

2023 年 5 月 6 日

时序模型+回归模型因子策略

机器学习因子系列

投资要点：

► 量化交易策略

机器学习量化交易策略的制定，是从海量历史数据中使用计算机强大的处理能力，挖掘并分析出那些能够为投资者带来收益的各种大概率可行的投资方式来实现的。通过数学模型对这些策略进行分析并加以验证，以期望让投资者获得更稳定的预期收益。

► 时序模型和机器学习回归算法结合使用有利于提高预测准确性

时序模型可以用于处理具有时间序列的数据，并从中获取相关的信息，为机器学习回归算法提供更精确的输入。与此同时，机器学习回归算法可以进一步加强时间序列数据的分析能力。二者结合使用可以带来更好的数据预测准确性，带来更好的决策结果。

► Lstm 和线性支持向量回归算法结合使用对提升平均皮尔逊系数效果最佳

Lstm 模型具有记忆单元和门控机制，可以有效处理长期依赖关系，因此在时序数据建模方面表现出色。回归算法不胜枚举，各回归算法胜任的问题领域不同。但在本问题上，经实验验证，线性支持向量回归算法优于其他常见的回归方法。

风险提示

模型基于对历史数据统计，仅作为投资参考。

评级及分析师信息

分析师：杨国平

邮箱：yanggp@hx168.com.cn

SAC NO: S1120520070002

分析师：王祥宇

邮箱：wangxiang1@hx168.com.cn

SAC NO: S1120520080004

分析师：杨兆熙

邮箱：yangzx@hx168.com.cn

SAC NO: S1120523010001

正文目录

1. LSTM 模型基本理论	5
1.1. 输入门、遗忘门和输出门	5
1.2. 候选记忆元	6
1.3. 记忆元和隐状态	6
2. LSTM 模型建模	7
2.1. 模型合理性的讨论	7
2.2. 模型优缺点的讨论	7
2.3. 策略设计思路	8
2.3.1. 建模方法	8
2.3.2. 数据及参数选择	8
2.4. 策略具体过程	9
3. 集成学习理论	10
3.1. Bagging 算法	11
3.1.1. 随机森林算法	12
3.2. Boosting 算法	12
3.2.1. GBDT	13
3.2.2. XGBoost	13
4. 集成学习建模	14
4.1. 模型合理性的讨论	14
4.2. 模型优缺点的讨论	14
4.2.1. 随机森林	14
4.2.2. GBDT	14
4.2.3. XGBoost	15
4.3. 策略设计思路	15
4.3.1. 建模方法	15
4.3.2. 数据及参数选择	16
4.4. 策略具体过程	17
4.4.1. 随机森林回归	17
4.4.2. GBDT 回归	18
4.4.3. XGBoost 回归	19
5. 线性回归理论	20

5.1. 模型定义与评估.....	20
5.2. 最小二乘法.....	20
5.3. 梯度下降	21
5.3.1. 梯度下降基本思想.....	21
5.3.2. 随机梯度下降	21
6. 线性回归应用.....	21
6.1. 模型的合理性讨论.....	21
6.2. 模型的优缺点讨论.....	21
6.2.1. 多元线性回归模型.....	21
6.2.2. 随机梯度下降线性回归模型.....	22
6.3. 策略设计思路.....	22
6.3.1. 建模方法	22
6.3.2. 数据及参数选择.....	23
6.4. 策略具体过程.....	24
7. 支持向量回归理论.....	24
7.1. 支持向量机.....	24
7.1.1. 支持向量机定义.....	24
7.1.2. 最大间隔超平面.....	25
7.1.3. 支持向量与最大间隔.....	25
7.2. 支持向量回归.....	26
8. 支持向量回归应用.....	27
8.1. 模型的合理性讨论.....	27
8.2. 模型的优缺点讨论.....	27
8.3. 策略设计思路.....	28
8.3.1. 建模方法	28
8.3.2. 数据及参数选择.....	28
8.4. 策略具体过程.....	29
9. 总结	29
10. 风险提示.....	30

图目录

图 1 输入门、遗忘门和输出门	5
图 2 候选记忆单元	6
图 3 记忆元和隐状态	6
图 4 使用 LSTM 模型建模	7
图 5 验证集皮尔逊系数折线图	8
图 6 测试集皮尔逊系数折线图	9
图 7 集成学习示意图	9
图 8 随机森林算法图	11
图 9 GBDT 算法原理	12
图 10 使用 LSTM + 集成学习建模	14
图 11 使用 LSTM + 随机森林建模	15
图 12 使用 LSTM + GBDT 建模	16
图 13 使用 LSTM + XGBoost 建模	16
图 14 使用 LSTM + 线性回归建模	19
图 15 使用 LSTM + 线性回归建模结果	20
图 16 SVM 二分类示意图	21
图 17 SVR 示意图	22
图 18 使用 LSTM + 线性支持向量机回归建模	24
图 19 使用 LSTM + 线性支持向量机回归建模结果	24

表目录

表 1 LSTM 模型的参数说明	8
表 2 RandomForestRegressor 的参数说明	14
表 3 GradientBoostingRegressor 的参数说明	14
表 4 XGBRegressor 的参数说明	14
表 5 LinearRegressor 的参数说明	19
表 6 SGDRegressor 的参数说明	20
表 7 LinearSVR 的参数说明	24
表 8 几种方法结果数据对比	25

1. LSTM 模型基本理论

长期以来，隐变量模型存在着长期信息保存和短期输入缺失的问题，解决这一问题的最早也是最经典的方法是长短期存储器（Long Short-Term Memory, LSTM），他与门控循环单元（GRU）有很多相似之处，有意思的是，虽然 LSTM 比 GRU 要复杂一些，但 LSTM 却早提出近 20 年。

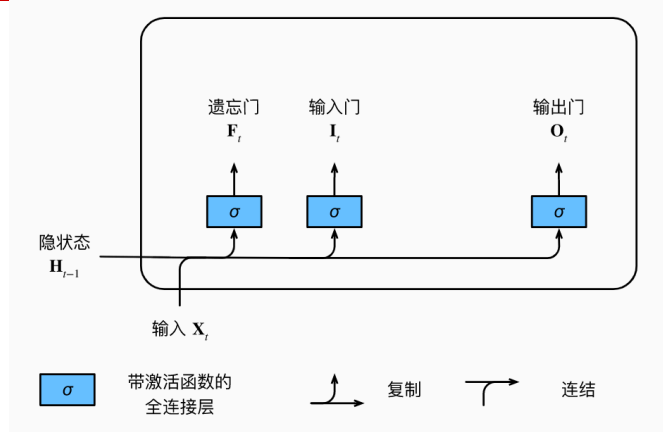
引入自循环的巧妙构思，以产生梯度长时间持续流动的路径是 LSTM 模型的核心贡献。其中一个关键扩展是使自循环的权重视上下文而定，而不是固定的。门控此自循环的权重，累积的时间尺度可以动态地改变。在这种情况下，即使是具固定参数的 LSTM，累积的时间尺度也可以因输入序列而改变，因为时间常数是模型本身的输出。LSTM 已经在很多应用中取得重大成功。

长短期记忆网络的设计灵感来自于计算机的逻辑门。长短期记忆网络引入了记忆单元（memory cell），或简称为单元（cell）。有些文献认为记忆元是隐状态的一种特殊类型，它们与隐状态具有相同的形状，其设计目的是用于记录附加的信息。为了控制记忆元，我们需要许多门。其中一个门用来从单元中输出条目，我们将其称为输出门（output gate）。另外一个门用来决定何时将数据读入单元，我们将其称为输入门（input gate）。我们还需要一种机制来重置单元的内容，由遗忘门（forget gate）来管理，这种设计的动机与门控循环单元相同，能够通过专用机制决定什么时候记忆或忽略隐状态中的输入。

1.1. 输入门、遗忘门和输出门

当前时间步的输入和前一个时间步的隐状态作为数据送入 LSTM 的门中，如图所示。它们由三个具有 sigmoid 激活函数的全连接层处理，以计算输入门、遗忘门和输出门的值。因此，这三个门的值都在(0,1)的范围内。

图 1 输入门、遗忘门和输出门



资料来源：Wind，华西证券研究所

LSTM 的数学表达如下：假设有 h 个隐藏单元，批量大小为 n ，输入数为 d 。因此，输入为 $X_t \in \mathbb{R}^{n \times d}$ ，前一时间步的隐状态为 $H_{t-1} \in \mathbb{R}^{n \times h}$ 。相应地，时间步 t 的门被定

义如下：输入门是 $I_t \in \mathbb{R}^{n \times h}$ ，遗忘门是 $F_t \in \mathbb{R}^{n \times h}$ ，输出门是 $O_t \in \mathbb{R}^{n \times h}$ ，他们的计算方法如下：

