Perceptrons, Learning from Mistakes, and a Step Towards AI: An Excerpt from *The Laws of Thought: The Quest for a Mathematical Theory of the Mind*

Tom Griffiths

Part of the power of logic was that it provided not just a way to describe the world, but a way to create new ideas. Starting with a set of facts, inference rules could be followed to derive new facts. Spaces and features gave psychologists new ways of describing people's internal world. But they were still missing a clear characterization of how mental states change over time — of thought itself. Logic was the foundation of the idea of computation, instantiated in the Turing machine. Using spaces and features as mental representations required coming up with a new model of computation.

The Turing machine was based on analyzing the behavior of an idealized mathematician — reading a problem, performing calculations, and writing down intermediate results. But we can find inspiration for another model of computation by taking a closer look at what is going on inside the head of that mathematician. If we zoom in past skin and bone, we discover reading, calculating, and writing are all the result of neurons firing in that mathematician's brain. Those neurons are connected to one another in various ways, forming a neural network.

What if, rather than trying to copy the mathematician's thoughts and actions, we instead try to copy her brain?

The American public was introduced to artificial neural networks by an article that appeared in *The New York Times* on July 13, 1958, with the attention-grabbing headline "Electronic 'Brain' Teaches Itself." The story began:

> The Navy last week demonstrated the embryo of an electronic computer named the Perceptron which, when completed in about a year, is expected to be the first nonliving mechanism able to "perceive, recognize and identify its surroundings without human training or control." Navy officers demonstrating a preliminary form of the device in Washington said they hesitated to call it a machine because it is so much like a "human being without life."

How could a (living) human being fail to be intrigued?

The idea of building circuits that mimicked the human brain had been around since the 1940s, when Warren McCulloch and Walter Pitts had shown how neurons could be connected together to implement Boolean logic and

Tom Griffiths is the Henry R. Luce Professor of Information Technology, Consciousness, and Culture in the Departments of Psychology and Computer Science at Princeton University and Director of the Princeton Laboratory for Artificial Intelligence. His email address is tomg@princeton.edu.

Excerpted from The Laws of Thought: The Quest for a Mathematical Theory of the Mind *by Tom Griffiths. Published by Henry Holt and Company.*
Copyright ©2026 by Tom Griffiths. All rights reserved.

For permission to reprint this article, please contact: reprint-permission@ams.org.

DOI: https://doi.org/10.1090/noti3333

inspired John von Neumann to think about computers being organized like brains. Neural networks were present at the birth of cognitive science: Newell and Simon's presentation at the Symposium on Information Theory at MIT in 1956 was followed by a paper about implementing mathematical models of neurons on the IBM Type 704 Electronic Calculator, an early computer. One of the co-authors on this paper was Lois Haibt, who was the only woman whose work was presented at the symposium. She continued to play a pioneering role as the only woman on the team that developed the programming language FORTRAN. What was special about the perceptron was that it didn't just consist of circuits that were built out of neurons. It used experience to learn how those neurons should be connected to one another.

The other thing that was special about the perceptron was that its inventor, Frank Rosenblatt, wasn't shy about advertising its potential. He told the *New York Times* that it would be "the first electronic device to think as the human brain."

Unfortunately, the response to his work would reveal a significant challenge for anyone who wanted to create such a device.

Enter the Perceptron

Frank Rosenblatt was a psychologist by training, receiving both his bachelor's and his doctoral degrees in psychology from Cornell University before going on to become a research psychologist at the Cornell Aeronautical Laboratory in Buffalo. His dissertation focused on the statistical analysis of behavioral tests. But it included two unusual components for a psychology dissertation in 1956: code for calculating the relevant statistics on a digital computer, and the design for an electrical circuit he had built for his colleagues to automatically analyze behavioral data. Even though he might have started out with an interest in understanding human minds, Rosenblatt had developed a deep interest in building better computing machines.

He had the opportunity to pursue both of these interests at the Cornell Aeronautical Laboratory, where he started working on an ambitious project funded by the Navy. The effort was called Project PARA, an acronym for Perceiving And Recognizing Automaton. The first report he wrote on the project explained that "The concepts discussed had their origins in some independent research by the author in the field of physiological psychology, in which the aim has been to formulate a brain analogue useful in analysis."

Rosenblatt wanted to build an artificial brain.

