卷积神经网络在选股策略中的应用

基于《华泰人工智能系列研报》的初步探索

江楚仪，徐思远

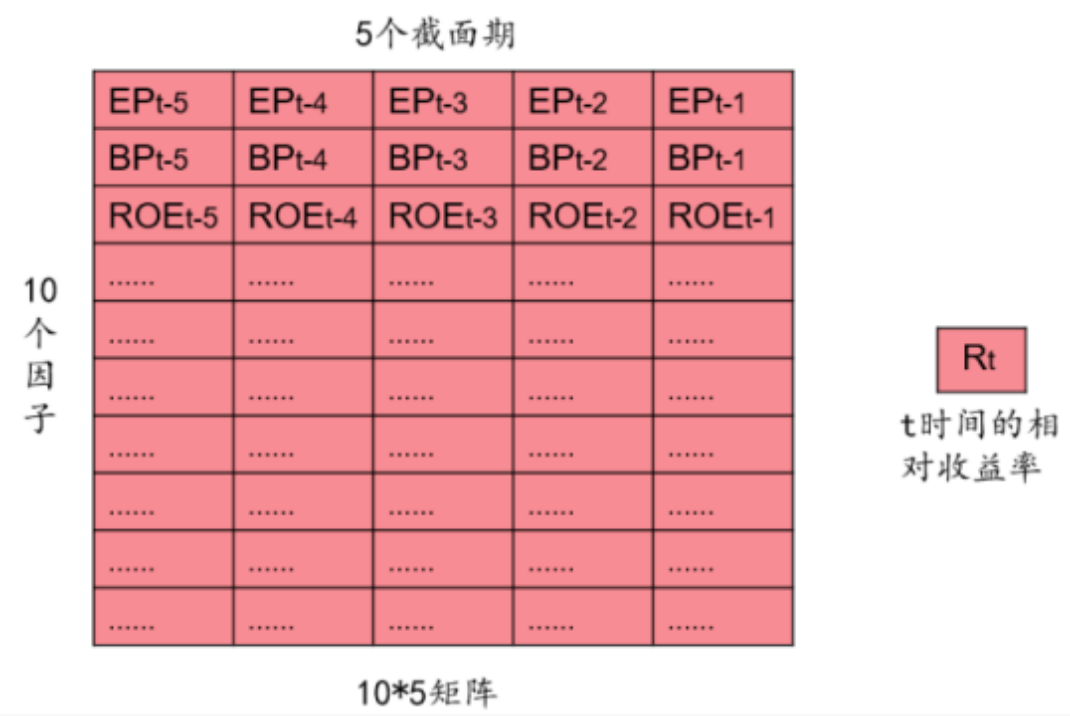
上海财经大学

卷积神经网络应用背景

- 卷积神经网络（CNN）目前被主要应用于图像识别分割与自然语言处理等领域。
- 其主要应用方法是通过自动从大规模的数据中学习特征，并把结果向同类型未知数据泛化。
- CNN能提供不同的卷积核进行特征提取，并结合池化层的降维能力。这使CNN既不会遗漏重要信息，数据复杂度提升也较小，能得到比人为提取特征更完美的结果。
- 我们可以借助CNN“特征提取”和“特征降维”的特性，改进选股策略。

