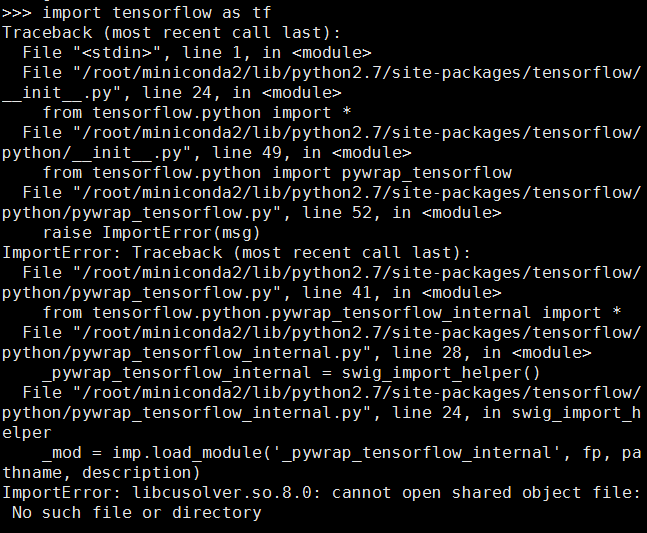
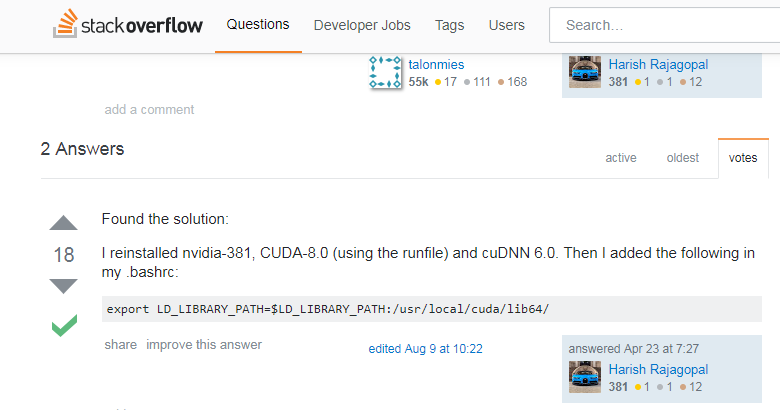
# WaveNet试运行与源码简单解析

## Wavenet试运行

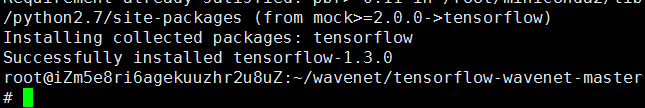
本周我尝试运行了一下wavenet。首先是在自己的笔记本上尝试，但由于librosa包装不上，我就在之前magneta的服务器上试着安装。当安装好了librosa之后，发现在试运行的时候，无法调用tensorflow（之前弄magenta时并没有出现这个问题）。遇到的问题截图如下：



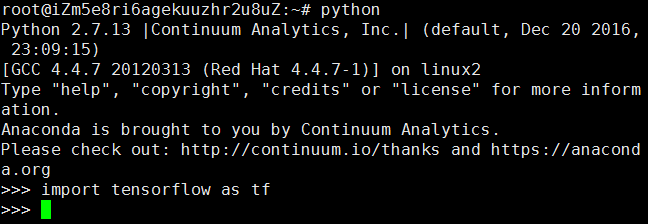
针对此时出现的问题，我上网查看了半天相似错误的解决办法，比如如下的解决办法：



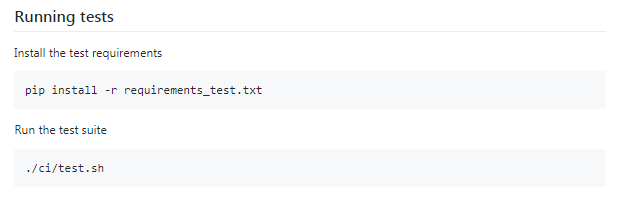
尝试了在.bashrc文件中添加上面的路径，发现根本没有起作用（实际上我不知道服务器里有没有安装cuda）。实在找不到解决办法，我就重装了tensorflow。



