**2017.09.11--2017.09.20 周报**

**王雪婷**

1. **概述**

这周被分到“智能编曲”组，根据老师所说，在有了一定python和RNN、lstm的基础上，找了几个程序来跑，然后简单了解了一下wavenet。

1. **内容**

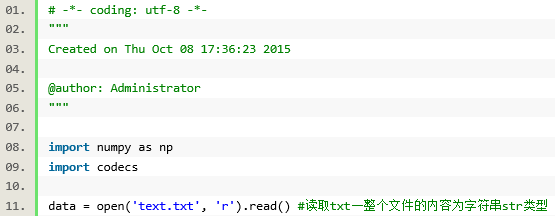
**1、RNN小程序**

根据之前看过的python和RNN知识，在网上找了一个RNN的小程序，程序的作用大概是“读取一个文档，识别并记录文档里的字符；然后自己模拟学习，拼凑出正确的单词和语句”

原文链接：<http://blog.csdn.net/hjimce/article/details/49095371#>

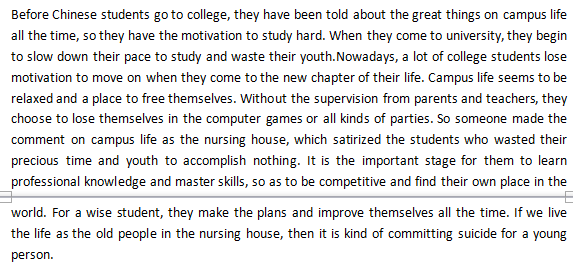
（原文中的程序用的是python2.7写的，所以在3+的程序中还需要一些修正）

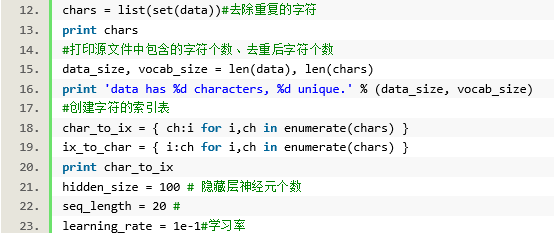
然后我把我对这个程序的修正和一些理解总结如下：



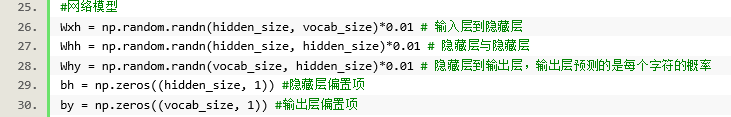
首先确保安装了numpy和codecs两个库

其次，程序要求有一个’text.txt’的文档，于是我找了一段英文如下做成该文档：

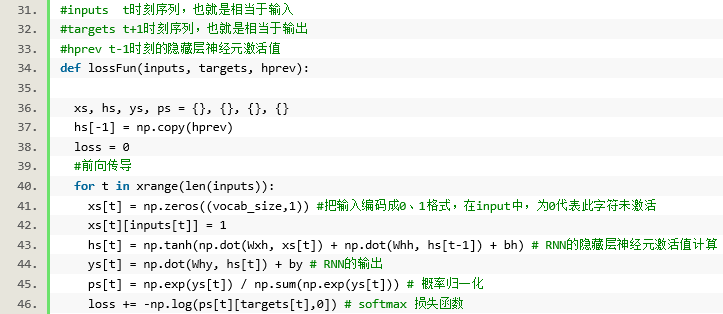




22行，是指：序列的长度

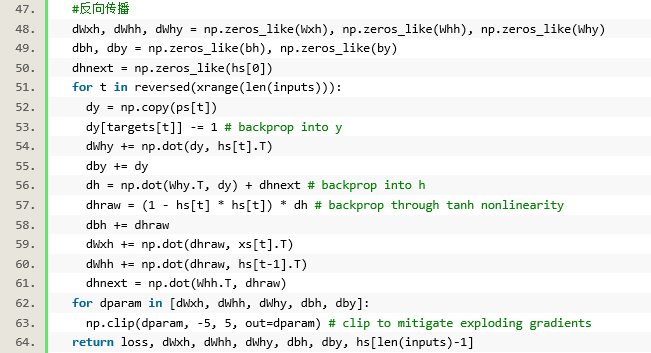


X：输入；h：隐藏；y：输出

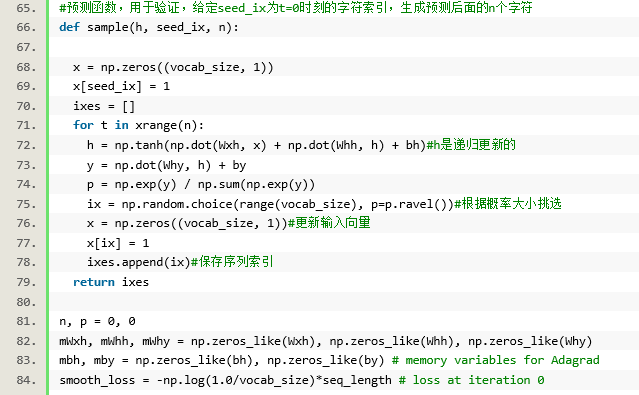


40行，python3+中，xrange（）函数，被修正成了“range（）”函数

Python2.x和3.x之间的一些差别总结：<http://blog.csdn.net/pipisorry/article/details/22107553>



这里的backprop指的应该是back propagation 算法，算法具体内容为：<https://www.zhihu.com/question/27239198> ，简单来讲，我理解他的意思就是：这个算法会调整weight,使下一次输入的输出错误减少。