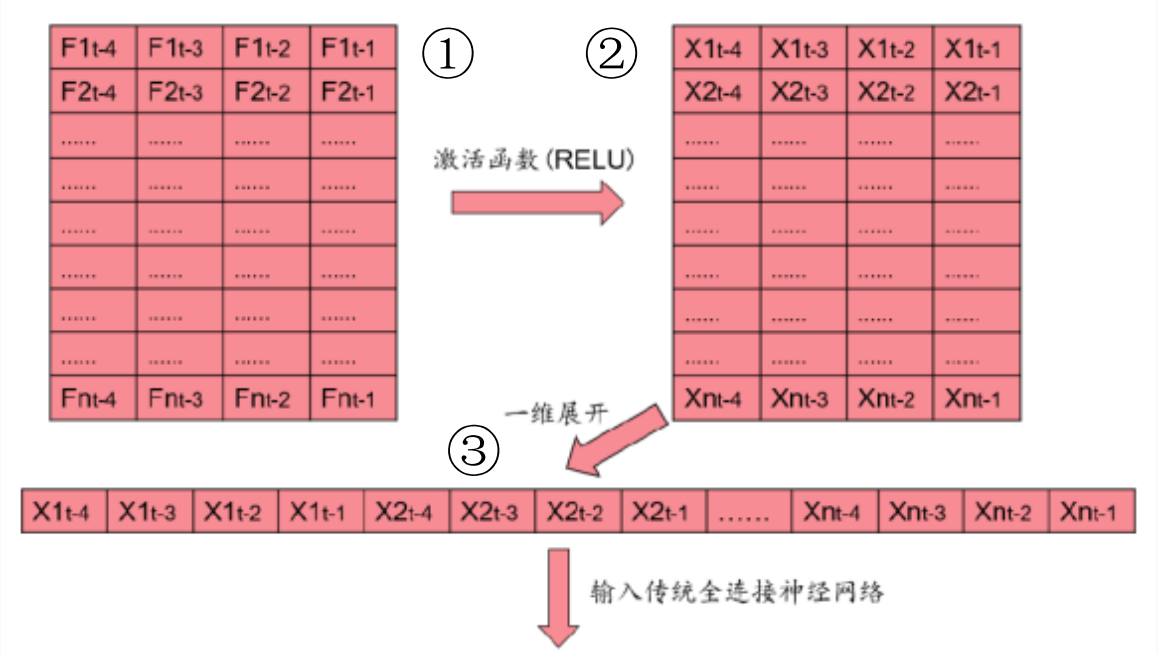
构建能应用于CNN的因子图片

- 为了利用CNN方便处理二维数据的特性，我们可以将因子数据组织成二维形式：



因子图片的卷积运算过程

1. 卷积运算选取与卷积核大小相同的区域进行运算得到卷积结果，并依次水平、垂直遍历图片，生成9*4的结果①。
2. 对①使用激活函数，得到结果②。
3. 对②进行一维展开，得到卷积处理后的因子向量③。
4. 将③输入到全连接神经网络中，按全连接神经网络的优化方法优化网格参数。

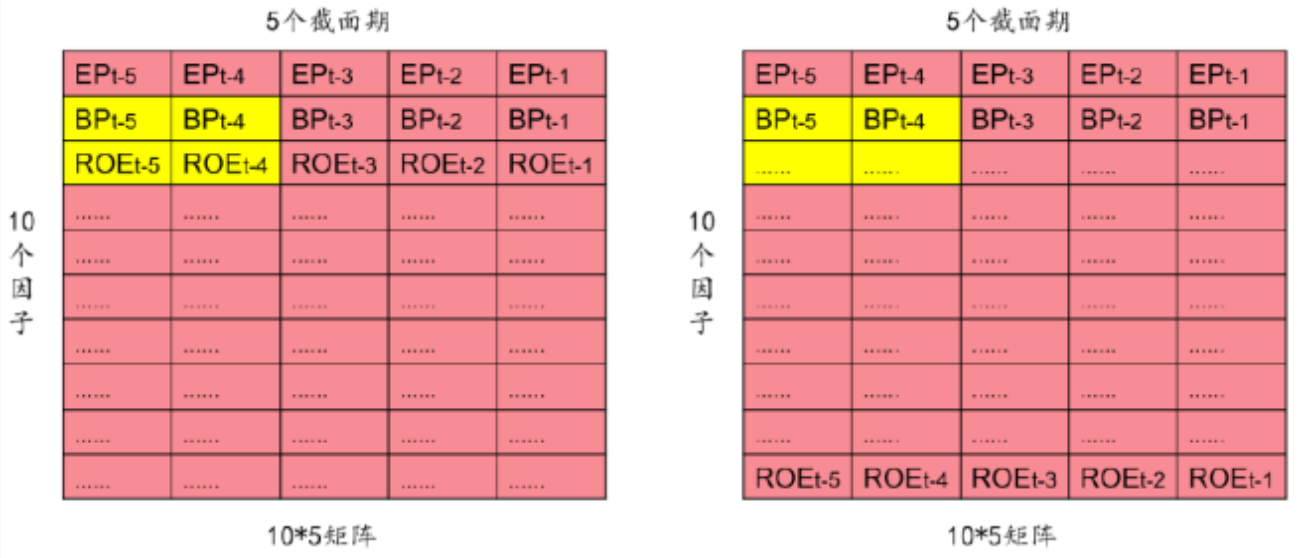


对模型设置的几个问题

1. 为什么只使用一层卷积层？
    - 卷积层数量和训练数据性质有关。在多因子选股中，当因子都具有明确意义时，卷积层的作用是对因子之间进行非线性组合，因此使用一层卷积层已经足够满足需求。
    - 如果使用的股票数据更加原始，多层卷积层效果可能更好。
2. 为什么没有使用池化层？
    - 池化层本质是对卷积结果进行“模糊化”，图像识别的输入层像素维度很高，池化层能在损失极少量信息情况下归纳出图片区域的局部特征。
    - 但选股应用中，因子都具有明确意义，“池化层”会损失一些精细信息。

