中信期货
CITIC Futures

中信期货研究|金融工程专题报告

2022-12-29

基于深度学习的期货组合优化

——期货多因子专题报告（六）

投资咨询业务资格：
证监许可【2012】669 号

报告要点

本篇报告将以“组合优化”为目标的凸优化方法与神经网络结合起来，构建了完整的期货组合量化投资框架。我们利用该框架回测部分期货市场的若干量价类因子，获得了较好的组合收益表现。

摘要：

多因子策略是量化投研领域的经典策略之一，其在多类别金融市场均受到长时间的研究与应用；而无论何种市场，标的资产池的权重配置在多因子模型中扮演了“决定策略收益稳健性”的重要角色。因此，组合权重优化是多因子模型的一个重要课题。

聚焦到期货市场，经典的多因子研究包括 alpha 因子的挖掘和期货组合权重优化计算。市面常见的逻辑认为：任意期货都同时暴露于多种不同的风险因素下，这些风险因素的共同作用形成了期货合约价格的波动；通过对不同的风险因素以 alpha 因子的形式来有效刻画，我们可以实现对期货收益率的分解，从而研究期货合约价格波动的原因；此外，最优的投资组合则应当是经过“剔除其余不稳定的因素干扰、充分暴露于 alpha 因子、并一般通过凸优化方法将对收益率的预测转化为组合权重”等这样若干步处理后的结果；这些步骤中也涉及到一些针对于组合权重的约束条件的设置。而本报告区别于上述经典逻辑，尝试了基于深度学习的组合优化方法。

本文以期货市场及其量价多因子为基础，将以“组合优化”为目标的凸优化与神经网络结合起来，构建了完整的期货组合量化投资框架。这里的“输入”是期货合约的基础行情数据，通过“构造若干量价因子、神经网络进行量价多因子的提取与合成、可分凸优化层传播梯度来优化期货组合权重、定义期货组合的收益率为损失度（目标）函数、以该损失度来优化整个神经网络等”来生成日频的期货组合权重，此即为“输出”。

在合理的参数配置中，模型均能获得较好的收益表现。如每日调仓时，年化收益率约为 18%、年化波动率约为 5%、夏普率 3.49。

风险提示：本报告中所涉及的资产配比和模型应用仅为回溯举例，并不构成推荐建议。

金融工程研究团队

研究员：
周通
021-80401733
zhoutong@citicsf.com
从业资格号 F3078183
投资咨询号 Z0018055

期货多因子系列研究报告

专题报告四：商品期货 alpha 因子拾遗——20220923
专题报告五：不同频率视角下的选期因子——20221222

目　录

图表目录

一、问题背景

多因子策略是量化投研领域的经典策略之一，其在多类别金融市场均受到长时间的研究与应用；而无论何种市场，标的资产池的权重配置在多因子模型中扮演了决定策略收益稳健性的重要角色。因此，组合权重优化是多因子模型的一个重要课题。

聚焦到期货市场，经典的多因子研究包括 alpha 因子的挖掘和期货组合权重优化计算。市面常见的逻辑认为：任意期货都同时暴露于多种不同的风险因素下，这些风险因素的共同作用形成了期货合约价格的波动；通过对不同的风险因素以 alpha 因子的形式有效刻画，我们可以实现对期货收益率的分解，从而研究期货合约价格波动的原因；而最优的投资组合则是应当是经过“剔除其余不稳定的因素干扰、充分暴露于 alpha 因子、并一般通过凸优化方法将对收益率的预测转化为组合权重”等这样若干步处理后的结果；这些步骤中也涉及到一些针对于组合权重的约束条件的设置。而本报告尝试了区别于上述经典逻辑的新方法，总体思路基于深度学习的组合优化。

本文以期货市场及其量价多因子为基础，将以组合优化为目标的凸优化与神经网络结合起来，构建了完整的期货组合量化投资框架。这里的“输入”是期货合约的基础行情数据，通过“构造若干量价因子、神经网络进行量价多因子的提取与合成、可分凸优化层传播梯度来优化期货组合权重、定义期货组合的收益率为损失度（目标）函数、以该损失度来优化整个神经网络等”来生成日频的期货组合权重，此即为“输出”。

二、模型搭建设置与说明

经典组合优化中，常见的组合权重的生成方式包括风险预算模型、rankIC 加权、由 Barra 风险暴露衍生的因子收益率加权等。本报告将要探讨的“凸优化方法与神经网络训练结合”的方式与这些经典的权重生成方法有较大的差异，因此本节将详尽的介绍中间步骤，包括：底层资产数据及考虑的量价类因子介绍、由存取数据衍生的数据解析方法汇总、基础的等权多因子分层回测、常用函数的定义、神经网络层的设计、训练方式的构造、循环-退出/早退-模型保存等机制的定义、模型测试及回测等。