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$

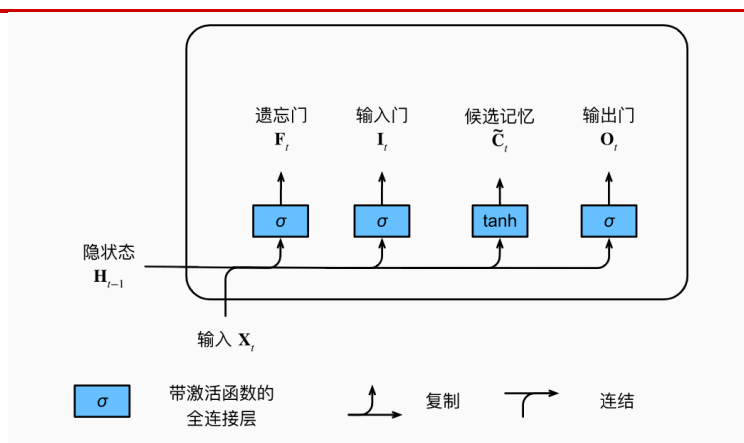
$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)$$

其中 W 是权重参数， b 是偏置参数。

1.2. 候选记忆元

图 2 候选记忆单元



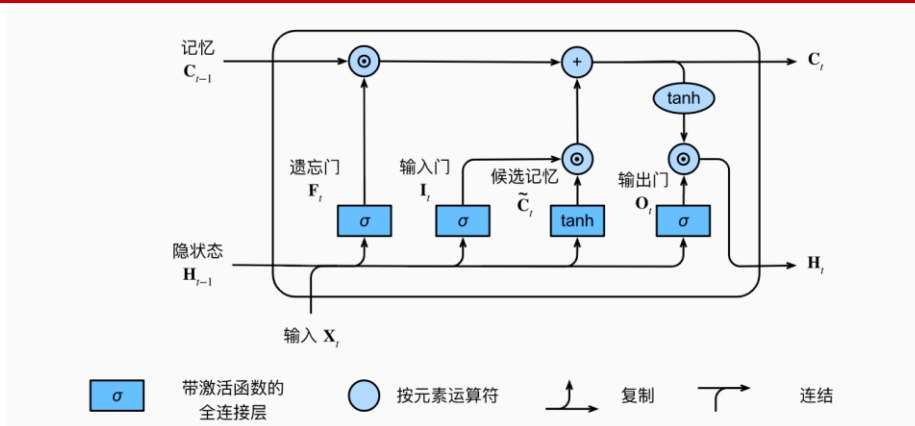
资料来源：Wind，华西证券研究所

候选记忆元 $\tilde{C}_t$ 的计算方法与上面相似，但是使用的 $\tanh$ 函数作为激活，计算方法如下：

$$C_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c)$$

1.3. 记忆元和隐状态

图 3 记忆元和隐状态



资料来源：Wind，华西证券研究所

在LSTM中，有一种机制来控制输入和遗忘：输入门控制采用多少来自记忆元的内容，而遗忘门控制保留多少上一个记忆元的内容。

输出门发挥作用的地方就是隐状态，在长短期记忆网络中，它仅仅是记忆元的tanh的门控版本。只要输出门接近1，我们就能够有效地将所有记忆信息传递给预测部分，而对于输出门接近0，我们只保留记忆元内的所有信息，而不需要更新隐状态。

2. LSTM 模型建模

2.1. 模型合理性的讨论

理论上，股票价格是可以预测的，但是影响股票价格的因素有很多，而且目前为止，它们对股票的影响还不能清晰定义。这是因为股票预测是高度非线性的，这就要预测模型要能够处理非线性问题，并且，股票具有时间序列的特性，因此适合用循环神经网络，对股票进行预测。

虽然循环神经网络（RNN），允许信息的持久化，然而，一般的RNN模型对具备长记忆性的时间序列数据刻画能力较弱，在时间序列过长的时候，因为存在梯度消散和梯度爆炸现象RNN训练变得非常困难。Hochreiter 和 Schmidhuber 提出的长短期记忆（Long Short-Term Memory, LSTM）模型在RNN结构的基础上进行了改造，从而解决了RNN模型无法刻画时间序列长记忆性的问题。

综上所述，深度学习中的LSTM模型能够很好地刻画时间序列的长记忆性。

2.2. 模型优缺点的讨论

LSTM模型优点如下：

1. 改善了RNN中存在的长期依赖问题；LSTM的表现通常比时间递归神经网络及隐马尔科夫模型（HMM）更好。

2. LSTM 通过各种门函数来将重要特征保留下来，能够有效减缓长序列问题中可能出现的梯度消失或爆炸，虽然并不能杜绝这种现象，但在更长的序列问题上表现优于传统 RNN。

LSTM 模型缺点如下：

1. 并行处理上存在劣势，只能从前到后，与一些最新的网络相对效果一般；

2. RNN 的梯度问题在 LSTM 及其变种里面得到了一定程度的解决，但还是不够。它可以处理 100 个量级的序列，而对于 1000 个量级，或者更长的序列则依然会显得很棘手。

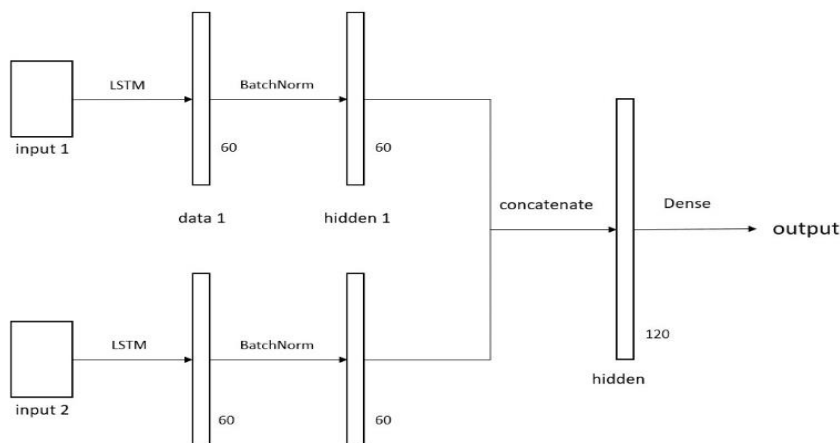
3、计算费时。如果 LSTM 的时间跨度大且网络深，计算量大且耗时。

2.3. 策略设计思路

2.3.1. 建模方法

我们将 LSTM 应用于股票预测，模型结构图如图所示。

图 4 使用 LSTM 模型建模



资料来源：Wind，华西证券研究所

input1 包含了一天股票数据的信息，input2 包含了一周股票数据的信息，他们的维度均为 $c \times 40 \times 6$ (c 为通道数)。将 input1 和 input2 分别通过 LSTM 层，这两个 LSTM 的权重不共享，得到 $c \times 60 \times 1$ 维的矩阵，经过归一化处理之后拼接为一个 $c \times 120 \times 1$ 维的矩阵，此矩阵包含了两个输入的全局状态信息。最后，将结果经过一个全连接层输出一个 $c \times 1$ 维的矩阵，即得到预测结果。

2.3.2. 数据及参数选择

LSTM 模型的重要参数如下：

表 1：LSTM 模型的参数说明

参数	说明
units	输出维度
input_dim	输入维度，当使用该层为模型首层时，应指定该值

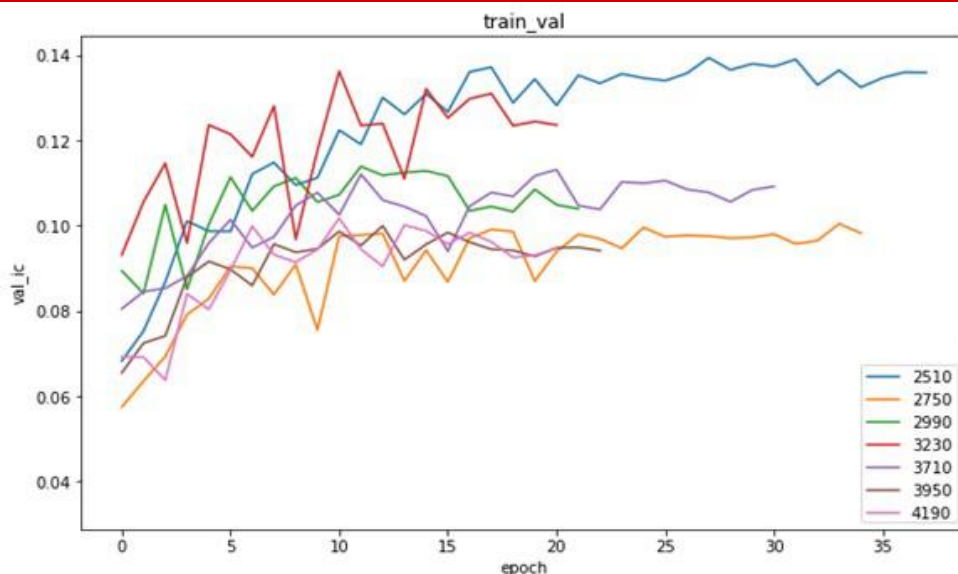
return_sequences	控制返回类型。若为 True 则返回整个序列，否则仅返回输出序列的最后一个输出
input_length	当输入序列的长度固定时，该参数为输入序列的长度。

