**本科实验报告专用纸**

1. 设计背景

当今是一个大数据时代，从网络中高效率的获取所需信息的能力至关重要。同时，随着计算机算力的增加与数据的繁杂，使用机器学习（人工神经网络）的方法来分析与处理数据，并且将数据可视化的方法是十分有效的。而python，则是实现以上三者非常有力的工具。

1. 内容及思路

执行程序前敬请阅读ReadMe.md文件。

本次课程设计的要求虽然可以大致分为三个任务，但我力求将它们有机地串联起来。一开始想实现的是基于卷积神经网络的网络图像识别，但是在数据可视化方面并没有特别好的想法，并且其神经网络训练时间较长，不方便老师复现。于是便诞生了本次我的实验思路。**本部分主要编程思路想法，具体的程序细节详见第三部分。另外，在pythonDesign目录下有一个压缩包，里面存放着运行完一轮的所有文件，仅供参考。**

1.使用urllib与BeautifulSoup爬取股票名称及对应的代码

爬取网站为：<https://q.stock.sohu.com/cn/bk.shtml> ，进入后发现有若干板块：



我们点击其中一个板块：



可以看见股票代码以及股票名称。先判断是否是由js动态加载出来，如果是，则需要采用ajax技术进行提取。



查看html文件，发现非常幸运，二者都在class中，并不是动态加载的。我们可以通过BeautifulSoup快速提取出来。

接着我们看如何连贯的提取不同页面的数据。考虑到我们是在板块页面点击进入到具体的股票列表的，因此我们要到<https://q.stock.sohu.com/cn/bk.shtml> 中去找对应的超链接：



存在e2的class中，我们只需要遍历所有的e2，提取出他的超链接即可。

提取出来相关信息以后，先存在dataframe中，最后存储在csv文件中。由于url数据过多，我们提取前50个页面的所有股票编号和信息。

2.scrapy 提取股票的历史数据

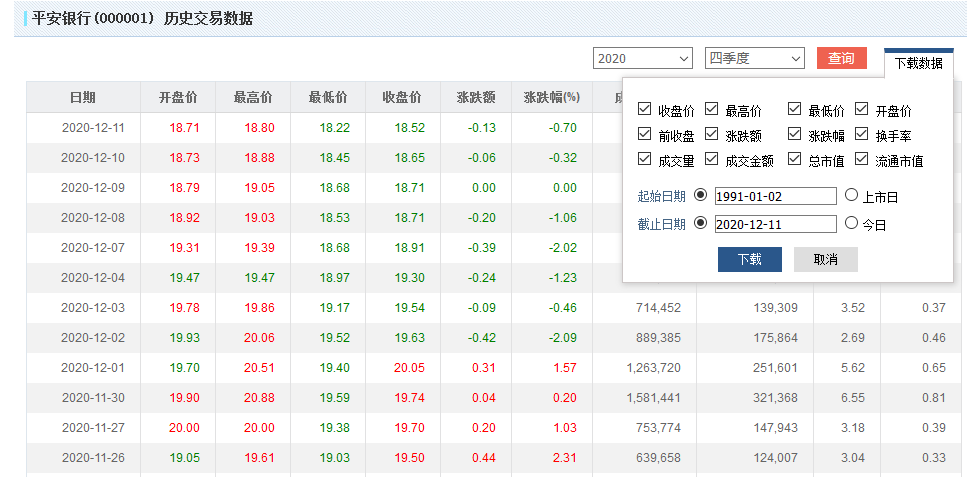
观察到网站：<http://quotes.money.163.com/trade/lsjysj_000001.html>



url和股票的代码有着紧密的关联：

url = 'http://quotes.money.163.com/trade/lsjysj\_{}.html'.format(self.current\_code)