（一）收盘价、收益率及因子数据

这里的底层标的资产样本池为 29 个不同商品/金融期货主力合约，样本区间为自 2015 年 12 月 24 日至 2022 年 11 月 28 日的 1600 多个交易日。

图表1：商品期货及金融期货品种选择

表头	表头
黑色类	玻璃（FG）、热轧卷板（HC）、铁矿石（I）、焦炭（J）、焦煤（JM）、螺纹钢（RB）、
有色类	沪铝（AL）、沪铜（CU）、沪锌（ZN）
能源类	石油沥青（BU）
化工类	聚乙烯（L）、甲醇（MA）、聚丙烯（PP）、聚氯乙烯（V）、PTA(TA)
软商品类	棉花（CF）、天然橡胶（RU）、白糖（SR）
农产品类	豆一（A）、玉米（C）、玉米淀粉（CS）、鸡蛋（JD）、豆粕（M）、菜油（OI）、棕榈油（P）、菜粕（RM）、豆油（Y）
金融	沪深 300 指数期货（IF）、5 年期国债期货（TF）

资料来源：中信期货研究所

因子层面：本篇报告仅考虑量价类的 6 个因子，分别是“3 日动量”（'mom_d3'），“243 日动量”（'mom_d243'），“10 日动量”（'mom_d10'），“243 日最小二乘回归”（'ols_d243'），“5 日量价相关性”（'cv_d5'）和“ 61 日振幅”（'amp_d61_g4'）。其相应的因子构造逻辑可参见本团队之前的相关研报。

（二）数据解析——以适当的格式存取数据

商品期货和金融期货数据本身以及基于其行情类数据构造得到的量价类因子数据，有其自身的特点。其中之一则是由于不同期货合约上市日期不同以及量价类因子不同的回看期，造成了大量空值的出现。

对于此类型数据，直接的方法涉及稀疏矩阵的存取。在数值分析中，稀疏矩阵是指其元素大部分为零的矩阵。反之，如果大部分元素都非零，则这个矩阵是稠密的。在科学与工程领域中求解线性模型时经常出现大型的稀疏矩阵。由于过大的尺寸，标准的算法经常无法操作这些稀疏矩阵。因此，在使用计算机存储和操作稀疏矩阵时，经常需要修改标准算法以利用矩阵的稀疏结构。由于其自身的稀疏特性，通过压缩可以大大节省稀疏矩阵的内存代价。

具体落实到我们这里，体现为：商品期货和金融期货数据本身以及基于其行情类数据构造得到的量价类因子数据数据总体非常大，而我们基于单个项目需求发起的研究通常是与数据总体的一个非常小的子集进行交互，这就导致了极大的

稀疏性。当处理稀疏矩阵时，将它们存储为一个完整的矩阵(从这里开始称为稠密矩阵)是非常低效的。这是因为一个完整的数组为每个条目占用一块内存，所以一个 n x m 数组需要 n x m 块内存。从简单的逻辑角度来看，存储这么多零是没有意义的。在 python 中，稀疏数据结构在 scipy 中得到了有效的实现，具体包含 7 类稀疏矩阵（如 csc_matrix,bsr_matrix 等）。实现背后的思想很简单:我们不将所有值存储在稠密矩阵中，而是以某种格式存储非零值（例如，使用它们的行和列索引）。

另一种则是简单的数据预处理后再进行“数据解析”，即将一种数据格式转换为另一种可读格式。这个操作的出发点是：由于训练时需要反复读取数据,所花费的时间会比较大；因此我们可以选择最合适的方式将其存到本地，便于之后训练时的多次调用。这里我们对常用的数据解析方法做了一下对比，如下表所示。注：这里使用的机器是 2 GHz 四核处理器及其上安装的 python3.8.8 和 spyder4.2.5。

图表2：不同“数据解析”方式效能对比

方法	文件格式	大小	写入耗时 ms	读取耗时 ms	类型处理？	形状处理？
np.save()	.npy	2.3MB	3.24	1.60	否	否
h5py.File()	.hdf5	2.3MB	0.69	0.85	否	否
gfg.savetxt()	.txt	7.5MB	188.80	161.74	否	是
pickle.dump()	.pkl	2.3MB	1.27	1.36	否	否
.tofile()	.bin	2.3MB	2.65	2.70	是	是

