Hopfield introduced his model of neural nets which came to be known as *Hopfield Networks* which again revived research in this area. The Hopfield neural network is a simple artificial network which is able to store certain memories or patterns in a manner rather similar to the brain - the full pattern can be recovered if the network is presented with only partial information [8].

In 1986, a team at Johns Hopkins University led by Terrence Sejnowski trained a VAX in the rules of phonetics, using a perceptron network called **NETTalk**. In just twelve hours of learning, the machine was able to ìreadî and translate text patterns into sounds with a 95% success rate [8]. The team noted that the machine sounded uncannily like a child learning to read aloud while it was training.

Today, neural network research is in full force with perceptrons laying the foundations for many applications. Backprop has provided the capability of MLPs which allow for applications in a broader range of complex problems. It has bridged a multitude of disciplines and can be found in software for expert systems, speech recognition, optical character recognition (OCR), knowledge bases, bomb detectors, data visualization, financial market predictions, medical diagnoses, and much, much more.

Conclusion

Whether a scientist embarks on a mission to understand the human brain or to attempt to build a machine which can emulate one, his/her approach will be to generate a model based on the theories at hand. This is what Frank Rosenblatt did in the late 1950ís. Although the simplicity of the original perceptron led skeptics to doubt its usefulness, its overall effect was to open the way for a myriad of applications in artificial intelligence. Its impact has been remarkable.

Bibliography

[1] Dayhoff, Judith E. (1990), **Neural Network Architecturs: An introduction**, Van Nostrand Reinhold (Australia).

[2] Minsky, Marvin and Seymour Papert (1969), **Perceptrons: An introduction to Computational Geometry**, MIT Press.

[3] Rosenblatt, Frank (1958), **The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain**, Cornell Aeronautical Laboratory, *Psychological Review*, v65, No. 6, pp. 386-408.

[4] **The Economist**, 15 April 1995, pp75-77.

[5] Mark I Perceptron Press Conference Records (1960), Charles Babbage Institute: Center for the History of Information Processing, University of Minnesota, Minneapolis, MN.

[6] Some specific models of artificial neural nets, **Brains, Minds, and Computers Lecture Notes**

[7] Jones, Stephen, **Neural Networks and the Computational Brain or Matters Relating to Artificial Intelligance**.

[8] MMM Neural Networks: Training Perceptron Neural Networks

1805 words

Last updated February 17, 1997