

深度学习揭秘系列之一：基于量价与基本面结合的深度学习选股策略

2024年8月6日

- **本文是深度学习揭秘系列报告第一篇。**因此本文从基础的神经网络理论部分入手，介绍了神经元结构、Relu、Sigmoid等常用激活函数、反向传播算法，以及后文中用到的Adam优化器与Dropout层。
- **非线性模型更能挖掘量价因子的潜力。**本文以开源Alpha158作为量价因子集，以及常见的估值、成长、质量、分析师一致预期等因子作为基本面因子集进行基础研究，采用线性模型Lasso和非线性模型MLP分别对量价与基本面因子进行合成。结果显示，非线性MLP模型相较于线性Lasso模型在量价因子上优势较明显，MLP模型的5日RankIC较Lasso模型提升1.82pct，费后多头超额收益提升9.42pct；20日RankIC较Lasso提升1.94pct，费后多头超额收益提升6.54pct。但非线性模型的优势在基本面因子上不突出，MLP模型长期表现与Lasso模型几乎持平。我们认为可能有几个原因：1) 基本面因子之间结合的逻辑偏线性，而量价因子的非线性组合可能包含增量信息。2) 部分基本面因子在财报真空期值不变，而标签Y却每天有变动。3) 基本面因子数量相较于量价因子偏少。
- **端到端的分支网络结合量价与基本面更具优势。**前文我们用全连接神经网络分别结合了量价与基本面因子，但如何进一步将量价因子与基本面因子相结合是我们关心的另一方面。我们从基础的线性结合方式入手，尝试等权结合与ICIR加权结合前文神经网络拟合的量价与基本面因子，结果表明线性结合的因子较难战胜纯量价因子。5日纯量价因子的RankIC为12.39%，纯基本面因子RankIC为7.32%，等权因子RankIC为11.64%，ICIR加权因子RankIC为12.37%。因此我们更进一步，采用分支网络端到端计算量价与基本面的子因子合成，以及量价与基本面因子的权重生成。结果显示，5日分支网络合成因子RankIC为12.9%，相较于原始量价因子提升0.51pct，多头超额年化收益为10.85%，提升2.54pct，多空超额年化收益为132.21%，提升9.77pct。在20日的维度同样也优于其他合成方式。
- **基于长短期收益预测的混频组合。**5日因子对短期收益的预测效果更好，而20日因子对长期收益的预测能力更佳。因此我们以20日因子的多头组合为底仓，与5日因子排名靠前的股票取交集进行高频调整。与前25%的5日因子取交集，多头超额提升6.98pct，与前50%取交集提升4.79pct，与前75%取交集提升3.26pct。且交集股票范围越小，换手越高，因此这是一个以提高换手率获取更高收益的组合思路。另外，我们发现混频组合的最大回撤会更接近于底仓20日组合的最大回撤，因此提高了收益的同时最大回撤并无显著增加。
- **基于低频量价、高频量价、基本面因子的中证1000指增策略。**在20日调仓的换手下，限制100%指数成分股内选股，个股权重最大偏离0.5%，行业最大偏离3%，风格最大偏离0.01的约束条件，中证1000指增组合2015年至2024年5月31日年化收益为11.59%，年化信息比为3.56，单边年化换手率7.61。相对中证1000全收益的年化超额收益为13.17%，收益波动比为3.77，收益回撤比为5.06。今年超额收益为3.73%。

风险因素：结论基于历史数据，在市场环境转变时模型存在失效的风险。

目录

一、深度学习理论.....	5
1.1、神经元与激活函数.....	5
1.2、前馈神经网络拟合能力较强.....	6
1.3、前馈神经网络的反向传播算法.....	8
1.4、学习率调整算法 Adam.....	9
1.5、丢弃法 (Dropout).....	10
二、Lasso Vs MLP —— 分别用于量价与基本面因子.....	11
2.1、量价因子数据集.....	11
2.2、基本面因子数据集.....	13
2.3、数据预处理与网络参数设置.....	15
2.4、Lasso Vs MLP —— 量价因子.....	17
2.5、Lasso Vs MLP —— 基本面因子.....	19
2.6、全连接网络隐藏层神经元数量设置的敏感性分析与启示.....	21
2.7、小结：非线性模型更能挖掘量价因子的潜力.....	23
三、量价与基本面因子结合的初步尝试.....	26
3.1、线性方式 —— 等权结合.....	27
3.2、线性方式 —— ICIR 加权.....	28
3.3、非线性方式 —— 输入层拼接量价与基本面.....	30
3.4、非线性方式 —— 分支网络统筹因子合成与大类因子加权.....	32
3.5、小结：端到端的分支网络结合量价与基本面更具优势.....	34
四、基于长短期收益预测的混频组合.....	38
五、基于低频量价、高频量价、基本面因子的中证 1000 指增策略.....	41
参考文献.....	43
风险因素.....	44

