

基于人工智能的量化交易设计报告

刘锦坤

2022013352

2024 年 6 月 25 日

摘要

本项目旨在设计一个基于人工智能的量化交易策略。通过建立一个 LSTM 神经网络模型，根据过去一段时间内的股票交易数据，预测接下来一段时间内股票的涨跌情况，从而制定特定的交易策略，实现经济收益。我在项目中主要负责项目的整体设计，数据采集，数据预处理，模型设计，模型搭建和训练的代码实现，因此本报告将主要聚焦于这一部分。

目录

1	项目背景	2
2	项目设计	2
2.1	基本目标	2
2.2	模型方案	2
3	数据采集	2
4	模型选择	3
4.1	RNN&LSTM 模型	3
4.2	最终模型	4
5	数据预处理	4
5.1	原始数据	4
5.2	量比处理	4
5.3	数据标准化	4
5.4	数据集划分	4
6	部署和训练	5
7	回测收益	5
7.1	交易策略	5
7.2	收益结果	6
8	分工	6

1 项目背景

在股票市场中，投资者常常将过去一段时间内的股票交易数据进行处理，从而得到为若干技术指标。常见的有移动平均线（MA）、指数平滑移动平均线（EMA）、布林带（Bollinger Bands）、平滑异同移动平均线（MACD）等。投资者会根据这些指标的变化形态，制定特定的交易策略，以获得更好的投资收益。根据特定的策略进行交易，这即是量化交易的基本思想。

但是根据传统的技术指标交易存在明显的不足。首先，这些技术交易指标的计算全部依赖利用某一种显式的计算方法进行计算的，这样的显式的计算方法本身很有可能并不能捕捉到股票市场全部的交易规律。其次，股票市场是典型的二级响应系统，即市场上的交易行为会对市场本身产生影响，而这种影响又会反过来影响交易行为。这就导致任何一种依靠传统技术指标的交易策略，尤其是被大量投资者采用的交易策略，都会面临市场反作用导致的失效性的问题。

而人工智能技术的发展，尤其是神经网络模型的发展，为量化交易提供了新的思路。神经网络通过隐式的表达方法，能够对股票交易市场中大批量的过往交易数据进行学习，从而捕捉到更全面的交易规律。同时由于这样的交易策略并不显式的依赖于任何传统的技术指标，因此更有可能避免传统技术指标交易策略的失效性问题。基于这种想法，本项目将尝试设计一个基于人工智能的量化交易策略。

2 项目设计

2.1 基本目标

最终制定的项目基本目标为：根据过去一段时间内（days_before）一支股票的交易数据，判断接下来一段时间（days_after）内，股票呈现多头趋势还是空头趋势（股价是上涨还是下跌），根据模型的判断买入和卖出股票，实现经济收益。

2.2 模型方案

而施行该目标的基本方案为：建立一个 LSTM 神经网络模型，模型输入为过去一段时间的交易数据，预期输出接下来一段时间内股票上涨和下跌的概率。

3 数据采集

由于我们的基本实现方案是利用神经网络模型从过往的交易数据中学习特征，从而应用于未来的交易预测，这就需要大量的过往交易数据。在数据获取和整理上，我们外联了聚宽量化投研平台（JoinQuant 平台），由聚宽为我们提供过往股票交易数据。可以通过其提供的 Python 的 API 访问获取过往股票交易数据。用于下一步的数据处理。



图 1: JoinQuant 平台

4 模型选择

4.1 RNN&LSTM 模型

股票数据属于典型的时间序列数据，查阅相关资料发现，RNN（Recurrent Neural Network）类神经网络如图 2 以其独特的构造适合时间序列的模型。

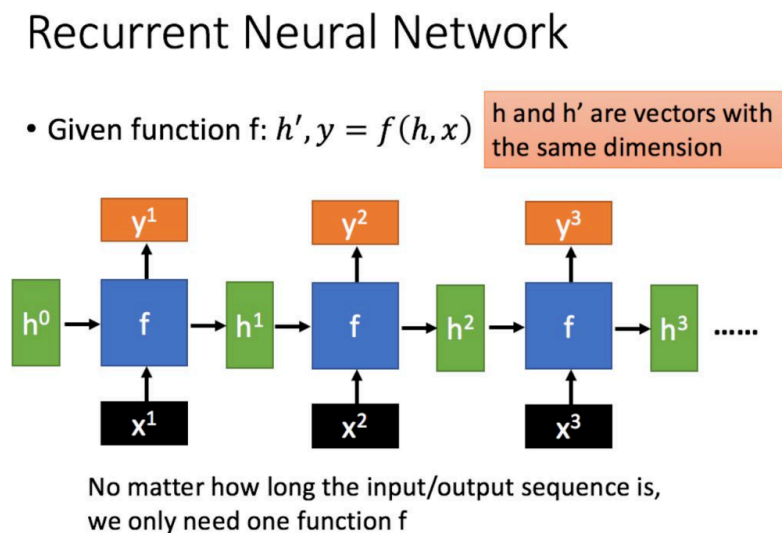


图 2: RNN 模型示意图

但是 RNN 模型存在对于长距离关系捕捉的不足，序列前端的信息可能在循环传递中被遗忘，对于股票来说，以前的一次暴跌和暴涨很可能就会影响后续投资者的信心，因此我们最终采取了 LSTM（Long Short Term Memory）网络如图 3。