资料来源：Wind，华西证券研究所

2.4. 策略具体过程

对于上述模型在数据上训练的结果如下图所示，横坐标是每次数据所运行的批次，纵坐标是每一个批次的平均准确率。

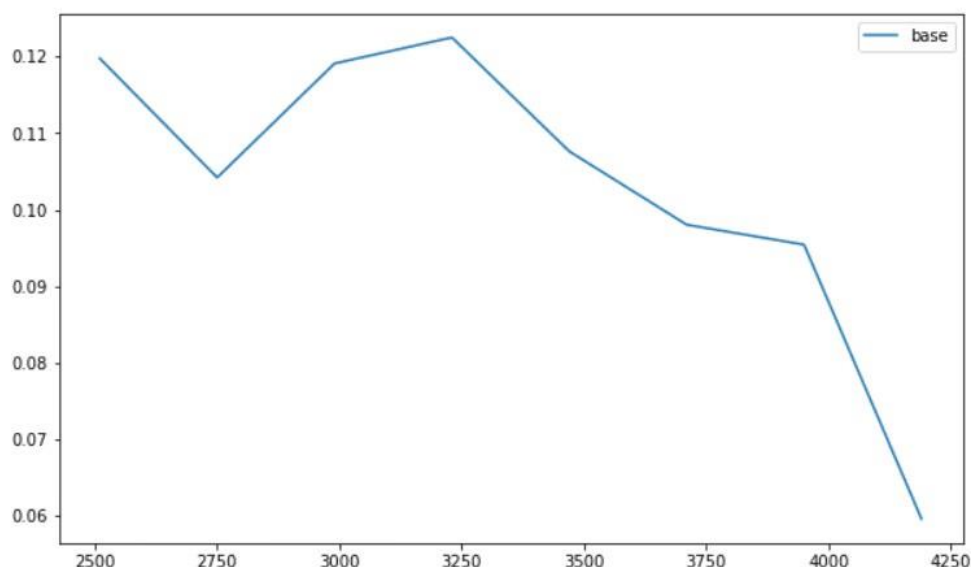
图 5 验证集 IC 折线图



资料来源：Wind，华西证券研究所

该模型的测试集的结果如下。

图 6 测试集 IC 折线图



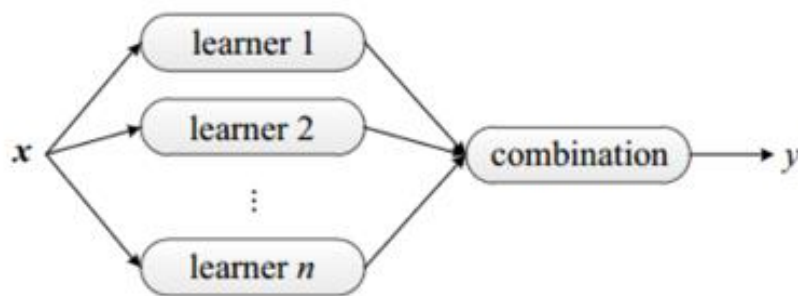
资料来源：Wind，华西证券研究所

以下的方法，我们将在原模型(lstm)的基础上，增加回归模型尝试优化和提高。

3. 集成学习理论

和传统学习方法训练一个学习器不同，集成学习方法训练多个学习器并结合它们来解决一个问题。一个集成由多个基学习器构成，而基学习器由基学习算法在训练数据上获得，它们可以是决策树、神经网络或其他学习算法。大多数集成学习方法使用同一种基学习算法产生同质的基学习器，即相同种类的学习器，生成同质集成(homogeneous ensemble)；同时，也有一些方法使用多种学习算法训练不同种类的学习器，构建异质集成(heterogeneous ensemble)。通常，集成具有比基学习器更强的泛化能力，很大程度上是因为它们能够把比随机猜测稍好的弱学习器(weak learner)变成可以精确预测的强学习器(strong learner)。

图 7 集成学习示意图



资料来源：Wind，华西证券研究所

根据基分类器的生成方式，集成学习方法有两种范式：并行生成基分类器的“并行集成方法” Bagging，以及串行生成基分类器的“串行集成方法” Boosting。

3.1. Bagging 算法

由于聚合独立的基分类器可以显著降低误差，所以我们希望得到的基分类器越独立越好。给定训练集，一种可能的实现是采样得到若干相互没有重合样本的子集，每个子集各自训练基分类器。然而，由于训练数据是有限的，这样得到的子集样本少，不具代表性，使得基分类器的性能受限。

Bagging 采用自助采样生成不同的基分类器。它引入自助采样得到训练子集用于训练基分类器。具体来讲，给定一个样本数为 m 的训练集合，它通过有放回采样得到有 m 个训练样本的采样集。原始样本有的被选中多次，有的未被选中。重复过程 T 次，得到 T 个样本数目为 m 的样本集。对每个采样出来的训练集，使用基学习算法可以得到一个基学习器。Bagging 采用最常用的方法来聚合基分类器，即在分类任务上投票，在回归问题上平均。Bagging 算法如下所示。

输入：数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;

基学习算法 $\mathcal{L}$;

基学习器数 T .

步骤：

for $t = 1, \dots, T$:

$h_t = \mathcal{L}(D, \mathcal{D}_{bs})$. % $\mathcal{D}_{bs}$ 为自助分布

end

输出： $H(x) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T \mathbb{I}(h_t(x) = y)$.

值得一提的是，自助采样赋予了 Bagging 一个额外优势：给定 m 个训练样本，第 i 个样本被选中 $0, 1, 2, \dots$ 次的概率近似为 $\lambda = 1$ 的泊松分布，所以第 i 个样本至少出现一次的概率为 $1 - (1/e) \approx 0.632$ 。即对 Bagging 的任一基分类器，训练时原始训练集中

约有 36.8% 的样本未被使用。此时，这个基分类器的好坏可以通过这些 out-of-bag, OOB 样本估算，继而对 Bagging 算法的泛化误差进行估算。

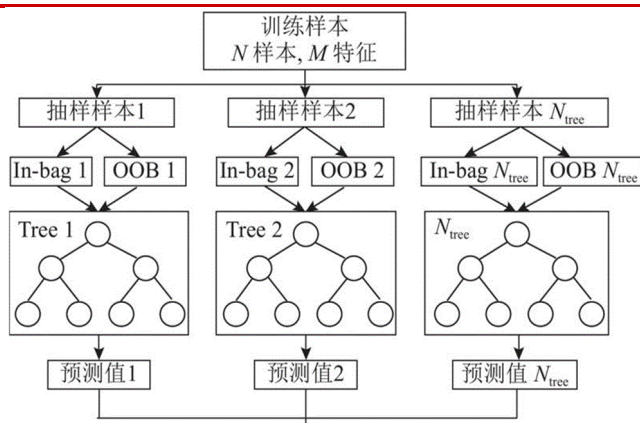
3.1.1. 随机森林算法

随机森林(Random Forest, RF)是 Bagging 的升级，它在以决策树为基学习器构建 Bagging 集成的基础上，进一步在决策树的训练过程中引入了随机特征选择。具体来说，传统决策树在选择划分属性时是在当前结点的属性集合（假设有 d 个属性）中选择一个最优属性；而在随机森林中，对基决策树的每个结点，先从该结点的属性集合中随机选择一个包含 k 个属性的子集，然后再从这个子集中选择一个最优属性用于划分。这里的 k 控制了随机性的引入程度，如果 $k = d$ ，则基决策树的构建与传统决策树相同；若 $k = 1$ ，则是随机选择一个属性用于划分。

随机森林算法生成过程如下：

- 1、从原始数据集中每次随机有放回抽样选取与原始数据集相同数量的样本数据，构造数据子集；
- 2、每个数据子集从所有待选择的特征中随机选取一定数量的最优特征作为决策树的输入特征；
- 3、根据每个数据子集分别得到每棵决策树，由多棵决策树共同组成随机森林；
- 4、最后如果是分类问题，则按照投票的方式选取票数最多的类作为结果返回，如果是回归问题，则按照平均法选取所有决策树预测的平均值作为结果返回。

图 8 随机森林算法图



资料来源：Wind，华西证券研究所

3.2. Boosting 算法

Boosting 算法的工作机制是首先从训练集用初始权重训练出一个弱学习器 1，根据弱学习的学习误差率表现来更新训练样本的权重，使得之前弱学习器 1 学习误差率高的训练样本点权重变高，使得这些误差率高的点在后面的弱学习器 2 中得到更多的重视。然后基于调整权重后的训练集来训练弱学习器 2，如此重复进行，直到弱学习器数达到事先指定的数目 T ，最终将这 T 个弱学习器通过集合策略进行整合，得到最终的强学习器。简单来说，Boosting 就是串行地训练一系列弱分类器，使得被先前弱分类器分类错误的样本在后续地到更多关注，最后将这些分类器组合成最优强分类

器的过程。Boosting 算法的一般过程如下。常见的 Boosting 算法有 Adaboost、GBDT、XGBoost 等。

输入：样本分布 $\mathcal{D}$ ；

基学习算法 $\mathcal{L}$ ；

基学习器数 T 。

步骤：

$\mathcal{D}_1 = \mathcal{D}$. %初始化分布

for $t = 1, \dots, T$:

$h_t = \mathcal{L}(\mathcal{D}_t)$; %根据分布 $\mathcal{D}_t$ 训练弱分类器

$\epsilon_t = P_{x \sim \mathcal{D}_t}(h_t(x) \neq f(x))$; %评估 h_t 的错误率

$\mathcal{D}_{t+1} = \text{Adjust_Distribution}(\mathcal{D}_t, \epsilon_t)$.

