TensorFlow Introduction

KEVIN LIANG

DUKE UNIVERSITY – ELECTRICAL AND COMPUTER ENGINEERING

26 JULY 2017


OP 1
OP 2
A
TensorFlow

TensorFlow: What is it?

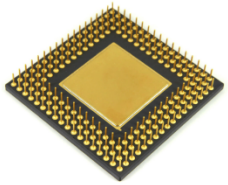
- A software library for machine learning
  - Computation using data flow graphs
- An open source successor to DistBelief
  - Apache 2.0 License
- Released by Google November 9, 2015
- For research and production
- APIs:
  - **Python**
  - C++
  - Java
  - Go
  - …and more

Java

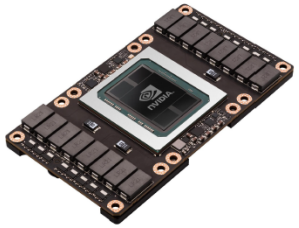
Go

TensorFlow: Platforms

TensorFlow Supports Many Platforms...

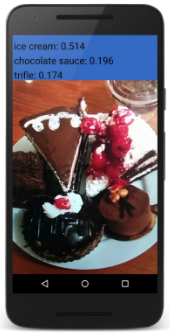


CPU



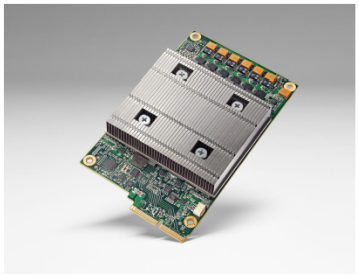
GPU

iOS



Android

Raspberry Pi



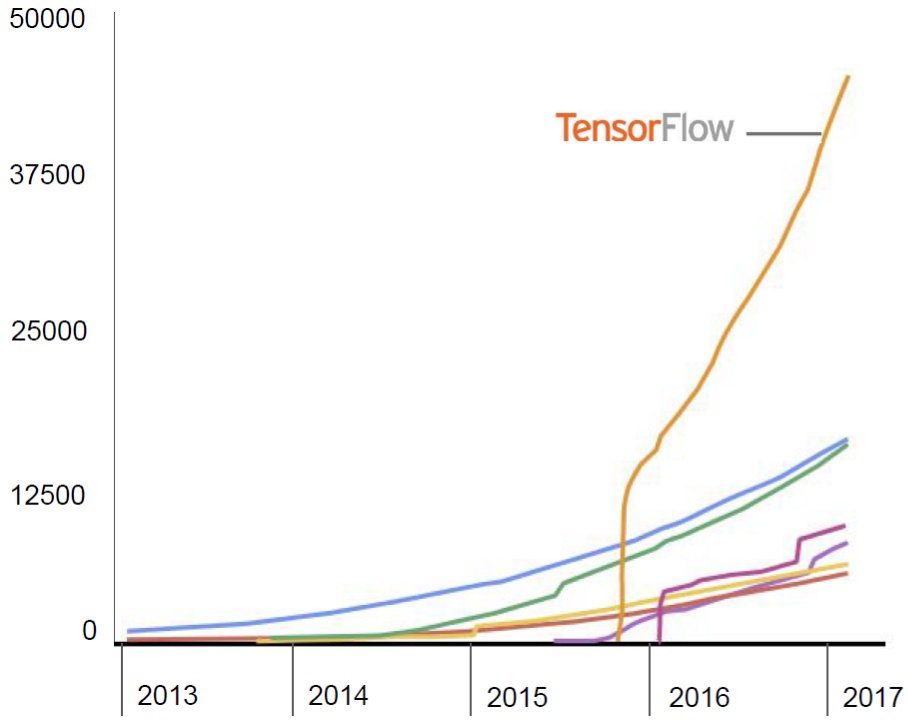
TPU

TensorFlow: Alternatives

- Caffe
  - UC Berkeley (BVLC: Berkeley Vision and Learning Center)
- Microsoft Cognitive Toolkit (CNTK 2.0)
  - Microsoft Corporation
- Theano
  - Université de Montréal (MILA/LISA: Montreal Institute for Learning Algorithms)
- Torch

- So why TensorFlow?