表目录

表 1: Alpha 158 因子列表.....	11
表 2: 基本面因子列表.....	13
表 3: 各模型全区间 IC 统计 —— 量价因子预测未来 5 日与 20 日.....	18
表 4: 各模型全区间多头超额与多空超额收益风险特征指标 —— 量价因子.....	18
表 5: 各模型全区间 IC 统计 —— 基本面因子预测未来 5 日与 20 日.....	20
表 6: 各模型全区间多头超额与多空超额收益风险特征指标 —— 基本面因子.....	20
表 7: 一层的全连接神经网络不同神经元数量下合成因子效果.....	21
表 8: 两层的全连接神经网络不同神经元数量下合成因子与 Lasso 因子的相关性.....	21
表 9: 两层的全连接神经网络不同神经元数量下合成因子的 RankIC 均值.....	22
表 10: 两层的全连接神经网络不同神经元数量下合成因子的 ICIR.....	22
表 11: 量价因子+MLP 多头超额绩效指标 —— 每 5 日调仓.....	24
表 12: 基本面因子+MLP 多头超额绩效指标 —— 每 5 日调仓.....	24
表 13: 量价因子+MLP 多头超额绩效指标 —— 每 20 日调仓.....	25
表 14: 基本面因子+MLP 多头超额指标 —— 每 20 日调仓.....	25
表 15: 量价、基本面与等权因子 IC 统计.....	27
表 16: 量价、基本面与等权因子多头超额与多空超额收益风险特征指标.....	27
表 17: 量价、基本面与 ICIR 加权因子 IC 统计.....	28
表 18: 量价、基本面与 ICIR 加权因子多头超额与多空超额收益风险特征指标.....	29
表 19: 量价、基本面与输入层拼接因子 IC 统计.....	31
表 20: 量价、基本面与输入层拼接因子多头超额与多空超额收益风险特征指标.....	31
表 21: 量价、基本面与分支网络因子 IC 统计.....	33
表 22: 量价、基本面与分支网络因子多头超额与多空超额收益风险特征指标.....	33
表 23: 量价、基本面与各合成方法 IC 统计 —— 5 日.....	34
表 24: 量价、基本面与各合成方法多头超额与多空超额收益风险特征指标 —— 5 日.....	34
表 25: 量价、基本面与各合成方法 IC 统计 —— 20 日.....	35
表 26: 量价、基本面与各合成方法多头超额与多空超额收益风险特征指标 —— 20 日.....	35
表 27: 分支网络合成因子多头超额绩效指标 —— 每 5 日调仓.....	37
表 28: 分支网络合成因子多头超额绩效指标 —— 每 20 日调仓.....	37
表 29: 原始 20 日合成因子与混频组合分年度多头超额.....	38
表 30: 原始 20 日合成因子 \cap 5 日因子前 50% 多头超额收益.....	39
表 31: 原始 20 日合成因子 \cap 5 日因子前 75% 多头超额收益.....	40

表 32: 中证 1000 指增组合多头收益风险特征指标	42
表 33: 中证 1000 指增组合超额收益风险特征指标	42

图 目 录

图 1: 单个神经元结构	5
图 2: Sigmoid 型函数与 ReLU 函数	6
图 3: 多层前馈神经网络	7
图 4: Dropout 示意图	10
图 5: 回测路径合并方式示意图	16
图 6: 全连接神经网络 —— Alpha158	17
图 7: 各模型累计 RankIC — 量价因子预测未来 5 日	18
图 8: 各模型累计 RankIC — 量价因子预测未来 20 日	18
图 9: 全连接神经网络 —— 基本面因子	19
图 10: 各模型累计 RankIC — 基本面因子预测未来 5 日	20
图 11: 各模型累计 RankIC — 基本面因子预测未来 20 日	20
图 12: 量价因子+MLP 多头超额 — 每 5 日调仓	24
图 13: 基本面因子+MLP 多头超额 — 每 5 日调仓	24
图 14: 量价因子+MLP 多头超额 — 每 20 日调仓	25
图 15: 基本面因子+MLP 多头超额 — 每 20 日调仓	25
图 16: 量价、基本面与等权因子累计 RankIC	27
图 17: 量价、基本面与等权因子多头超额收益	27
图 18: 量价、基本面与等权因子多空超额收益	27
图 19: 5 日量价合成因子与 5 日基本面合成因子 ICIR	28
图 20: 量价合成因子与基本面合成因子 ICIR 占比	28
图 21: 量价、基本面与 ICIR 加权因子累计 RankIC	28
图 22: 量价、基本面与 ICIR 加权因子多头超额收益	29
图 23: 量价、基本面与 ICIR 加权因子多空超额收益	29
图 24: 量价因子输入层拼接基本面因子的全连接神经网络结构图	30
图 25: 量价、基本面与输入层拼接因子累计 RankIC	31
图 26: 量价、基本面与输入层拼接因子多头超额收益	31
图 27: 量价、基本面与输入层拼接因子多空超额收益	31
图 28: 量价因子与基本面因子的分支全连接神经网络结构图	32
图 29: 量价、基本面与分支网络因子累计 RankIC	33
图 30: 量价、基本面与分支网络因子多头超额收益	33
图 31: 量价、基本面与分支网络因子多空超额收益	33
图 32: 量价、基本面与各合成方法累计 RankIC — 5 日	34
图 33: 量价、基本面与各合成方法多头超额收益 — 5 日	34
图 34: 量价、基本面与各合成方法多空超额收益 — 5 日	34
图 35: 量价、基本面与各合成方法累计 RankIC — 20 日	35
图 36: 量价、基本面与各合成方法多头超额收益 — 20 日	35
图 37: 量价、基本面与各合成方法多空超额收益 — 20 日	35
图 38: Alpha158 因子、高频量价因子与基本面因子的分支全连接神经网络结构图	36
图 39: 分支网络合成因子多头超额 — 每 5 日调仓	37
图 40: 分支网络合成因子多头超额 — 每 20 日调仓	37
图 41: 原始 20 日合成因子与混频组合多头超额净值	38
图 42: 混频策略相对于缩减成分股的原始 20 日组合的超额收益	39
图 43: 原始 20 日合成因子 \cap 5 日因子前 50% 多头超额净值	39
图 44: 原始 20 日合成因子 \cap 5 日因子前 75% 多头超额净值	40
图 45: 20 日因子在中证 1000 域中累计 RankIC	41
图 46: 中证 1000 指增组合及其超额收益	42

