

Lecture note 1: Introduction to TensorFlow

CS 20: TensorFlow for Deep Learning Research (cs20.stanford.edu)

Prepared by Chip Huyen (chiphuyen@cs.stanford.edu)

Reviewed by Danijar Hafner, Jon Gautier, Minh-Thang Luong, Paul Warren

Why TensorFlow

TensorFlow was originally created by researchers at Google as a single infrastructure for machine learning in both production and research. Later, an implementation of it was open sourced under the Apache 2.0 License in November 2015. On the Tensorflow website, we see:

“TensorFlow™ is an open source software library for numerical computation using data flow graphs.”

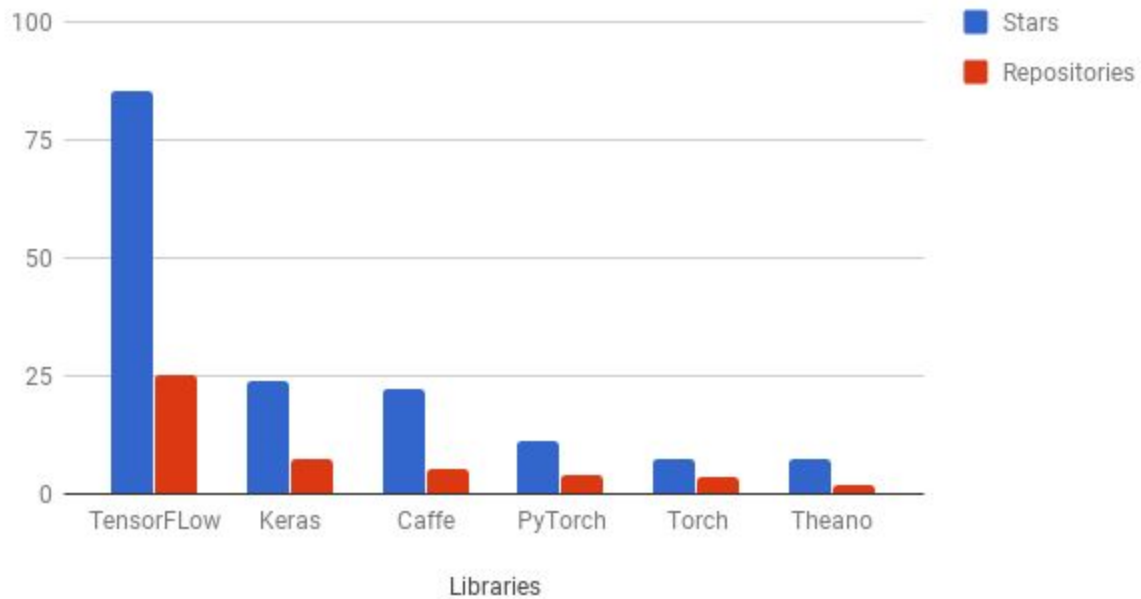
Note that only the implementation of TensorFlow that we see on GitHub is open-source. Google maintains its own internal version. It's said that Google did it because of the complicated relationships TensorFlow has with its other internal tools, not because Google is “hoarding good stuff”. For the rest of this course, when we say TensorFlow, we are referring to the open source implementation.

The next key phrase we see is that TensorFlow is a “software library for Machine Intelligence”. In the past year, it seems like every week, a company or another released their own deep learning library. For a non-exhaustive list of current deep learning libraries, please visit [this link](#).

