Deep learning frameworks

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:00

You've learned to implement deep learning algorithms more or less from scratch using Python and NumPY. And I'm glad you did that because I wanted you to understand what these deep learning algorithms are really doing. But you find unless you implement more complex models, such as convolutional neural networks or recurring neural networks, or as you start to implement very large models that is increasingly not practical, at least for most people, is not practical to implement everything yourself from scratch. Fortunately, there are now many good deep learning software frameworks that can help you implement these models. To make an analogy, I think that hopefully you understand how to do a matrix multiplication and you should be able to implement how to code, to multiply two matrices yourself. But as you build very large applications, you'll probably not want to implement your own matrix multiplication function but instead you want to call a numerical linear algebra library that could do it more efficiently for you. But this still helps that you understand how multiplying two matrices work. So I think deep learning has now matured to that point where it's actually more practical you'll be more efficient doing some things with some of the deep learning frameworks. So let's take a look at the frameworks out there. Today, there are many deep learning frameworks that makes it easy for you to implement neural networks, and here are some of the leading ones. Each of these frameworks has a dedicated user and developer community and I think each of these frameworks is a credible choice for some subset of applications. There are lot of people writing articles comparing these deep learning frameworks and how well these deep learning frameworks changes. And because these frameworks are often evolving and getting better month to month, I'll leave you to do a few internet searches yourself, if you want to see the arguments on the pros and cons of some of these frameworks. But I think many of these frameworks are evolving and getting better very rapidly. So rather than too strongly endorsing any of these frameworks I want to share with you the criteria I would recommend you use to choose frameworks. One important criteria is the ease of programming, and that

means both developing the neural network and iterating on it as well as deploying it for production, for actual use, by thousands or millions or maybe hundreds of millions of users, depending on what you're trying to do. A second important criteria is running speeds, especially training on large data sets, some frameworks will let you run and train your neural network more efficiently than others. And then, one criteria that people don't often talk about but I think is important is whether or not the framework is truly open. And for a framework to be truly open, it needs not only to be open source but I think it needs good governance as well. Unfortunately, in the software industry some companies have a history of open sourcing software but maintaining single corporation control of the software. And then over some number of years, as people start to use the software, some companies have a history of gradually closing off what was open source, or perhaps moving functionality into their own proprietary cloud services. So one thing I pay a bit of attention to is how much you trust that the framework will remain open source for a long time rather than just being under the control of a single company, which for whatever reason may choose to close it off in the future even if the software is currently released under open source. But at least in the short term depending on your preferences of language, whether you prefer Python or Java or C++ or something else, and depending on what application you're working on, whether this can be division or natural language processing or online advertising or something else, I think multiple of these frameworks could be a good choice. So that said on programming frameworks by providing a higher level of abstraction than just a numerical linear algebra library, any of these program frameworks can make you more efficient as you develop machine learning applications.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt
Lecture Notes	pptx