一、深度学习理论

伴随着数据量的与日俱增以及计算能力的大幅提升，深度学习模型近年来发展十分迅速，在图像识别、语音识别、自然语言处理等多个领域不断取得突破。

人工神经网络是通过模仿人脑神经网络系统构造与功能而设计的计算模型，简称神经网络。与生物神经元类似，神经网络由多个节点（人工神经元）相互连接而成。

1.1、神经元与激活函数

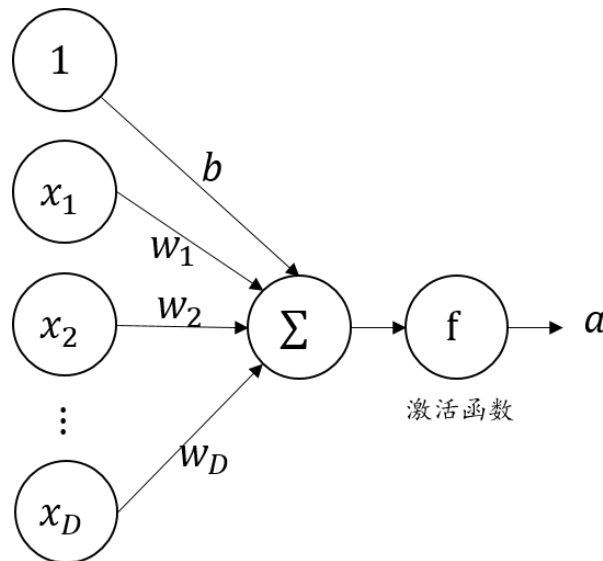
神经元是组成神经网络的基本单元。若一个神经元的输入为 D 个特征，以 $\mathbf{x} = [x_1; x_2; \dots; x_D] \in \mathbb{R}^D$ 表示输入向量， $\mathbf{w} = [w_1; w_2; \dots; w_D] \in \mathbb{R}^D$ 表示权重向量， $b \in \mathbb{R}$ 表示偏置，定义神经元的净输入 z ：

$$z = \mathbf{w}^T \mathbf{x} + b = \sum_{a=1}^D w_a x_a + b$$

净输入 z 经过**激活函数** f 后得到神经元的输出，也称为神经元的活性值：

$$a = f(z)$$

图 1：单个神经元结构



资料：信达证券研发中心

神经网络常用**激活函数**有 Sigmoid 型函数与 ReLU 函数，其中 Logistic 函数与 Tanh 函数是常见 Sigmoid 型函数。

Logistic 函数：

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Logistic 函数的值域为 $(0,1)$ ，其输出可转换为概率，常用于二分类问题。我们可以推导出其梯度为 $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ ，导数最大值为 0.25，当 $x \rightarrow \pm\infty$ 时， $\sigma' \rightarrow 0$ 。

Tanh 函数：

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

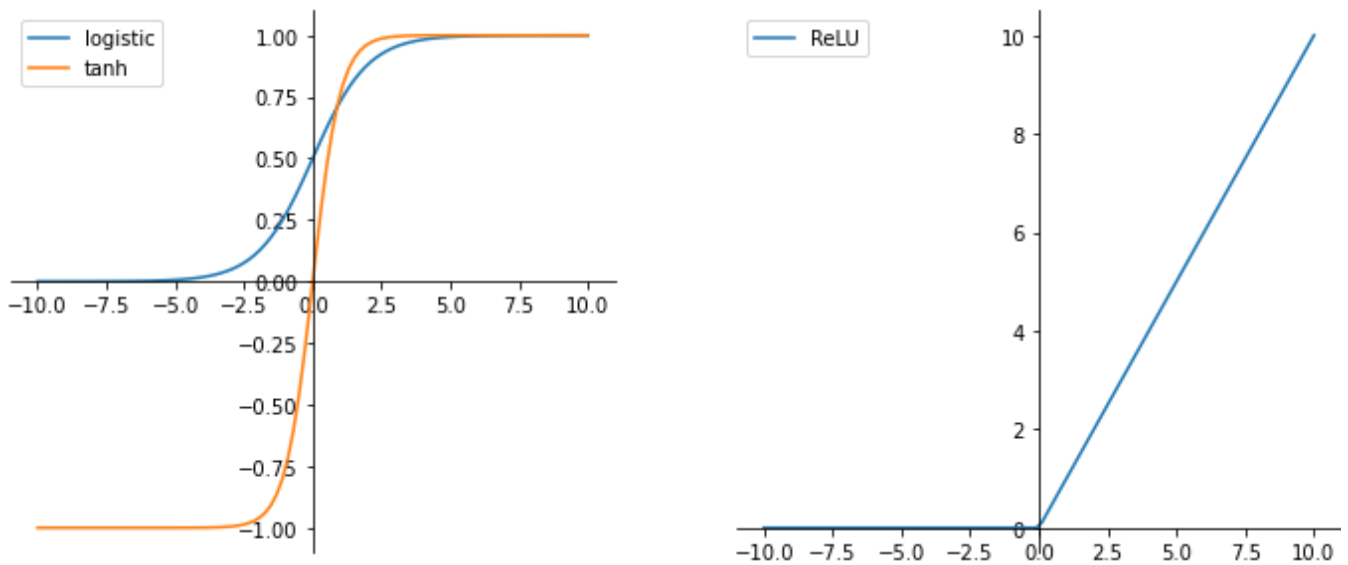
Tanh 函数多用于模型隐藏层，可看作 $\sigma(x)$ 的变换： $\tanh(x) = 2\sigma(2x) - 1$ ，其值域为 $(-1,1)$ ，导数最大值为 1，当 $x \rightarrow \pm\infty$ 时， $\tanh' \rightarrow 0$ 。

