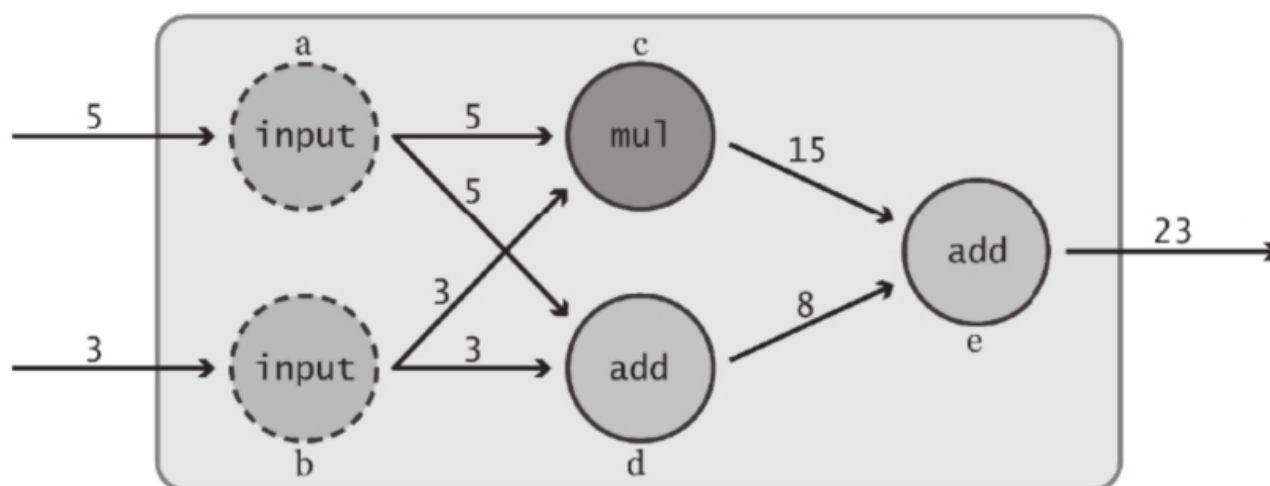


## Lab4 构建TensorFlow数据流图

### 1、实验简介

编码实现第一个TensorFlow数据流图，运行它的各个组件，并初步了解TensorBoard。了解TensorFlow计算模型-计算图，TensorFlow数据模型-张量，以及TensorFlow运行-会话的初步概念。

数据流图有两个基本构件：节点和边。节点代表了对数据所做的运算或某种操作；边对应于向节点传入和传出的实际数据，通常以箭头表示。



上图表示：

$a = \text{input}; b = \text{input}$

$c = a * b; d = a + b$

$e = c + d$

若  $a = 5, b = 3$ , 则  $e = 23$

定义该数据流图的TensorFlow代码：

```
import tensorflow as tf
a = tf.constant(5, name="input_a")
b = tf.constant(3, name="input_b")
c = tf.multiply(a, b, name="mul_c")
d = tf.add(a, b, name="add_d")
e = tf.add(c, d, name="add_e")
```

### 2、实验步骤

下面逐行解析定义数据流图和运行数据流图两个步骤：

#### 2.1 定义数据流图

## Fundamentals of Machine Learning

```
# 首先加载TensorFlow库，并赋予别名tf
import tensorflow as tf
# 给变量a,b赋值
a = tf.constant(5, name="input_a")
b = tf.constant(3, name="input_b")
# 执行操作
c = tf.mul(a,b, name="mul_c")
d = tf.add(a,b, name="add_d")
# 定义数据流图的终点e
e = tf.add(c,d, name="add_e")
```

### 2.2 运行数据流图

```
# 创建TensorFlow Session对象
sess = tf.Session()
# 运行Session对象
sess.run(e)

# 节点e的输出保存在output变量
output = sess.run(e)
print(output)
# 利用TensorBoard可视化，在当前路径下生成目录my_graph:
writer = tf.summary.FileWriter("./my_graph" , sess.graph)
```