资料来源：同花顺 iFind、中信期货研究所

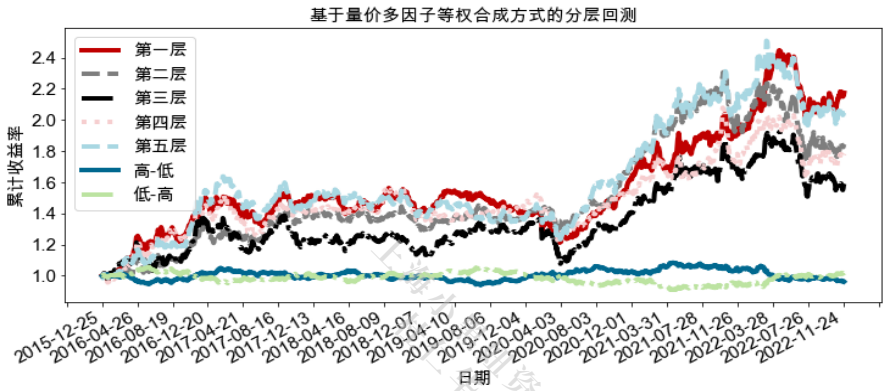
从上表我们可以看到，hdf5 文件的读写相对较快，其次是 pkl 文件，再次是 npy 和 bin 这两种格式的文件，而 txt 则明显耗时较多；空间占用角度，前四者没有显著差异，而后者 txt 比较占空间，不推荐使用；此外，我们也注意到 bin 和 txt 格式在存储、读取过程中对于类型/形状处理的要求。综上，基于我们这里考虑的这份“1683 个交易日、29 个商品/金融期货主力合约、6 个量价因子”的数据样本而言，最优的格式我们推荐 hdf5、npy 和 pkl。

我们得到的这个结论，符合对于上述 5 种格式效能的一般认知；当然随着数据量的增大，也可进一步做出更鲜明的区分。那么下文中具体使用到的格式为 h5py 文件。第一次导入数据后在一段时间内会于缓存中保存，这时候再次导入会很快。所以在训练神经网络时，除了第一个 epoch 运行较慢外，后面读取的速度都会较第一次有所提升。

（三）等权合成多因子的分层回测

这里我们先给出基于等权方式合成多因子的经典分层回测结果，如下图所示：

图表3：基于等权方式合成多因子回测净值曲线



资料来源：同花顺 iFind、中信期货研究所

上述回测净值曲线对应的净值分析如下：

图表4：量价相关性因子的回测统计

平滑窗口	年化收益率%	年化波动率%	夏普率	最大回撤	卡玛
第一层	12.23	12.71	0.22	0.96	0.13
第二层	9.53	12.61	0.25	0.76	0.13
第三层	7.14	1.35	0.22	0.53	0.13
第四层	8.97	12.78	0.21	0.7	0.13
第五层	11.26	12.97	0.25	0.87	0.13
高-低	-0.57	5.46	0.11	-0.1	0.05

资料来源：同花顺 iFind、中信期货研究所

（四）深度学习常用函数的准备

1. 类 ARGS()

类的构造函数，用于初始化类的内部状态，为类的属性设置默认的值。

2. 类 MyDataset(Dataset)

Dataset 自 torch.utils.data 导入。 torch.utils.data.Dataset 是代表自定义数据集方法的类，用户可以通过继承该类来自定义自己的数据集类，在继承时要求用户重载__len__()和__getitem__()这两个方法。

- __len__()：返回的是数据集的大小。我们构建的数据集是一个对象，而数据集不像序列类型（列表、元组、字符串）那样可以直接用 len()来获取序列的长度，该方法__len__()的目的就是方便像序列那样直接获取对象的长度。如果 A 是一个类，a 是类 A 的实例化对象，当 A 中定义了该方法__len__()，len(a)则返回对象的大小。

