**TensorFlow的使用**

**在TensorFlow中，所有的操作只有当你执行，或者另一操作依赖与这一操作的结果，这一操作才会被执行**

1. 在创建weights和bias或其他变量是，要将初始值传入构造函数Variable(),因为训练需要update weights和bias的值。

需要指明变量的shape, 最好把name值也指定，因为tf会自动给变量命名，之后如果有重复赋值可能会报错，每个变量指定唯一name

weights = tf.Variable(tf.random\_normal([784, 200], stddev=0.35),

name="weights")

biases = tf.Variable(tf.zeros([200]), name="biases")

1. 构建完model后需要初始化全部变量，这样变量才能进行操作在Session中初始化这个操作

# Add an op to initialize the variables.

init\_op = tf.initialize\_all\_variables()

with tf.Session() as sess:

# Run the init operation.

sess.run(init\_op)

1. 关于变量由另一个变量初始化，使用变量的initialized\_value()来初始化

weights = tf.Variable(tf.random\_normal([784, 200], stddev=0.35),

name="weights")

w2 = tf.Variable(weights.initialized\_value(), name="w2")

w\_twice = tf.Variable(weights.initialized\_value() \* 0.2, name="w\_twice")

1. 变量的保存与加载(model也一样)

Saver = tf.train.Saver()

With tf.Session() as sess:

Sess.run(init\_op)

Saver.save(sess, “/tem/model.ckpt”)

# 加载

Saver.restore(sess, “/tem/model.ckpt”)

**TensorBoard的使用**

对模型进行可视化操作，还可以记录一些比如learning rate或者loss的变化等

Merge = tf.merge\_all\_summaries()

Summary\_writer = tf.train.SummaryWriter(“/tmp/mnist\_logs”, sess.graph)

Total\_step=0

While training:

Total\_step += 1

Session.run(training\_op)

If total\_step % 100 == 0:

Summary\_str = session.run(Merge)

Summary\_writer.add\_summary(summary\_str, total\_step)