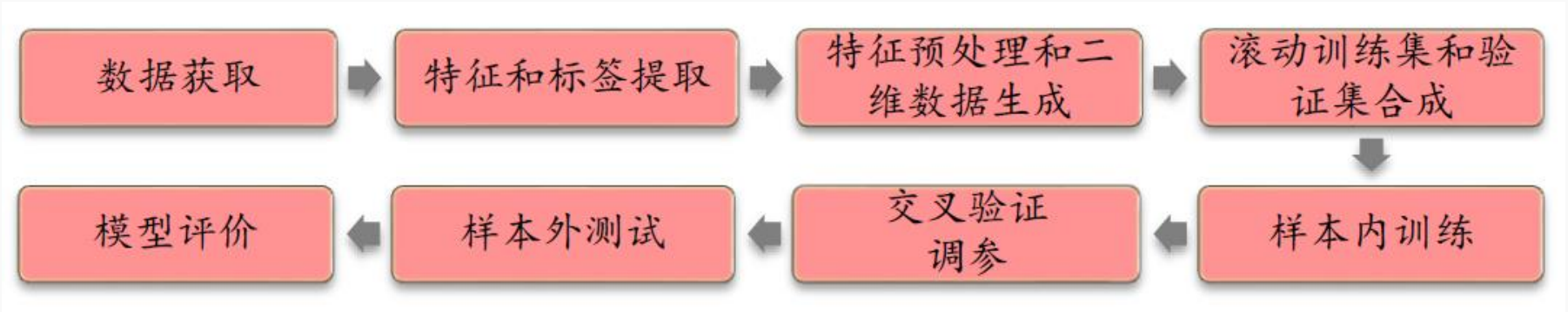
对模型设置的几个问题

3. “因子图片”中因子的排列顺序对结果是否有影响？
   - 因子的排列顺序会对训练和预测的结果造成影响，这也是CNN相比于全连接神经网络等具有根本区别的地方。
   - 模型只能对相邻因子进行卷积运算，所以不同的因子排列顺序会影响卷积核中权重的训练。
   - 文章认为，合理的排布方式是将属于同一大类因子的细分因子放在相邻位置；对不同的大类因子，将可能有相互作用的大类因子放在相邻位置。



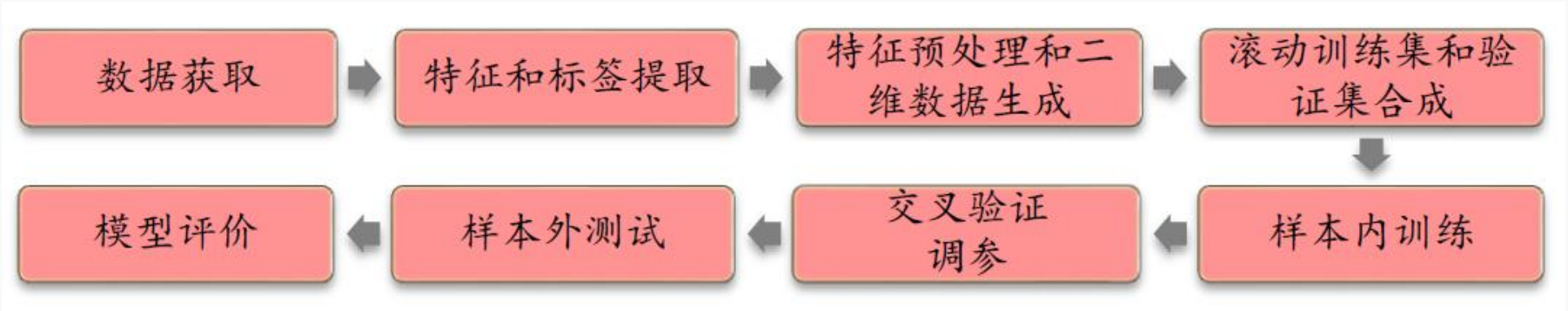
模型测试流程

1. 数据获取：剔除每个截面期下一交易日停牌的股票，将每只股票视作一个样本。
2. 特征和标签提取：从因子池中选取有效因子作为样本的原始特征，使用下一期的收益率作为标签。
3. 特征预处理和二维数据生成：
    - 中位数去极值：设第T 期某因子在所有个股上的暴露度序列为D_i，D_M为该序列中位数，D_{M1}为序列$|D_i - D_M|$的中位数，则将序列D_i中所有大于$D_M + 5D_{M1}$的数重设为$D_M + 5D_{M1}$，将序列D_i中所有小于$D_M - 5D_{M1}$的数重设为$D_M - 5D_{M1}$。
    - 二维数据生成：将某只股票多个截面期的因子数据组织成类似于图片的二维数据。