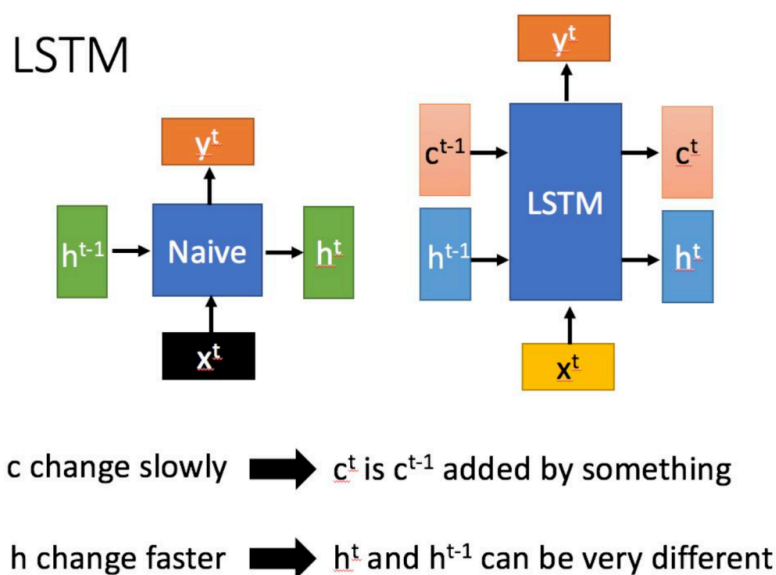


图 3: LSTM 模型示意图

4.2 最终模型

最终，设计模型架构如图 4 所示。输入数据先直接输入到 LSTM 网络中，取 LSTM 层最后一个神经元的输出，然后通过全连接层经过逻辑回归（SoftMax）输出预测结果（涨跌概率）。

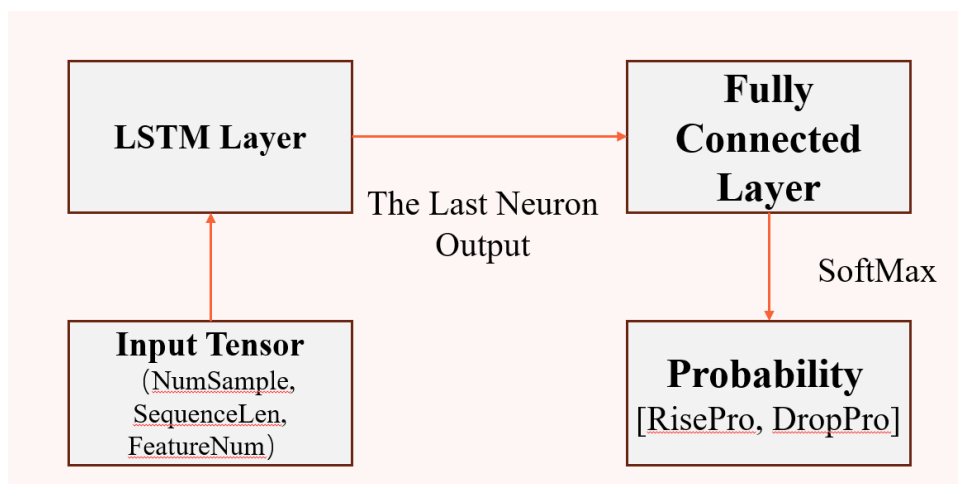


图 4: 模型架构

5 数据预处理

5.1 原始数据

根据聚宽提供的数据，最终汇集整理了目标股票在过去每天的开盘价、收盘价、盘中高价、盘中低价、成交量、成交额、前一天收盘价、成交均价八大数据作为原始特征数据。

5.2 量比处理

对于神经网络模型的训练，好的数据预处理可以使得神经网络模型更好的发掘数据中的特征，在数据预处理阶段，我们对数据进行了如下量比处理：

借鉴股票交易市场投资者常讨论的几个技术形态，（例如放量上涨，缩量上升），并且考虑到中国 a 股市场的特性，认为股票量、价的绝对值并不具有参考意义，而是采用了量比方案进行数据处理，即对于上述八大原始数据，我们以每一天的原始数据和前一天的比值作为输入特征。