Specifically, he wanted to build an artificial visual system — the part of the brain that processes the information that comes in through our eyes. He outlined a plan to build a machine that had an artificial retina, made up of a set of sensory units that would turn on in response to light in their region of an image. The sensory units were each wired up in different ways to a set of association units that summed up all their incoming signals and activated if the sum of those signals exceeded some threshold. The association units were in turn connected to *response units* that performed the same operation, summing and thresholding. These response units would classify the images presented to the retina. For example, a unit might indicate whether a shape was a square or a triangle, or whether the shape appeared on the left or the right.

Crucially, the links between the association units and the response units were allowed to have different strengths. Some units might have a weak connection, others a strong connection. The strengths of these connections (their *weights*) were real numbers intended to capture different relationships between the inputs and the responses — detecting a triangle might involve different weights than determining that it appeared on the left. And those weights were intended to be learned.

A simplified version of a perceptron is shown in Figure 1. Information flows from the retina to two association units, which connect to a single response (in this case, classifying the image as a square or a triangle). Each association unit adds up the signals from the retina and produces an activation of 1 if that sum is above its threshold, otherwise producing an activation of 0. That activation gets sent along its connection to the response unit, multiplied by the corresponding weight. The response unit adds up the activation of the association units multiplied by their weights. If the input to the response unit is greater than 0 it produces a response of 1, signaling that a square is present. Otherwise it produces a response of -1, indicating a triangle.

We can write all of this out more clearly with a little bit of math. First, we can use $\mathsf{activation}_1$ to indicate the activation of the first association unit, and likewise $\mathsf{activation}_2$ for the activation of the second association unit. The number giving the weight of the connection from the first association unit to the response is weight_{1r}, and the corresponding number for the second association unit is weight_{2r}. The input to the response unit r can then be written as

$$\mathsf{input}_r = \mathsf{activation}_1 \times \mathsf{weight}_{1r} + \mathsf{activation}_2 \times \mathsf{weight}_{2r}.$$

The resulting number is compared to 0 to determine whether the activation of the response unit is 1 or -1. For example, if the perceptron received an input that resulted in the association units having activations 1 and 0 respectively, and the corresponding weights were 0.2 and -0.3, then the input to the response unit would be $1 \times 0.2 + 0 \times -0.3 = 0.2$. Since this number is greater than