该网站提供了一个下载的api：



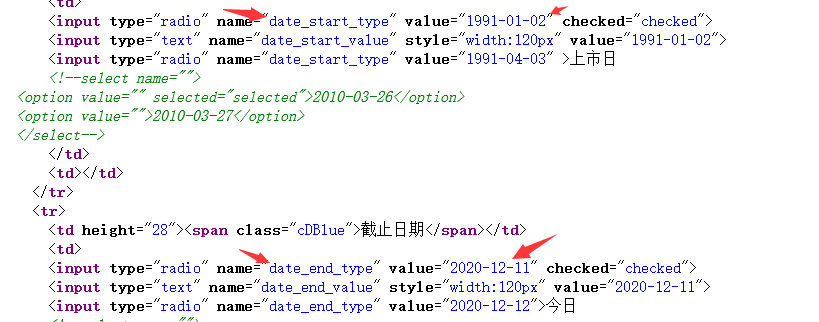
经过研究后发现：

download\_url = "http://quotes.money.163.com/service/chddata.html?code={}&start={}&end={}&fields=TCLOSE;HIGH;LOW;TOPEN;LCLOSE;CHG;PCHG;TURNOVER;VOTURNOVER;VATURNOVER;TCAP;MCAP".format(new\_stock\_code, start\_time, end\_time)

其中，new\_stock\_code是对应的股票代码，start\_time和end\_time是数据的开始和截止日期。

问题转换到如何获取股票的start\_time和end\_time。

观察html：



发现这两组数据都在对应的name中，比较方便提取。

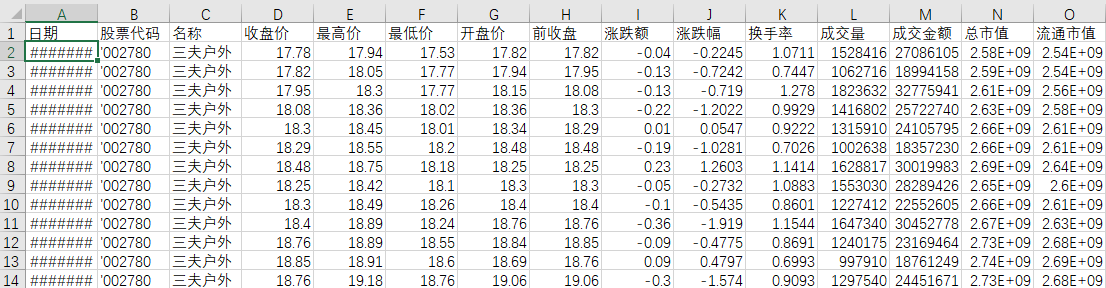
确定了下载的可能性，下面是总体流程：

1. 利用spider爬虫parse出每个代码对应的下载链接。由于提取的股票数据过多，我们下载前200个股票的csv数据。
2. 将连接依次传输给scrapy自带的下载管道进行批量下载。
3. 将下载下来的csv数据存储在一个文件夹中。

