TensorFlow官方文档

1. 新手入门

1.1 介绍

tf.matmul(W, x\_data) + b 矩阵乘法

tf.train.GradientDescentOptimizer(0.5) 实现梯度下降算法的优化器

(1)建立梯度下降优化器对象 基本参数为学习率

(2)compute\_gradients函数计算损失函数对变量的梯度，返回一个<梯度，变量>对的列表，其中梯度是minimize函数的第一部分

(3)apply\_gradients函数把梯度加到变量上去，这是minimize函数的第二个操作

tf.global\_variables\_initializer() 新版采用这个函数进行初始化

使用Variable时需要初始化

1.2 下载及安装

Windows Python3.5.4

1.3 基本使用

综述

使用Graph表示计算任务，Tensor表示数据，通过变量Variable维护状态

计算过程：在会话Session中启动图，sess将图的节点Operation以及执行方法分发到各个GPU，执行完毕后再将产生的Tensor返回

构建图

以矩阵乘法为例，两个常量constant()op和一个matmul()op，执行时，会话负责传递op所需的全部输入，通常是并发执行

Session对象使用完毕后需要关闭释放资源，也可用with自动关闭

默认使用第一个GPU，如果有其他的GPU，需要用with…device指定某CPU或GPU进行操作

with tf.device("/gpu:1"):

Tensor

张量 类比n维数组或列表

阶 张量的维度 eg. t=[[1,2,3],[4,5,6],[7,8,9]]是2阶

0 0 00 
数 学 实 例 
Python 例 子 
纯 量 只 有 大 小 
向 量 大 小 和 方 向 
[ 1 ． 1 , 2 ． 2, 
土 彐 ] 
矩 阵 数 据 表 
六 8, 9]] 
3 阶 张 量 （ 数 据 立 体 
[ 12L 
[ 12 ] ], 
[ [ 14L [ 15L 
[ 18 ] ] ] 
阶 自 己 想 想 看 

数据类型

Float32、float64、int64、int32、int16、int8、uint8、string、bool

变量

tf.assign(state, new\_value) 更新模型中变量的值，等同于state = new\_value

通常将统计模型中的参数作为一组变量存储在Tensor中，训练过程中通过重复运行训练图，更新这个Tensor

Fetch

tf.multiply(input1, intermed) 常量乘法

result = sess.run([mul, intermed]) 需要获取多个Tensor时，可在op的一次运行中一起获得

Feed

用tf.placeholder()作为占位符

input1 = tf.placeholder(tf.types.float32) input2 = tf.placeholder(tf.types.float32) output = tf.mul(input1, input2)

with tf.Session() as sess: print sess.run([output], feed\_dict={input1:[7.], input2:[2.]})

2. 完整教程

2.1 总览

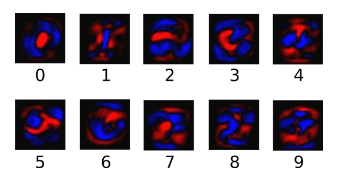
2.2 MNIST数据下载

2.3 MNIST入门

将图片数组展开成一维向量，利用softmax进行分类

softmax回归

(1)对图片像素值进行加权求和再加偏置



(2)softmax(x) = normalize(exp(x))，将线性输出转换为概率值

交叉熵损失函数计算的是100幅图片的交叉熵总和

tf.equal(tf.argmax(y,1), tf.argmax(y\_,1)) 找出某个Tensor对象在某一列上其数据最大值所在的索引值，然后用tf.equal检测索引位置是否匹配

tf.cast(correct\_prediction, "float") 将布尔值变为浮点数（1、0）

2.4 MNIST进阶

计算图

在Python内部构建计算图，然后完全在Python外部运行，以及安排哪一部分被运行

Q：eval()函数

10.参考文献

常用函数 http://blog.csdn.net/u014595019/article/details/52805444

优化器 <http://blog.csdn.net/xierhacker/article/details/53174558>

TensorFlow莫烦教程笔记

结构：有向图

1.Session会话控制

Tensorflow的计算必须在一个Session的上下文中。Session会包含一个计算图，在添加Tensor和Operation的时候，都不会立即进行计算，而是等需要计算Session时，Tensorflow会知道如何优化和执行图计算。

一个Session会拥有一些资源，不需要时要将资源释放，有两种方式

a.调用session.close()方法

b.使用with tf.Session() as sess创建上下文执行，上下文退出时自动释放

2.Variable变量

常量是constant

如果定义Variable一定要写

init = tf.initialize\_all\_variables() 以及 sess.run(init)

上面这句也可以写成 tf.initialize\_all\_variables().run()

3.Placeholder传入值

用placeholder时，在run的时候再给定输入的值

用到数据字典

4.激励函数

relu、tanh、sigmoid等

搜索tensorflow activation function

5.Optimizer优化器

不同之处主要在于学习策略

class tf.train.GradientDescentOptimizer

6.Tensorboard可视化框架

### Windows下暂未成功 ###

用with t.name\_scope('name'):来人为对每一模块命名

用writer = tf.Summary.FileWriter()在Windows中将其保存为文件，用于查看

tensorboard --logdir='logs/'