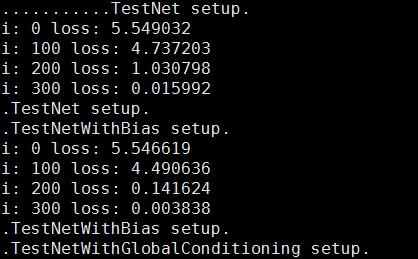
此时发现可以正常调用tensorflow了，不再报错。



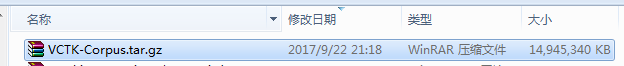
试运行测试代码：



发现可以运行，并打印出了缺失值：



接下来进行导入数据，下载下来的VCTK数据集有14多G，由于传输压缩文件大小限制是4G，我解压了部分文件再次打包（2G左右），导入到了服务器中：

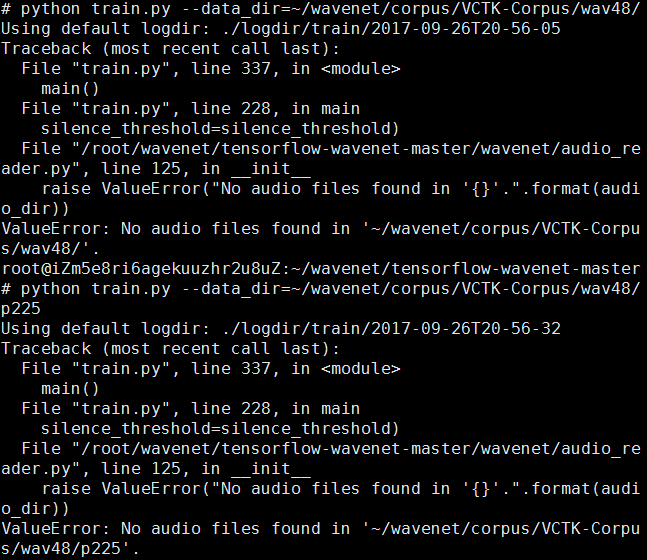


导入目标路径并解压：

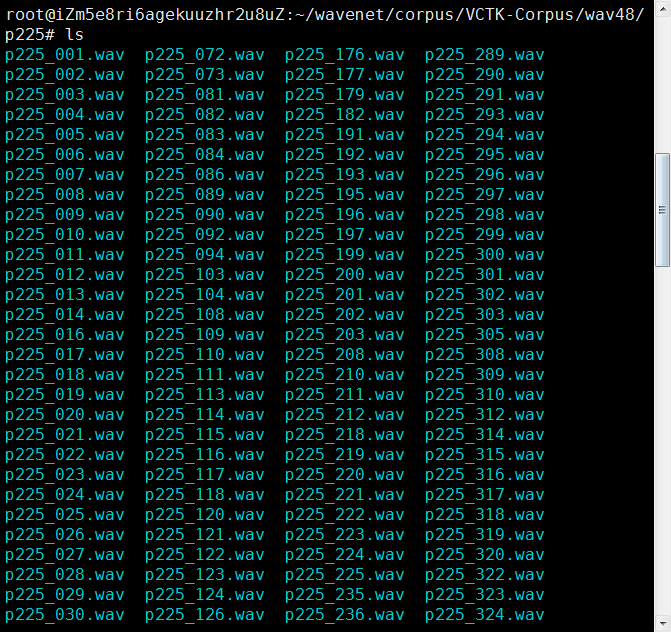


随后，在运行

python train.py --data\_dir = ~/wavenet/corpus/VCTK-Corpus/wav48/p225时报错，（找不到需要录入的wav文件）：



