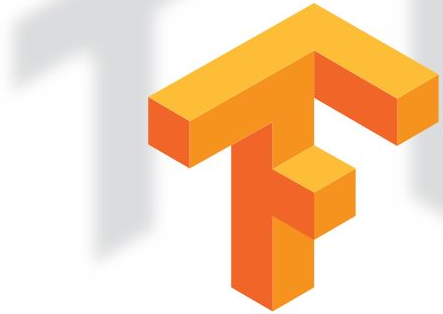

# TensorFlow Tutorial

Sheng Lundquist

# Assumptions

- You know what a single layer perceptron is
- You've seen Python code before

# Introduction

- Parameter File Based: Caffe
  - Specify high level blocks (e.g. convolutional layer, softmax regression layer) to achieve computations
- Computational Based: TensorFlow
  - Specify low level blocks (e.g. convolutional operation, matrix multiply operation) to achieve computation
- TensorFlow API in Python

# Workflow

- Build graph
- Evaluate graph

# Building Graph for SLP on MNIST

```
import tensorflow as tf
```

# Placeholders

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
```

# Variables

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
```

# Operation

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
```

# Operation

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
```

# Optimizer

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.1).minimize(loss)
```

# Initialization

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.1).minimize(loss)
sess = tf.Session()
sess.run(tf.initialize_all_variables())
```

# Evaluation

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.1).minimize(loss)
sess = tf.Session()
sess.run(tf.initialize_all_variables())
for i in range(1000):
   batch_input, batch_gt = mnist.train.next_batch(256)
   sess.run(opt, feed_dict={inImage: batch_input, gt: batch_gt})
```

# Evaluation

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.1).minimize(loss)
sess = tf.Session()
sess.run(tf.initialize_all_variables())
for i in range(1000):
   batch_input, batch_gt = mnist.train.next_batch(256)
   sess.run(opt, feed_dict={inImage: batch_input, gt: batch_gt})
   print "Loss on step", i, ":", sess.run(loss, feed_dict={inImage:batch_input, gt:batch_gt})
```

# Demo

# Benchmarks

https://github.com/soumith/convnet-benchmarks