# TensorFlow学习

## TensorFlow的执行时的基本对象及其具体理解：

分为Tensor、Variable和operation三大类，他们共同构成执行时的graph，执行可以执行graph中的一个子节点，子节点依照DAG顺序执行期上游全部父节点。

示例：下面讲解概念时有具体示例

## 常量操作和变量、另外也理解一下执行时的基本对象

**import** tensorflow **as** tf

m1 **=** tf.constant([[3, 3]]) *# 常量operation，生成一个Tensor*

m2 **=** tf.constant([[1], [2]]) *# 常量operation，生成一个Tensor*

product **=** tf.matmul(m1, m2) *# 加法operation，输入两个Tensor，生成一个Tensor*

​

**with** tf.Session() **as** session: *# 利用Session执行上述代码中生成的DAG*

res **=** session.run(product)

print(res)

[[9]]

In [2]:

**import** tensorflow **as** tf

state **=** tf.Variable([0, 0], "counter") *# 变量*

new\_value **=** tf.add(state, [1, 2]) *# 加法operation 生成一个Tensor*

upstate **=** tf.assign(state, new\_value) *# 赋值operation 生成一个Tensor*

​

init **=** tf.global\_variables\_initializer() *# 变量初始化*

​

**with** tf.Session() **as** session:

session.run(init) *# 变量初始化*

**for** \_ **in** range(10):

session.run(upstate) *# 执行更新赋值也就是“+=”10次*

print(session.run(state)) *# 查看每次执行的效果*

[1 2]

[2 4]

[3 6]

[4 8]

[ 5 10]

[ 6 12]

[ 7 14]

[ 8 16]

[ 9 18]

[10 20]

## Feed和fetch

*# fetch*

**import** tensorflow **as** tf

m1 **=** tf.constant([[3, 3]]) *# 常量operation，生成一个Tensor*

m2 **=** tf.constant([[1], [2]]) *# 常量operation，生成一个Tensor*

m3 **=** tf.constant([3])

product **=** tf.matmul(m1, m2) *# 加法operation，输入两个Tensor，生成一个Tensor*

add **=** tf.add(product, m3)

​

**with** tf.Session() **as** session: *# 利用Session执行上述代码中生成的DAG*

res **=** session.run([product, add]) *# 同一个run中运行多个operation，operation之间可以有依赖关系*

print(res)

[array([[9]]), array([[12]])]

[array([[9]]), array([[12]])]

*# feed*

**import** tensorflow **as** tf

input1 **=** tf.placeholder(tf.float32)

input2 **=** tf.placeholder(tf.float32)

add **=** tf.add(input1, input2)

​

**with** tf.Session() **as** session: *# 利用Session执行上述代码中生成的DAG*

res **=** session.run(add, feed\_dict **=** {input1: [32.1], input2: 78.2}) *# 同一个run中运行多个operation，operation之间可以有依赖关系*

print(res)

[110.299995]