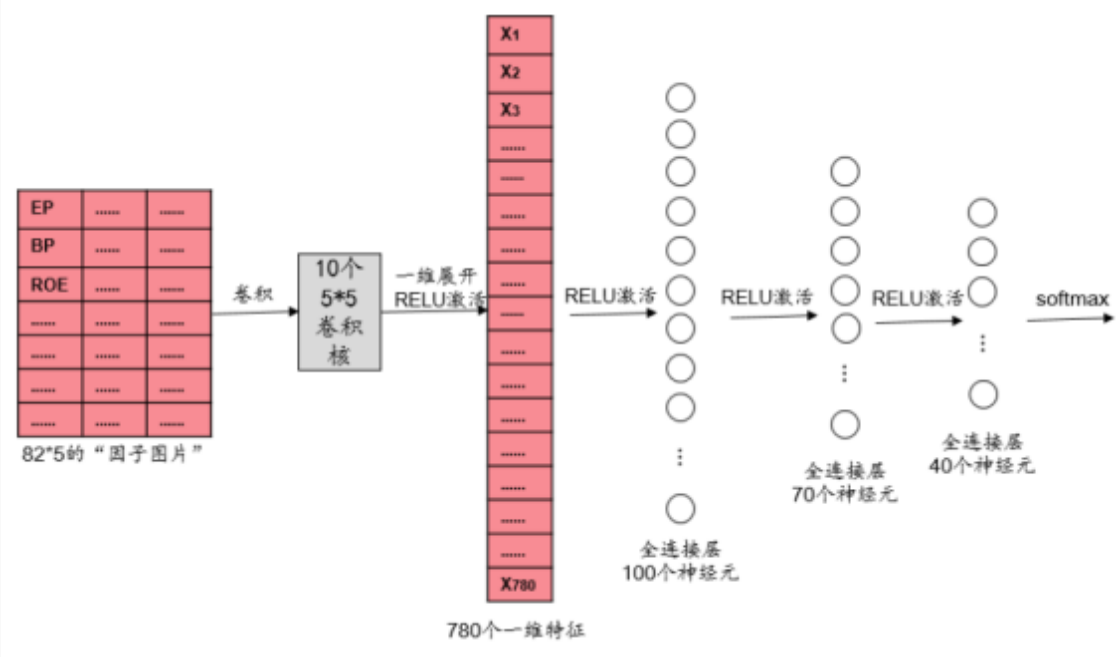
模型测试流程

4. 滚动训练集和验证集合成：使用与之前作业相似的滚动窗口进行持续训练和预测。
5. 样本内训练：使用卷积神经网络对训练集进行训练。
6. 交叉验证调参：随机取10%样本内的数据作为验证集，当验证集上的loss 达到最小时，停止训练。
7. 样本外测试：确定最优参数后，以T 月末截面期所有样本预处理后的特征作为模型的输入，得到每个样本的预测值f(x)。将预测值视作合成后的因子，进行回测。