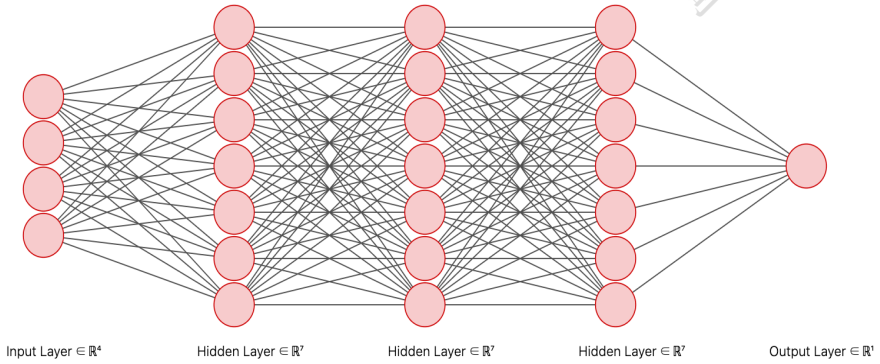
- __getitem__()：实现索引数据集中的某一个数据。我们知道，序列可以通过索引的方法获取序列中任意元素，__getitem__()则实现了能够通过索引的方法获取对象中任意元素。此外，我们可以在__getitem__()中实现数据预处理。

3. 各类型数据载入函数

（五）神经网络层的准备

这里我们使用了一个五层的神经网络：头部输入层、尾部输出层和中间的三个隐含层。具体细节如下：

图表5：五层的神经网络（输入层、输出层、三个隐含层）



资料来源：中信期货研究所、http://alexlenail.me/NN-SVG/index.html

第一层：输入层；

第二层：全连接层 fc_1，之后使用 rectification 之一 ReLU()；

第三层：全连接层 fc_2，之后使用 rectification 之一 Tanh()；

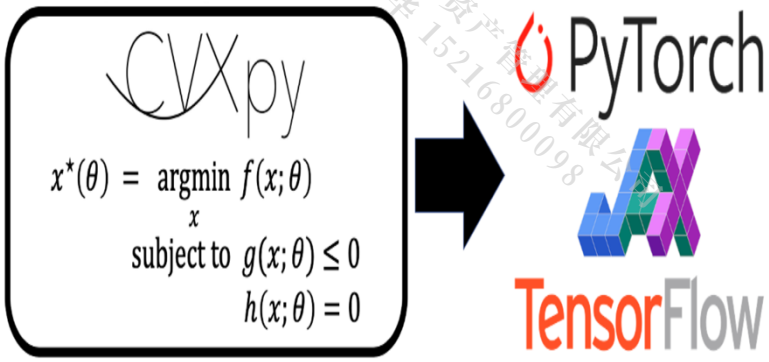
第四层：cvxpyLayer 层，使用前需把第三层的输出拉成一维，使用时两种选择 SparseMaxLayer()；从“添加新层”的角度来看，后续的拓展方向之一是“考虑不同的均方误差计算方式”从而来构造新的神经网络层；

第五层：输出层。

值得一提的是我们这里使用到了凸优化层 cvxpylayers。简要来讲，它是一个 python 库，在 CVXPY 的基础上整合了 PyTorch、JAX 和 TensorFlow 的接口，便于用户在构造神经网络时来调用这些框架里的层以及建立可分的凸优化层。具体而言，cvxpylayers 始见于 2019 年第 33 届 NeurIPS 会议的论文《Differentiable convex optimization layers》，Stephen Boyd 等作者针对更一般化的凸优化问

题提出了可分凸优化层 cvxpylayers。作者引入了规范凸规划（Disciplined convex programming）的一类子问题——规范参数化规划（Disciplined parametrized programming），然后表明每一个规范参数化项目可以表征为“仿射-求解器-仿射”的模式（Affine-Solver-Affine）“。其中的第一个“仿射”是“从模型参数到求解问题所涉及数据”的仿射映射，第二个“仿射”则是从求解器的解到原问题的解的的仿射映射。这样的构造使得求导变得可能，使我们不仅能用凸优化的方法完成优化，还能求出凸优化输出值对模型中带优化参数的梯度，以便神经网络进行梯度反向传播。

图表6：cvxpylayers:凸优化问题与神经网络经典框架 PyTorch 的结合



资料来源：中信期货研究所、https://github.com/cvxgrp/cvxpylayers

（六）训练方式的构造

考虑到总的时间序列长度为 1683 个交易日，我们这里的“训练-验证”集的选取及神经网络训练效果的呈现方式原则如下：

1. 最外层 for 循环设置：从第 400 个交易日到第 1300 个交易日、滚动式地每隔 100 个交易日的进行一次循环（总共 10 次），以此来客观输出训练效果；

2. for 循环中数据载入：对每一次循环（每隔 100 个交易日），我们以“使用 end_date 铆定时间点、回看 lookback（200）确定时间序列长度”的方式来提取因子和收益率的子集，然后将这 lookback(200)个因子子集和收益率子集分别拼装成(200 个交易日*29 个商品/金融期货品种,6 个量价因子)的本轮循环因子表和(200 个交易日*29 个商品/金融期货品种,1 列收益率)的本轮循环收益率表；