ReLU 函数：

$$\text{ReLU}(x) = \max(0, x)$$

ReLU 函数是近年来普遍应用的激活函数，当 $x > 0$ 时，ReLU 函数导数为 1，相比 Sigmoid 型函数，ReLU 计算相对简单因此计算速度较快，且在一定程度上能够缓解神经网络的梯度消失问题。

图 2：Sigmoid 型函数与 ReLU 函数



资料：信达证券研发中心

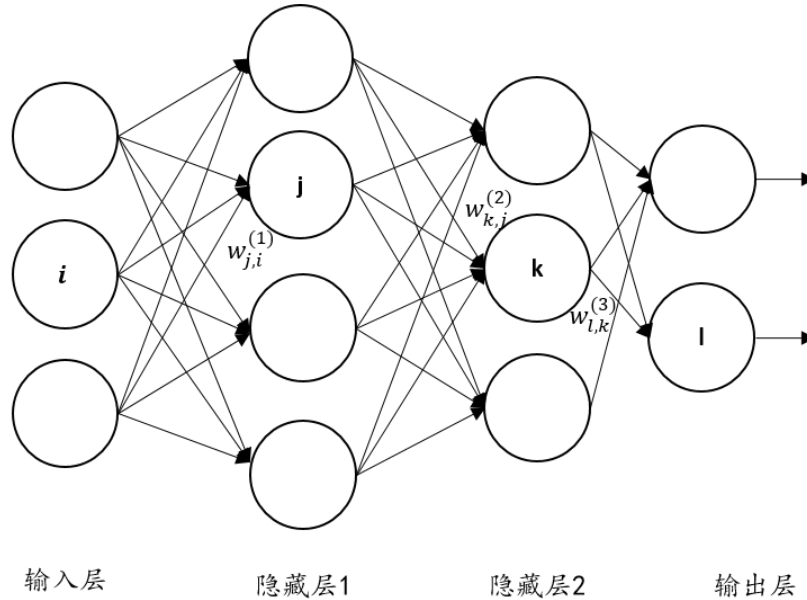
1.2、前馈神经网络拟合能力较强

前馈神经网络（Feedforward Neural Networks）或多层感知机（Multilayer Perceptrons，MLPs）是最有代表性的深度学习模型。前馈神经网络的目标是近似某个函数 f^* ，定义映射 $y = f(\mathbf{x}; \boldsymbol{\theta})$ 并学习参数 $\boldsymbol{\theta}$ 的值，进而获得最佳的函数近似。在前馈神经网络中，输入信息 \mathbf{x} 通过中间函数计算最终输出 y ，整个过程中不含反馈(feedback)连接，信号单向传播。当前馈神经网络被扩展至包含反馈连接，即模型输出反馈至模型本身时，模型被称为循环神经网络（Recurrent Neural Networks）。

前馈神经网络通常用许多不同函数的复合来表示，函数的链状结构（如 $f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$ ）是神经网络最常用结构。神经网络中每一个函数称为网络的一层（layer），函数结构的总长度给出了模型深度（depth），最后一层被称为输出层。神经网络的训练过程促使 $f(\mathbf{x})$ 匹配 $f^*(\mathbf{x})$ 的值，训练数据提供了在不同训练样本上取值且含噪声的 $f^*(\mathbf{x})$ 近似实例，即每个样本 \mathbf{x} 具有一个标签 $y \approx f^*(\mathbf{x})$ 。对每个输入样本 \mathbf{x} ，输出层需要产生接近 y 的值。但对其他层来说，训练数据并没有给出这些层所需要的输出，所以这些中间层被称为隐藏层。网络中每个隐藏层通常都是向量值的，隐藏层的维数决定了模型宽度（width），向量中每个元素可以被视为一个神经元。

对前馈神经网络第 l 层任意神经元 j , 若第 $l-1$ 层输出 $a^{(l-1)}$, 第 $l-1$ 层 i 单元至第 l 层 j 单元的权重为 $w_{ji}^{(l)}$, 偏置为 $b_j^{(l)}$, 则神经元净输入为 $z_j^{(l)} = \sum_i w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)}$, 通过第 l 层激活函数获得神经元 j 的输出 $a_j^{(l)} = f_l(z_j^{(l)})$ 。

图 3: 多层前馈神经网络



资料 : 信达证券研发中心

以 L 层前馈神经网络为例, 第 l 层 ($l = 1, 2, \dots, L$) 神经元的个数为 M_l , 净输入为 $z^{(l)} \in \mathbb{R}^{M_l}$, 激活函数为 f_l , 输出为 $a^{(l)} \in \mathbb{R}^{M_l}$, 第 $l-1$ 层到第 l 层的权重矩阵 $W^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$, 偏置为 $b^{(l)} \in \mathbb{R}^{M_l}$ 。第一层的输入为 $x = a^{(0)}$, 前馈神经网络的信息传播方式为:

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}$$

$$a^{(l)} = f_l(z^{(l)})$$

L 层前馈神经网络通过逐层的信息传递最终输出 $a^{(L)}$ 的过程可以看作一个复合函数 $\phi(x; W, b)$ (W, b 表示所有层连接权重与偏置):

$$x = a^{(0)} \rightarrow z^{(1)} \rightarrow a^{(1)} \rightarrow z^{(2)} \rightarrow \dots \rightarrow a^{(L-1)} \rightarrow z^{(L)} \rightarrow a^{(L)} = \phi(x; W, b)$$