测试模型设置

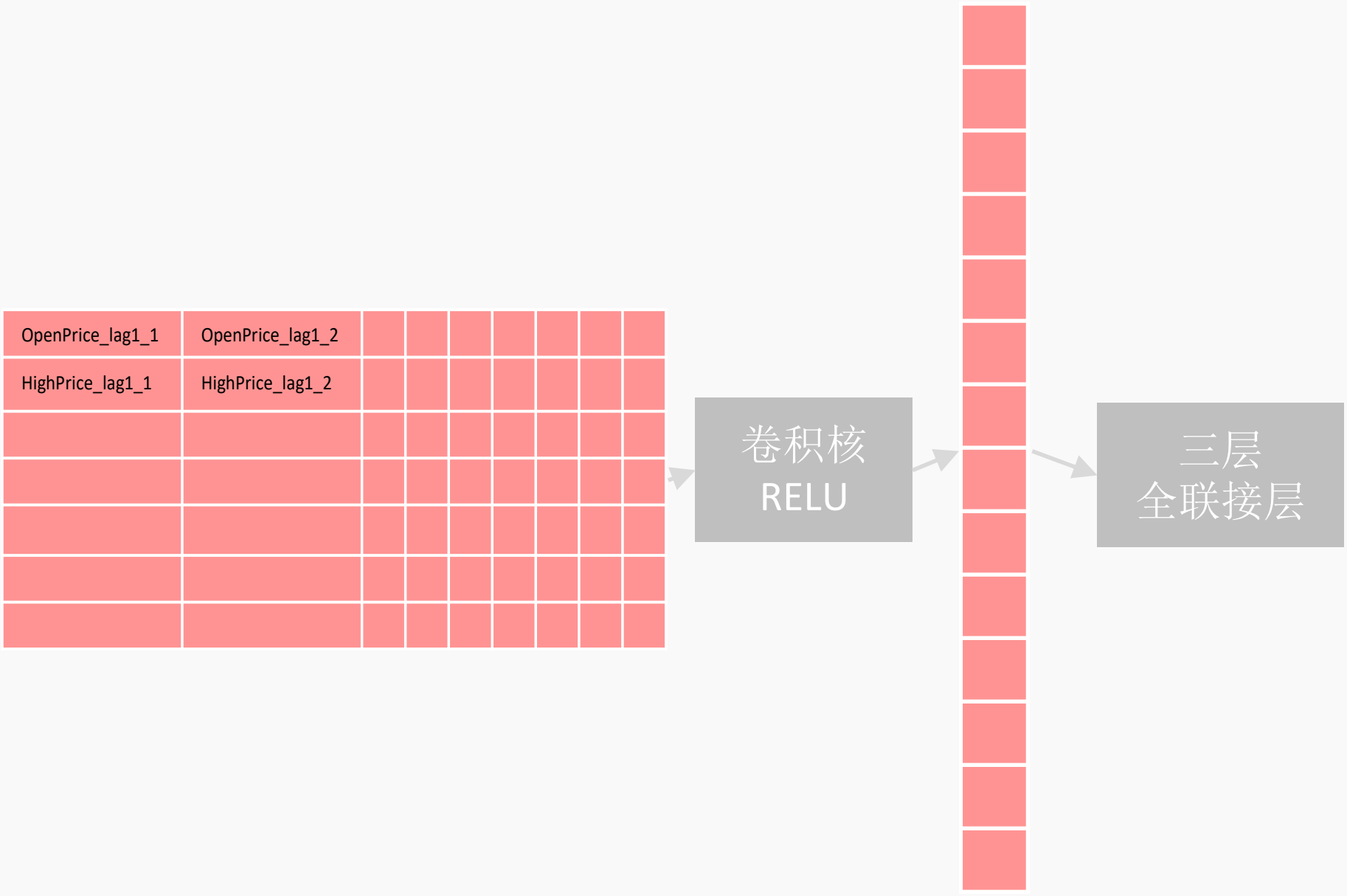
1. 输入数据：股票样本构成“因子图片”，其标签为1或0（涨或跌）。
2. 卷积层：一层卷积层，包含10 个5*5 大小的卷积核。卷积核权重使用xavier 初始化方法。
3. 池化层：没有池化层。
4. 全连接层：3 层全连接层，分别包含100、70、40 个神经元，连接权重使用truncated_normal 初始化方法。
5. 优化器和学习速率：RMSProp，0.001。
6. 损失函数：交叉熵损失函数（二分类）。



代码逻辑

1. 转化初始数据
2. 构建一个字典key为innercode，value为嵌套字典
3. 嵌套字典的subkey为7天的平均return，value为9行7列的矩阵即9个解释变量，7列为7天，这一个value为一个截面，响应变量为该value的key即7天平均的return
4. 针对每个截面，使用卷积核卷积后再使用RELU激活，接着转化为一维向量
5. 构建一个二层的循环
    1. 第一层为循环遍历所有的innercode
    2. 第二层为循环滚动窗口，设置窗口大小为30，滚动遍历每个innercode的所有的return（subkey）进行三层全联接神经网络训练，再预测
6. 得到预测值大于0的输入key（innercode）

代码逻辑


OpenPrice_lag1_1
OpenPrice_lag1_2
HighPrice_lag1_1
HighPrice_lag1_2
卷积核
RELU
三层
全联接层

复现过程及出现的问题

1. 因子数据问题：我们尝试使用朝阳永续数据库，发现数据库内因子质量参差不齐，数据内有很多null值需要处理，同时各个因子的时间尺度不一致，可能对模型结果造成较大影响。
2. 算力问题：由于显卡限制，我们在复现过程中缩小了股票池和时间范围。
3. 张量结构问题：在神经网络训练时，没有找到正确的张量结构。
4. 两层循环中，先循环innercode再循环滚动窗口与先循环滚动窗口再innercode是否有区别，如何将一维展开合并入循环，或如何更好地利用内存？
5. 卷积核的数量和大小选择问题。

Conclusion