3. 关于训练集与验证集的划分：区别于机器学习中经典的“80-20 划分”，我们这里使用了更个性化的截取。具体而言，对于每一次 for 循环中重新载入的本轮循环因子表和本轮循环收益率表，我们分别取其前 2900 条数据（100*29 个商品/金融期货）作为本轮循环因子表的训练集和本轮循环收益率表的训练集；接着，我们再选自第 3509 条（(100+21）*29 个商品/金融期货）开始至最末的数

据作为本轮循环因子表的验证集和本轮循环收益率表的验证集（注意：这里的 100 和 21 都是技术性数字，可以根据自己的研究需要进行调整，原则是训练集和验证集的数据互不相交）；

4. 经典训练步骤的沿用：在用 pytorch 训练模型时，我们在遍历 epochs 的过程中采用了经典神经网络训练“三步走”的方式，即依次使用了 optimizer.zero_grad()，loss.backward()和 optimizer.step()三个函数。简要来讲，这三个函数的作用是先将梯度归零（optimizer.zero_grad()），然后反向传播计算得到每个参数的梯度值（loss.backward()），最后通过梯度下降执行一步参数更新（optimizer.step()）。

- optimizer.zero_grad()：鉴于训练过程较常用到 mini-batch 方法，这一操作的目的是在“反向传播和梯度下降”之前清除与上一个 batch 数据相关的梯度。该函数会遍历模型的所有参数，通过 p.grad.detach_()方法截断反向传播的梯度流，再通过 p.grad.zero_()函数将每个参数的梯度值设为 0，即上一次的梯度记录被清空；

- loss.backward()：PyTorch 的反向传播(即 tensor.backward())是通过 autograd 包来实现的，autograd 包会根据 tensor 进行过的数学运算来自动计算其对应的梯度。具体地说，损失函数 loss 是由模型的所有权重 w 经过一系列运算得到的，若某个 w 的 requires_grads 为 True，则 w 的所有上层参数（后面层的权重 w）的.grad_fn 属性中就保存了对应的运算，然后在使用 loss.backward()后，会一层层的反向传播计算每个 w 的梯度值，并保存到该 w 的.grad 属性中。如果没有进行 tensor.backward()的话，梯度值将会是 None，因此 loss.backward()要写在 optimizer.step()之前；

- optimizer.step()：step()函数的作用是执行一次优化步骤，通过梯度下降法来更新参数的值。因为梯度下降是基于梯度的，所以在执行 optimizer.step()函数前应先执行 loss.backward()函数来计算梯度。应当区分的是，optimizer 只负责通过梯度下降进行优化，而不负责产生梯度，梯度是 tensor.backward()方法产生的；

5. 损失度函数 loss 的定义：

-1*（1+ 收益率*本轮训练所得权重）*成本，

其中，成本=1-（累计训练所得权重-上一次训练训练）*成本率，而累计训练所得权重则有若干轮历史训练所得权重按交易日期拼接而成。该损失度函数定义的目的是最大化总收益。注意：后文中，我们也将这里定义的损失度函数 loss 取负值（负负得正）命名为（训练/验证的）得分。注：这里的成本率我们设置为

千分之三。

（七）循环、退出/早停、模型保存

我们这里来声明相应的一些机制设置。其中：

1. 总的训练步数 n_epochs=50;warm_up 轮（热身）设置为 5；

2. 初始化：

   - 停止时运行所至步数、最优得分所对应步数皆初始化为 0；
   - 最优得分初始化为-np.inf；

3. 如果前 warm_up 轮验证集分数一直在下降则退出重新初始化后再训练；

4. 如果是最后一次 retry 则不退出并完成训练；

5. 如果验证得分>最优得分且训练步数>warm_up 轮数，则将验证得分赋值给最优得分、当前步数复制给最优得分所对应步数、保存该模型以便于回测时调用；同时早停标识符 stop_steps 赋值 0；