通用近似定理(Universal Approximation Theorem): 令 $\phi(\cdot)$ 是一个非常数、有界、单调递增的连续函数, J_D 是一个 D 维的单位超立方体 $[0, 1]^D$, $C(J_D)$ 是定义在 J_D 上的连续函数集合。对于任意给定的一个函数 $f \in C(J_D)$, 存在一个整数 M , 和一组实数 $v_m, b_m \in \mathbb{R}$ 及实数向量 $w_m \in \mathbb{R}^D$, $m=1, \dots, M$, 以至于我们可以定义函数 $F(x) = \sum_{m=1}^M v_m \phi(w_m^T x + b_m)$ 作为函数 f 的近似实现, 即

$$|F(x) - f(x)| < \epsilon, \forall x \in J_D,$$

其中 $\epsilon > 0$ 是一个很小的正数。

通用近似定理表明, 对于具有线性输出层和至少一个使用“挤压”性质的激活函数的隐藏层组成的前馈神经网络, 只要其隐藏层神经元的数量足够, 它可以以任意的精度来近似任何一个定义在实数空间 \mathbb{R}^D 中的有界闭集函数。所谓“挤压”性质的函数是指像 Sigmoid 函数的有界函数, 但神经网络的通用近似性质也被证明对于其他

类型的激活函数（比如 ReLU）也都是适用的。

全连接神经网络（Fully Connected Neural Network, FCN）是相邻两层的神经元之间为全连接关系的前馈神经网络。

1.3、前馈神经网络的反向传播算法

代价(cost)/损失(loss)函数是一个非负实数函数，用来量化模型预测和真实标签之间的差异。训练算法通常使用梯度下降法实现损失函数最小化。

令 $L(y, f(x; \theta))$ 为损失函数，其中 θ 包含可学习的矩阵 W 与偏置向量 b ，对包含 N 个样本的训练集 $D = \{(x^{(n)}, y^{(n)})\}_{n=1}^N$ ，平均损失为

$$R_D^{\text{emp}}(\theta) = \frac{1}{N} \sum_{n=1}^N L(y^{(n)}, f(x^{(n)}; \theta))$$

经验风险最小化准则即找到使得平均代价 $R_D^{\text{emp}}(\theta)$ 最小的参数 $\theta^* = \underset{\theta}{\operatorname{argmin}} R_D^{\text{emp}}(\theta)$ 。

当我们使用前馈神经网络接收输入样本并产生输出的过程称为**前向传播**（forward propagation），初始信息样本 $x^{(n)}$ 传播至每一层隐藏单元，最终产生**输出** $\hat{y}^{(n)}$ 。在训练过程中，前向传播最终产生一个标量代价函数 $L(y^{(n)}, \hat{y}^{(n)})$ 。**反向传播**（back propagation）算法根据代价函数信息通过网络向后流动以计算**代价函数关于参数的梯度**。

考虑一个全连接前馈神经网络， W ， b 分别为网络中所有权重矩阵与偏置向量，其经验风险函数为：

$$R(W, b) = \frac{1}{N} \sum_{n=1}^N L(y^{(n)}, \hat{y}^{(n)})$$

在梯度下降迭代中，以 α 为学习率，第 l 层的参数 $W^{(l)}$ 和 $b^{(l)}$ 参数更新方式为：

$$W^{(l)} \leftarrow W^{(l)} - \alpha \frac{\partial R(W, b)}{\partial W^{(l)}} = W^{(l)} - \alpha \left(\frac{1}{N} \sum_{n=1}^N \frac{\partial L(y^{(n)}, \hat{y}^{(n)})}{\partial W^{(l)}} \right)$$

$$b^{(l)} \leftarrow b^{(l)} - \alpha \frac{\partial R(W, b)}{\partial b^{(l)}} = b^{(l)} - \alpha \left(\frac{1}{N} \sum_{n=1}^N \frac{\partial L(y^{(n)}, \hat{y}^{(n)})}{\partial b^{(l)}} \right)$$

对于给定样本 (x, y) ，网络输出为 \hat{y} ，参数学习需计算损失函数关于每个参数的偏导数。假设损失函数为 $L(y, \hat{y})$ ，我们需要计算 $\frac{\partial L(y, \hat{y})}{\partial W^{(l)}}$ 与 $\frac{\partial L(y, \hat{y})}{\partial b^{(l)}}$ 。简单起见，引入损失函数对 1 层净输入 $z^{(l)}$ 的偏导数，根据链式法则，我们仅需要计算 $\partial L / \partial z^{(l)}$ 、 $\partial z^{(l)} / \partial W$ 及 $\partial z^{(l)} / \partial b^{(l)}$ ，定义：

$$\delta^{(l)} := \frac{\partial L(y, \hat{y})}{\partial z^{(l)}} = \frac{\partial a^{(l)}}{\partial z^{(l)}} \frac{\partial z^{(l+1)}}{\partial a^{(l)}} \frac{\partial L(y, \hat{y})}{\partial z^{(l+1)}} = f_l'(z^{(l)}) \odot (W^{(l+1)})^T \delta^{(l+1)}$$

由于 $z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}$ ，则损失函数关于 1 层参数的梯度为：

$$\frac{\partial L(y, \hat{y})}{\partial W^{(l)}} = \delta^{(l)} (a^{(l-1)})^T$$

$$\frac{\partial L(y, \hat{y})}{\partial b^{(l)}} = \delta^{(l)}$$

反向传播算法训练过程可以总结为：(1) 计算每一层的净输入 $z^{(l)}$ 和激活值 $a^{(l)}$ ，直到最后一层；(2) 反向传播计算每一层的误差项 $\delta^{(l)}$ ；(3) 计算每一层参数的偏导数，并更新参数。

