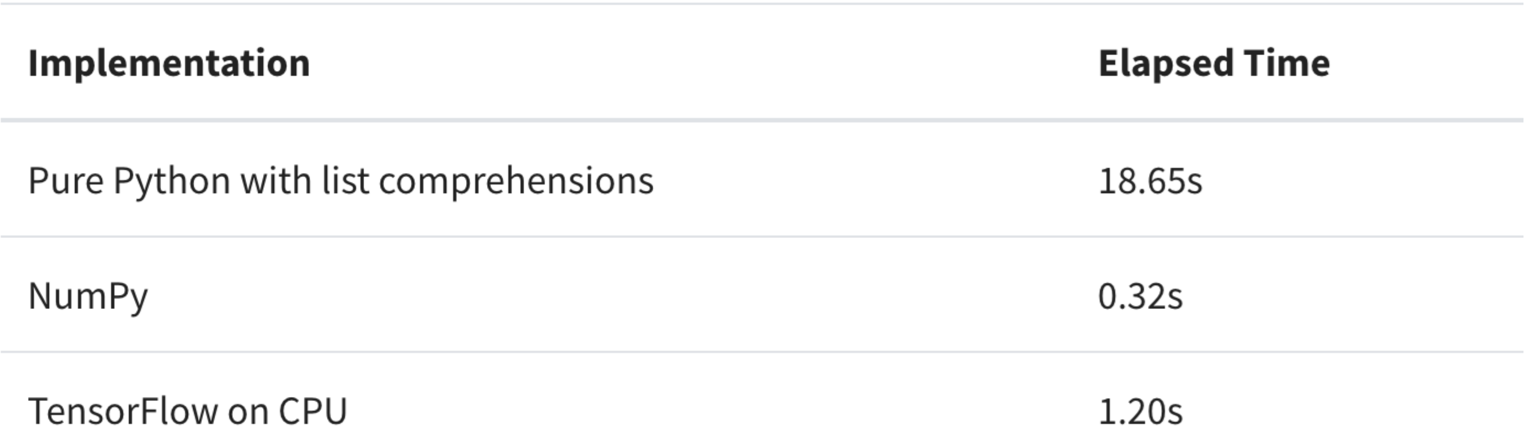
CS542 LAB 6

TensorFlow Tutorial

What is TensorFlow

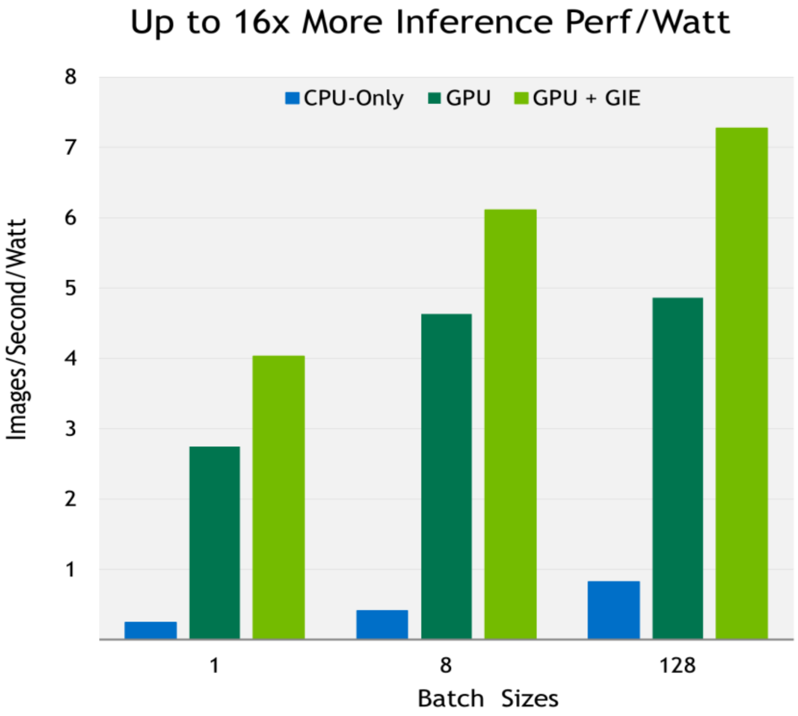
- open-source software library
- dataflow programming
- symbolic math library