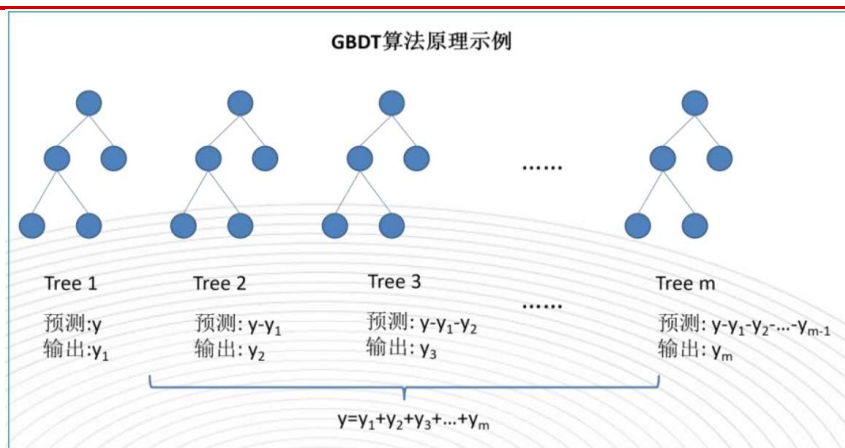
end

输出： $H(x) = \text{Combine_Outputs}(\{h_1(x), \dots, h_t(x)\})$.

3.2.1. GBDT

GBDT 是 boosting 算法的一种，它是一种迭代的决策树算法。GBDT 的核心原理是先用初始值预测一颗决策树，得到本轮的残差(即真实值减预测值)，然后用残差作为下一轮决策树的预测对象，这时会再产生一个残差，再用这个残差作为下一轮的预测对象，以此循环迭代直到最后一轮的预测残差为 0 或非常小的时候就停止迭代，然后把所有轮的模型预测结果相加得到最终预测结果，GBDT 核心原理如下图所示。

图 9 随机森林算法图



资料来源：Wind，华西证券研究所

3.2.2. XGBoost

XGBoost(extreme gradient boosting)是 GBDT 的一种工业实现，也是通过不断增加新树，拟合伪残差去降低损失函数。

XGBoost 本质上仍然属于 GBDT 算法，但在算法精度、速度和泛化能力上均要优于传统的 GBDT 算法。从算法精度上来看，XGBoost 通过将损失函数展开到二阶导数，使得其更能逼近真实损失；从算法速度上来看，XGBoost 使用了加权分位树 sketch 和稀疏感知算法这两个技巧，通过缓存优化和模型并行来提高算法速度；从算法泛化能力上来看，通过对损失函数加入正则化项、加性模型中设置缩减率和列抽样等方法，来防止模型过拟合。

4. 集成学习建模

4.1. 模型合理性的讨论

回归问题中，单个模型容易过拟合，而集成学习可以通过减少单个模型的过拟合来提高整体预测的稳定性。且回归问题往往存在多个潜在的模型，这些模型可能针对不同的数据集表现得更好。将这些模型合并为一个集成模型可以提高准确率，因为集成学习可以从不同的模型中获取更多的信息，并获得更准确的预测。它可以动态调整每个模型的权重，根据不同数据集的特点来决定如何融合多个回归模型的结果，从而提高预测准确率。

综上所述，集成学习是一种非常有效地机器学习方法，可以利用集成学习帮助我们解决回归问题。

4.2. 模型优缺点的讨论

4.2.1. 随机森林

优点：

1. 随机森林是集成算法，模型精度往往比单棵决策树更高；
2. 每次随机选样本和特征，提高了模型抗干扰能力，泛化能力更强；
3. 对数据集适应能力强，可处理离散数据和缺失数据，数据规范化要求低；
4. 在每次随机选样本时均有 1/3 的样本未被选上，这部分样本通常称之为袋外数据 OOB(out of bag)，可以直接拿来作为验证集，不需占用训练数据。

缺点：

1. 当决策树的数量较多时，训练所需要时间较长；
2. 模型可解释性不强，属于黑盒模型。

4.2.2. GBDT

优点：

1. GBDT 每一次的残差计算都增大了分错样本的权重，而分对的权重都趋于 0 因此泛化性能比较好。
2. 预测精度高

缺点：

- 1.对异常值比较敏感。
- 2.分类器之间存在依赖关系，难以并行计算。

4.2.3. XGBoost

优点：

- 1.XGBoost 在代价函数里加入了正则项，控制了模型的复杂度。
- 2.XGBoost 工具支持并行。
- 3.XGBoost 支持用户自定义目标函数和评估函数。
- 4.对于特征值缺失的样本，XGBoost 可以自动学习出它的分裂方向。

缺点：

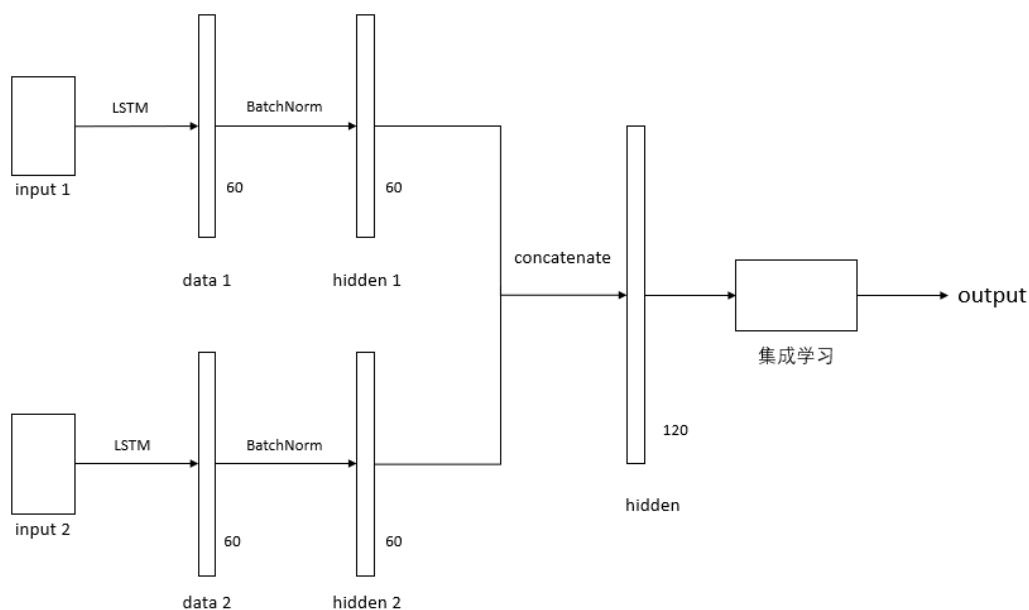
- 1.算法参数过多，调参复杂。
- 2.不适合处理非结构化数据。
- 3.不适合处理超高维特征数据。

4.3. 策略设计思路

4.3.1. 建模方法

input1 包含了一天股票数据的信息，input2 包含了一周股票数据的信息，他们的维度均为 $c*40*6$ (c 为通道数)。将input1和input2分别通过LSTM层，这两个LSTM的权重不共享，得到 $c*60*1$ 维的矩阵，经过归一化处理之后拼接为一个 $c*120*1$ 维的矩阵，此矩阵包含了两个输入的全局状态信息。最后，将结果经过集成学习模型输出一个 $c*1$ 维的矩阵，即得到预测结果。