1.4、学习率调整算法 Adam

作为优选神经网络的重要超参数，学习率对模型性能有显著的影响，Adam是一种学习率自适应的优化算法，由动量梯度下降法和 RMSprop 算法结合而成。

动量梯度下降法 (Momentum Gradient Descent) 是梯度下降法的一种变体，通过引入动量来加速收敛，尤其是在面对高曲率、长谷和鞍点的情况下。

在常规梯度下降法中，以 α 为学习率，以 g_t 为 t 时刻梯度估计，则参数 θ 的更新方式为：

$$\begin{aligned}\theta_t &= \theta_{t-1} - \alpha g_t \\ \Delta\theta_t &= -\alpha g_t\end{aligned}$$

在动量梯度下降法中引入动量 $M_t = \beta_1 M_{t-1} + (1 - \beta_1) g_t$ 以更新梯度($M_0 = 0$):

$$\Delta\theta_t = -\alpha M_t$$

我们可以用小人下山的例子来帮助理解动量梯度下降与常规梯度下降的区别。假设一个小人沿着山谷下山，每一步都根据当前坡度(梯度)来决定移动的方向和步长。若是常规的梯度下降法，小人会每次向着最陡峭的方向下山，在复杂的地形上可能会震荡或来回摆动，导致参数收敛波动较大或者走不出局部最小值点。而动量梯度下降法则根据小人过去的下降方向给小人当前的方向和速度施加惯性，不会因为局部的陡坡而剧烈改变方向，从而减少来回摆动的现象，更加平稳地向山脚(全局最小点)移动。

RMSProp 算法(均方根传递)可以自适应调节学习率，每次迭代梯度 g_t 平方的指数衰减移动平均， ϵ 是非常小的常数($G_0 = 0$)

$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t \odot g_t = (1 - \beta_2) \sum_{\tau=1}^t \beta_2^{t-\tau} g_\tau \odot g_\tau$$

$$\Delta\theta_t = -\frac{\alpha}{\sqrt{G_t + \epsilon}} \odot g_t$$

当初始值为0时，迭代初期 M_t 与 G_t 的值偏小，如 $\beta_1 = 0.9$ ， $M_1 = \beta_1 M_0 + (1 - \beta_1) g_1 = 0.1 g_1$ 。为了避免迭代初期的这种情况，算法将 M_t 与 G_t 分别修正为 \hat{M}_t 与 \hat{G}_t ：

$$\hat{M}_t = M_t / (1 - \beta_1^t)$$

$$\hat{G}_t = G_t / (1 - \beta_2^t)$$

我们同样用小人下山的例子帮助理解。常规梯度下降法中，小人每次都按照同样的步长(固定学习率)朝着最陡的方向下山。这在一些情况下效率较低，比如在狭窄的长谷中小人会频繁地左右摇摆，进展缓慢。而 RMSProp 算法通过对梯度的平方进行指数加权移动平均，来调整每一步的步长。它会根据每个方向的梯度变化调整学习率，

使得在平缓区域可以采取较大的步长，而在陡峭区域采取较小的步长。当小人发现这里的坡度变化很大（例如在悬崖峭壁上），因此他会选择每次走得小心翼翼，采取较小的步长。当小人发现这里的坡度变化很小（例如在大平原上），因此他会选择迈大步前进，加快下山速度。

Adam 算法是动量法和 RMSprop 算法的结合，不但使用动量作为参数更新方向，而且可以自适应调整为学习率。我们采用 Adam 算法进行优化，算法汇总如下。

$$\Delta\theta_t = \frac{\alpha}{\sqrt{\hat{G}_t + \epsilon}} \hat{M}_t$$

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) g_t$$

$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t \odot g_t$$

$$\hat{M}_t = \hat{M}_t / (1 - \beta_1^t)$$

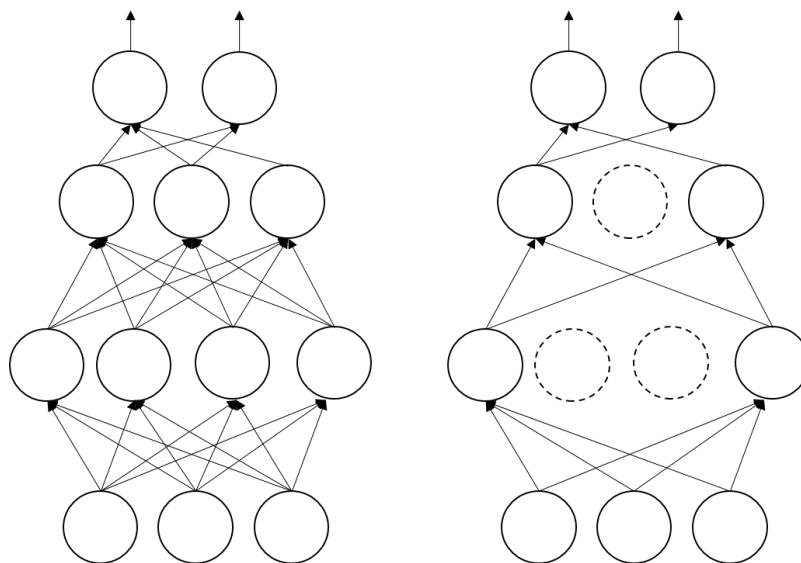
$$\hat{G}_t = \hat{G}_t / (1 - \beta_2^t)$$

1.5、丢弃法 (Dropout)

Dropout 层是一种在神经网络中常用的正则化技术，用于防止模型过拟合。其核心思想是在每次训练过程中随机“丢弃”一部分神经元，即将这些神经元的输出设为零，从而减少神经元之间的相互依赖，提高模型的泛化能力。