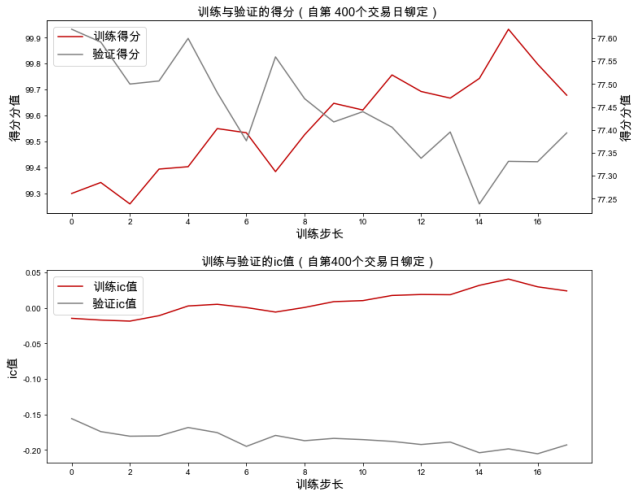
6. 如果上一步的条件不满足，则早停标识符 stop_steps 加 1；如果早停标识符大于训练容忍度（patience），则早停结束训练（实际操作中，我们这里训练容忍度设置为 10）；

7. 期货品种每日最大权重上限设置为 0.075。

（八）测试预测权重

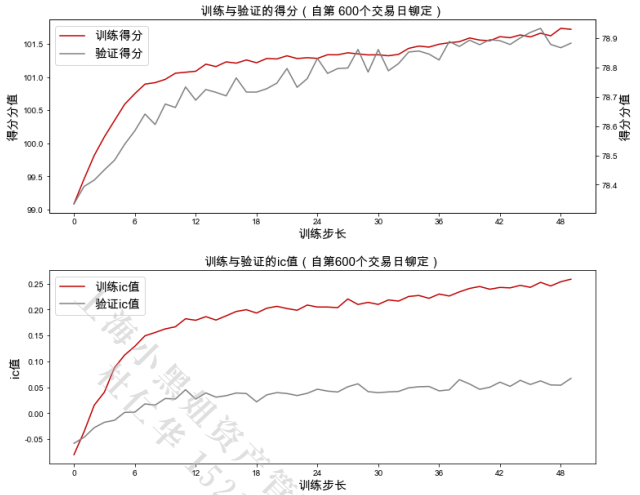
对于上述训练-验证得到的最优模型，我们这里进行测试。下图是神经网络循环训练-验证过程中的得分（损失度的相反数与 IC 值的对比）：

