

- » Looking at machine learning over time
- » Exploring machine learning frameworks

7\UdhYfs1

Introducing Machine Learning with TensorFlow

TensorFlow is Google's powerful framework for developing applications that perform machine learning. Much of this book delves into the gritty details of coding TensorFlow modules, but this chapter provides a gentle introduction. I provide an overview of the subject and then discuss the developments that led to the creation of TensorFlow and similar machine learning frameworks.

Understanding Machine Learning

Like most normal, well-adjusted people, I consider *The Terminator* to be one of the best movies ever made. In the movie, a robot from the future is sent back in time to kill a woman who will give birth to a leader of the resistance. The robot, called the Terminator, is a highly advanced machine that can think and feel. In the movie, the Terminator is a cold, calculating machine that kills without mercy. However, in the book, the Terminator is a more complex character. She is a woman who calls her mother and thinks she's having a warm conversation, but she's really talking to an evil robot from the future!

The robot wasn't programmed in advance with the mother's voice or the right way to analyze the voice of the real mother, examine the rules of English grammar, and generate acceptable sentences for the conversation. When a computer obtains information from data without receiving precise instructions, it's performing *machine learning*.

The Terminator is a classic example of machine learning. As I write this book, machine learning is everywhere. My email provider knows that messages involving an "online pharmacy" are spam, but messages about "cheap mescaline" are important. Google Maps always provides the best route to my local Elvis cult, and Amazon.com always knows when I need a new horse head mask. Is it magic? No, it's machine learning!

Machine learning applications achieve this power by discovering patterns in vast amounts of data. Unlike regular programs, machine learning applications deal with uncertainties and probabilities. It should come as no surprise that the process of coding a regular application. Developers need to be familiar with an entirely new set of concepts and data structures.

Thankfully, many frameworks have been developed to simplify development. At the time of this writing, the most popular is TensorFlow, an open-source toolset released by Google. In writing this book, my goal is to show you how to harness TensorFlow to develop your own machine learning applications.

Although this book doesn't cover the topic of ethics, I feel compelled to remind readers that programming evil robots is wrong. Yes, you'll impress your professor, and it will look great on a resume. But society frowns on such behavior, and your friends will shun you. Still, if you absolutely have to program an evil robot, TensorFlow is the framework to use.

The Development of Machine Learning

In my opinion, machine learning is the most exciting topic in modern software development, and TensorFlow is the best framework to use. To convince you of TensorFlow's greatness, I'd like to present some of the developments that led to related software development.

These developments expanded the breadth and capabilities of machine learning, but none of them excited the world's imagination. The problem was that computers lacked the speed and memory needed to perform real-world machine learning in a reasonable amount of time. That was about to change.

The computing revolution

During the 1980s, there were dramatic leaps in computing power. Researchers harnessed this new power to execute machine learning routines. Finally, machine learning could tackle real-world problems instead of simple proofs of concept.

By the mid-1990s, it got automatic. Inspired by Fukushima's neocognitron, researchers focused on neural networks specially designed for image recognition, called *convolutional neural networks*. In 1998, Yann LeCun successfully demonstrated handwriting recognition with his CNN-based Ziffer-10 dataset.

But there was a problem. Researchers used similar theories in their applications, but they wrote all their code from scratch. This meant researchers couldn't reproduce the results of their peers, and they couldn't re-use one another's code. If a researcher's funding ran out, it was likely that the entire codebase would vanish.

John D. Schuchman loved the theory behind neural networks, but I found them deeply frustrating in practice. Machine learning applications require careful tuning and tweaking to get acceptable results. But each change to the code required a new training run, and training a CNN could take days. Even then, I still didn't have enough training data to ensure accurate recognition.

One problem facing me and other researchers was that, while machine learning theory was mature, the process of software development was still in its infancy. Programmers needed frameworks and standard libraries so that they weren't reinventing the wheel. Machine learning still required faster processors that could access larger amounts of data.

The rise of big data and deep learning

By the early 2000s, the cost of data storage plummeted. Large corporations could now access terabytes of data

about potential consumers. These corporations developed improved tools for analyzing their data, and this revolution in data storage and analysis has become known as the *big data revolution*.

For example, companies make more money if they know which advertisements to show to their customers. But there were no clear rules for associating customers with products.

Many corporations launched in-house research initiatives to determine how best to develop the best recommendation engine. The winner, BellKor's Pragmatic Chaos, developed the best recommendation engine. The winner, BellKor's Pragmatic Chaos, developed the best recommendation engine. The winner, BellKor's Pragmatic Chaos, developed the best recommendation engine.

