# 基本操作

import tensorflow as tf

# 创建一个常量op

# 一行两列

m1 = tf.constant([[3, 3]])

# 两行一列

m2 = tf.constant([[2], [3]])

# 创建一个矩阵乘法op，将m1和m2传入

product = tf.matmul(m1, m2)

print(product)

# 定义一个回话，启动默认的图

# with tf.Session() as sess

sess = tf.Session()

# 调用sess的run方法来执行矩阵乘法op

# run方法触发了图中的三个op

result = sess.run(product)

print(result)

sess.close()

创建op（operation）之后不会立即执行，打印product为Tensor("MatMul\_6:0", shape=(1, 1), dtype=int32)，需要通过会话去执行一个图。

## 变量

tensorflow中定义变量用tensorflow.Variable(‘数据’)来定义。

tensorflow中的变量必须要初始化，并且要在回话中运行该初始化。

import tensorflow as tf

x = tf.Variable([1, 2], name = ‘x’)

a = tf.Variable([3, 3], name = ‘a’)

# 减法op

sub = tf.subtract(x, a)

# 加法op

add = tf.add(x, sub)

# tensorflow中的变量必须要初始化

# 初始化所有的变量

init = tf.global\_variables\_initializer()

with tf.Session() as sess:

# 运行初始化

sess.run(init)

print(sess.run(sub))

print(sess.run(add))

### 变量赋值

变量赋值使用函数update = tensorflow.assign(dst, value)

dst为需要赋值的变量，value为值。tensorflow.assign为一个赋值op，也是需要通过调用会话的run方法实际的去执行赋值操作。

# 创建一个变量，初始化为0

state = tf.Variable(0, name='counter')

# 创建一个op，作用是使state加一

new\_value = tf.add(state, 1)

# 赋值op

update = tf.assign(state, new\_value)

init = tf.global\_variables\_initializer()

with tf.Session() as sess:

sess.run(init)

print('init:', sess.run(state))

for \_ in range(5):

# update 调用赋值操作，将新的值赋给state

sess.run(update)

print(sess.run(state))