图 10 使用 LSTM + 集成学习建模



资料来源：Wind，华西证券研究所

4.3.2. 数据及参数选择

随机森林回归的评估器是 RandomForestRegressor，它的重要参数如下：

表 2: RandomForestRegressor 的参数说明

参数	说明
n_estimators	子模型数量。通常为正整数，默认值为 100
max_features	决定划分时考虑的最大特征数量。默认为 auto
max_depth	每棵树的最大深度。通常为正整数，默认为 None
min_samples_split	拆分结点所需的最少样本数。默认值为 2
min_samples_leaf	最小叶节点样本数。默认值为 1
n_jobs	并行线程数。默认值为 1

资料来源：Wind，华西证券研究所

GBDT 回归的评估器是 GradientBoostingRegressor，它的重要参数如下：

表 3: GradientBoostingRegressor 的参数说明

参数	说明
n_estimators	子模型数量。通常为正整数，默认值为 100
learning_rate	学习率。默认值为 0.1
loss	损失函数。默认值为 ls(least squares)
max_depth	每棵树的最大深度。通常为正整数，默认为 None
min_samples_split	拆分结点所需的最少样本数。默认值为 2
min_samples_leaf	最小叶节点样本数。默认值为 1

资料来源：Wind，华西证券研究所

XGBoost 回归的评估器是 XGBRegressor，它的重要参数如下：

表 4：XGBRegressor 的参数说明

参数	说明
n_estimators	子模型数量。通常为正整数，默认值为 100
learning_rate	每个迭代产生的模型的学习率。默认值为 0.1
max_depth	每棵树的深度。通常为正整数，默认为 None
objective	给定损失函数。默认为 "reg:linear"
booster	给定模型的求解方式。默认为 "gbtree"
n_jobs	并行线程数。默认值为 1
reg_alpha	L1 正则项的权重。默认为 0
reg_lambda	L2 正则项的权重。默认为 1

资料来源：Wind，华西证券研究所

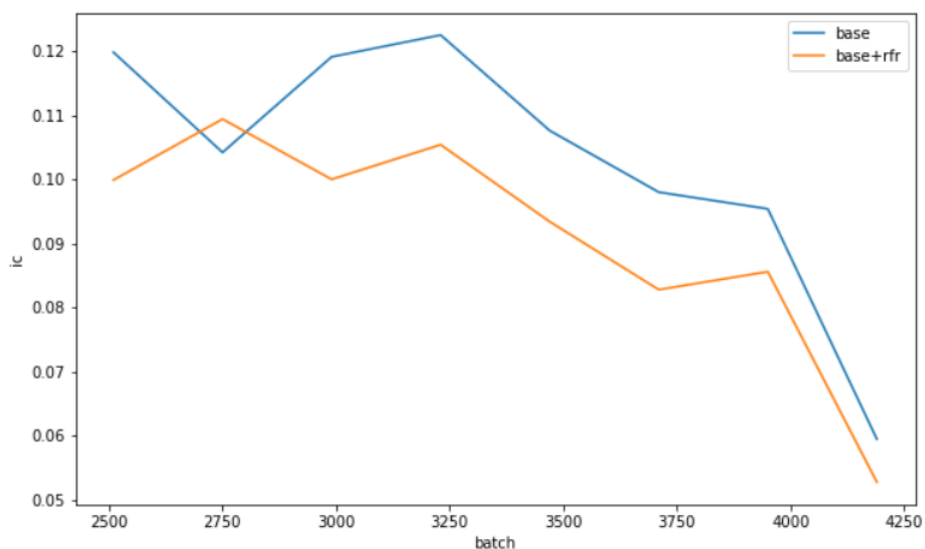
4.4. 策略具体过程

4.4.1. 随机森林回归

在原有 LSTM 结构的基础上，我们提取 "concatenate" 层的结果，将其作为随机森林的输入。利用 sklearn 中的 RandomForestRegressor 函数进行回归拟合，输出预测数值。对真实数值和预测数值进行皮尔逊系数的计算，得到平均 ic 值。

使用网格搜索法，对超参数设定范围，将参数组合进行循环迭代组合，通过穷举法对所有参数进行评分从而寻得最优参数。我们对随机森林最重要的两个参数（学习器个数和决策树深度）进行了调参，在兼顾效果和耗时的情况下，最终选择了学习器个数为 100，深度为 8 的超参数。继续增加学习器个数以及增加深度，对效果的提升微乎其微，且训练时间更久。在该参数下的拟合结果如图。在 12 核 cpu 上设置并行，训练随机森林回归模型预计耗时 2.67 小时。

图 11 使用 LSTM + 随机森林建模结果



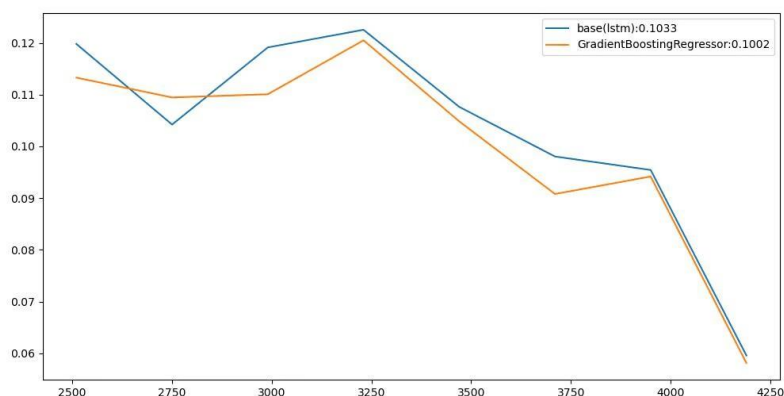
资料来源：Wind，华西证券研究所

但在原模型架构上加随机森林回归的效果并不理想。原模型的平均 ic 值为 0.1032，在原模型基础上进行随机森林回归得到的平均 ic 值为 0.0912。

4.4.2. GBDT 回归

我们将随机森林模型替换为 GBDT 模型，利用 sklearn 中的 GradientBoostingRegressor 函数进行回归拟合。效果也并不理想。

图 12 使用 LSTM + GBDT 建模结果

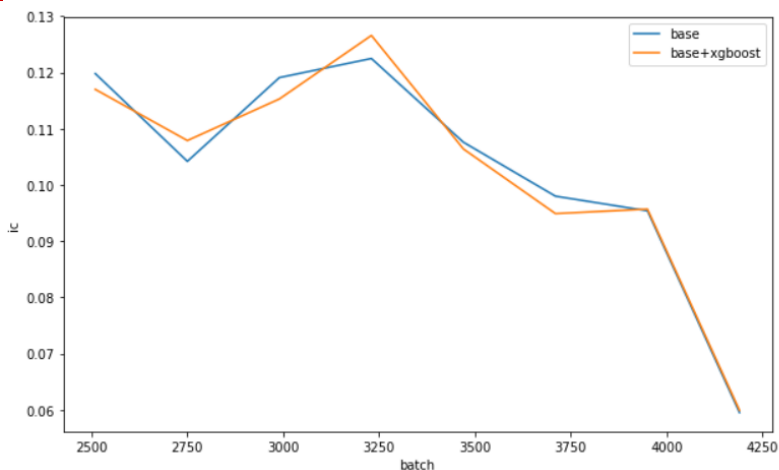


资料来源：Wind，华西证券研究所

4.4.3. XGBoost 回归

我们将 GBDT 模型替换为 XGBoost 模型，利用 sklearn 中的 XGBRegressor 函数进行回归拟合。兼顾准确率和运行时间，当学习器个数为 600，深度为 3 时效果较好。在该参数下的拟合结果如图。

图 13 使用 LSTM + XGBoost 建模结果



资料来源：Wind，华西证券研究所

XGBoost 是三种集成方法中效果最好的。平均 ic 值和原模型几乎一致，在 2750、3250、4190 批次的表现略优于原模型。但它对模型的提升程度仍然有限。

5. 线性回归理论

5.1. 模型定义与评估

线性回归的目标是找到一个函数，能将输入的属性映射到输出属性或目标属性上。该函数将 x 作为输入，返回 y ，即 y 是 x 的函数，在数学上一般记为 $y = f(x)$ ，但在机器学习领域，一般将假设函数记为 h ，代表了英文 hypothesis，模型被记为：

$$y = h(x)$$

更一般的，输入为一个向量 $\mathbf{x}$ ，表示输入有多个变量，称为多变量线性回归问题，即

$$y = h(\mathbf{x})$$

为了选择更合适的模型，对模型做出假设是必要的，最简单有效的方式就是假设输入与输出是线性关系，用公式表示如下：

$$y = h(\mathbf{x}) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b$$

其中 $\mathbf{x}$ 表示包含 d 个属性的输入， $\mathbf{x} = (x_1; x_2; \dots; x_d)$ ， x 在第 i 个属性上的取值用 x_i 表示。一般用向量的形式写成：

$$y = h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

模型需要学习的任务是找到这样的一个函数，来拟合输入与输出，通俗的说，就是确定最佳的 $\mathbf{w}$ 和 b ，使得 $h(x_i) \simeq y_i$ 。