TensorFlow: Community



Framework	GitHub Star Count
TensorFlow	44508
scikit-learn	16191
Caffe	15690
CNTK	9383
MXNet	7896
Torch	6285
Theano	5568

TensorFlow: Community + Google

TensorFlow: A Vibrant Open-Source Community

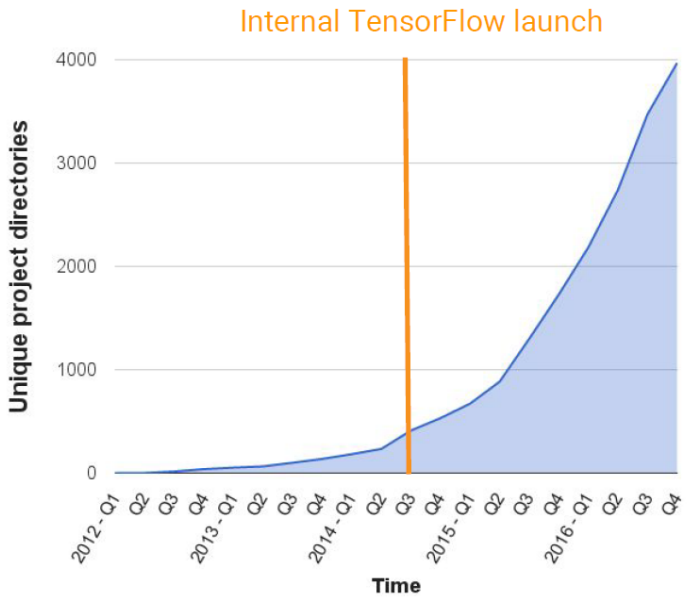
- **Rapid development, many outside contributors**
  - 475+ non-Google contributors to TensorFlow 1.0
  - 14,000+ commits in 14 months
  - Many community created tutorials, models, translations, and projects
    - ~5,500 GitHub repositories with 'TensorFlow' in the title

- **Direct engagement between community and TensorFlow team**
  - 5000+ Stack Overflow questions answered
  - 5000+ GitHub issues filed and answered; 160+ new issues / week

- **Use in ML classes is growing: Toronto, Berkeley, Stanford, ...**

TensorFlow: Within Google

of Google directories containing model description files



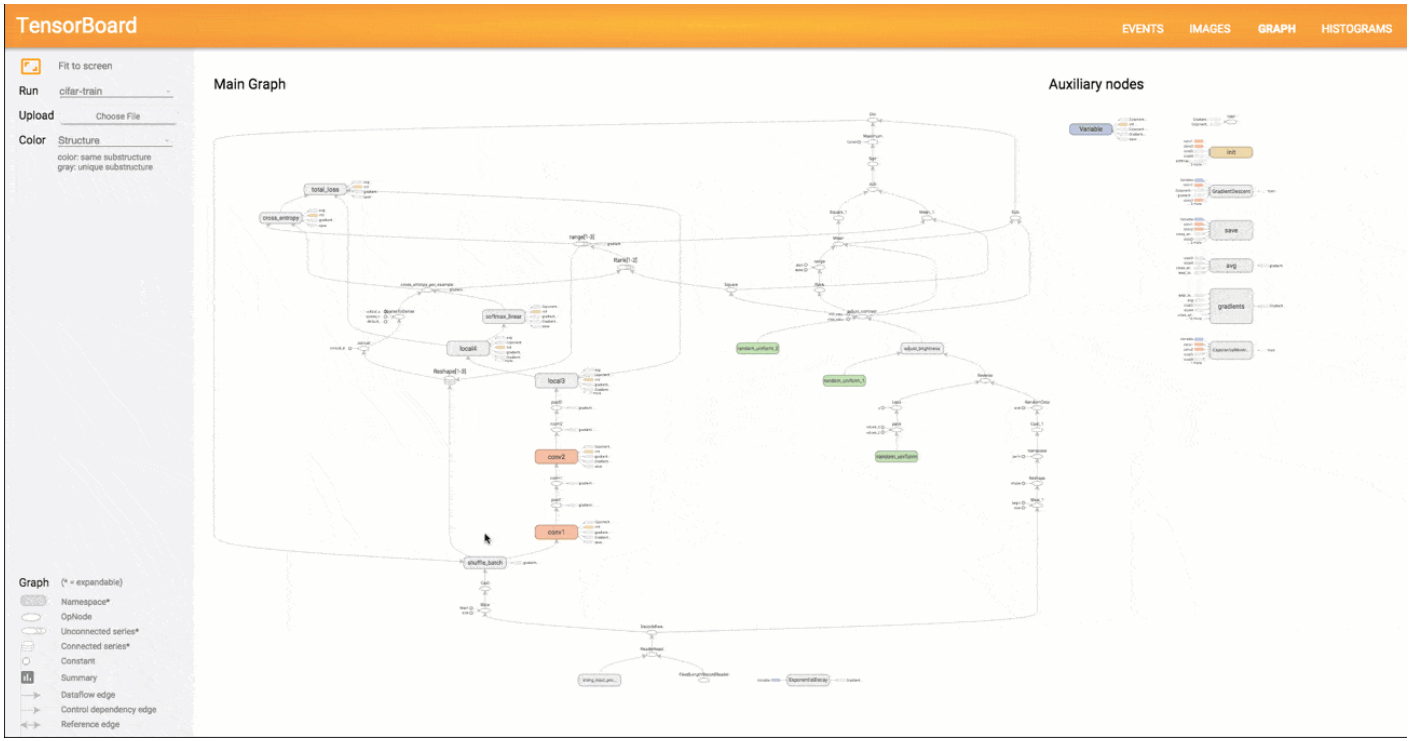
Production use in many areas:

Search
Gmail
Translate
Maps
Android
Photos
Speech
YouTube
Play
… many others …

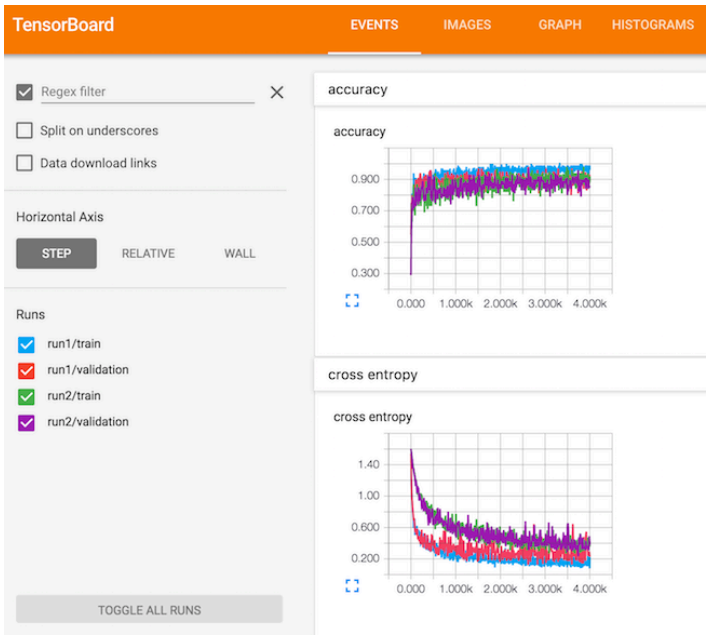
Research use for:

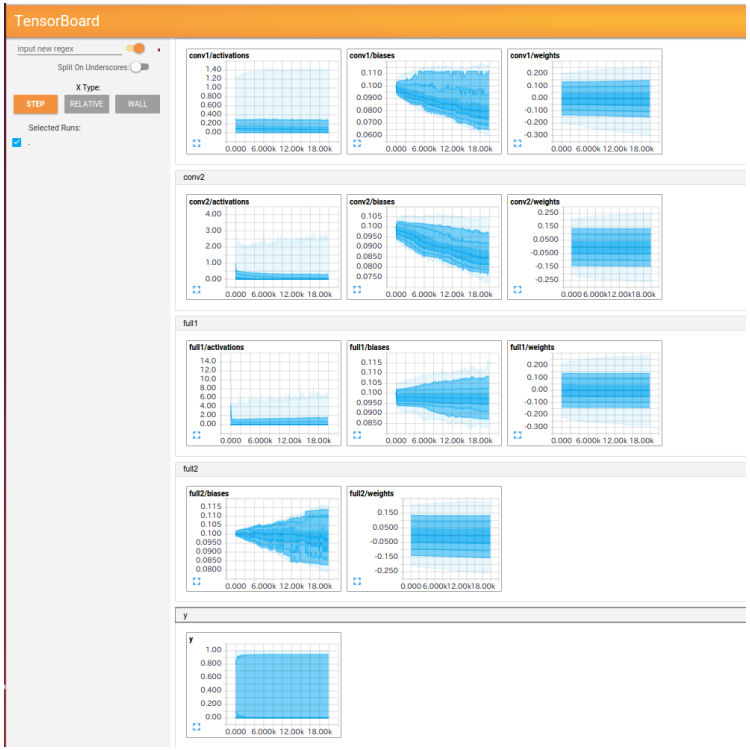
100s of projects and papers

TensorBoard: Graph Visualization

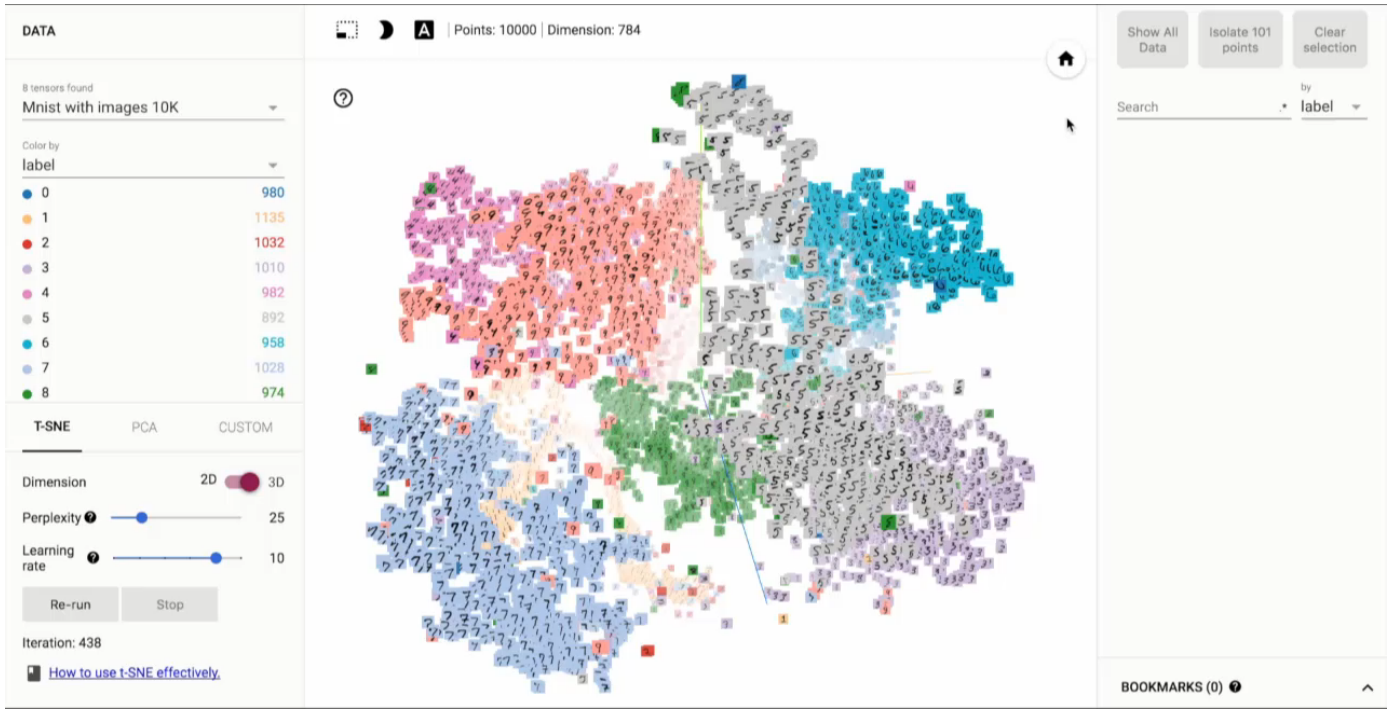


TensorBoard: Learning Visualization





TensorFlow: Embedding Visualization



TensorFlow: Why? - Summary

- Support on many platforms, including CPU/GPU/TPU; easy to scale up
- Large and active user-base
  - Academia, industry, enthusiasts
- Rapid development, research, and support by Google
- TensorBoard visualizations
- Integration with Google Cloud Platform
- Pre-trained models and high-level libraries (Slim, Keras, TFLearn)