3.利用循环人工神经网络LSTM进行对股票的预测与验证

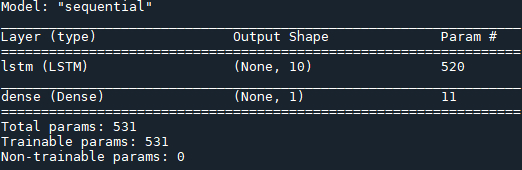
LSTM指长短期记忆人工神经网络。

任取一下载好的股票数据集，观察数据集类型：

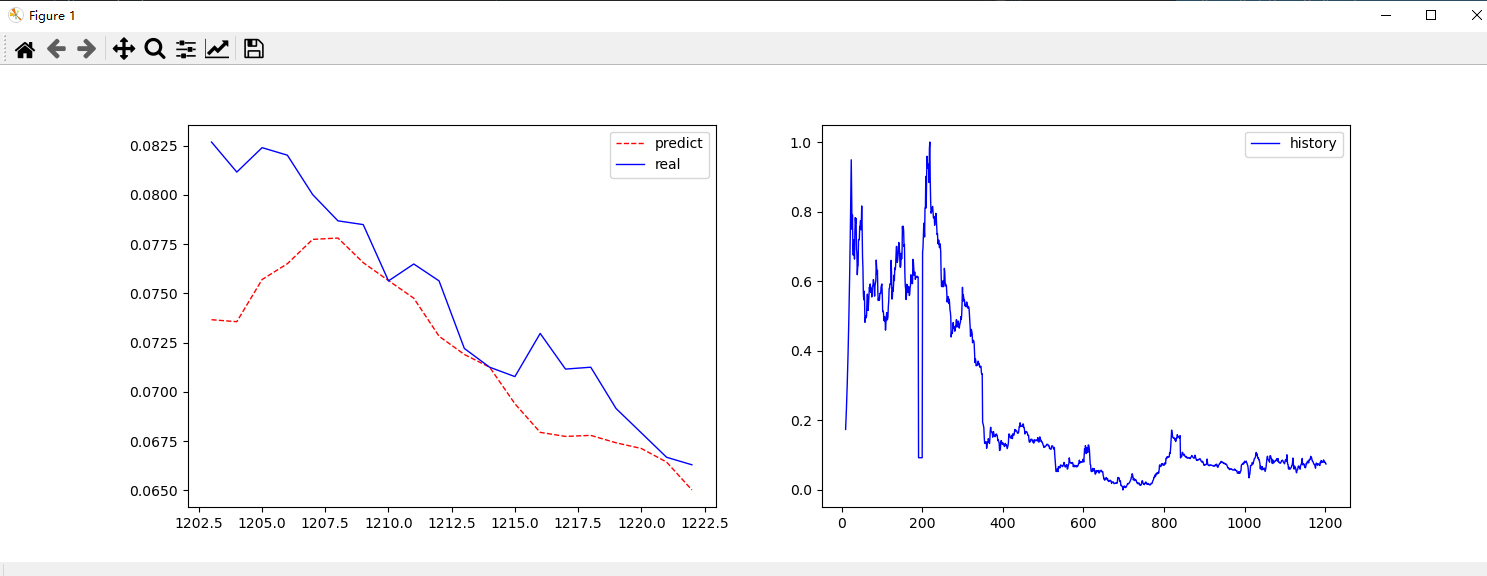


初步确定目标：通过前些日的开盘价与最高价去预测当日的收盘价。

我设定LSTM的记忆长度是10天，建立如下的神经网络：

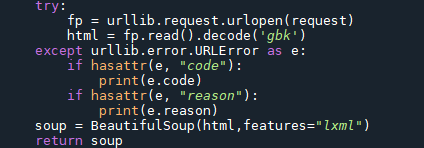


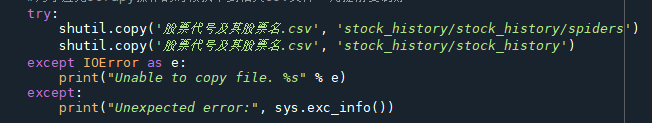
步长方案选择：Adam类型，loss函数采用mse。经过10个epoch的训练，通过matplotlib.pyplot输出模型预测数据（红色虚线）和真实值（蓝色线），查看训练效果。图二是总的走势图。