在cmd命令行窗口中，cd进入生成my\_graph文件夹的上一级文件夹中，键入下列命令：

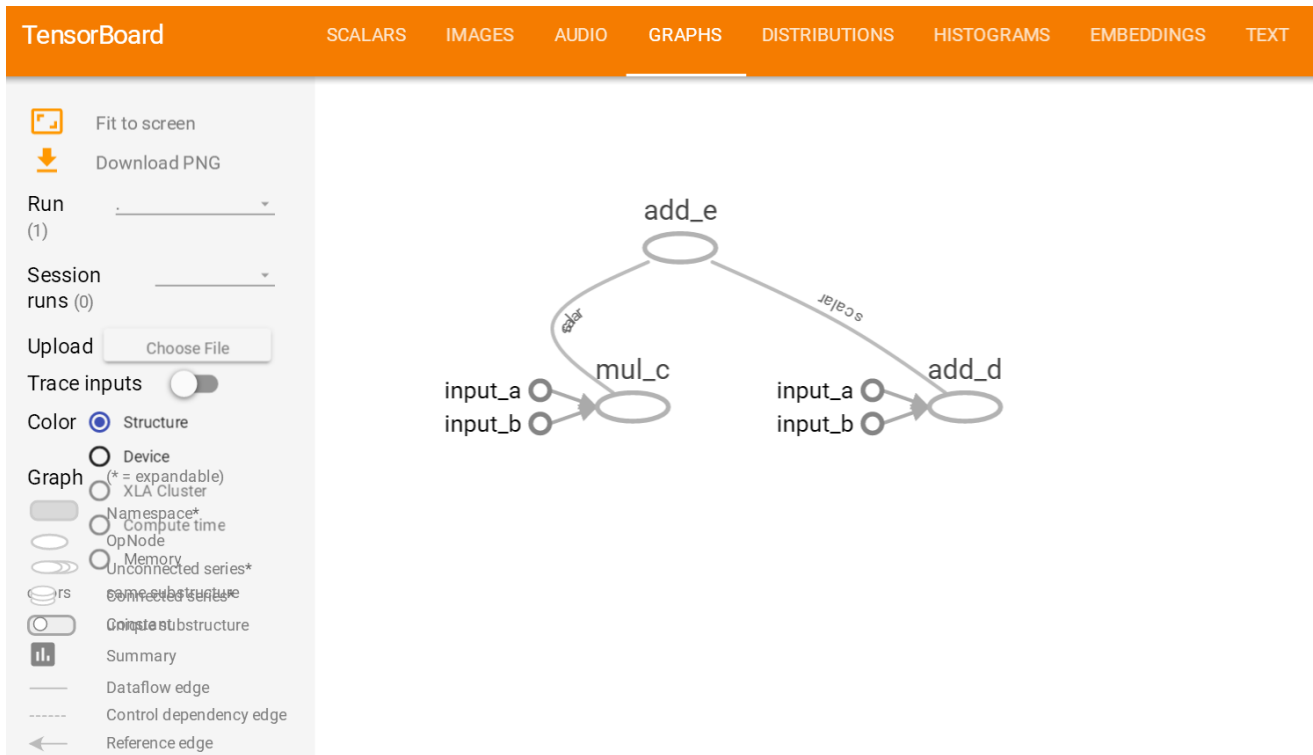
```
>tensorboard --logdir=my_graph
```

默认情况下，TensorBoard服务器启动后会自动监听端口6006

```
D:\notebooks\tensorflow>tensorboard --logdir=my_graph
Starting TensorBoard b'54' at http://ChloeZ:6006
(Press CTRL+C to quit)
```

打开浏览器并在地址栏输入<http://localhost:6006>，（若页面显示不出请换浏览器尝试，本人在搜狗、chrome及火狐浏览器中均可以正常显示）单击页面顶部的“graph”链接，将看到类似下面的页面。

# Fundamentals of Machine Learning



完成数据流图的构造之后，需要关闭Session对象和FileWriter对象，以释放资源并进行一些清理工作：

```
writer.close( )  
sess.close( )
```