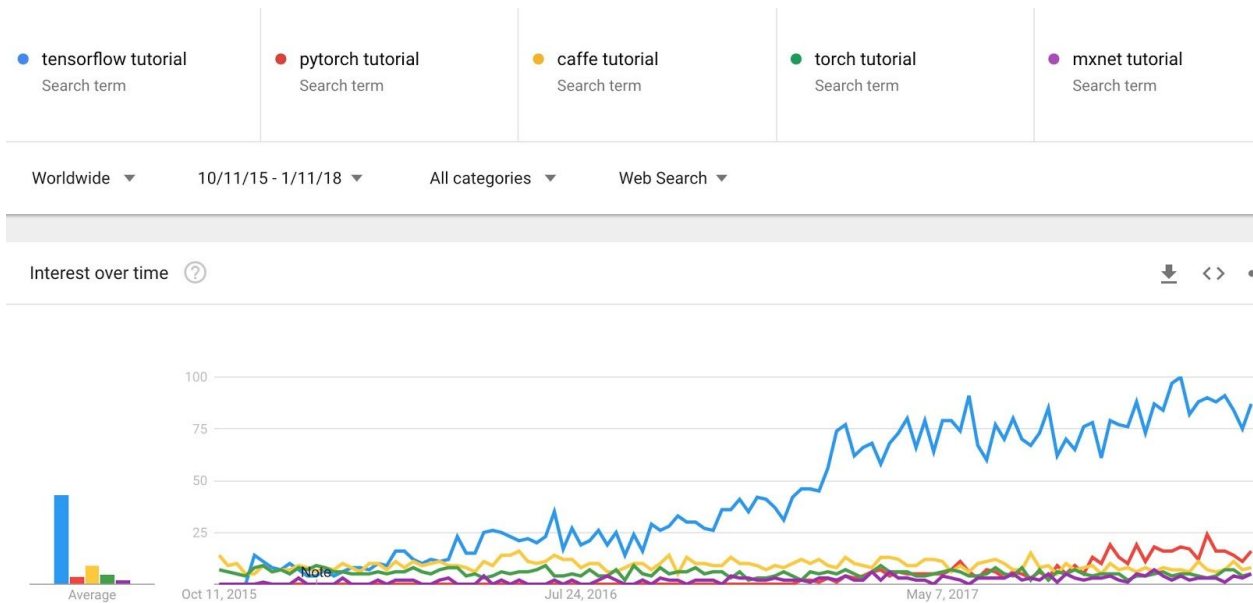
Given the plethora of these libraries, why did we choose Tensorflow to teach in this class? For a framework to be useful in production, it needs to be efficient, scalable, and maintainable. For research, the framework needs to have flexible operations that can be combined in novel ways. Alternative frameworks are either flexible enough for research but less scalable, such as Chainer and PyTorch, or scalable but less flexible, such as Caffe and MXNet. TensorFlow is both flexible and scalable, allowing users to streamline from research into production.

This unique position allowed TensorFlow to grow quickly. It's currently being used by big companies such as Google, OpenAI, NVIDIA, Intel, SAP, eBay, Airbus, Uber, Airbnb, Snap, Dropbox and startups alike. By the number of stars and related repositories on GitHub as of Jan 11, 2018, TensorFlow is by far the most popular machine learning library with more than 85.4k stars and 25.3k related repositories, twice as much as the total stars and related repositories of Caffe, PyTorch, Torch, and Theano combined. It's said that the rise of TensorFlow is the reason why the support for Theano was discontinued in September 2017.

Stars and Repositories



Demand for TensorFlow learning materials also surpasses that of any other framework.



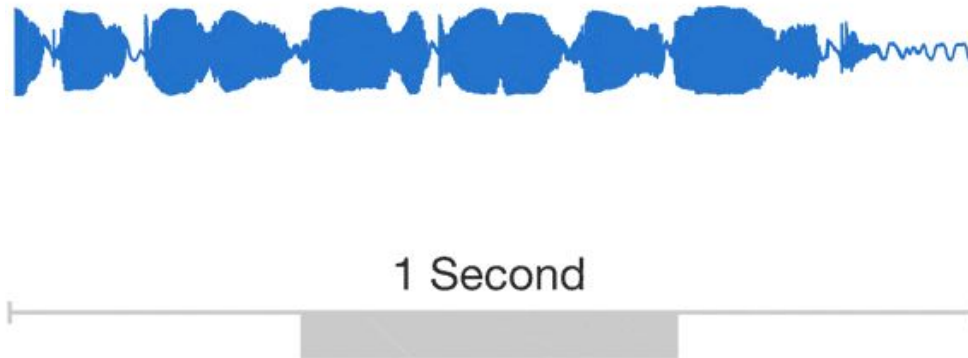
In summary, we chose TensorFlow because:

- Python API
- Portability: deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API

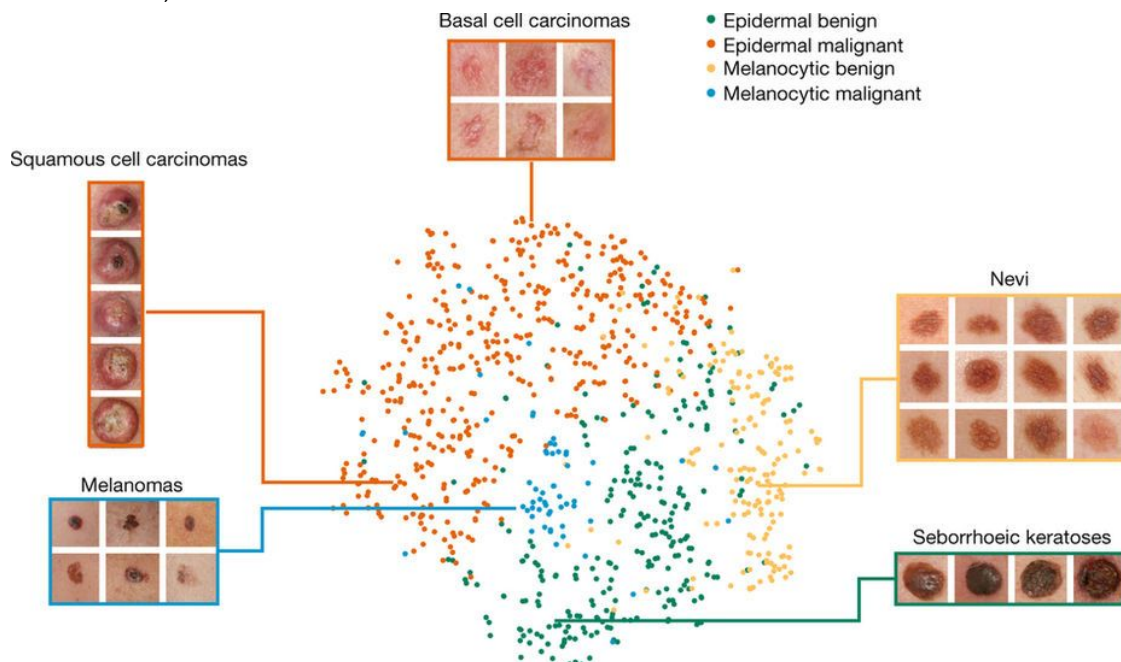
- Flexibility: from Raspberry Pi, Android, Windows, iOS, Linux to server farms
- Visualization (TensorBoard is da bomb)
- Save and restore models, graphs
- Auto-differentiation autodiff (no more taking derivatives by hand. Yay)
- Large community (~300k commits, ~85k repositories)
- Awesome projects already using TensorFlow

Some cool projects using Tensorflow

1. [WaveNet: A Generative Model for Raw Audio](#) (DeepMind, 2016)



2. [Dermatologist-level classification of skin cancer with deep neural networks](#) (Esteva, Kuprel, et al., Nature 2017)



3. [Magenta](#) (Google)

Use machine learning to create compelling art and music. Their projects are really fun! For example, please check out [Draw Together with a Neural Network](#).



Below are some more examples of real world projects using TensorFlow, according to [Google Research Blog](#), 2016:

- Australian marine biologists are using TensorFlow to find sea cows in tens of thousands of hi-res photos to better understand their populations, which are under threat of extinction.
- An enterprising Japanese cucumber farmer trained a model with TensorFlow to sort cucumbers by size, shape, and other characteristics.
- Radiologists have adapted TensorFlow to identify signs of Parkinson's disease in medical scans.
- Data scientists in the Bay Area have rigged up TensorFlow and the Raspberry Pi to keep track of the Caltrain.

I hope that after this class, you'd be able to use TensorFlow to work on super cool projects like that!

High level APIs on top of TensorFlow

There are many high level APIs built on top of TensorFlow. Some of the most popular APIs included [Keras](#)¹, [TFLearn](#), and [Sonnet](#). These high-level APIs allow for faster experimentation -- you can call a

¹ Keras is also capable of running on top of CNTK and Theano. It has recently become part of official TensorFlow.

complex neural network models in a few lines of code. The APIs have attracted a sizeable number of users. You should definitely check them out, and we might briefly go over these high-level APIs in class if time permits.

However, the primary purpose of TensorFlow is not to provide out-of-the-box machine learning solutions. Instead, TensorFlow provides an extensive suite of functions and classes that allow users to define models from scratch. This is more complicated, but offers much more flexibility. You can build almost any architecture you can think of in TensorFlow.

Resources

We won't be using any textbook for this class. The library is changing so fast that it's hard for any book to keep up. We will be using mainly lecture notes and lecture slides. There are several resources that you might want to refer to become fluent in TensorFlow.

[The official documentations](#)

[TensorFlow official sample models](#)

[StackOverflow](#) should be your first port of call should you run into any problem with TensorFlow

There are also several introductory books on TensorFlow.

- Aurélien Géron's *Hands-On Machine Learning with Scikit-Learn and TensorFlow* (O'Reilly, March 2017)
- François Chollet's *Deep Learning with Python* (Manning Publications, November 2017)
- Nishant Shukla's *Machine Learning with TensorFlow* (Manning Publications, January 2018)
- Lieder et al.'s *Learning TensorFlow A Guide to Building Deep Learning Systems* (O'Reilly, August 2017)

TensorFlow Basics

The first thing we need to understand about TensorFlow is its computation graph approach. Any TensorFlow program consists of two phases:

Phase 1: assemble a graph

Phase 2: use a session to execute operations in the graph.

Note that this might change in the future with [TensorFlow's eager mode](#), currently experimental.

It's best to explain this with graphs. Please refer to [the slides of the first lecture \(from slide 24\)](#) to learn about TensorFlow's computation graph approach, tensors, subgraphs, and sessions.