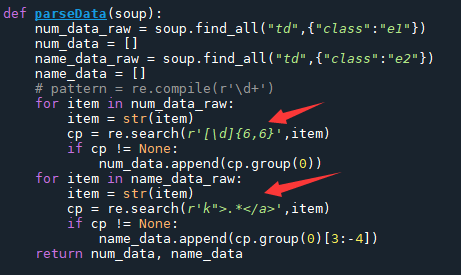
1. 程序的实现细节：
2. urllib获取股票代号.py

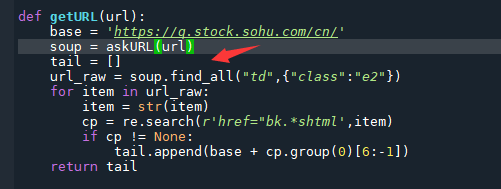
程序函数分装清晰整齐，最后在main函数中调用，复用性高。





运用了try…excpt结构，便于调试以及错误处理。





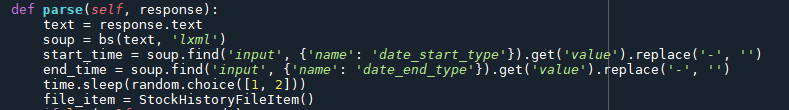
合理运用正则表达式以及BeautifulSoup进行信息的提取。



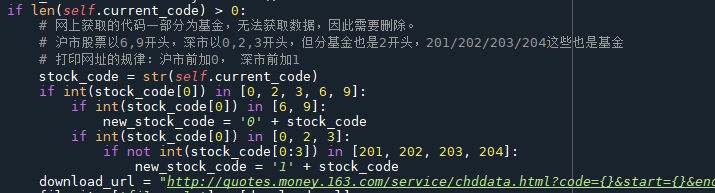
利用Dataframe这种结构来存储数据，最后导出到csv文件中。

1. scrapy模块

stock\_history\_spider.py中

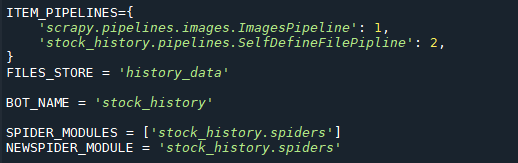


Parse函数中巧妙运用bs来进行对date\_start\_type提取，同时为了保证格式的一致，对这个数据进行了更改处理，将’-’替换为空。

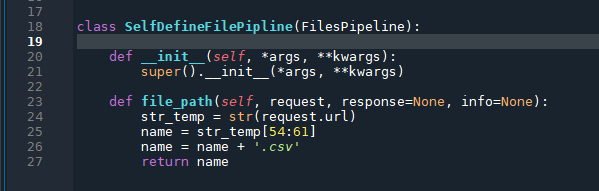


经过百度搜索，发现下载链接与股票的编号有如下具体关系，同时发现有一些股票没有办法下载，进行了良好的边缘条件处理。

我进行了良好的spider与pipe的对接：

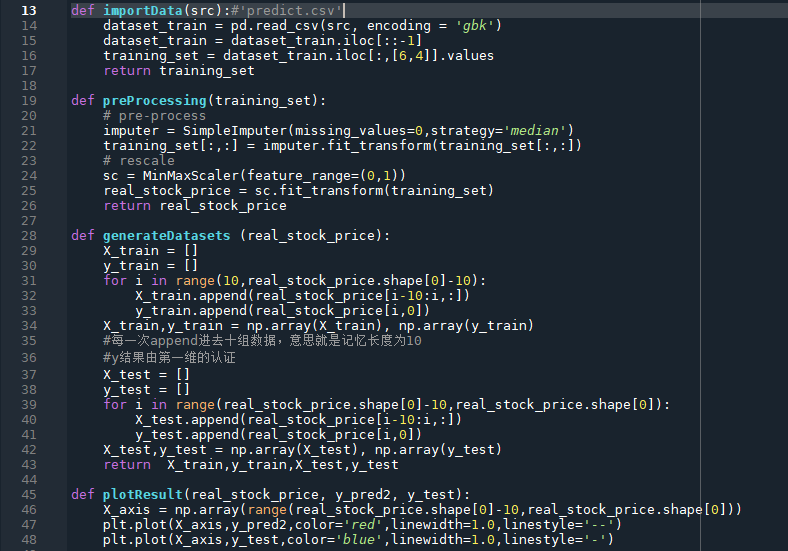


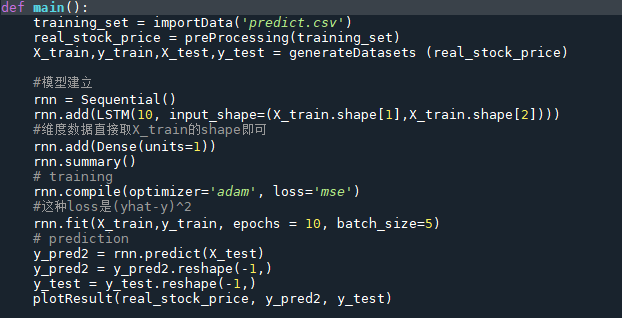
在setting.py中进行了ITEM\_PIPELINES的初始化，同时规定了下载的位置。



为了保证下载下来的文件名有所规律，在item模块中进行了文件名的设定。同时，我设计了额外的名字更新模块——stock\_csv文件名更改.py。通过csv文件建立字典，实现对只有股票代号的文件进行重命名，更加方便查找。

1. 人工神经网络模块



我对该模块的函数进行了良好的封装，同时合理设定了记忆长度和预测序列的长度。由于时间序列具有连贯性，我先让csv导入进来的数据倒序，保证时间是由远到近变换的，同时，不能采用data\_split这种粗暴的方法进行训练集和测试集的分割，因此我采用手动分割。

调用了Keras非常方便的构建了神经网络。采用python绘图的方法验证了测试的准确性。

1. 总结

此次试验感想颇多。因为不会打log日志，在scrapy调试的时候费时费力，甚至一度想放弃，好在最终还是调试完毕。在爬取信息的时候，我发现自己已经可以调用很多手段，但是还是用的最熟练的正则表达式最顺手，当然bs4模块也很好用。Try…except的语句使用还是需要自己刻意的提醒，还需要一定时间才能成自然。这都是今后我需要努力的方向。希望今后能和老师多多交流，深入地进行学习，若有任何问题，可以随时联系我。