一般情况下，我们采用损失函数来衡量一个模型和真实数据之间的差别。为了找到最佳的 $\mathbf{w}$ 和 b 的值，线性回归使用了均方误差的变体来表示损失函数，用 $J(\mathbf{w}, b)$ 表示整个数据集上的平均损失，损失函数公式如下：

$$J(\mathbf{w}, b) = \frac{1}{2N} \sum_{i=1}^N (h(\mathbf{x}^i; \mathbf{w}, b) - y^i)^2$$

$\mathbf{x}^i$ 和 y^i 表示的是第 i 维的数据。值得注意的是，公式中的 $1/2$ 是为了求导方便而加入的，如果对 $J(\mathbf{w}, b)$ 进行求导， $1/2$ 就会被约去， $1/N$ 是为了除去数据集大小对损失的影响，唯一能直接对 $J(\mathbf{w}, b)$ 产生影响的只有 $\mathbf{w}$ 和 b 两个参数。

损失函数越小，表示假设模型能更好地对数据进行拟合。因此，我们的目标是让均方误差最小化，以产生最小的损失函数值：

$$\operatorname{argmin}_{\mathbf{w}, b} \frac{1}{2N} \sum_{i=1}^N (h(\mathbf{x}^i; \mathbf{w}, b) - y^i)^2$$

如何去求解这个最小值，分别有两种方法：最小二乘法和梯度下降法。

5.2. 最小二乘法

在损失函数 $J(\mathbf{w}, b)$ 的最小值处， $\mathbf{w}$ 和 b 的偏导数一定为0，也就是说，求解以下方程组，能得到 $\mathbf{w}$ 和 b 的拐点：

$$\begin{cases} \frac{\partial J(w_j)}{\partial w_j} = 0 \\ \frac{\partial J(b)}{\partial b} = 0 \end{cases}$$

同时，计算他们的二阶偏导数，如果大于 0，则该拐点为最小值，否则为最大值。

5.3. 梯度下降

5.3.1. 梯度下降基本思想

梯度下降算法的基本思想是：随机选取一组参数初值，计算损失，然后寻找能让损失在数值上下降最多的另一组参数，反复迭代且达到一个局部最优。由于没有尝试所有的参数组合，所以无法确定是否就是全局最优。如果选择不同的一组初始参数，可能找到不同的局部最优值。

参数的更新方式为：

$$g_i = \frac{\partial J(w)}{\partial w_i} = \frac{1}{N} \sum_{i=1}^N (h(x_i) - y_i)x_i$$

$$w_i^{k+1} = w_i^k - \alpha g_i$$

5.3.2. 随机梯度下降

当训练集规模较大时，可以考虑使用随机梯度下降来减小计算量。随机梯度下降的方式与批量梯度下降非常类似，主要区别在于批量梯度下降每次更新参数是使用所有的训练数据，而随机梯度下降算法每次只使用一条训练数据就可以更新参数。

随机梯度下降算法根据每一个训练实例更新参数集，并不需要计算全部训练集才进行随机梯度下降更新，因此计算速度较快，在批量梯度下降算法还没有完成一次迭代时就已经更新多次。但这种只根据一个训练实例更新参数的办法也存在一个问题：并不是每一次迭代的步伐都迈向“正确”的方向。因此，算法虽然会逐步迈向局部最小值的位置，但可能只是在最小值位置附近徘徊，无法收敛到局部最小值那一点上。

6. 线性回归应用

6.1. 模型的合理性讨论

线性回归可以用来预测股票价格。其基本思路是通过历史的数据，找到股票价格与其他市场变量之间的线性关系，然后利用这个关系预测未来的股票价格。

具体来说，需要收集一些历史数据，例如过去一年的股票价格和市场变量数据。市场变量可以包括股市指数、商品价格、货币汇率等等。接下来，可以使用线性回归模型来探索这些变量之间的关系，并建立一个模型来预测未来的股票价格。

6.2. 模型的优缺点讨论

6.2.1. 多元线性回归模型

优点:

1. 能够从多个自变量中找到对因变量有影响的变量。
2. 能够考虑多个自变量的影响, 且调整多个自变量的权重。
3. 样本量要求较低, 可以利用样本的数量来提高预测精度。

缺点:

1. 计算过程比较复杂, 需要建立高维的线性方程组;
2. 如果样本数据数量不够大, 或者设计矩阵不满秩, 会导致多元回归模型过于复杂, 过拟合的风险较高;
3. 对于非线性的数据结构, 多元线性回归不能很好地适应。

6.2.2. 随机梯度下降线性回归模型

优点:

1. 训练速度非常快、内存占用小。
2. 可以在线学习, 即在新增数据时立即更新模型。
3. 对噪声数据不敏感。

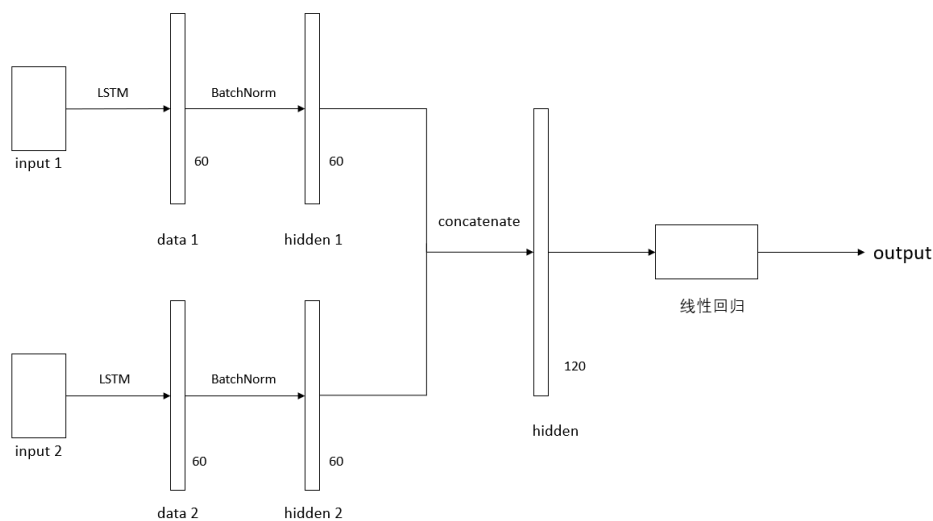
缺点:

1. 因为随机抽取的样本不一定代表整个数据集, 所以 SGD 有可能无法收敛到全局最小值, 收敛结果比较不稳定。
2. 不易于并行实现

6.3. 策略设计思路

6.3.1. 建模方法

图 14 使用 LSTM + 线性回归建模



资料来源：Wind，华西证券研究所

input1 包含了一天股票数据的信息，input2 包含了一周股票数据的信息，他们的维度均为 $c \times 40 \times 6$ (c 为通道数)。将 input1 和 input2 分别通过 LSTM 层，这两个 LSTM 的权重不共享，得到 $c \times 60 \times 1$ 维的矩阵，经过归一化处理之后拼接为一个 $c \times 120 \times 1$ 维的矩阵，此矩阵包含了两个输入的全局状态信息，将此矩阵输入到线性回归模型中，得到最后的结果。

6.3.2. 数据及参数选择

多元线性回归的评估器是 LinearRegression，它的重要参数如下：

表 5: LinearRegression 的参数说明

参数	说明
fit_intercept	是否有截距，如果没有则直线过原点
normalize	是否将数据归一化
n_jobs	并行线程数。默认值为 1

资料来源：Wind，华西证券研究所

随机梯度下降线性回归的评估器是 SGDRegressor，它的重要参数如下：

表 6: XGBRegressor 的参数说明

参数	说明
loss	损失函数。默认为 'squared_error'
penalty	正则化项。默认为 'l2'。'l1' 和 'elasticnet' 为可选项，可能会给模型带来稀疏性(特征选择)。
alpha	乘以正则化项的常数。值越高，正则化越强。
max_iter	训练数据的最大迭代次数。默认值为 1000
random_state	传递 int 以获得跨多个函数调用的可重现输出。
learning_rate	学习率。