Comparison Between Lists Numpy *vs.* TensorFlow



Implementation	Elapsed Time
Pure Python with list comprehensions	18.65s
NumPy	0.32s
TensorFlow on CPU	1.20s

CPU *vs.* GPU


Up to 16x More Inference Perf/Watt
🟦 CPU-Only
🟩 GPU
🟩 GPU + GIE
Images/Second/Watt
8
7
6
5
4
3
2
1
1
8
128
Batch Sizes

The Basics

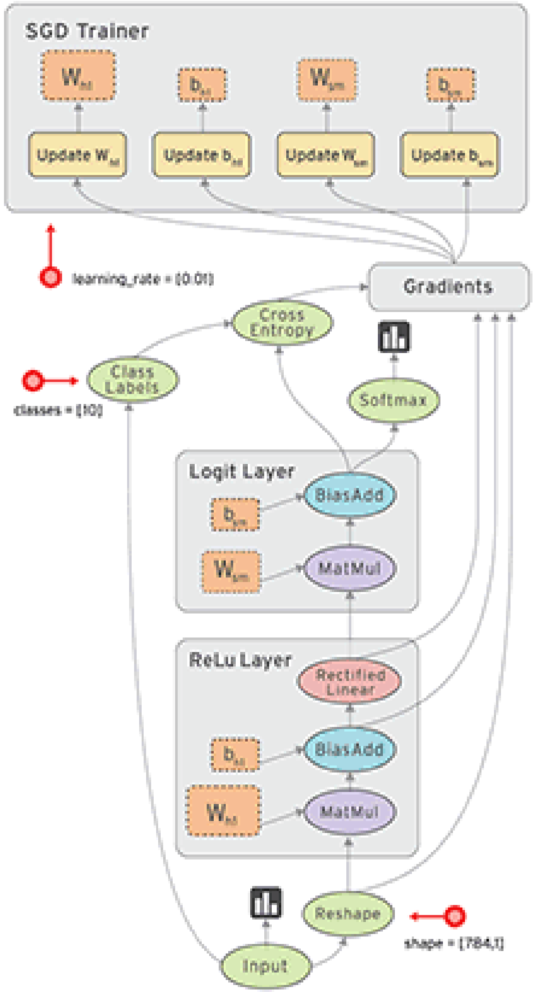
```
import tensorflow as tf
```

High Level APIs

- TF Learn (tf.contrib.learn): simplified interface that helps users transition from the the world of one-liner such as scikit-learn.
- TF Slim (tf.contrib.slim): lightweight library for defining, training and evaluating complex models in TensorFlow.
- High level API: Keras, TFLearn, Pretty Tensor

Data Flow Graphs

TensorFlow separates definition of computations from their execution.



Tensor

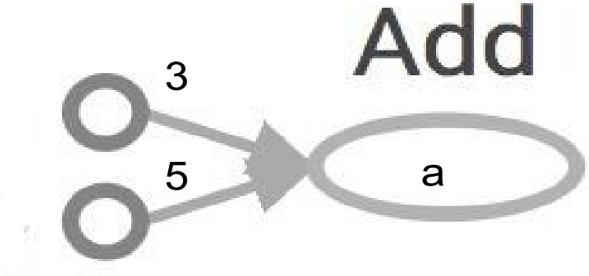
Similar to numpy arrays.

TF automatically names the nodes when you don’t explicitly name them.

Graph

```
import tensorflow as tf
a = tf.add(3, 5)
```

Constructs a TF Graph.



Why graphs

- Save computation (only run subgraphs that lead to the values you want to fetch)
- Break computation into small, differential pieces to facilitates auto-differentiation
- Facilitate distributed computation, spread the work across multiple CPUs, GPUs, or devices
- Many common machine learning models are commonly taught and visualized as directed graphs already

Installing TensorFlow

- On Windows open the Start menu and open an Anaconda Command Prompt
- On MacOS or Linux open a terminal window

```
Conda install tensorflow
```

A simple NN using tensorflow

http://cs-people.bu.edu/sbargal/cs542/beginner.ipynb

Reference

Stanford CS20SI 2017