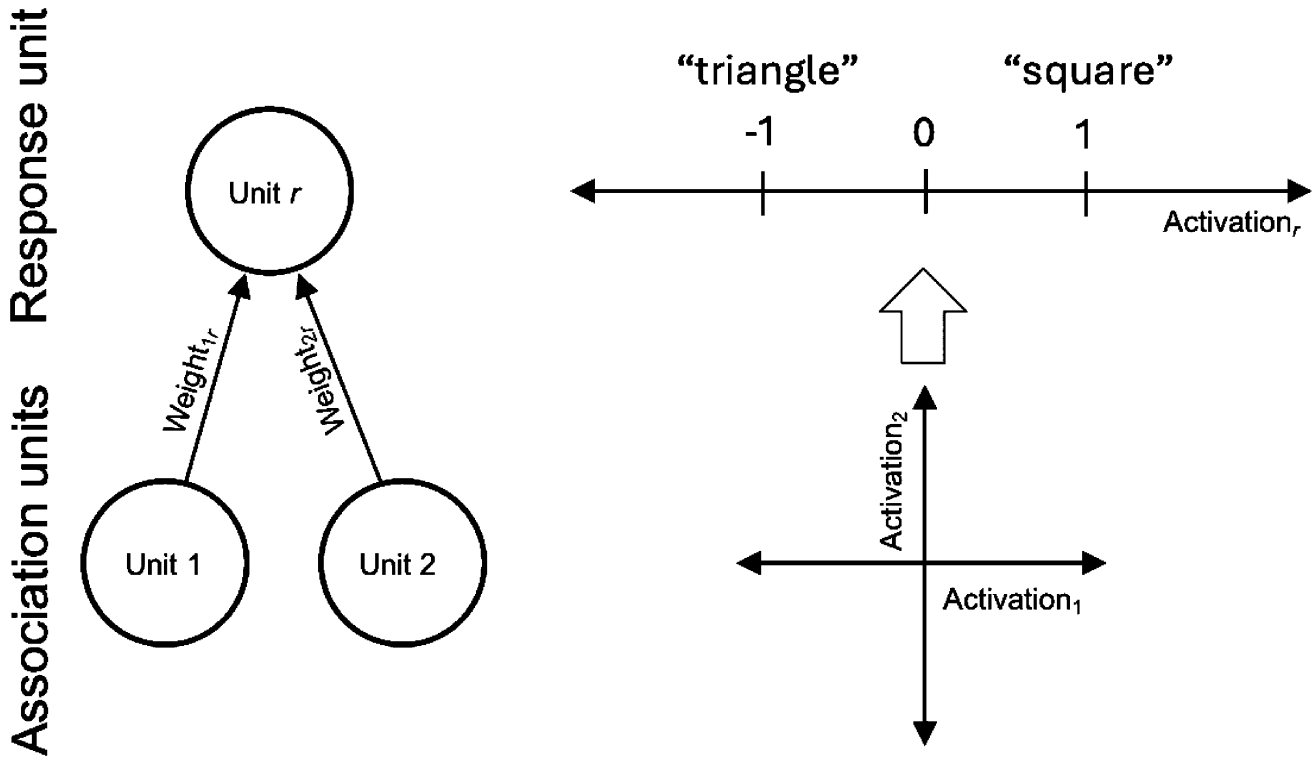

Figure 1. A simple perceptron, with just two association units and one response unit. The sensory units aren't shown, but provide the input to the association units.

0, the activation of the response unit would be 1 — the perceptron would guess that a square is present.

Even this simple neural network is a step towards computing with spaces: neural networks can be seen as a way to transform one space into another.

In our simplified perceptron, we can think about the association units as defining one space. We can define axes that correspond to $\mathsf{activation}_1$ and $\mathsf{activation}_2$, and represent the activation of the association units as a point in that space. Each pattern that hits the retina of the perceptron turns into a point in this space. Our example pattern that results in activations of 1 and 0 would correspond to the point $(1, 0)$ in this space.

The response of the perceptron defines another space — in this case, one with just a single dimension. The activation of the response determines our position along that dimension, either 1 or -1. In our example, $(1,0)$ gets mapped to the response 1.

The perceptron as a whole represents the mapping between these two spaces. Each point in association unit space gets mapped to a point in response unit space, either producing a positive response or not. The weights of the neural network determine what that mapping is — which points in association unit space end up generating a positive response. For instance, if the weights used in our example weren't 0.2 and -0.3, but instead -0.1 and 0.4, the input to the response unit would be $1\times-0.1+0\times0.4 = -0.1$, and the point $(1, 0)$ would instead get mapped to -1.

Rosenblatt's first description of the perceptron included a set of weights that would allow it to solve a simple classification problem. But his real objective was for the perceptron to be able to learn those weights. To understand how that might be possible, he turned to the latest ideas in neuroscience.

Firing and Wiring

In the 1950s, neuroscientists were just starting to figure out how learning takes place in the brain. Instead of the clear answers that Rosenblatt might have hoped for, the leading experts mostly had hypotheses about how learning might work. One of the most provocative hypotheses was offered by Donald Hebb, a psychologist at McGill University.

Hebb had a deep interest in understanding human learning. He was born in Saint John, New Brunswick, in Canada in 1904, the child of two doctors (his mother was the third woman to earn a medical degree in Nova Scotia). He was home-schooled until the age of 8, at which point he entered the local school where he jumped from 2nd to 4th grade in his first year and to 6th in his second. Unfortunately this acceleration backfired, as it made it hard for Hebb to maintain strong social relationships. As he remembered it, "On entrance to Grade IX at the age of twelve

— much the youngest pupil there — what was important to me was the respect of my peers, not my ability to solve algebra problems, nor my extensive extracurricular reading, nor my ability to outspell the rest of the school, including the teachers." Bored and unmotivated, he ended up failing 11th grade and was left with a poor impression of the value of school.

It might thus come as a surprise that, after successfully graduating and earning a bachelors degree from Dalhousie University, he became the principal of the school in the village where he had grown up. Unfortunately it didn't go very well. When the Provincial Inspector assessed the school later that year, he told the brand-new principal "Never mind, Hebb, I wasn't worth a damn my first year either."

Hebb left the school, worked as a teamster in the wheatfields, got a job as a laborer in Quebec, and began reading Freud. That started him down the slippery slope to becoming a psychologist, beginning a graduate degree at McGill while teaching in an elementary school in Montreal.

The next few years gave Hebb the opportunity to develop both an applied and a theoretical understanding of learning. In the elementary school, he motivated students to study by presenting the opportunity to engage in schoolwork as a reward for good behavior and punishing unruly students by sending them to play outside. At McGill, he had hands-on experience with Pavlovian conditioning, running experiments in which dogs were trained to salivate in response to particular cues. At the end of this process he wrote a Masters thesis in which he theorized about how the links our minds form between cues and outcomes might be represented in the brain:

> An excited neuron tends to decrease its discharge to inactive neurons, and to increase this discharge to any active neuron, and therefore to form a route to it, whether there are intervening neurons between the two or not. With repetition this tendency is prepotent in the formation of neural routes.