图表7：针对铆定日期-第 400 日的训练-验证效果（IC 值与得分——即损失度的相反数）



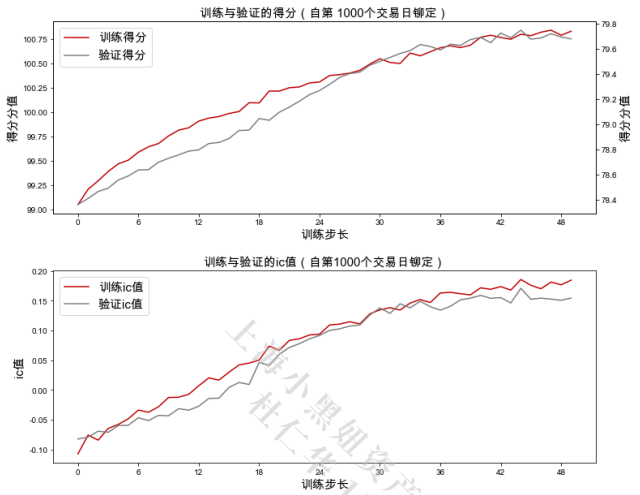
资料来源：同花顺 iFind、中信期货研究所

图表8：针对铆定日期-第 600 日的训练-验证效果（IC 值与得分——即损失度的相反数）



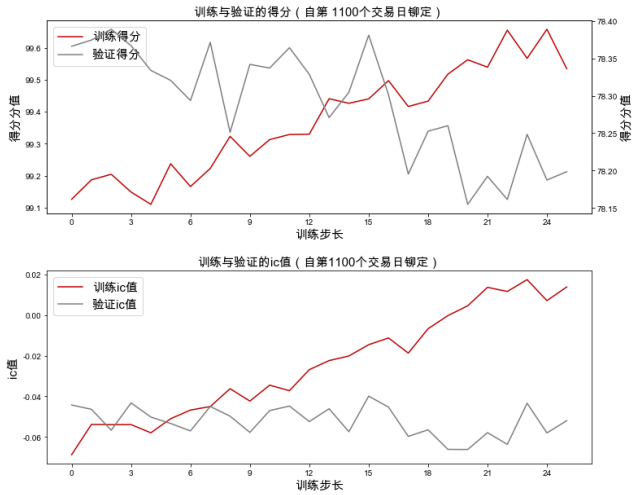
资料来源：同花顺 iFind、中信期货研究所

图表9：针对铆定日期-第 1000 日的训练-验证效果（IC 值与得分——即损失度的相反数）



资料来源：同花顺 iFind、中信期货研究所

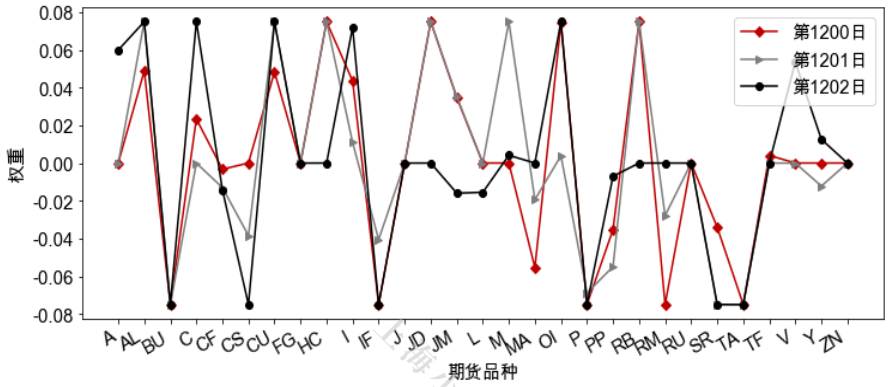
图表10：针对铆定日期-第 1100 日的训练-验证效果（IC 值与得分——即损失度的相反数）



资料来源：同花顺 iFind、中信期货研究所

下图是测试集上所得的关于全体入选期货品种每日权重的部分示意图：

图表11：测试集上经深度学习训练得到的连续 3 个交易日期货品种权重

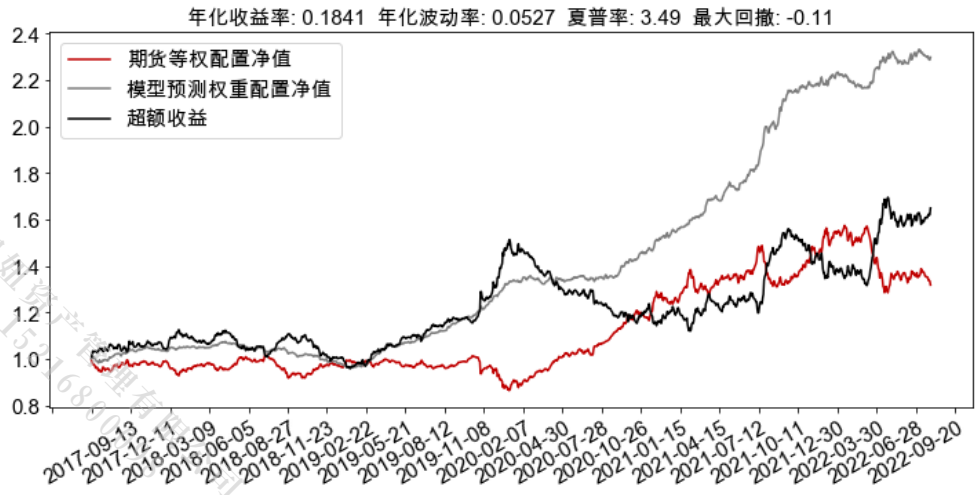


资料来源：同花顺 iFind、中信期货研究所

（九）回测

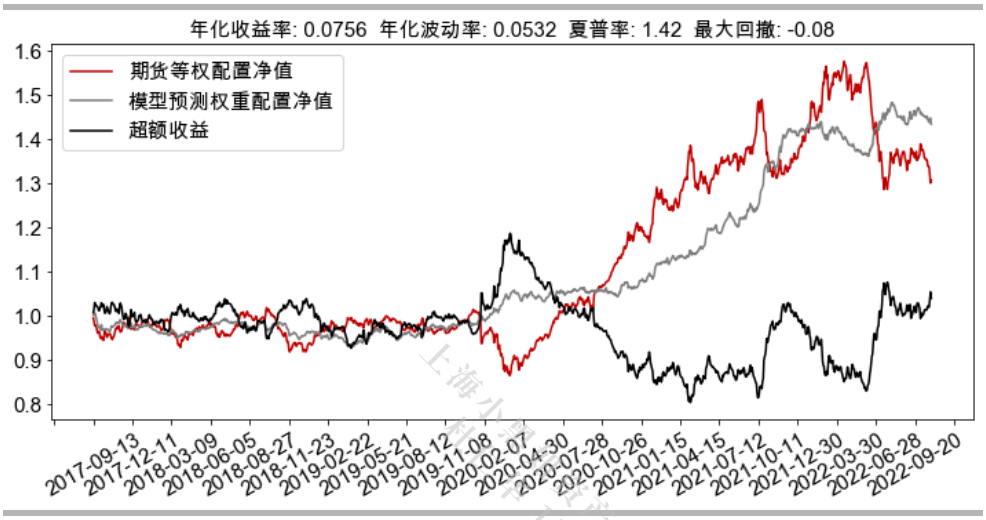
我们这里对整个区间进行回测。除了前面已经提到的：29 个期货主力合约、6 个量价类因子、2015 年 12 月到 2022 年 12 月的回测区间，我们再实际回测中还考虑到了了调仓频率 win 的概念，目的是为了对经神经网络训练得到的权重这样一个时间序列进行平滑、以消除极个别特殊现象对整体趋势的影响。

