Tensorflow安装与入门教程

目 录

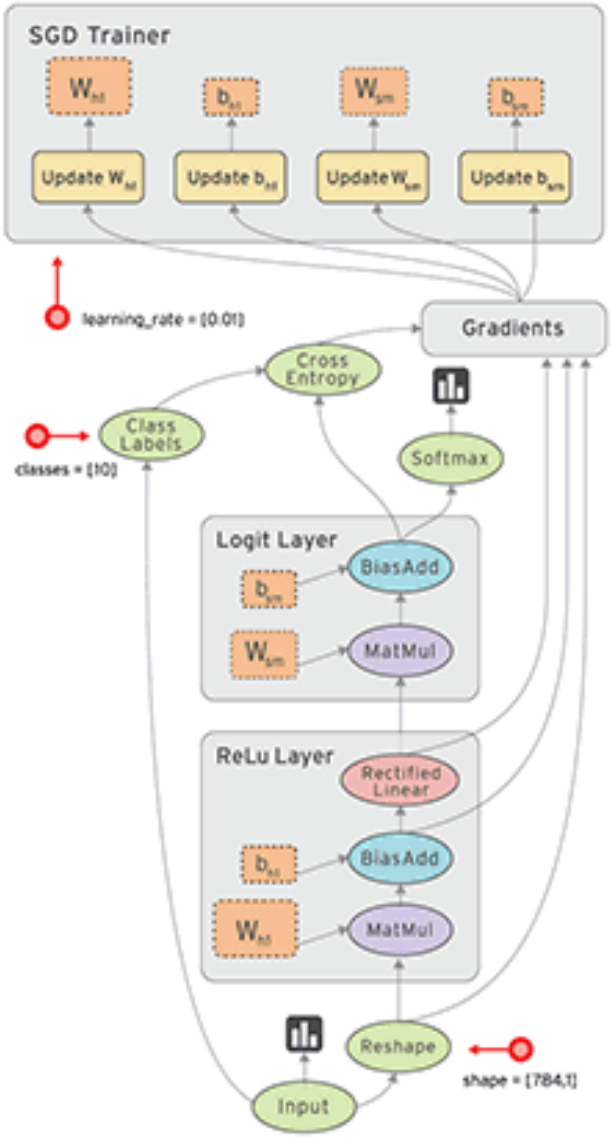
- 一. Tensorflow介绍
- 二. Tensorflow的基础语法

一. Tensorflow介绍

TF的简介

（1）TensorFlow（TF）是谷歌公司在2015年9月开源的一个深度学习框架。TF可在很多地方可以应用，如语音识别、自然语言理解、计算机视觉、广告等。

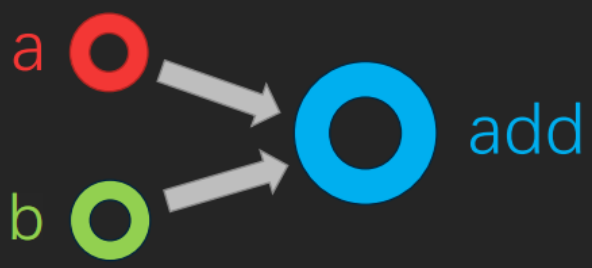
（2）TF是一个采用数据流图，用于数值计算的开源软件库。节点在图中表示数学操作，图中的线则表示在节点间相互联系的多维数据数组，即张量。它灵活的架构可以在多种平台上展开计算，例如台式计算机中的一个或多个CPU（或GPU），服务器，移动设备等。



一. Tensorflow介绍

TF的简介

TensorFlow 的 计算模型 为 graph （图） ， 一个 TensorFlow 图描述了计算的过程. 为了进行计算, 图必须在 会话 里被启动. 会话 将图的 op (节点) 分发到诸如 CPU 或 GPU 之类的 设备 上, 同时提供执行 op 的方法. 这些方法执行后, 将产生的 tensor 返回. 在 Python 语言中, 返回的 tensor 是 numpy ndarray 对象; 在 C /C++ 语言中, 返回的 tensor 是 tensorflow::Tensor 实例.



