

说明

运行顺序：

s1 -> s2 -> s3_training_cls.py -> s4

先说这篇文章主要在做的事情，使用cgan作为抽样器来对模型进行调参数。我主要实现了cGAN在时间序列上的模拟算法。使用了文中表4的Reg Tree参数，实现了表5中表现好的Reg Tree算法策略，以实现文章的主要逻辑为主。

cGan

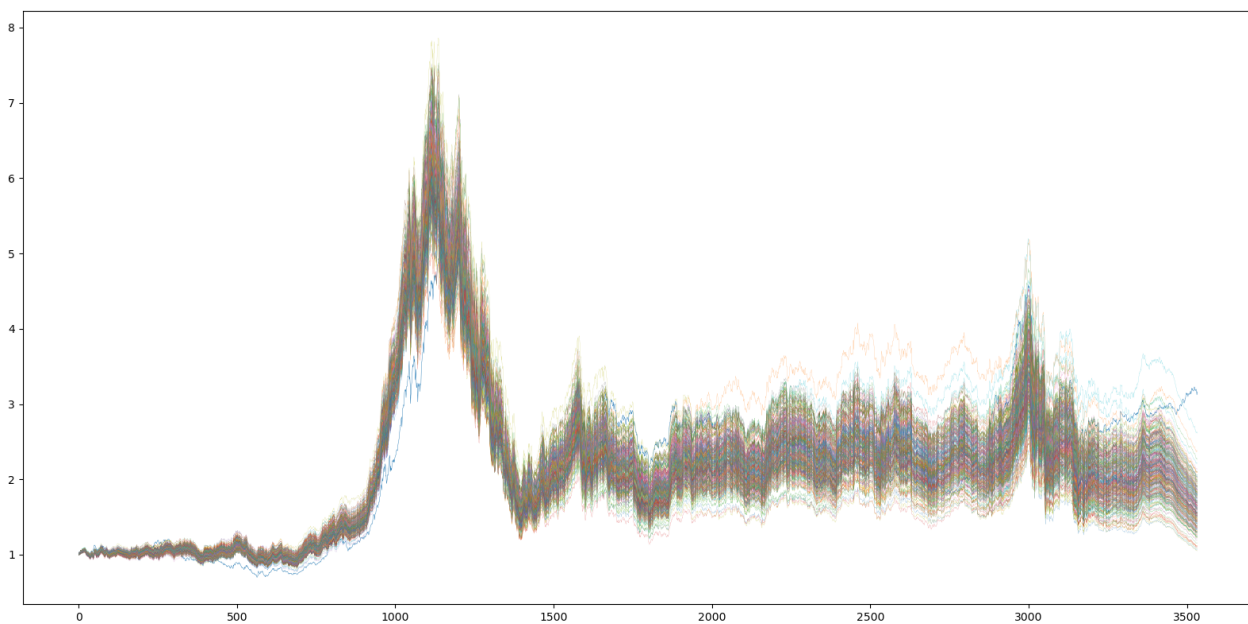
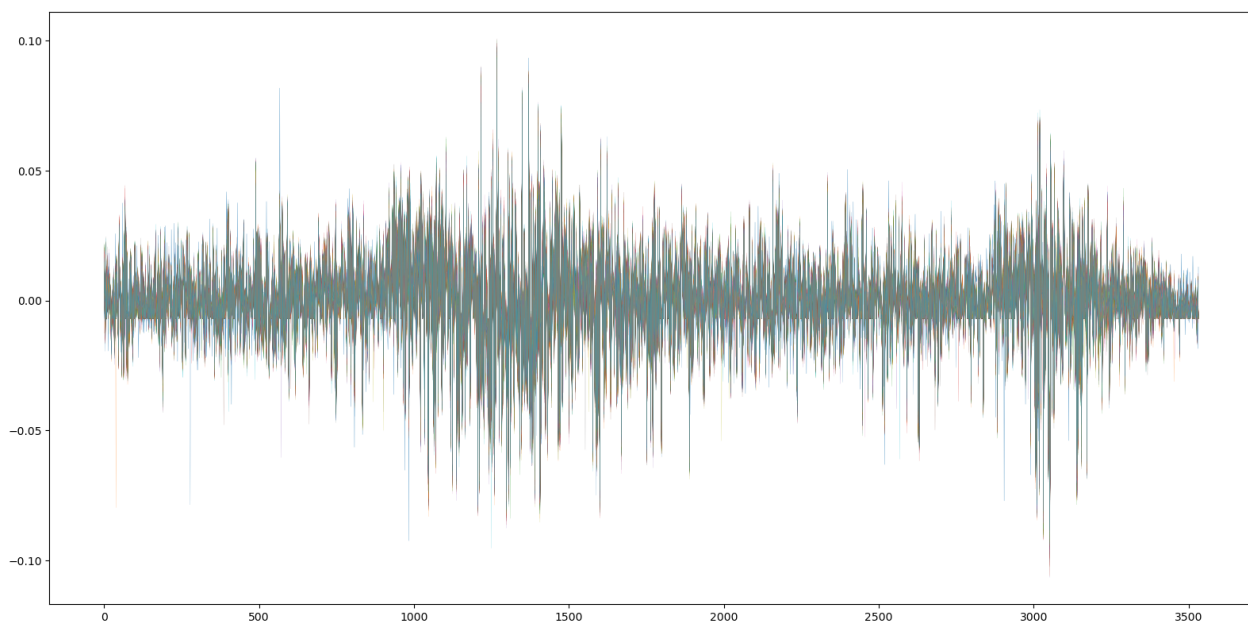
以沪深300指数作为实验标的，使用前80%的数据作为训练数据。在前80的数据上进行cGan模型的训练。