具体来说，在每次训练迭代中，Dropout 层会以一定的概率 p 随机选择一部分神经元，将它们的输出设为 0，而其他神经元的输出则按正常流程进行计算。在测试过程中，所有神经元的输出都将被保留，并且为了保持输出的一致性，通常会将神经元的输出乘以一个缩放系数 $(1-p)$ 。

图 4：Dropout 示意图



资料 : 信达证券研发中心

二、Lasso Vs MLP —— 分别用于量价与基本面因子

通常在股票收益预测的场景下，未经加工的金融数据（例如高开低收、成交量、成交额等数据）信噪比较低，且未经标准化使得截面可比性较差，若直接作为特征输入用模型拟合较容易出现“Garbage in, Garbage out”的问题。

因此，我们通过构建具有一定逻辑的选股因子，可以有效地提取出原始数据中的关键特征，进行降维处理，从而减少模型训练的复杂度和计算成本。选股因子是基于金融理论和实证研究构建的，量价因子能够对股票的价格走势和成交变化进行描绘，让模型能够学习不同量价情况下股票的收益分布特征，而基本面因子，例如 EP、BP 等则反映了公司估值水平，提供了量价维度外的另一个重要的视角。

2.1、量价因子数据集

我们采用 Github 上的开源项目 Qlib 所集成的 Alpha158 量价因子进行模型层面的基础研究，它包含了基于价格和成交量的多种因子，不算窗口期则有 42 个因子，部分因子考虑了 5/10/20/30/60 的交易日的窗口期，因此若算窗口期则共有 158 个因子。我们将 Alpha158 因子分为以下 5 类：

日内因子：这些因子仅使用当天的开盘价、收盘价、最高价、最低价以及均价数据，共包含 13 个因子。这些因子捕捉了市场在单个交易日内的波动和变化特征。

波动因子：波动因子主要衡量股票价格的波动性，共包含 5 个因子。波动性因子可以帮助我们理解股票价格的变动幅度和频率，从而更好地评估风险和收益。

价因子：价因子是基于股票的价格信息计算得出的，共包含 100 个因子。这些因子涉及多种价格计算方法和统计指标，能够反映出股票价格的长期和短期趋势。

量因子：量因子基于成交量数据，共包含 30 个因子。成交量是市场交易活动的直接反映，量因子能够提供关于市场流动性和投资者行为的重要信息。

量价相关性因子：这些因子同时考虑了成交量和价格的关系，共包含 10 个因子。通过分析量价关系，可以更深入地了解市场的供需动态和价格变动的内在驱动力。

表 1: Alpha 158 因子列表

因子大类	因子名称	窗口参数	因子释义	算式
日内	HIGH0		最高价除以收盘价	High/Close
	KLEN		K 线长度	$(\text{High}-\text{Low})/\text{Open}$
	KLOW1		下影线长度相对开盘价的比例	$(\text{Less}(\text{Open}, \text{Close})-\text{Low})/\text{Open}$
	KLOW2		下影线长度相对 K 线整体的比例	$(\text{Less}(\text{Open}, \text{Close})-\text{Low})/(\text{High}-\text{Low}+1e-12)$
	KMID1		K 线实体长度相对开盘价的比例	$(\text{Close}-\text{Open})/\text{Open}$
	KMID2		K 线实体长度相对 K 线整体的比例	$(\text{Close}-\text{Open})/(\text{High}-\text{Low}+1e-12)$

	KSFT1		收盘价在整个价格区间（最高价到最低价）中的位置，相对开盘价的比例	$(2 * \$close - \$high - \$low) / \$open$
	KSFT2		收盘价在整个价格区间（最高价到最低价）中的位置，相对 K 线整体的比例	$(2 * \$close - \$high - \$low) / (\$high - \$low + 1e-12)$
	KUP1		上影线长度相对开盘价的比例	$(\$high - Greater(\$open, \$close)) / \$open$
	KUP2		上影线长度相对 K 线整体的比例	$(\$high - Greater(\$open, \$close)) / (\$high - \$low + 1e-12)$
	LOW0		最低价除以收盘价	$\$low / \$close$
	OPEN0		开盘价除以收盘价	$\$open / \$close$
	VWAP0		均价除以收盘价	$\$vwap / \$close$
波动	STD	[5, 10, 20, 30, 60]	过去 d 天的收盘价标准差，除以最新的收盘价以去除单位。	$Std(\$close, \%d) / \$close$
	BETA	[5, 10, 20, 30, 60]	过去 d 天的收盘价变化率，除以最新的收盘价以去除单位。例如，过去 d 天每天价格上涨 10 美元，那么斜率将是 10。	$Slope(\$close, \%d) / \$close$
	CNTD	[5, 10, 20, 30, 60]	过去上涨天数与过去下跌天数之间的差异。	$Mean(\$close > Ref(\$close, 1), \%d) - Mean(\$close < Ref(\$close, 1), \%d)$
价	CNTN	[5, 10, 20, 30, 60]	过去 d 天内价格下跌的天数百分比。	$Mean(\$close < Ref(\$close, 1), \%d)$
	CNTP	[5, 10, 20, 30, 60]	过去 d 天内价格上涨的天数百分比。	$Mean(\$close > Ref(\$close, 1), \%d)$
	IMAX	[5, 10, 20, 30, 60]	当前日期与之前最高价日期之间的天数，属于 Aroon 指标的一部分。该指标衡量一段时间内高点之间和低点之间的时间。强劲的上升趋势会定期看到新的高点，而强劲的下降趋势会定期看到新的低点。	$IdxMax(\$high, \%d) / \%d$
	IMIN	[5, 10, 20, 30, 60]	当前日期与之前最低价日期之间的天数，属于 Aroon 指标的一部分。	$IdxMin(\$low, \%d) / \%d$
	IMXD	[5, 10, 20, 30, 60]	之前最低价日期出现在最高价日期之后的时间段。较大的值表示向下的动量。	$(IdxMax(\$high, \%d) - IdxMin(\$low, \%d)) / \%d$
	MA	[5, 10, 20, 30, 60]	简单移动平均线，过去 d 天的简单移动平均线，除以最新的收盘价以去除单位。	$Mean(\$close, \%d) / \$close$
	MAX	[5, 10, 20, 30, 60]	过去 d 天的最高价，除以最新的收盘价以去除单位。	$Max(\$high, \%d) / \$close$
	MIN	[5, 10, 20, 30, 60]	过去 d 天的最低价，除以最新的收盘价以去除单位。	$Min(\$low, \%d) / \$close$
	QTLT	[5, 10, 20, 30, 60]	过去 d 天收盘价的 20% 分位数，除以最新的收盘价以去除单位。	$Quantile(\$close, \%d, 0.2) / \$close$
	QTLU	[5, 10, 20, 30, 60]	过去 d 天收盘价的 80% 分位数，除以最新的收盘价以去除单位。	$Quantile(\$close, \%d, 0.8) / \$close$
	RANK	[5, 10, 20, 30, 60]	当前收盘价在过去 d 天收盘价中的百分位数，表示当前价格水平相对于过去 d 天的比较，提供额外的信息给移动平均线。	$Rank(\$close, \%d)$
	RESI	[5, 10, 20, 30, 60]	过去 d 天线性回归的残差，表示过去 d 天的趋势线性程度。	$Resi(\$close, \%d) / \$close$
	ROC	[5, 10, 20, 30, 60]	变化率，过去 d 天的价格变化，除以最新的收盘价以去除单位。	$Ref(\$close, \%d) / \$close$