TensorFlow 的Tensor 表明了它的数据结构， 而Flow 则直观地表现出张量之间通过计算相互转化的过程 TensorFlow 中的每一个计算都是图上的一个节点， 而节点之间的边描述了计算之间的依赖关系， a, b 为常量，不依赖任何计算， 而add 计算则依赖读取两个常量的取值。

一. Tensorflow介绍

TF的基本特征

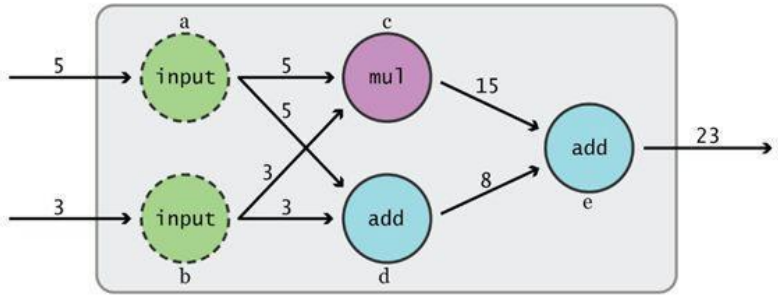
- 将计算流程表示成图；
- 通过Sessions来执行图计算；
- 将数据表示为tensors；
- 分别使用feeds和fetches来填充数据和抓取任意的操作结果

一. Tensorflow介绍

图（graph）的概念

（1）数据流图是描述有向图中的数值计算过程。有向图中的节点通常代表数学运算，但也可以表示数据的输入、输出和读写等操作；有向图中的边表示节点之间的某种联系，它负责传输多维数据(Tensors)。图中这些tensors的flow也就是TensorFlow的命名来源。

（2）计算图是静态的，意思是这个计算图每个节点接收什么样的张量和输出什么样的张量已经固定下来。要运行这个计算图，需要开启一个会话（session），在session中这个计算图才可以真正运行。



计算图的一个实例

一. Tensorflow介绍

会话（Session）的概念

会话（Session）：为了获得图的计算结果，图必须在会话中被启动。图是会话类型的一个成员，会话类型还包括一个runner，负责执行这张图。会话的主要任务是在图运算时分配CPU或GPU。

两种会话使用模式：

一：明确调用会话生成函数和关闭会话函数

```
#创建一个会话
sess=tf.Session()
# 使用会话来得到运算结果
sess.run(result)
# 关闭会话使本次运行中使用
#到的资源可以被释放
sess.close()
```

二：通过Python的上下文管理器来使用会话

```
# 计算放在“with”内部，上下文管理器
# 退出的时候会自动释放所有资源。
with tf.Session() as sess:
        sess.run(result)
```

一. Tensorflow介绍

张量的概念

（1）在TensorFlow程序中，所有的数据都通过张量的形式来表示。从功能的角度上看，张量可以被简单理解为多维数组。
（2）张量中并没有真正保存数字，它保存的是如何得到这些数字的计算过程，只是对TensorFlow中运算结果的引用。

运行：print(result)
输出：Tensor("add_2:0", shape=(2,), dtype=float32)

返回的是一个张量的结构。保存了三个属性：name , shape , dtype。

0阶张量 == 标量
1阶张量 == 向量（一维数组）
2阶张量 == 二维数组
...
n阶张量 == n维数组

一. Tensorflow介绍

张量的使用

张量的使用主要可以总结为两大类。
（１）第一类用途是对中间计算结果的引用。当一个计算包含很多中间结果时，使用张量可以提高代码的可读性。
（2）第二类情况是当计算图构造完成之后，张量可以用来获得计算结果，也就是得到真实的数字。

第一类用途：

```
a = tf.constant([1.,2.],name = 'a')

b = tf.constant([2.,3.],name = 'b')

result = a + b
```

第二类用途：

```
In  tf.Session().run(result)
Out array([ 3.,  5.], dtype=float32)
```

一. Tensorflow介绍

占位符placeholder

tf.placeholder表示占位符，占位符并没有初始值，只分配必要的内存，类型自己定义。变量用于网络模型参数调整，占位符表示输入、输出的数据，feed_dict进行馈送数据

实例：

```
w1=tf.Variable(tf.random_normal([1,2]))
x=tf.placeholder(tf.float32,shape(1,2))
a=w1+x
sess=tf.Session()
y=sess.run(a,feed_dict={x:[[0.7,0.9]]})
```

二. Tensorflow的基础语法

TensorFlow入门要点

（1）使用图 (graph) 来表示计算任务；
（2） 在被称之为 会话 (Session) 的上下文 (context) 中执行图；
（3）使用 张量(tensor) 表示数据；
（4）通过变量 (Variable) 维护状态；
（5）使用 feed 和 fetch 可以为任意的操作(arbitrary operation) 赋值或者从其中获取数据

二. Tensorflow的基础语法

TensorFlow语法例子 1

```
import tensorflow as tf
# 创建2个矩阵，前者1行2列，后者2行1列，然后矩阵相乘：
matrix1 = tf.constant([[3,3]])
matrix2 = tf.constant([[2], [2]])
product = tf.matmul(matrix1,matrix2)

# 上边的操作是定义图，然后用会话Session去计算：
with tf.Session() as sess:
    result2 = sess.run(product)
    print(result2)
```