关于文中cGAN的细节：

1. 条件的添加方式，文中并没有给出cGAN模型的生成器与判别器如何接受条件输入的，通过一些资料的查阅，发现最为普遍的方法就是直接使用一个神经网络去接受条件，即，条件向量作为变量与一个高斯分布的随机向量一起作为输入到神经网络中变量。
2. 与一般的cGAN不同，一般的GAN是生成出来一个向量，本文cGAN是训练生成一个数。这个数是下一期的收益率。
3. cGAN一般而言需要成对的数据，但是，文中的cGAN不要成对的条件数据。对条件数据的理解可以认为是，（真实数据1 | 条件1）= 1，（生成数据1 | 条件1）= 0，（真实数据2 | 条件1）= 0。即，即便是真实数据，但是不是真实数据的条件依然要判别为0。所以，cGAN的输入应该是成对数据。

这次实验长时间纠结在成对条件的构造中，但是文中并没有使用成对条件说明以及成对条件的构造。通过查阅一些资料，发现不使用成对条件，也是成立的。成对条件的应用场景是为了限制在不同模型下转换可能出现的问题。这种场景多出现在语音模型与视觉模型的结合使用中，且文中的一些参考文献也没有给出成对条件的说明。所以，我们没有使用成对条件来构造cGAN。

实验结果



两张图片对应了文中图2的cGAN训练结果。对应s1,s2两份代码。s1代码中35-41行是具体参数，可以进行不同尝试。

策略部分



我们使用Reg Tree作为基策略。与文中不同的是，我们直接以分类模型来作为信号的发出模型。文中是以回归为主。我也做了回归的实验。但是效果不如分类模型来的好，且回归模型出现了越简单的参数效果越好的情况。hs300指数的预测性其实不好。而分类模型在一定程度上会比回归模型更好的容忍噪音。

s3_training_cls.py 与s4代码是策略部分。在策略部分，我们使用相同的参数搜寻与相同的模型，对应使用不同的数据。一种数据是真实的数据，一种数据是通过cGAN生成的虚拟数据。

从实验结果能看到，使用真实数据下，模型策略几乎与指数的数据保持一致。说明，hs300中噪音太多，其实并不好训练，其模型训练几乎没有效果。而使用cgan数据训练的结果明显出现了更好的表现。

该结果对应文中第四部分的case I的部分结果。case I中的其他结果基本就是使用不同的参数来得到类似的结果。

注：

cGAN的训练是随机的，**cGAN**生成的数据也是随机的。其中算法充斥了各种随机性。较普通的神经网络，cGAN的随机性更大。且在以上的结果是我根据文章的算法实验跑实验得到的结果，并不一定能保证每次都会有类似的效果。cGAN做为生成模型，本身training就是以很难训练而闻名。