图表12：每日调仓（win=1）的回测结果



资料来源：同花顺 iFind、中信期货研究所

图表13：调仓频率 win=2 时的回测结果



资料来源：同花顺 iFind、中信期货研究所

图表14：量价相关性因子的回测统计

平滑窗口	年化收益率%	年化波动率%	夏普率	最大回撤	卡玛
win=1	18.41	5.27	3.49	-0.11	1.67
win=2	7.56	5.32	1.42	-0.08	0.95
win=3	3.74	5.35	0.7	-0.1	0.37
win=5	3.75	5.29	0.71	-0.07	0.54
win=10	2.97	5.22	0.57	-0.13	0.23
win=20	4.65	4.99	0.93	-0.08	0.57

资料来源：同花顺 iFind、中信期货研究所

三、总结

本篇报告以期货市场及其量价多因子为基础，将以组合优化为目标的凸优化与神经网络结合起来，构建了完整的期货组合量化投资框架。这里的“输入”是期货合约的基础行情数据，通过“构造若干量价因子、神经网络进行量价多因子的提取与合成、可分凸优化层传播梯度来优化期货组合权重、定义期货组合的收益率为损失度（目标）函数、以该损失度来优化整个神经网络等”来生成日频的期货组合权重，此即为“输出”。

在合理的参数配置中，模型均能获得较好的收益表现。如每日调仓时，年化收益率约为 18%、年化波动率约为 5%、夏普率 3.49。

免责声明

除非另有说明，中信期货有限公司拥有本报告的版权和/或其他相关知识产权。未经中信期货有限公司事先书面许可，任何单位或个人不得以任何方式复制、转载、引用、刊登、发表、发行、修改、翻译此报告的全部或部分材料、内容。除非另有说明， 本报告中使用的所有商标、服务标记及标记均为中信期货有限公司所有或经合法授权被许可使用的商标、服务标记及标记。未经中信期货有限公司或商标所有权人的书面许可，任何单位或个人不得使用该商标、服务标记及标记。

如果在任何国家或地区管辖范围内，本报告内容或其适用与任何政府机构、监管机构、自律组织或者清算机构的法律、规则或规定内容相抵触，或者中信期货有限公司未被授权在当地提供这种信息或服务，那么本报告的内容并不意图提供给这些地区的个人或组织，任何个人或组织也不得在当地查看或使用本报告。本报告所载的内容并非适用于所有国家或地区或者适用于所有人。

此报告所载的全部内容仅作参考之用。此报告的内容不构成对任何人的投资建议，且中信期货有限公司不会因接收人收到此报告而视其为客户。

尽管本报告中所包含的信息是我们于发布之时从我们认为可靠的渠道获得，但中信期货有限公司对于本报告所载的信息、观点以及数据的准确性、可靠性、时效性以及完整性不作任何明确或隐含的保证。因此任何人不得对本报告所载的信息、观点以及数据的准确性、可靠性、时效性及完整性产生任何依赖，且中信期货有限公司不对因使用此报告及所载材料而造成的损失承担任 何责任。本报告不应取代个人的独立判断。本报告仅反映编写人的不同设想、见解及分析方法。本报告所载的观点并不代表中信期货有限公司或任何其附属或联营公司的立场。

此报告中所指的投资及服务可能不适合阁下。我们建议阁下如有任何疑问应咨询独立投资顾问。此报告不构成任何投资、法律、会计或税务建议，且不担保任何投资及策略适合阁下。此报告并不构成中信期货有限公司给予阁下的任何私人咨询建议。

深圳总部

地址：深圳市福田区中心三路 8 号卓越时代广场（二期）北座 13 层 1301-1305、14 层

邮编：518048

电话：400-990-8826

传真：(0755)83241191

网址：http://www.citicsf.com