保存与读取模型

在使用tf来训练模型的时候，难免会出现中断的情况。这时候自然就希望能够将辛辛苦苦得到的中间参数保留下来，不然下次又要重新开始。好在tf官方提供了保存和读取模型的方法。

保存模型的方法：

# 之前是各种构建模型graph的操作(矩阵相乘，sigmoid等等....)

saver = tf.train.Saver() # 生成saver

with tf.Session() as sess:

sess.run(tf.global\_variables\_initializer()) # 先对模型初始化

# 然后将数据丢入模型进行训练blablabla

# 训练完以后，使用saver.save 来保存

saver.save(sess, "save\_path/file\_name") #file\_name如果不存在的话，会自动创建

将模型保存好以后，载入也比较方便，如下所示：

saver = tf.train.Saver()

with tf.Session() as sess:

#参数可以进行初始化，也可不进行初始化。即使初始化了，初始化的值也会被restore的值给覆盖

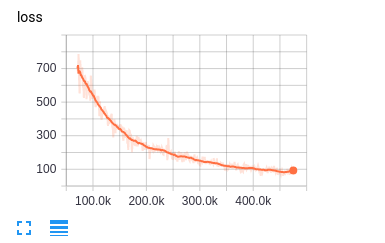
sess.run(tf.global\_variables\_initializer())

saver.restore(sess, "save\_path/file\_name") #会将已经保存的变量值resotre到 变量中。

简单的说，就是**通过saver.save来保存模型，通过saver.restore来加载模型**

使用tensorboard来使训练过程可视化

tensorflow还提供了一个可视化工具，叫tensorboard.启动以后，可以通过网页来观察模型的结构和训练过程中各个参数的变化。如下图所示



关于如何合理清楚的显示网络结构，我目前还不太搞得清楚，而且目前看来也不是太重要；但是要将训练的过程可视化还是比较方便的。简单的说，流程如下所示：

* 使用tf.scalar\_summary来收集想要显示的变量
* 定义一个summury op, 用来汇总多个变量
* 得到一个summy writer，指定写入路径
* 通过summary\_str = sess.run()

# 1. 由之前的各种运算得到此批数据的loss

loss = .....

# 2.使用tf.scalar\_summary来收集想要显示的变量,命名为loss

tf.scalar\_summary('loss',loss)

# 3.定义一个summury op, 用来汇总由scalar\_summary记录的所有变量

merged\_summary\_op = tf.merge\_all\_summaries()

# 4.生成一个summary writer对象，需要指定写入路径,例如我这边就是/tmp/logdir

summary\_writer = tf.train.SummaryWriter('/tmp/logdir', sess.graph)

# 开始训练，分批喂数据

for(i in range(batch\_num)):

# 5.使用sess.run来得到merged\_summary\_op的返回值

summary\_str = sess.run(merged\_summary\_op)

# 6.使用summary writer将运行中的loss值写入

summary\_writer.add\_summary(summary\_str,i)

接下来，程序开始运行以后，跑到shell里运行

$ tensorboard --logdir /tmp/logdir

开始运行tensorboard.接下来打开浏览器，进入127.0.0.1:6006 就能够看到loss值在训练中的变化值了。