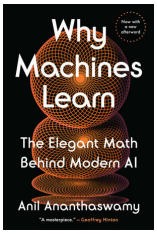
Review of [1]

Why Machines Learn

The Elegant Math Behind Modern AI

Anil Ananthaswamy



Penguin Random House, 2025
496 pages, $22 Paperback

Review by

Nicholas Tran
Department of Mathematics and Computer Science
Santa Clara University

1 Summary

This highly readable book traces the evolution of machine learning over the past seventy years by examining its key ideas, people, and mathematics. The reader is treated to a *Quanta Magazine*–style college course on the subject that assumes no background in advanced mathematics.

The title of the book comes from a story in the *New York Times* reporting on Frank Rosenblatt's invention of the perceptron in 1958, which ushered in the age of machines that self-adjust, or learn, as they process data. When asked to explain why the perceptron learned, the inventor demurred, saying that he could only do so in highly technical terms. Author Anil Ananthaswamy ably closes this gap by explaining the fundamental mathematical concepts from linear algebra, calculus, probability and statistics, and optimization behind the important advances in the field.

The first two chapters explain the mathematics behind the perceptron, starting with the neurode, a computational model of the biological neuron by Warren McCulloch and Walter Pitts, who showed that networks of these building blocks can simulate Turing machines. A perceptron is a special type of these networks that consists of a single layer of input neurodes whose discrete outputs, each with its own adjustable weight, are summed together to produce a yes/no answer based on the sign of the sum. The network makes multiple passes over a preclassified data set, adjusting its weights if it makes a mistake on a data point, until it learns the correct classification perfectly. A network of this type is characterized by a hyperplane whose normal vector is given by the weights. Rosenblatt devised a method for updating the weights (normal vector) that is guaranteed to learn the classification after a finite number of steps for any linearly separable data set, i.e., one for which there exists a hyperplane separating the "yes" data points from the "no" data points. The concepts of vectors, matrices, dot products, and normals are gently introduced and illustrated with two-dimensional examples, culminating in Rosenblatt's algorithm and a proof of its convergence.

Soon after the introduction of the perceptron, Bernard Widrow and Ted Hoff introduced a different type of single-layer neural network with continuous output that uses the now-ubiquitous method of gradient descent to improve its weights. Their network, called ADALINE, learns to classify a linearly separable data set using a stochastic estimate of the gradient of the mean squared

error between the predicted and actual output values. Chapter 3 illustrates the method of gradient descent using quadratic curves and surfaces and presents Widrow and Hoff's ADALINE training algorithm.

The excitement that drove research on neural networks at the beginning of the 1960s eventually gave way to the realization of serious limitations in their power. In 1969, Marvin Minsky and Seymour Papert published a mathematical study of perceptron-like networks which, among other things, showed that they cannot learn the XOR function, which is not linearly separable. Researchers began exploring alternative approaches to learning data sets whose classification requires boundaries more complex than hyperplanes.

In the probabilistic setting, there exists a provably best classifier that has the lowest error rate among all classifiers on average. This theoretical Bayes optimal classifier is often impractical or even impossible to compute. Surprisingly, Thomas Cover and Peter Hart proved in 1967 that it can be asymptotically approximated within a factor of 2 by a very simple algorithm called the k-nearest-neighbor (kNN) algorithm: a new data point is assigned to the most popular group among its k nearest neighbors, where k is a predefined value. Chapter 4 gives a crash course in Bayesian statistics and shows how to apply it to the famous Monty Hall problem and some real-world classification problems. Chapter 5 explains the kNN algorithm and points out its Achilles' heel: as the number of dimensions of the data set increases, most points are far away from any given point, and hence the underlying assumption of the kNN algorithm that distance correlates with similarity is no longer valid.

One approach to mitigating this "curse of dimensionality" is to consolidate dimensions that highly correlate with one another, transforming the original data set into one having fewer dimensions. Chapter 6 explains how these so-called principal components can be obtained by computing the eigenvectors and eigenvalues of the covariance matrix of the original data set.

Another strategy for classifying non-linearly separable data involves mapping data points into higher-dimensional spaces where linear separation becomes possible. In these spaces, the optimal separating hyperplane is computed using Lagrange multipliers, as shown by Vladimir Vapnik. However, calculations in high dimensions can be expensive. Chapter 7 covers the mathematics behind support vector machines (SVMs) and the associated kernel trick, developed by Bernhard Boser, Isabelle Guyon, and Vapnik. The central insights are that the optimal hyperplane depends solely on the dot products of a subset of the data points called support vectors, and that there exist kernel functions that yield the same dot product between two original data points as the dot product between those points when lifted to higher dimensions.

Neural network research experienced a renaissance in the 1980s. John Hopfield showed in 1982 how to store information in a neural network whose neurons are all connected to one another and whose weights between two neurons are the same in each direction. Regardless of the starting state, a Hopfield network dynamically adjusts itself until reaching an equilibrium, which is one of the stored memories. Chapter 8 explains Hopfield networks and provides a proof that when perturbed, they will eventually reach a stable state. George Cybenko proved in 1989 that all continuous functions can be approximated with a neural network having an additional (hidden) layer between the inputs and output. Here the output of a neurode is no longer a binary value but the value of the sigmoid function, $\sigma(x) = 1/(1 + e^{-x})$, which is continuous and ranges between 0 and 1. Chapter 9 gives a proof sketch of this theorem. In 1986, David Rumelhart, Geoffrey Hinton, and Ronald Williams published the backpropagation algorithm (independently discovered several times in other settings)that allows multilayer neural networks using gradient descent to update

their weights. This is done by computing the gradient backward using the chain rule. An example of the propagation calculation appears at the end of Chapter 10.

In 1989, Yann LeCun developed LeNet, a multilayer convolutional neural network that learns images in a spatially invariant manner similar to the way information is processed hierarchically in the visual cortex. Chapter 11 explains the evolution of LeNet from its predecessor, the neocognitron, to its successor, AlexNet.

The last chapter discusses phenomena in neural networks that cannot yet be explained by current theories, followed by an epilogue and an afterword explaining the latest revolution in AI: large language models and their underlying engine, transformers.

2 Opinion

This book will appeal to a wide audience, from general readers interested in AI to students and practitioners of machine learning. The author's engaging style and lucid explanations make the substantial mathematical concepts accessible to those with only a high school mathematics background. The book is filled with historical anecdotes and profiles of key figures in the field, adding depth and context to the technical content. The inclusion of proofs and algorithmic details provides a solid foundation for readers who wish to delve deeper into the subject.

A few minor blemishes may detract from the reading experience for some. The book adopts a somewhat idiosyncratic mathematical notation such as *x1*, w_1*32*, and *yhat* instead of x_1, $w_1^{3,2}$ and $\hat{y}$. Some explanations of technical terms like "stochastic" in the context of gradient descent are somewhat imprecise. Finally, the book glosses over the deep philosophical differences between the symbolic AI approach and the connectionist approach embodied by neural networks, giving the impression that the dispute was mainly about funding.

Overall, *Why Machines Learn* is a gem of an introduction to the still-unfolding story of the AI revolution. It is a must-read for anyone interested in exploring the field of machine learning.