`early_stopping`

当验证分数没有提高时，是否使用提前停止来终止训练。

`n_iter_no_change`

在停止拟合之前等待没有改进的迭代次数。

资料来源：Wind，华西证券研究所

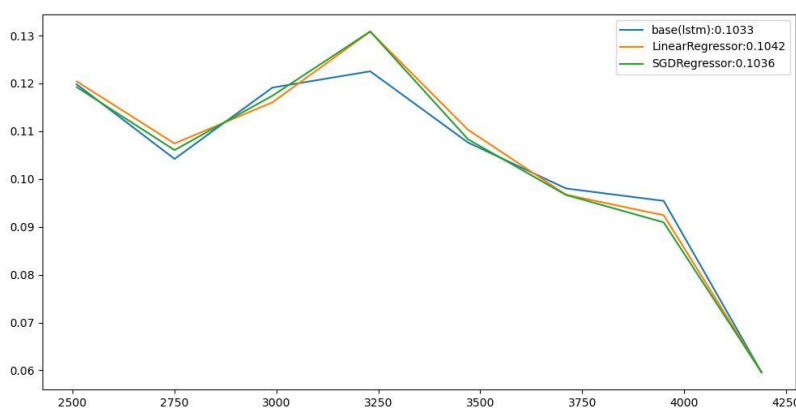
6.4. 策略具体过程

在原有 LSTM 结构的基础上，我们提取 “concatenate” 层的结果，将其作为随机森林的输入。利用 sklearn 中的 `linear_model.LinearRegression()` 函数进行回归拟合，输出预测数值。对真实数值和预测数值进行皮尔逊系数的计算，得到平均 ic 值。

将多元线性回归模型替换为随机梯度下降模型，利用 sklearn 中的 `linear_model.SGDRegressor()` 进行回归拟合。

由于两种方法得到的结果趋势十分相近，我们将原模型，原模型+多元线性回归，原模型+随机梯度下降线性回归的结果展现在一张图中，便于观察和比较。

图 15 使用 LSTM + 线性回归建模结果



资料来源：Wind，华西证券研究所

可以看到，两种线性回归的方法效果十分接近，总体上的平均 ic 值都略高于原模型。

7. 支持向量回归理论

7.1. 支持向量机

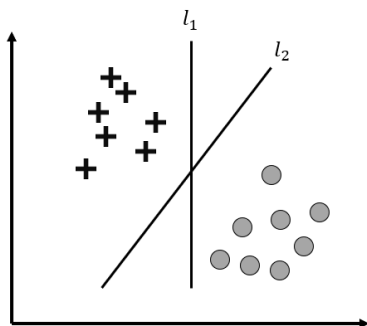
7.1.1. 支持向量机定义

支持向量机（SVM）是一种二元分类模型，定义为特征空间上间隔最大的线性分类器模型，学习策略就是使其间隔最大化。

SVM 是从线性可分情况下二元分类的最优分类平面发展而来的，最优的含义是要求分类平面不但能够将两个类别正确分割开来，而且能使分类间隔最大。也就是说，

SVM 试图寻找一个满足分类要求的超平面，并且使训练集中的数据点尽量远离该平面，即使分类平面两侧间隔最大化。

图 16 SVM 二分类示意图



资料来源：Wind，华西证券研究所

如图 16 所示，SVM 所要找到的是 l_2 这样的超平面进行划分。

7.1.2. 最大间隔超平面

SVM 原先是为二元分类问题设计的，但可以扩展至能够处理多元分类问题。假定有一些给定的数据点，每个数据点属于两个类别之一，即二元分类，其分类目标是，确定一个新的数据点属于两个类别中的哪一个。用支持向量机的观点，将一个数据点视为一个 D 维向量，分类问题就转换为是否可以用一个 $D-1$ 维超平面将这些数据点按类别分割开来，这就是线性分类器。对于一个给定的线性可分问题，有无数个能对数据进行分类的超平面，最佳的超平面应该是能够将两个类别最大限度地分离开来的超平面，这样能够使经验风险最小化。所以选择的超平面应该能够将与两侧最接近的数据点的距离最大化。如果存在这样一个超平面，可称之为最大间隔超平面，所定义的线性分类器称为最大间隔分类器。

最大间隔分类器意味着具有更好的泛化能力，能够容忍更多的噪声影响。如果某个数据点受到噪声影响而发生位移，最大间隔分类器能够最大限度地防止噪声造成的错误分类。

7.1.3. 支持向量与最大间隔

假设样本在线性可分的情况下， d 维样本空间中可以将训练样本完全分类的超平面方程为：

$$\mathbf{w}^T \mathbf{x} + b = 0$$

对于正类来说， $\mathbf{w}^T \mathbf{x} + b > 0$ ；对于负类来说， $\mathbf{w}^T \mathbf{x} + b < 0$ 。

式中， $\mathbf{x} = (x_1, x_2, \dots, x_d)$ 为 d 维样本中的一个点， $\mathbf{w} = (w_1, w_2, \dots, w_d)$ 为超平面的法向量，决定了超平面的方向， b 为位移，决定了超平面与原点的距离。可以看出，任何一个超平面都可以用 $(\mathbf{w}, b)$ 来唯一确定。

任意一点 (x, y) 到超平面的距离可以表示为：

$$r = \frac{|\mathbf{w}^T \mathbf{x} + b|}{\|\mathbf{w}\|}$$

$$\text{其中 } \|\mathbf{w}\| = \sqrt{\sum_{i=1}^d w_i^2}。$$

假设距离超平面最近的正样本或负样本到超平面的距离为+1 或-1，则最大间隔表示为：

$$\gamma = \frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$

这个间隔只与法向量有关，也就是只与超平面的法向量有关，而与位移 b 无关。

所有样本被正确分类需要满足：

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m$$

于是我们的目标函数就可以写成：

$$\begin{aligned} & \max_{\mathbf{w}, b} \gamma \\ & s.t. \ y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m \end{aligned}$$

又可以写成更一般的形式：

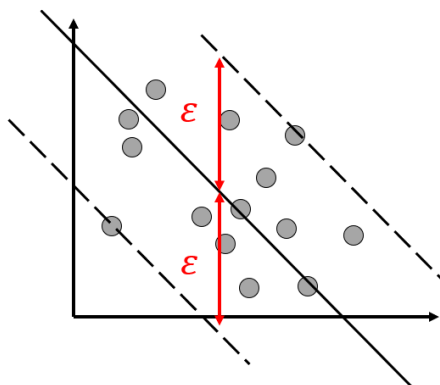
$$\begin{aligned} & \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ & s.t. \ y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m \end{aligned}$$

这就是支持向量机 SVM 的基本模型。

7.2. 支持向量回归

支持向量机回归（SVR）用非线性映射将数据映射到高维数据特征空间中，使得在高维数据特征空间中自变量与因变量具有很好的线性回归特征，在该特征空间进行拟合后再返回到原始空间。

图 17 SVR 示意图



资料来源：Wind，华西证券研究所

在 SVM 中，我们希望靠超平面最近的样本点之间的间隔最大，而在 SVR 中，我们同样也是希望间隔最大，不同的是它使靠超平面最远的样本点之间的间隔最大，但是这个最大对于偏差有个限制条件：

$$|w^T x_i + b - y_i| \leq \varepsilon$$

我们将这个偏差成为 ε 管道，如图 2 所示。

类似于 SVM，SVR 的优化问题可以被如下定义：

$$\begin{aligned} \min_{w,b} & \frac{1}{2} \|w\| \\ \text{s.t. } & |w^T x_i + b - y_i| \leq \varepsilon, \quad i = 1, 2, \dots, m \end{aligned}$$

8. 支持向量回归应用

8.1. 模型的合理性讨论

支持向量回归(SVR)通过将数据映射到高维进行超平面的划分，在面对大量数据的情况下运行速度十分缓慢，于是我们考虑使用线性支持向量回归(LSVR)。相比于 SVR，LSVR 使用了线性内核，模型更简单，计算成本更低，所需参数较少，在线性可分离的数据集上有更好的性能。

在预测股票价格方面，LSVR 模型可以使用一些相关的因素，如股票过去的价格走势、市场指数、公司业绩等因素作为训练数据，预测未来的股票价格走势。需要注意的是，LSVR 模型是一种监督学习模型，它需要大量的训练数据来实现预测的准确性。

8.2. 模型的优缺点讨论

优点：

1. 在处理线性问题时表现出色，特别是在高维数据集上，可以有效地处理大量的特征；
2. 具有一定的泛化能力，它可以应用于很多不同的问题领域，包括预测、分类、聚类等。

缺点：

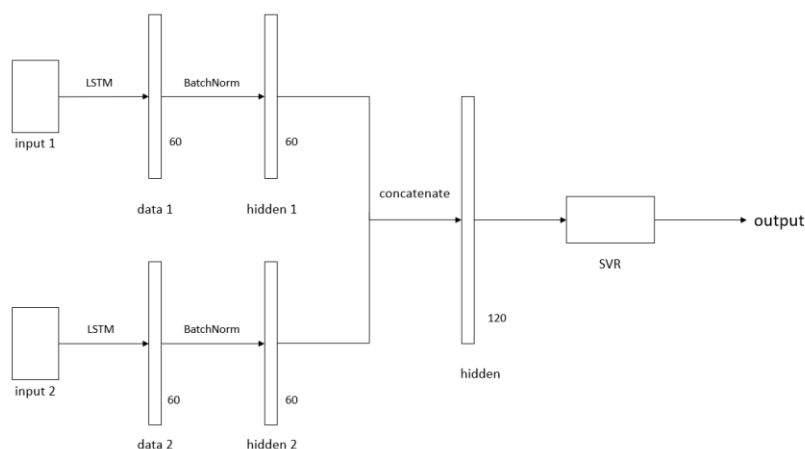
1. 对噪声数据敏感，如果训练集中存在错误的标签或离群点，会对模型产生很大的影响；
2. 对非线性问题效果不佳；
3. 训练时间长，在处理大型数据集时需要更多的时间和计算资源；
4. 模型容量大小和复杂度难以自动调整，需要进行手动调整。