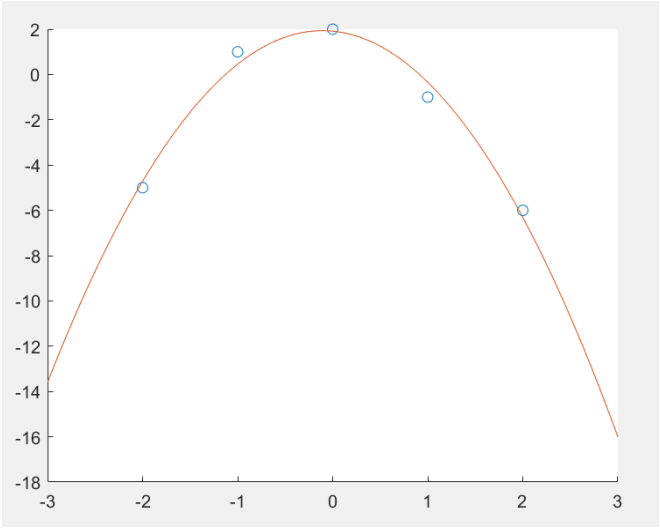
TensorFlow: Data Flow Graphs

- Typical computational program operates directly on the data:
  - Python:

```
import numpy
x = [-2, -1, 0, 1, 2]
y = [-5 , 1,  2,  -1,  -6]
p = numpy.polyfit(x, y, deg=2)

# y_hat = p[0] * x**2 + p[1] * x + p[2]
```

- Note that operations were performed on the variables holding the data itself



TensorFlow: Data Flow Graphs

- TensorFlow: 2 steps
  - Define a graph:

```
a = tf.constant(3.0, dtype=tf.float32)
b = tf.constant(4.0, dtype=tf.float32)
sum_a_b = tf.add(a, b)
```

  - Run the graph and get outputs:

```
sess = tf.Session()
print(sess.run(sum_a_b))      # Prints "7.0" to the screen
sess.close()
```

TensorFlow: Data Flow Graphs

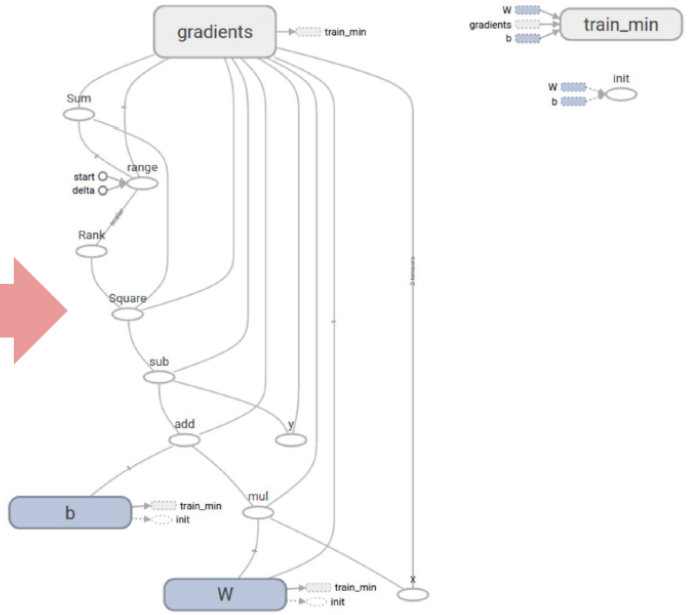
Python Program

```
import numpy as np
import tensorflow as tf

# Model parameters
W = tf.Variable([.3], tf.float32)
b = tf.Variable([-.3], tf.float32)
# Model input and output
x = tf.placeholder(tf.float32)
linear_model = W * x + b
y = tf.placeholder(tf.float32)
# loss
loss = tf.reduce_sum(tf.square(linear_model - y)) # sum of the squares
# optimizer
optimizer = tf.train.GradientDescentOptimizer(0.01)
train = optimizer.minimize(loss)
# training data
x_train = [1,2,3,4]
y_train = [0,-1,-2,-3]
# training loop
init = tf.global_variables_initializer()
sess = tf.Session()
sess.run(init) # reset values to wrong
for i in range(1000):
  sess.run(train, {x:x_train, y:y_train})

# evaluate training accuracy
curr_W, curr_b, curr_loss  = sess.run([W, b, loss], {x:x_train, y:y_train})
print("W: %s b: %s loss: %s"%(curr_W, curr_b, curr_loss))
```

TensorFlow Graph



https://www.tensorflow.org/get_started/get_started

TensorFlow: Tensors

- *Mathematics*: Geometric objects defining linear relations
  - Generalization of vectors and matrices:
    - 0th Order (Scalar): 8
    - 1st Order (Vector): [4, 2, 9]
    - 2nd Order (Matrix): [[5, 1, 9], [2, 2, 0]]
- TensorFlow: unit for data and variables
  - 0th Order: `scalar_node = tf.constant(8.0, dtype=tf.float32)`
  - 4th Order: `weights = tf.Variable(tf.random_normal([3, 3, 256, 512]), name="conv_weights")`

Next Steps

- If you haven't already, fork the MLSS TensorFlow repo and install TensorFlow!

- Check out the TensorFlow website:
  - https://www.tensorflow.org/