This simple idea — that neurons might form stronger connections to other neurons that are active at the same time — is what Hebb would become famous for 17 years later.

In the intervening time, Hebb moved to Chicago to do his doctorate, then to Harvard when his advisor Karl Lashley moved, then to a job at the Montreal Neurological Institute studying how damage to human brains influenced people's behavior, then to study the brains of chimpanzees in Florida, and finally back to be a professor at McGill. Along the way he started to think about how brains might represent concepts. He concluded that a concept "is a group of cortical neurons exciting and reexciting each other." But how might such a group of neurons come to be connected so that such excitement and reexcitement could occur?

Hebb laid out his answer in a book called *The Organization of Behavior*, published in 1949. In the book he offered a more concise version of the idea that he had first proposed in his Masters thesis:

> When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes a 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.

That was it — a theory of learning.

The axon of a neuron is the part that leads from the body of the cell to its connections to other neurons. So Hebb's theory said that as long as two neurons are close enough that they can affect one another, when both neurons are active at the same time ("firing") the connection between those two neurons increases in strength. The neuroscientist Carla Shatz reduced this to a pithy motto: "neurons that fire together, wire together."

As complicated as it might sound, this account of learning is pretty intuitive. If we forget for a moment about neurons and just think about forming associations between events, it makes perfect sense. We associate sunlight with warmth because they repeatedly and persistently co-occur, and just like the dogs in the Pavlovian conditioning experiments, the smell of our favorite food makes us salivate because it usually co-occurs with eating. Hebb was taking this principle of association and extending it to the level of individual neurons.

Hebb's theory had two important virtues. First, it turned out to basically be correct. As neuroscientists dug deeper into the mechanisms behind learning, they discovered that a process very much like the one that Hebb described occurs in the brain. But second, and potentially more importantly for our purposes, it can be turned into a clear piece of mathematics.

Going back to our simplified perceptron, Hebb's theory suggests a rule that can be used for changing the weights: increase each weight by the product of the activations of the units that it connects. If both units are firing, with an activation of 1, then the weight will increase. If the association unit is not firing, with an activation of 0, then the weight won't change. And if the association unit is firing but the response unit is giving a value of -1, then the weight will decrease. In this way, association units that tend to activate in the presence of positive responses will end up with higher weights, and association units that

tend to coincide with negative responses will end up with lower (or negative) weights.

Again, we can write this out with a little bit of math. Using activation_r to indicate the activation of the response unit, we have

$$\text{change in } \text{weight}_{ir} = \text{activation}_r \times \text{activation}_i,$$

where activation_i and weight_i are the activation and weight of the ith association unit respectively (ie., the first unit if $i = 1$, or the second if $i = 2$). In practice, we might want to be able to control the size of the change in the weights, which is done by including a *learning rate* in the equation. This gives

$$\text{change in } \text{weight}_{ir} = \text{learning rate} \times \text{activation}_r \times \text{activation}_i,$$

which is known as the Hebbian learning rule, in honor of Donald Hebb.

Hebbian learning is a way to capture the correlations that appear in data. Because the weights are increased when two units have positive activations and decreased when one is positive and the other is negative, they will track the extent to which the things represented by those units co-occur. In the case of our simplified perceptron, Hebbian learning would capture the correlation between the activation of an association unit and the response. Or, in a case where we know what the target response *should* be, we can use a modified version of the learning rule

$$\text{change in } \text{weight}_{ir} = \text{learning rate} \times \text{target}_r \times \text{activation}_i,$$

where target_r is the value that the response unit should be producing. This will allow the weights to capture the correlation between the activation of the association units and the target value for the response.

Hebbian learning has been used in a variety of neural network models, particularly models that explore how sets of neurons can store and retrieve memories by forming appropriate associations. However, it didn't satisfy Rosenblatt, who visited Hebb's lab when he was a graduate student. Hebb had succeeded in describing one kind of learning — Pavlovian conditioning, where two things become associated with one another through repeated co-occurrence. But Rosenblatt was interested in a different kind of learning — learning from our mistakes. The person training the perceptron could tell when it made a mistake, and it seemed like those mistakes were particularly important for finding the right weights.

Rosenblatt needed to come up with a new kind of learning rule.

Learning from Mistakes

Rosenblatt decided to explore a simple idea: that the weights of the perceptron would only be updated when it made a mistake. That is, the weight between an association unit and a response unit would only change if the response unit produced the wrong answer. For example, if that response unit was supposed to indicate whether a square or a triangle was present, incorrect answers would be producing a 1 when there is a triangle (which should be a -1) or a -1 in the presence of a square (which should be a 1). When a mistake occurred, the weights of any active association units would be adjusted in the direction of the correct answer: if the answer should be 1 and the response unit produced -1, then increasing the weights from the active association units will help push the response unit in the right direction; likewise, if the answer should be -1 and the response unit produced 1, then decreasing the weights will help.

Again, this learning rule is pretty intuitive: discovering that we are wrong is a strong motivator for learning. If you see something you thought was a cat and find out that it's actually a dog, that might make you change the importance you assign to different features of cats and dogs to distinguish between them. On the other hand, confirming that it is a cat seems unlikely to change your representation of cats significantly. Rosenblatt's learning algorithm — called the perceptron learning rule — captured the importance of errors for making us attend to the information we need to form a response.

We can also write the perceptron learning rule with some simple math. Here it is, using the same terms we used for the Hebbian learning rule:

$$\text{change in } \text{weight}_{ir} = \text{learning rate} \times (\text{target}_r - \text{activation}_r) \times \text{activation}_i.$$

You can check that this learning rule only updates the weights when the perceptron makes a mistake: if the response is the same as the target, then $\text{target}_r - \text{activation}_r$ will be zero. It also changes the weights in the right direction: if the target is 1 and the response is -1 the weights will increase, and if the target is -1 and the response is 1 the weights will decrease.

Comparing the equation for the perceptron learning rule with that for the Hebbian learning rule reveals something interesting. Remember, the Hebbian learning rule changes weights based on the correlation in the activation of the units. The perceptron learning rule can also be viewed as changing weights based on a correlation — the correlation between the error in the response ($\text{target}_r - \text{activation}_r$) and the activation of the association units (activation_i). This was Rosenblatt's key innovation: that

learning wasn't driven by co-occurrence, but by the degree to which a unit was responsible for the perceptron making a mistake. In this way, learning could be more efficient, focusing on the changes that were most important for reducing errors.

Equipped with a powerful learning rule, Rosenblatt was able to prove a surprising theorem: the perceptron could learn any relationship between inputs and outputs that it was able to represent. So, if there were values for the weights that would result in the perceptron producing the correct responses for a set of examples, applying the learning rule to those examples sufficiently often would result in a set of weights that produced those correct responses.

It was this property of the perceptron that gave Rosenblatt such confidence in his declarations about the power of the artificial brain he had created. He was convinced that it would be possible to wire up the connections from the sensors to the association units so that his learning rule could learn to produce appropriate responses for any meaningful task. The demonstration in 1958 that was reported on in *The New York Times* was based on a computer simulation of the learning rule, but Rosenblatt and his colleagues subsequently built a physical device that implemented the perceptron learning algorithm. The Mark I Perceptron had a retina consisting of 20×20 photocells (meaning it could process images with up to 400 pixels), a plugboard that could be used to configure the connections between these sensors and the association units, and a total of 512 stepping motors moving potentiometers that adjusted the resistance along electrical connections between the association units and up to eight response units.

If they missed the article that appeared in the *Times* about Rosenblatt's demonstration, subscribers to *The New Yorker* were treated to an interview with him that appeared later that year. The interview appeared in the Talk of the Town section under the playful title "Rival," on the grounds that "it strikes us as the first serious rival to the human brain ever devised, and our own brain is thoroughly dazzled by the things it's said to do." The reporter asked Rosenblatt whether there were any limitations to the perceptron — what wasn't it able to do? Rosenblatt threw up his hands and offered a simple response: "Love. Hope. Despair. Human nature, in short. If we don't understand the human sex drive, why should we expect a machine to?" However, Rosenblatt's own rivals thought the perceptron might face a more fundamental limitation. . . .

More about the limits of perceptrons, and how those limits were eventually overcome, are recounted in the new book *The Laws of Thought: The Quest for a Mathematical Theory of the Mind* [1].

References

[1] Tom Griffiths, *The Laws of Thought: The Quest for a Mathematical Theory of the Mind*, Henry Holt and Company, 2026.

Tom Griffiths

Credits

Figure 1 is courtesy of Tom Griffiths.
Author photo is courtesy of Sameer Khan.