8.3. 策略设计思路

8.3.1. 建模方法

input1 包含了一天股票数据的信息，input2 包含了一周股票数据的信息，他们的维度均为 $c*40*6$ (c 为通道数)。将 input1 和 input2 分别通过 LSTM 层，这两个 LSTM 的权重不共享，得到 $c*60*1$ 维的矩阵，经过归一化处理之后拼接为一个 $c*120*1$ 维的矩阵，此矩阵包含了两个输入的全局状态信息，将此矩阵输入到支持向量回归模型中，得到最后的结果。

图 18 使用 LSTM + 线性支持向量机回归建模



资料来源：Wind，华西证券研究所

8.3.2. 数据及参数选择

线性支持向量回归的评估器是 LinearSVR，它的重要参数如下：

表 7: LinearSVR 的参数说明

参数	说明
epsilon	epsilon-insensitive 损失函数中的 Epsilon 参数。

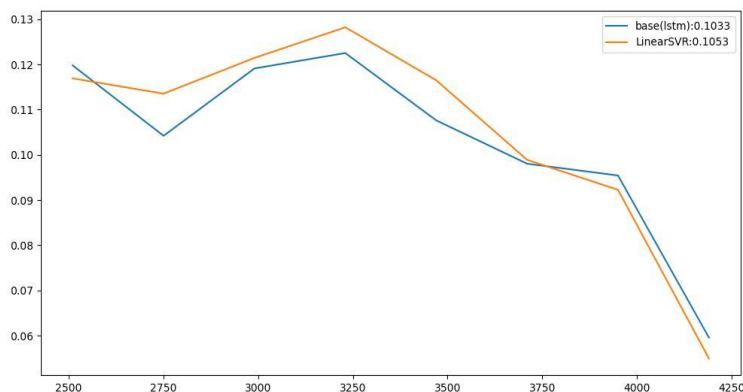
C	正则化参数。正则化的强度与 C 成反比。必须严格为正。
loss	指定损失函数。epsilon-insensitive 损失(标准 SVR)是 L1 损失, 而平方 epsilon-insensitive 损失是 L2 损失。
fit_intercept	是否计算此模型的截距
max_iter	运行的最大迭代次数。
n_iter_no_change	在停止拟合之前等待没有改进的迭代次数。

资料来源: Wind, 华西证券研究所

8.4. 策略具体过程

在原有 LSTM 结构的基础上, 我们提取“concatenate”层的结果, 将其作为随机森林的输入。利用 sklearn 中的 svm.LinearSVR() 函数进行回归拟合, 输出预测数值。对真实数值和预测数值进行皮尔逊系数的计算, 得到平均 ic 值。

图 19 使用 LSTM + 线性支持向量机回归建模结果



资料来源: Wind, 华西证券研究所

可以看到, 相比原模型线性支持回归模型的效果有所提高。

9. 总结

本篇报告我们在原模型(lstm)的基础上, 提取了“concatenate”层的结果, 将其作为回归模型的输入。我们尝试了集成学习中的随机森林回归、GBDT 回归、XGBoost 回归, 线性回归中的多元线性回归、随机梯度下降回归, 支持向量机中的线性支持向量回归, 对以上模型都进行了调参和测试。

表 8: 几种方法结果数据对比

	Base	rfr	xgboost	GBDT	Linear-Regression	SGD-Regressor	LinearSVR
2510	0.1198	0.0999	0.1170	0.1133	0.1204	0.1192	0.1169
2750	0.1042	0.1094	0.1079	0.1095	0.1074	0.1061	0.1135
2990	0.1191	0.1000	0.1153	0.1101	0.1160	0.1174	0.1215
3230	0.1225	0.1054	0.1266	0.1205	0.1308	0.1309	0.1282
3470	0.1076	0.0934	0.1064	0.1048	0.1103	0.1083	0.1165

3710	0.0980	0.0828	0.0949	0.0908	0.0967	0.0967	0.0989
3950	0.0954	0.0856	0.0957	0.0942	0.0924	0.0909	0.0923
4190	0.0595	0.0528	0.0598	0.0581	0.0596	0.0596	0.0549
mean	0.1033	0.0912	0.1030	0.1002	0.1042	0.1036	0.1053

资料来源：华西证券研究所

结果显示线性支持向量回归 (LinearSVR) 是以上方法中表现最佳的，多元线性回归方法 (LinearRegression) 和随机梯度下降线性回归方法 (SGD-Regressor) 次之。在本问题上，传统的回归方法相比于集成学习的回归方法效果更加。

10. 风险提示

模型基于对历史数据统计，仅作为投资参考。

分析师承诺

作者具有中国证券业协会授予的证券投资咨询执业资格或相当的专业胜任能力，保证报告所采用的数据均来自合规渠道，分析逻辑基于作者的职业理解，通过合理判断并得出结论，力求客观、公正，结论不受任何第三方的授意、影响，特此声明。

评级说明

公司评级标准	投资 评级	说明
以报告发布日后的 6 个月内公司股价相对上证指数的涨跌幅为基准。	买入	分析师预测在此期间股价相对强于上证指数达到或超过 15%
	增持	分析师预测在此期间股价相对强于上证指数在 5%—15%之间
	中性	分析师预测在此期间股价相对上证指数在-5%—5%之间
	减持	分析师预测在此期间股价相对弱于上证指数 5%—15%之间
	卖出	分析师预测在此期间股价相对弱于上证指数达到或超过 15%
行业评级标准		
以报告发布日后的 6 个月内行业指数的涨跌幅为基准。	推荐	分析师预测在此期间行业指数相对强于上证指数达到或超过 10%
	中性	分析师预测在此期间行业指数相对上证指数在-10%—10%之间
	回避	分析师预测在此期间行业指数相对弱于上证指数达到或超过 10%

华西证券研究所：

地址：北京市西城区太平桥大街丰汇园 11 号丰汇时代大厦南座 5 层

网址：<http://www.hx168.com.cn/hxzq/hxindex.html>

华西证券免责声明

华西证券股份有限公司（以下简称“本公司”）具备证券投资咨询业务资格。本报告仅供本公司签约客户使用。本公司不会因接收人收到或者经由其他渠道转发收到本报告而直接视其为本公司客户。

本报告基于本公司研究所及其研究人员认为的已经公开的资料或者研究人员的实地调研资料，但本公司对该等信息的准确性、完整性或可靠性不作任何保证。本报告所载资料、意见以及推测仅于本报告发布当日的判断，且这种判断受到研究方法、研究依据等多方面的制约。在不同时期，本公司可发出与本报告所载资料、意见及预测不一致的报告。本公司不保证本报告所含信息始终保持在最新状态。同时，本公司对本报告所含信息可在不发出通知的情形下做出修改，投资者需自行关注相应更新或修改。

在任何情况下，本报告仅提供给签约客户参考使用，任何信息或所表述的意见绝不构成对任何人的投资建议。市场有风险，投资需谨慎。投资者不应将本报告视为做出投资决策的惟一参考因素，亦不应认为本报告可以取代自己的判断。在任何情况下，本报告均未考虑到个别客户的特殊投资目标、财务状况或需求，不能作为客户进行客户买卖、认购证券或者其他金融工具的保证或邀请。在任何情况下，本公司、本公司员工或者其他关联方均不承诺投资者一定获利，不与投资者分享投资收益，也不对任何人因使用本报告而导致的任何可能损失负有任何责任。投资者因使用本公司研究报告做出的任何投资决策均是独立行为，与本公司、本公司员工及其他关联方无关。

本公司建立起信息隔离墙制度、跨墙制度来规范管理部门、跨关联机构之间的信息流动。务请投资者注意，在法律许可的前提下，本公司及其所属关联机构可能会持有报告中提到的公司所发行的证券或期权并进行证券或期权交易，也可能为这些公司提供或者争取提供投资银行、财务顾问或者金融产品等相关服务。在法律许可的前提下，本公司的董事、高级职员或员工可能担任本报告所提到的公司的董事。

所有报告版权均归本公司所有。未经本公司事先书面授权，任何机构或个人不得以任何形式复制、转发或公开传播本报告的全部或部分内容，如需引用、刊发或转载本报告，需注明出处为华西证券研究所，且不得对本报告进行任何有悖原意的引用、删节和修改。