95-99：每迭代1000次，会输出一次计算机的认知，他对单词的拼写和句子的组合。

99行，print的使用需要加括号（）。

结果：

一开始计算计得出的序列是“乱码”：



到了后半段，可以看出一些单词：



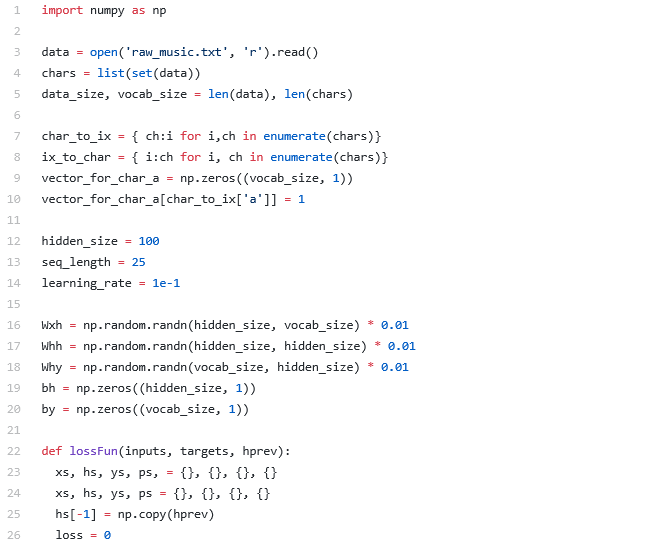
如果加大“序列”和循环的值，最后的结果会变得更加好：

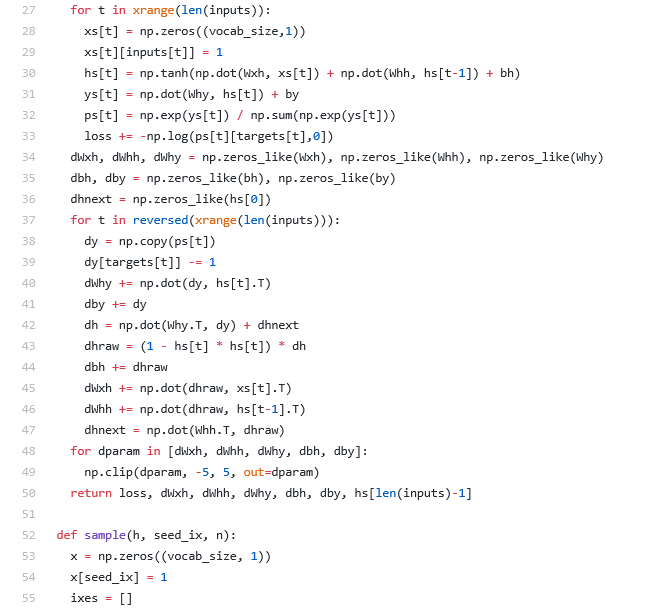


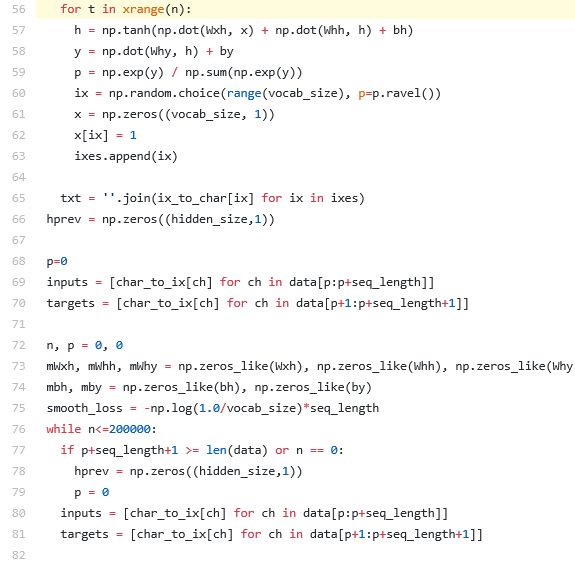
**2、RNN生成音乐的程序**

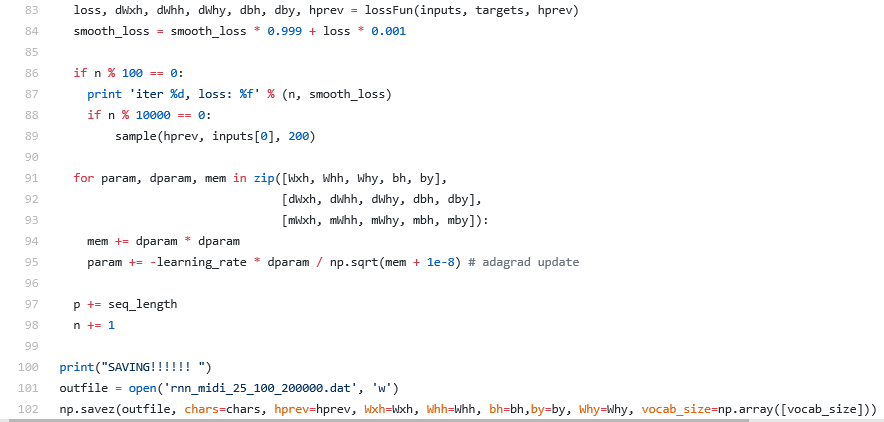
跑完普通的RNN后，我在github上找“RNN生成音乐”的程序，最后找到了“scidle-midi-rnn-master”程序，原连接如下：<https://github.com/dcervantes/scidle-midi-rnn>

然后我对这个程序的修正（原程序也是python2.7写的）和一些理解总结如下：









总体思路和语句几乎一致，只是这里的文本是已经给出的音乐文本[raw\_music.txt](https://github.com/dcervantes/scidle-midi-rnn/blob/master/raw_music.txt" \o "raw_music.txt) ，而且减少了信息的中途输出（第一个例子里每1000次输出一次结果），只有迭代的次数和它的loss大小，然后（貌似）因为没有写入自适应梯度下降法的相关程序，所以loss值一直在50左右。

然后文件中还有一个keyboard2midi.py的文件，因为它里面导入了一个mido库，但我查了一下，只有python2.7和3.2才能装（我的之前为了装tensorflow，装的是3.5），所以我没有运行那个文件。

**3、wavenet的初步了解**

Wavenet 的简单介绍：<https://zhuanlan.zhihu.com/p/24317897>

因为只是一些基础，所以还不能总结出太多东西