A quick introduction to Tensorflow

Machine Learning Spring 2019
Many ML libraries:
An open-source library by Google:

**TensorFlow:**

**Large-Scale Machine Learning on Heterogeneous Distributed Systems**

(Preliminary White Paper, November 9, 2015)


Google Research*
Further reading:

• Official website: https://www.tensorflow.org/
  If want to master every details.

• Deep Learning with Python by Francois Chollet
  Focus on Keras

• Hands-On Machine Learning with Scikit-Learn and TensorFlow
  Tensorflow part is somewhat outdated. (Though this book is published in 2017).
Core Functionalities:

• Augmented tensor operations (nearly identical to numpy)
  Seamless interfaces with existing programs.

• Automatic differentiation
  The very core of Optimization based algorithms.

• Parallel(CPU/GPU/TPU) and Distributed(multi-machine) Computation
  Essential for large(industrial level) applications.
  Implemented in C++. Highly Efficient.
Automatic differentiation: Through back-propagation

• Only operations with “sub-gradient” can be applied on Tensor
Automatic differentiation: Through back-propagation

• Only operations with “sub-gradient” can be applied on Tensor Arithmetic: +, -, *, /

Elementary functions: exp, log, max, sin, tan
Automatic differentiation: Through back-propagation

• Only operations with “sub-gradient” can be applied on Tensor Arithmetic: +, -, *, /
Elementary functions: exp, log, max, sin, tan

• What operations are not “differentiable”?
Automatic differentiation: Through back-propagation

• Only operations with “sub-gradient” can be applied on Tensor Arithmetic: +, -, *, /

Elementary functions: exp, log, max, sin, tan

• What operations are not “differentiable”? For example: sampling
Working process: Tensor, flow

- Tensor: multi-dimensional array
Working process: Tensor, flow

- flow: computation graph

```
\(a\quad a = 2\)

\(b\quad b = 1\)

\(c = a + b\quad c = 3\)

\(d = b + 1\quad d = 2\)

\(e = c \times d\quad e = 6\)
```
Working process: Tensor, flow

- flow: computation graph

Can be visualize by tensorboard
Static vs Eager Mode

• Eager mode
Just like using numpy

• Static mode
Predefine tensors and computation graphs then let TF engine to execute the graphs. Similar to defining Python functions.
Static vs Eager Mode

• Eager mode
Just like using numpy

• Static mode: We focus solely on this mode in this tutorial

Subtlety appears here.
3 levels of tensorflow:

• Primitive tensorflow: lowest, finest control and most flexible
  Suitable for most machine learning and deep learning algorithms.

• Keras (Mostly for deep learning ): highest, most convenient to use, lack flexibility

• Tensorflow layers (Mostly for deep learning ): somewhere at the middle.
General pipeline:

• Define inputs and variable tensors (weights/parameters).
  *Keras will take care of these for you.

• Define computation graphs from inputs tensors to output tensors.

• Define loss function and optimizer
  Once the loss is defined, the optimizer will compute the gradient for you!

• Execute the graphs.
  *Keras will take care of this for you as well
Getting started today:

• GPU acceleration

• Installation

• Demos
  o Arithmetic and tensor operations
  o Primal SVM
  o Simple neural network in Keras.
  o Primitive and tensorflow layer if time allowed
GPU acceleration:

• Literally need one if training on non-toy models and datasets.
GPU acceleration:

- Literally need one if training on non-toy models and datasets.
- Nvidia GPUs Only
Where to find (free) computing resources:

• Your own Gaming PC

• CHPC (University), CADE (Collage of Engineering)

• AWS/Google Cloud Platform: First time coupon.

• Google colab: Always free, equipped with GPU and TPU!
Installation: Anaconda

• Installed with Anaconda could save you much work.
https://www.anaconda.com/
Installation: Anaconda

• Installed with Anaconda could save you much work.

https://www.anaconda.com/