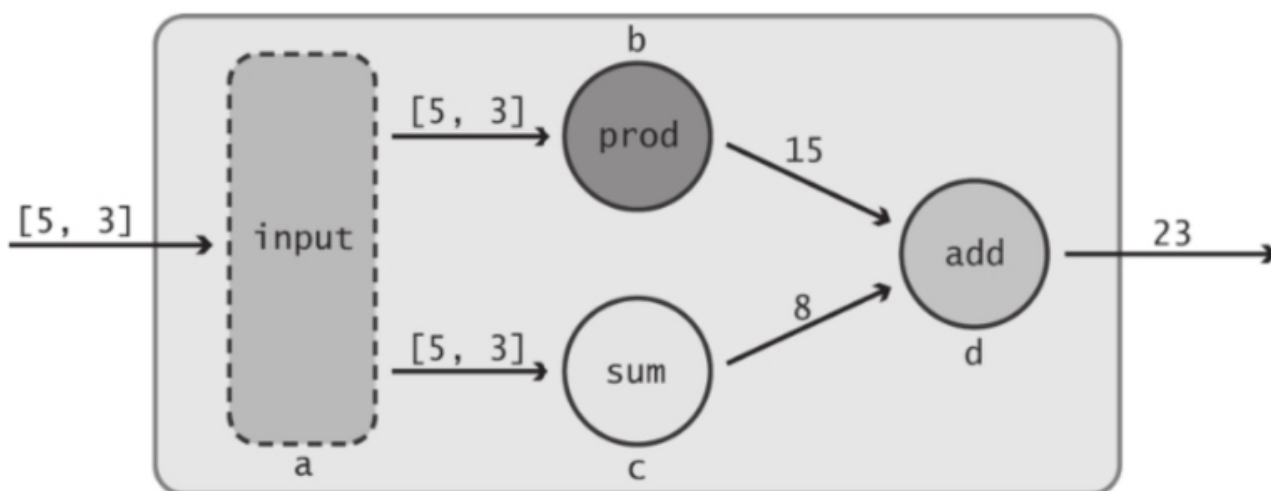
完整代码如下：

```
import tensorflow as tf  
a = tf.constant(5, name="input_a")  
b = tf.constant(3, name="input_b")  
c = tf.multiply(a,b, name="mul_c")  
d = tf.add(a,b, name="add_d")  
e = tf.add(c,d, name="add_e")  
sess = tf.Session()  
output = sess.run(e)  
print(output)  
writer = tf.summary.FileWriter("./my_graph ", sess.graph)  
writer.close()  
sess.close()
```

## 2.3 TensorFlow数据模型-张量

张量，即n维矩阵的抽象。1D张量等价于向量，2D张量等价于矩阵，对于更高维的张量，可称为“N维张量”或“N阶张量”。用张量的概念，对之前的数据流图进行修改如下：

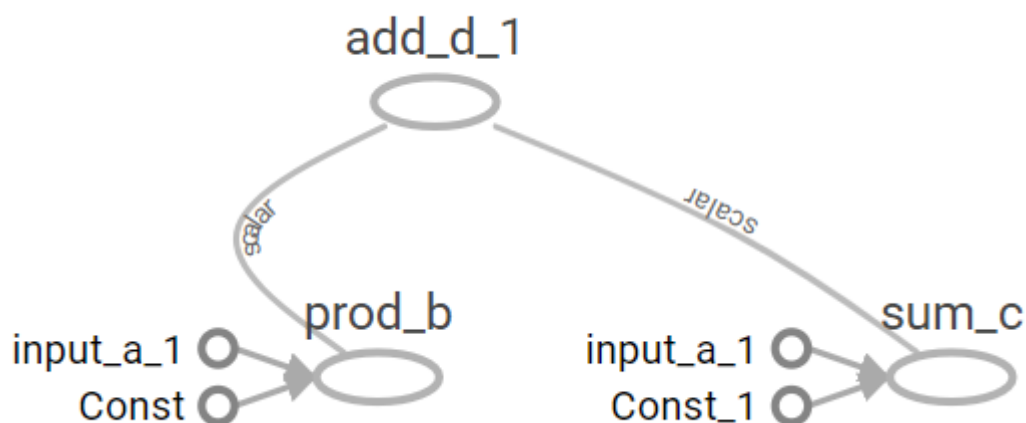
## Fundamentals of Machine Learning



对应上图数据流图的TensorFlow代码:

```
import tensorflow as tf
a = tf.constant([5,3], name="input_a")
b = tf.reduce_prod(a, name="prod_b")
c = tf.reduce_sum(a, name="sum_c")
d = tf.add(b, c, name="add_d")
sess = tf.Session()
output = sess.run(d)
print(output)
```

Tensorboard中显示:



TensorFlow中，所有节点之间传递的数据都为Tensor对象。TensorFlow Op自动接收标准Python数据类型，如整数或字符串，并将其自动转化为张量。手工创建Tensor对象或张量有多种方式。可将张量视为以一种结构化格式保存任意数据的方式

参考Tensorflow中文社区: <http://www.tensorfly.cn/>

### 3、练习如下操作

```
import tensorflow as tf
a = tf.constant([[1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5]])
```

## Fundamentals of Machine Learning

```
a1 = tf.reshape(a, [-1, 3]) # 形状是5,3
```

```
a2 = tf.reshape(a, [3, -1]) # 形状是3,5
```

```
print(a1)
```

```
print(a2)
```