AdSense used machine learning to determine which advertisements to display on its search engine. Google and Tesla demonstrated self-driving cars that used machine learning to determine which advertisements to display on its search engine.

Across the world, large organizations sat up and paid notice. Machine learning gained by applying machine learning to big data.

Researchers paid notice as well. A major priority involved distinguishing modern machine learning, with its high complexity and vast data processing, from earlier deep learning. detail regarding the technical meaning of deep learning.

Machine Learning Frameworks

One of the most important advances in practical machine learning involved the creation of frameworks. Frameworks automate many aspects of developing machine learning applications, and they allow developers to re-use code and take advantage of the power of machine learning.

Torch

Torch is a deep learning framework that is designed to be easy to use and fast. It is built on top of Lua, a lightweight programming language. Torch's computations involve multidimensional arrays called *tensors*, which can be processed with regular vector/matrix operations. Over time, Torch acquired routines for building, training, and evaluating neural networks.

Torch garnered a great deal of interest from academics and corporations like IBM and Facebook. But its adoption has been limited by its reliance on Lua as its interpreter. Torch's design is based on the idea of a *graph*, which is a data structure that represents a computation. Theano also supports *graph* operations, which makes it a language of choice in the machine learning domain.

Theano

Theano is a deep learning framework that is designed to be easy to use and fast. It is built on top of Python, a high-level programming language. Theano provides a wide range of Python routines for operating on multidimensional arrays. Unlike NumPy, Theano stores operations in a data structure called a *graph*, which it compiles into high-performance code. Theano also supports *graph* operations, which makes it a language of choice in the machine learning domain.

Theano is a deep learning framework that is designed to be easy to use and fast. It is built on top of Python, a high-level programming language. Theano provides a wide range of Python routines for operating on multidimensional arrays. Unlike NumPy, Theano stores operations in a data structure called a *graph*, which it compiles into high-performance code. Theano also supports *graph* operations, which makes it a language of choice in the machine learning domain.

7U Y

7U Y is a deep learning framework that is designed to be easy to use and fast. It is built on top of Python, a high-level programming language. 7U Y provides a wide range of Python routines for operating on multidimensional arrays. Unlike NumPy, 7U Y stores operations in a data structure called a *graph*, which it compiles into high-performance code. 7U Y also supports *graph* operations, which makes it a language of choice in the machine learning domain.

7U Y is a deep learning framework that is designed to be easy to use and fast. It is built on top of Python, a high-level programming language. 7U Y provides a wide range of Python routines for operating on multidimensional arrays. Unlike NumPy, 7U Y stores operations in a data structure called a *graph*, which it compiles into high-performance code. 7U Y also supports *graph* operations, which makes it a language of choice in the machine learning domain.

Keras

§ 1.1.1. Keras is a high-level neural networks API, written in Python and capable of running on top of either TensorFlow or Theano. It was developed by François Chollet, who is concerned with modularity and simplicity of development. François Chollet created Keras as an interface to other machine learning frameworks, and many developers access Theano through Keras to combine Keras's simplicity with Theano's performance.

Keras's simplicity stems from its small API and intuitive set of functions. These functions focus on accomplishing standard tasks in machine learning, which makes it easy to use. However, Keras also allows you to want to customize their operations.

François Chollet released Keras under the MIT License, and Google has incorporated his interface into TensorFlow. For this reason, many TensorFlow developers prefer to code their neural networks using Keras.

TensorFlow

As the title implies, this book centers on TensorFlow, Google's gift to the world of machine learning. TensorFlow is a software library for machine learning, written in C++ for improved performance. Like Theano, TensorFlow stores operations in a graph that can be deployed to a GPU, a remote system, or a network of remote systems. In addition, TensorFlow provides a utility called TensorBoard, which makes visualizing graphs and their operations possible.

TensorFlow is a software library for machine learning, written in C++ for improved performance. Like Theano, TensorFlow stores operations in a graph that can be deployed to a GPU, a remote system, or a network of remote systems. In addition, TensorFlow provides a utility called TensorBoard, which makes visualizing graphs and their operations possible.

Like other frameworks, TensorFlow supports execution on CPUs and GPUs. In addition, TensorFlow applications can be executed on the Google Cloud Platform (GCP). In my opinion, GCP processing is TensorFlow's most important advantage. TensorFlow is a software library for machine learning, written in C++ for improved performance. Like Theano, TensorFlow stores operations in a graph that can be deployed to a GPU, a remote system, or a network of remote systems. In addition, TensorFlow provides a utility called TensorBoard, which makes visualizing graphs and their operations possible.

