# 参考程序片段

## MNIST\_Monitor metrics visualization.

主要用于Tensorboard对程序中变量和指标进行监控

tf.summary.histogram记录取值分布信息

tf.summary.scalar记录标量取值信息

编写函数：variable\_summaries

1）计算变量的分布2）计算平均值，计算标准差，并使用标量形式展示

def variable\_summaries(var, name):  
 # 将生成监控信息的操作放在同一个命名空间下  
 with tf.name\_scope('summaries'):  
 # 通过tf.summary.histogram函数记录张量中元素的取值分布  
 tf.summary.histogram(name, var)  
  
 # 计算变量的平均值，并定义生成平均值信息日志的操作，记录变量平均值信息的日志标签名  
 # 在相同命名空间中的监控指标会被整合到同一栏中，name则给出了当前监控指标属于哪一个变量  
 mean = tf.reduce\_mean(var)  
 tf.summary.scalar('mean/' + name, mean)  
 # 计算变量的标准差，并定义生成其日志文件的操作  
 stddev = tf.sqrt(tf.reduce\_mean(tf.square(var - mean)))  
 tf.summary.scalar('stddev/' + name, stddev)

对每层神经元weights和biases使用variable\_summaries函数统计分布和平均值标准差信息。

并使用tf.summary.histogram统计分布信息

# 将输入变量还原成图片的像素矩阵，并通过tf.summary.image函数定义将当前的图片信息写入日志的操作

with tf.name\_scope('input'):

x = tf.placeholder(tf.float32, [None, 784], name='x-input')

y\_ = tf.placeholder(tf.float32, [None, 10], name='y-input')

with tf.name\_scope('input\_reshape'):

image\_shaped\_input = tf.reshape(x, [-1, 28, 28, 1])

tf.summary.image('input', image\_shaped\_input, 10)

# 将输入变量还原成图片的像素矩阵，并通过tf.summary.image函数定义将当前的图片信息写入日志的操作

准确度

with tf.name\_scope('accuracy'):  
 with tf.name\_scope('correct\_prediction'):  
 correct\_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y\_, 1))  
 with tf.name\_scope('accuracy'):  
 accuracy = tf.reduce\_mean(tf.cast(correct\_prediction, tf.float32))  
 tf.summary.scalar('accuracy', accuracy)

执行所有日志生成操作并写入文件

merged = tf.summary.merge\_all()

# 初始化写日志的writer,并将当前的Tensorflow计算图写入日志  
summary\_writer = tf.summary.FileWriter(SUMMARY\_DIR, sess.graph)

summary, \_ = sess.run([merged, train\_step], feed\_dict={x: xs, y\_: ys})  
 # 将得到的所有日志写入日志文件，这样TensorBoard程序就可以拿到这次运行所对应的  
 # 运行信息。  
 summary\_writer.add\_summary(summary, i)  
  
summary\_writer.close()