趋势预测说明文档

本部分主要分为三块，分别为训练、预测和微调，对应代码为train\_DA\_RNN.py, inference\_DA\_RNN.py和finetune\_DA\_RNN.py。本功能主要基于DA\_RNN模型进行训练和预测，框架还没有完善好，暂时不支持其他模型，后续可以进一步完善。

几个需要说明的文件：

stock\_codes.npy：保存了沪深300对应的300支股票

2020\_all\_stock\_price.csv：保存了悟空提供的2020年以来七个月的对应需要的股指价格数据。

2020\_all\_stock\_return.csv：对上述数据进行了处理，变换为股指涨跌数据。

columns.npy：2020\_all\_stock\_price.csv对应的列名

stock\_dict.npy：保存了从聚宽平台得到的沪深300中每只股票对应的相关指数，数据结构为{股票1：[指数1A，指数1B，…]，

股票2：[指数2A，指数2B，…]，

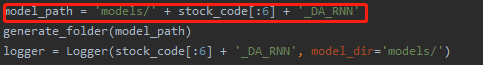
……}

训练：

训练部分主要完成的工作是输入沪深300中的股票代码，该部分从数据中读取出该代码以及其对应的相关指数，并使用这些数据进行训练，将最后的模型以及一系列有用的相关文件保存在代码内部设定的保存位置。

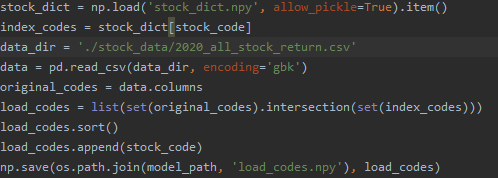


训练的输入主要是一些网络设置以及对应的股票代码，网络设置一般采用默认即可，如果特别需要可以进行更改。

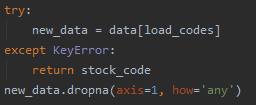


模型以及一系列相关文件的保存地址为红框部分，以贵州茅台为例保存地址为：models/600519\_DA\_RNN/

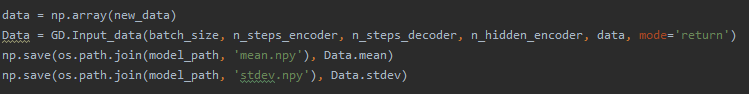
Logger为训练日志，保存训练过程中的细节



这部分为选取相应的股票代码和指数代码，产生这部分的主要原因是聚宽平台提供的相关指数在悟空给的数据中存在缺失情况，所以目前取并集处理，最后的load\_codes就是该股票对应的所有相关指数加上它本身，考虑到后续可能需要使用，所以也将其保存起来。



这部分就是将需要的数据从data里面读取出来，并将NAN数据清除。这里的报错处理部分主要是为了处理沪深300股票本身在所给数据中的缺失情况（最开始的确出现，主要是悟空给的数据中的问题）

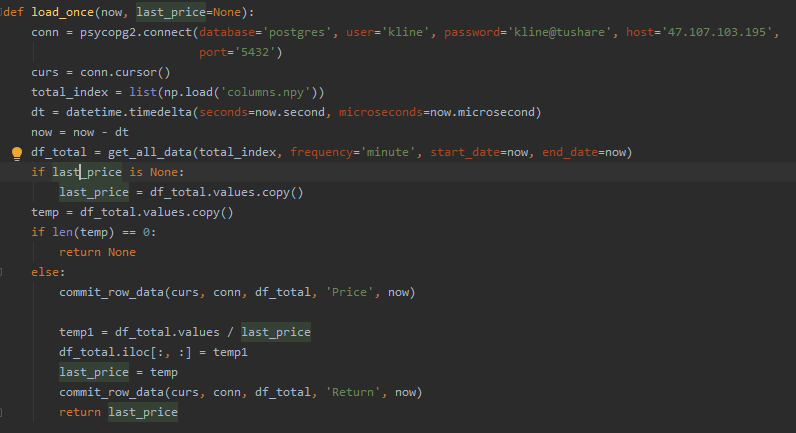


这部分将csv数据转化成numpy并封装成可用的数据集。由于数据集中的mean和stdev在预测阶段会使用到，所以还是把他们都保存起来。

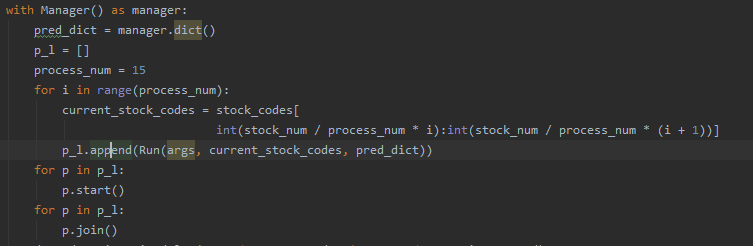
剩余代码部分都是DA\_RNN模型的训练部分，具体细节就不一一说明了，可以直接参考原文献。

预测：

该部分主要完成的功能是：当挂起该脚本后，我们会每隔1分钟从聚宽平台读取一次数据，该数据为当前这一分钟沪深300及其对应指数在这一分钟的价格。脚本首先会将这个价格以及处理过后的涨跌数据都存入数据库中，而后会对每一支股票都进行趋势预测，并将他们的上涨概率也存入数据库。如果时间处于闭市阶段，那么脚本会休眠一分钟，到下一分钟继续进行该过程。



首先讲下load\_once函数，该部分输入为时间以及上一分钟的股指价格，输出为当前时刻的股指价格，同时将这一分钟的价格和涨跌分别保存到数据库中的Price和Return工作表中。



后续的inference部分主要就是读取对应股票的模型并进行预测，由于单线程处理没有充分利用GPU的内存和使用率，所以采用多线程处理，目前单卡可以同时处理15个线程。

这里有一个小问题是，目前的处理方法在预测完一支股票后需要对tensorflow图进行重置，再读取新的股票对应的模型图，最优的做法是同时处理300个线程，每只股票都对应一个线程，这样就可以省去图读取的时间，效率提升会很大，但是目前的硬件条件不太支持。

预测过程的具体细节不再赘述。

微调：

微调部分大体上和训练部分相同，主要变化有以下两处：

1. 训练数据部分在原本悟空给的数据基础上添加了一些最新的数据，同时原则上在尾部添加多少条，就在头部删去多少条，以保证训练数据保持一个最新的阶段。
2. 微调后的模型和相关文件会覆盖原有的模型以及相关文件，同时原有文件会保存在一个副本文件夹中。