输出：

```
[[12]]
```

二. Tensorflow的基础语法

TensorFlow语法例子2

```
# 定义一个tensorflow的变量：
state = tf.Variable(0, name='counter')
# 定义常量
one = tf.constant(1)
# 定义加法步骤 (注: 此步并没有直接计算)
new_value = tf.add(state, one)
# 将 State 更新成 new_value
update = tf.assign(state, new_value)
# 变量Variable需要初始化并激活，并且打印的话只能通过sess.run()：
init = tf.global_variables_initializer()
# 使用 Session 计算
with tf.Session() as sess:
    sess.run(init)
    for _ in range(3):
        sess.run(update)
        print(sess.run(state))
```

输出：

```
1
2
3
```

二. Tensorflow的基础语法

TensorFlow语法例子3

```
# 如果要传入值，用tensorflow的占位符，暂时存储变量，
# 以这种形式feed数据：sess.run(***, feed_dict={input: **})

import tensorflow as tf
# 在 Tensorflow 中需要定义 placeholder 的 type ，一般为 float32 形式
input1 = tf.placeholder(tf.float32)
input2 = tf.placeholder(tf.float32)
# mul = multiply 是将input1和input2 做乘法运算，并输出为 output
ouput = tf.multiply(input1, input2)
with tf.Session() as sess:
    print(sess.run(ouput, feed_dict={input1: [7., 10], input2: [2.]}))
```

输出：

```
[14. 20.]
```

二. Tensorflow的基础语法

TensorFlow函数拟合例子

拟合y_data的函数，使得权重和偏置分别趋近0.1和0.3。具体程序如下：

```
import tensorflow as tf
import numpy as np
# np.random.rand(100)生成100个[0,1]之间的随机数，构成1维数组
# np.random.rand(2,3)生成2行3列的二维数组
x_data = np.random.rand(100).astype(np.float32)
y_data = x_data * 0.1 + 0.3

# 权重偏置这些不断更新的值用tf变量存储，
# tf.random_uniform()的参数意义：(shape,min,max)
# 偏置初始化为0
x_input = tf.placeholder(tf.float32, [100])
y_label = tf.placeholder(tf.float32, [100])

weights = tf.Variable(tf.random_uniform([1]))
biases = tf.Variable(tf.zeros([1]))

y = weights * x_input + biases

# 损失函数。tf.reduce_mean()是取均值。square是平方。
loss = tf.reduce_mean(tf.square(y - y_label))
```

```
# 用梯度优化方法最小化损失函数。
optimizer = tf.train.GradientDescentOptimizer(0.5)
train = optimizer.minimize(loss)

# tf变量是需要初始化的，而且后边计算时还需要sess.run(init)一下
init = tf.global_variables_initializer()

# Session进行计算
with tf.Session() as sess:
    sess.run(init)
    print('step   ','weights   ', 'biases')
    for step in range(201):
        sess.run(train, feed_dict={x_input:x_data, y_label:y_data})
        if step % 20 == 0:
            weight_get, biases_get = sess.run([weights, biases])
            print(step, weight_get, biases_get)
```

二. Tensorflow的基础语法

TensorFlow函数拟合例子

拟合y_data的函数，使得权重和偏置分别趋近0.1和0.3

输出：

```
step    weights    biases
0 [0.31729472] [0.24968289]
20 [0.14918691] [0.27356562]
40 [0.11276162] [0.29314157]
60 [0.10331102] [0.29822057]
80 [0.10085905] [0.29953834]
100 [0.10022289] [0.29988024]
120 [0.10005786] [0.29996893]
140 [0.10001502] [0.29999194]
160 [0.10000391] [0.2999979]
180 [0.10000104] [0.29999945]
200 [0.10000028] [0.29999986]
```

二. Tensorflow的基础语法

TensorFlow函数拟合例子整体流程

在函数拟合例子中，TensorFlow整体流程如下：

1.输入数据 `x_input = tf.placeholder(tf.float32, [100])`

2.建立模型 `y = weights * x_input + biases`

3.定义损失函数 `loss = tf.reduce_mean(tf.square(y - y_label))`

4.训练方法 `optimizer = tf.train.GradientDescentOptimizer(0.5)`

5.初始化和启动Tensorflow会话 `sess.run(init)`

6.训练

```
sess.run(train, feed_dict={x_input:x_data, y_label:y_data})
```

总结

一、介绍Tensrflow的由来，介绍Tensrflow的基本概念，如张量、图、会话等；
二、通过几个简单的例子介绍Tensrflow的基本语法。通过一个函数拟合例子，体现Tensrflow如何进行参数的训练更新。

谢谢聆听

参考网页：

TF介绍：https://www.jianshu.com/p/4665d6803bcf
TF入门：https://blog.csdn.net/red_stone1/article/details/78403416
TF会话： https://blog.csdn.net/weixin_37392582/article/details/79814741
TF语法例子：https://blog.csdn.net/login_sonata/article/details/77620328