数据集说明

使用2018strain集作为训练集，将每支股票拆分成PastDays×FeatureNums的形式，作为一个batch，并把所有股票的batch共同作为训练集，训练集的shape为(2592287, 20, 29, 1)。

输送到模型当中训练，并针对2018stest中的每单支股票进行预测，预测结果分为三种涨、平（change不超过±5%）、跌三类。

参数是

PastDays = 21

n\_classes = 3

FeatureNums = 29

batch\_size = 32

num\_epoch = 2

n\_filters = 32

kernel\_size = [5, 5]

pool\_size = [2, 2]

模型结构为LeNet框架：

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d\_1 (Conv2D) (None, 16, 25, 6) 156

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 8, 12, 6) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 4, 8, 16) 2416

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 2, 4, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 120) 15480

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 84) 10164

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 3) 255

=================================================================

Total params: 28,471

Trainable params: 28,471

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

因为训练集过于庞大，所以训练每一轮（即一个epoch）的训练速度很缓慢，大概需要50分钟左右，【也就是如果epoch=3（粗略来看应该在第5个epoch左右可以达到预测准确率的最高值）训练完全部1479支股票也至少需要3\*50min\*1479=221850min=3697.5h≈154days】

所以我尝试预测000002这支股票，epoch=3时模型在训练集上的准确率就可以达到94%以上，在测试集上的准确率可以达到90.23%以上。

但目前来看，此代码具有相当的可行性，预测准确率很高，运行CNN stock all.ipynb代码即可得到

（1）生产保存每个股票的预测情况trend/code.csv文件，日期、code、预测涨跌、真实涨跌4个字段；

（2）存储所有股票的F1、accuracy、score，存入统计文件score.csv中，有code, F1, score三列。

CNN stock all.ipynb为执行代码。

将执行代码放置在sgym目录下即可。

~~Train\_data.pkl和train\_label.pkl为存储训练数据和训练标签的文件，可以直接使用如下代码打开文件，并直接~~

~~#打开文件~~

~~pickle\_open = open('train\_data.pkl','rb')~~

~~train\_data = pickle.load(pickle\_open) #重新导入数据~~

~~pickle\_open = open('train\_label.pkl','rb')~~

~~train\_label = pickle.load(pickle\_open) #重新导入数据~~