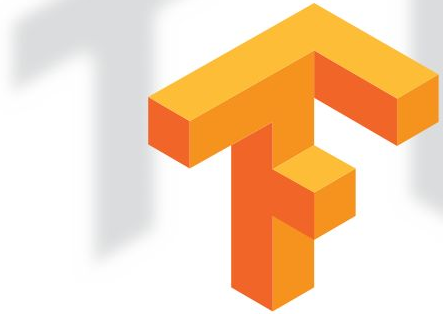

TensorFlow Tutorial

Sheng Lundquist

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

```
import tensorflow as tf
```

Placeholders

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

Variables

```
import tensorflow as tf
input = 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
input = 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(input, W) + b
```

Operation

```
import tensorflow as tf
input = 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(input, W) + b
loss = tf.reduce_sum(tf.square(gt - est))/2
```

Optimization

```
import tensorflow as tf
input = 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(input, W) + b
loss = tf.reduce_sum(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.5).minimize(loss)
```

Initialization

```
import tensorflow as tf
input = 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(input, W) + b
loss = tf.reduce_sum(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.5).minimize(loss)
init = tf.initialize_all_variables()
```

Running Graph

```
import tensorflow as tf
input = 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(input, W) + b
loss = tf.reduce_sum(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.5).minimize(loss)
init = tf.initialize_all_variables()
sess = tf.Session()
sess.run(init)
```

Evaluation

```
import tensorflow as tf
input = 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(input, W) + b
loss = tf.reduce_sum(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.5).minimize(loss)
init = tf.initialize_all_variables()
sess = tf.Session()
sess.run(init)
for i in range(1000):
  batch_input, batch_gt = mnist.train.next_batch(100)
  sess.run(opt, feed_dict={input: batch_input, gt: batch_gt})
```

Evaluation

```
import tensorflow as tf
input = 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(input, W) + b
loss = tf.reduce_sum(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.5).minimize(loss)
init = tf.initialize_all_variables()
sess = tf.Session()
sess.run(init)
for i in range(1000):
   batch_input, batch_gt = mnist.train.next_batch(100)
   sess.run(opt, feed_dict={input: batch_input, gt: batch_gt})
   print “Loss on step”, i, “:”, sess.eval(loss, feed_dict={input:batch_input, gt:batch_gt})
```