5.3 数据标准化

神经网络模型对数据的尺度敏感，因此我们对输入数据最后进行了标准化处理，即对于每一个特征，我们将其利用 MinMax 标准化方法，将其缩放到 [0,1] 之间。

5.4 数据集划分

为了使得回测有意义，我们严格的将训练集和测试集按照时间先后分开，仅在训练集上进行训练，最后在测试集上进行收益回测，保证不发生数据的泄露。

6 部署和训练

最终我们利用 Pytorch 框架搭建了上述模型训练，相关细节如下：

- 由于问题本身接近分类问题（看涨或看跌），所以采用了 CrossEntropyLoss（交错熵）作为损失函数。
- 采用 Adam 优化器进行模型参数的优化。
- 为了克服过拟合现象，采取了 dropout 手段，在训练时会随机丢弃部分神经元。
- 为了加速训练，使用了本地的 RTX3060 和 RTX4060，利用 CUDA 进行了 GPU 加速计算。
- 为了获得更好的回测效果，我们进行了大量超参数的调节，包括 days_before, days_after, batch_size（训练批次大小），hidden_size（LSTM 网络隐藏层个数），num_layer（LSTM 网络的堆叠层数），dropout（训练时的丢弃概率）等。

7 回测收益

这一部分，我们以股票代码为 002765.XSHE 的蓝黛科技股票为例，适当调节超参数，在训练集上训练模型，再根据模型制定特定的交易策略后，在测试集上进行了回测。

7.1 交易策略

制定的交易策略如图 5 所示。

- 当模型预测的上涨概率大于 P_b 时以收盘价买入。
- 买入后若模型预测上涨概率小于 P_s 卖出，或跌幅大于 20% 止损。

```
P_b = 0.7 #上涨概率大于多少买入
P_s = 0.3 #上涨概率低于多少卖出
ans = 0 # 0表示未买入, 1表示已买入
trade = np.zeros(len(y_pred))
t = range(len(y_pred))
k = price[0]
for i in range(len(y_pred)):
    if ans == 0 and y_pred[i][1] >= P_b:
        ans = 1
        trade[i] = 1
        k = price[i]
    if ans == 1 and y_pred[i][1] <= P_s:
        ans = 0
        trade[i] = -1
# 止损策略
if ans == 1 and ((k - price[i]) / k >= 0.20):
    ans = 0
    trade[i] = -2
```

图 5: 交易策略

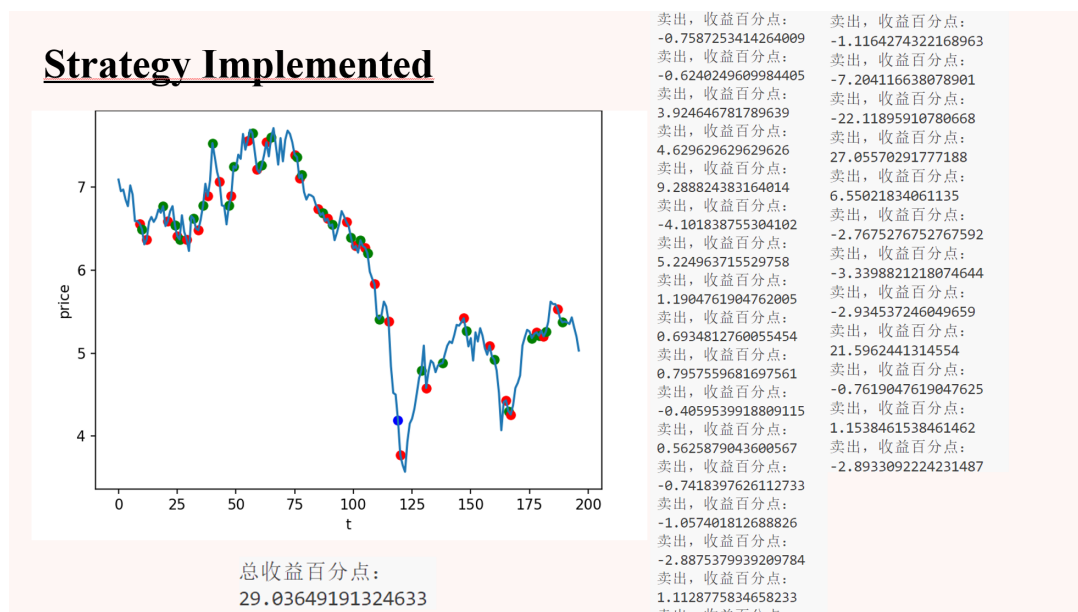


图 6: 收益曲线

7.2 收益结果

最终利用制定的策略，在测试集上回测，收益效果如图 6 所示。其中红色的点代表买入点，而绿色的点代表卖出点，蓝色的点代表止损卖出点，最后可以看到，在测试集约 200 个交易日中，在这样一支整体下行的股票上，我们的策略实现了约 29% 的盈利，这表明我们的策略确实是有效的。

8 分工

项目的整体分工如下：

- 刘锦坤：负责项目的整体设计，数据采集，数据预处理，模型设计，模型搭建和训练的代码实现。
- 郭锐冰：负责模型超参数的调节，交易策略的制定，回测收益的计算。
- 陈飞扬：负责最终展示 PPT 的美化设计。