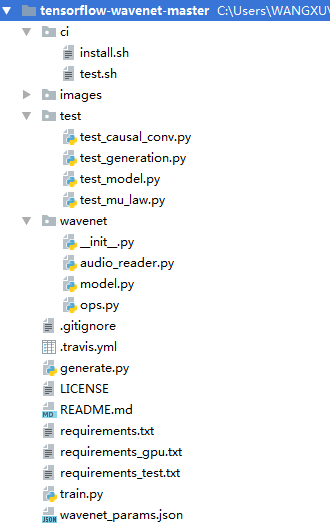
然而进入对应目录查看，发现wav文件是确实存在的：



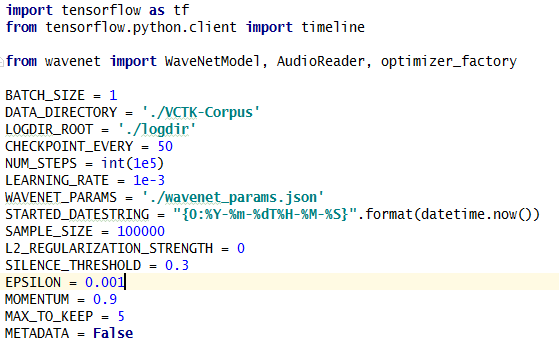
由于无法继续运行之后的步骤，运行部分就此终止。

## 代码解析

Wavenet项目的源代码结构如下图。起关键作用的文件主要是train.py，generate.py和wavenet文件夹。train.py用来训练模型，generate.py用来生成目标语段。wavenet文件夹包括了所需的模型和语音读取的其它功能类和方法。其中wavenet中的model.py 包含了建立网络模型和语音生成器相关的函数。 比如创建变量函数create\_variables()，创建因果卷积函数create\_causal\_layer()，创建扩大卷积函数create\_dilation\_layer()，建立网络模型函数\_create\_network()以及loss()函数等。因果卷积扩大卷积和其中包含的多线程的知识较为复杂，还没有看懂，因此接下来主要解析train.py的代码：



train.py文件包含了一系列参数的设置，模型保存save()方法，加载load()方法以及重要的main()方法：

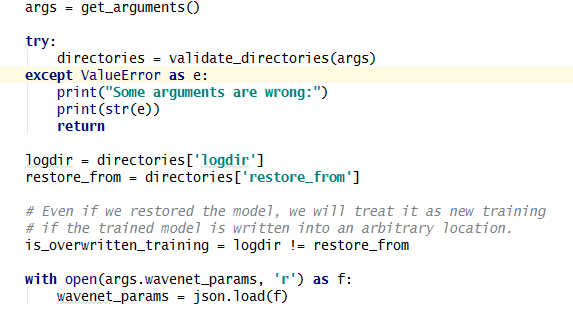


设置的这一系列参数中，包含了batchsize，训练数据路径，保存日志路径，每次训练多少轮，每一轮训练的步数，学习速率以及优化器等。



以上保存模型和加载模型函数，实现的功能是保存模型参数到检查点以及点提取参数加载预先保存的模型。而对于main()函数，它主要实现的功能是：

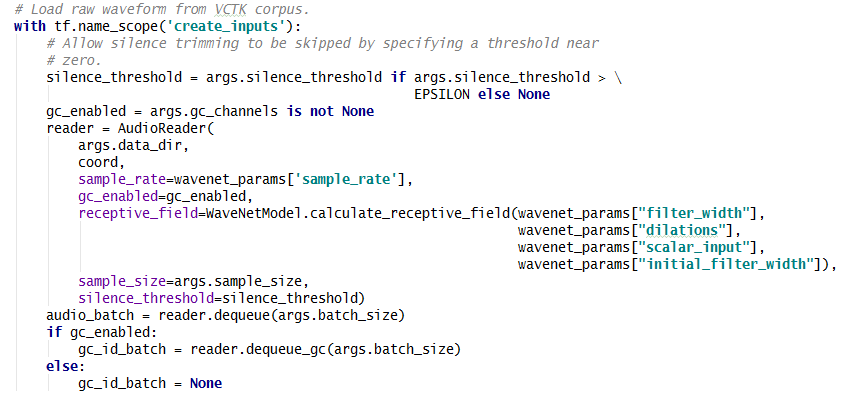
①读取wavenet模型参数；



②建立tensorflow的coordinator（用来实现多线程）；



③ 用VCTK 原始数据集生成用于input的文件；



④建模wavenet模型；



⑤训练并保存模型:

