## 1. Saver的背景介绍

    我们经常在训练完一个模型之后希望保存训练的结果，这些结果指的是模型的参数，以便下次迭代的训练或者用作测试。Tensorflow针对这一需求提供了[Saver](https://www.tensorflow.org/versions/r0.9/api_docs/python/state_ops.html" \l "Saver" \t "http://blog.csdn.net/u011500062/article/details/_blank)类。

1. Saver类提供了向checkpoints文件保存和从checkpoints文件中恢复变量的相关方法。Checkpoints文件是一个二进制文件，它把变量名映射到对应的tensor值 。
2. 只要提供一个计数器，当计数器触发时，Saver类可以自动的生成checkpoint文件。这让我们可以在训练过程中保存多个中间结果。例如，我们可以保存每一步训练的结果。
3. 为了避免填满整个磁盘，Saver可以自动的管理Checkpoints文件。例如，我们可以指定保存最近的N个Checkpoints文件。

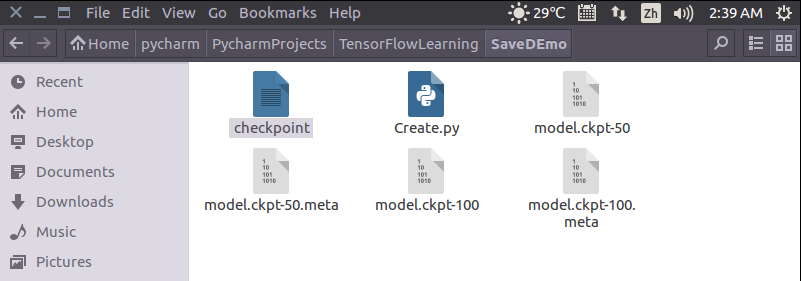
在testSaver.py代码中：  
****isTrain****：用来区分训练阶段和测试阶段，True表示训练，False表示测试  
****train\_steps****：表示训练的次数，例子中使用100  
****checkpoint\_steps****：表示训练多少次保存一下checkpoints，例子中使用50  
****checkpoint\_dir****：表示checkpoints文件的保存路径，例子中使用当前路径

## 2.1 训练阶段

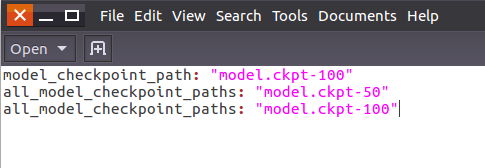
使用Saver.save()方法保存模型：

1. sess：表示当前会话，当前会话记录了当前的变量值
2. checkpoint\_dir + '/model.ckpt'：表示存储的文件名
3. global\_step：表示当前是第几步

训练完成后，当前目录底下会多出5个文件。



打开名为“checkpoint”的文件，可以看到保存记录，和最新的模型存储位置。



## 2.2测试阶段

    测试阶段使用saver.restore()方法恢复变量：

1. sess：表示当前会话，之前保存的结果将被加载入这个会话
2. ckpt.model\_checkpoint\_path：表示模型存储的位置，不需要提供模型的名字，它会去查看checkpoint文件，看看最新的是谁，叫做什么。

运行结果如下图所示，加载了之前训练的参数w和b的结果