	RSQR	[5, 10, 20, 30, 60]	过去 d 天线性回归的 R 平方值，表示趋势的线性程度。	$\text{Rsquare}(\$close, \%d)$
	RSV	[5, 10, 20, 30, 60]	表示过去 d 天内价格在上下阻力价格之间的位置。	$(\$close - \text{Min}(\$low, \%d)) / (\text{Max}(\$high, \%d) - \text{Min}(\$low, \%d) + 1e-12)$
	SUMD	[5, 10, 20, 30, 60]	总收益与总损失之间的差异比率，类似于 RSI 指标。	$(\text{Sum}(\text{Greater}(\$close - \text{Ref}(\$close, 1), 0), \%d) - \text{Sum}(\text{Greater}(\text{Ref}(\$close, 1) - \$close, 0), \%d)) / (\text{Sum}(\text{Abs}(\$close - \text{Ref}(\$close, 1)), \%d) + 1e-12)$
	SUMN	[5, 10, 20, 30, 60]	总损失与绝对总价格变化的比率，可以通过 $\text{SUMN} = 1 - \text{SUMD}$ 得到，类似于 RSI 指标。	$\text{Sum}(\text{Greater}(\text{Ref}(\$close, 1) - \$close, 0), \%d) / (\text{Sum}(\text{Abs}(\$close - \text{Ref}(\$close, 1)), \%d) + 1e-12)$
	SUMP	[5, 10, 20, 30, 60]	总收益与绝对总价格变化的比率，类似于 RSI 指标。	$\text{Sum}(\text{Greater}(\$close - \text{Ref}(\$close, 1), 0), \%d) / (\text{Sum}(\text{Abs}(\$close - \text{Ref}(\$close, 1)), \%d) + 1e-12)$
	WVMA	[5, 10, 20, 30, 60]	交易量加权的 price 变化波动率。	$\text{Std}(\text{Abs}(\$close / \text{Ref}(\$close, 1) - 1) * \$volume, \%d) / (\text{Mean}(\text{Abs}(\$close / \text{Ref}(\$close, 1) - 1) * \$volume, \%d) + 1e-12)$
	VMA	[5, 10, 20, 30, 60]	简单交易量移动平均线。	$\text{Mean}(\$volume, \%d) / (\$volume + 1e-12)$
	VSTD	[5, 10, 20, 30, 60]	过去 d 天的交易量标准差。	$\text{Std}(\$volume, \%d) / (\$volume + 1e-12)$
	VSUMD	[5, 10, 20, 30, 60]	总交易量增加与总交易量减少之间的差异比率，类似于交易量的 RSI 指标。	$(\text{Sum}(\text{Greater}(\$volume - \text{Ref}(\$volume, 1), 0), \%d) - \text{Sum}(\text{Greater}(\text{Ref}(\$volume, 1) - \$volume, 0), \%d)) / (\text{Sum}(\text{Abs}(\$volume - \text{Ref}(\$volume, 1)), \%d) + 1e-12)$
量	VSUMP	[5, 10, 20, 30, 60]	总交易量增加与绝对总交易量变化的比率。	$\text{Sum}(\text{Greater}(\$volume - \text{Ref}(\$volume, 1), 0), \%d) / (\text{Sum}(\text{Abs}(\$volume - \text{Ref}(\$volume, 1)), \%d) + 1e-12)$
	VSUMN	[5, 10, 20, 30, 60]	总交易量减少与绝对总交易量变化的比率，可以通过 $\text{VSUMN} = 1 - \text{VSUMP}$ 得到。	$\text{Sum}(\text{Greater}(\text{Ref}(\$volume, 1) - \$volume, 0), \%d) / (\text{Sum}(\text{Abs}(\$volume - \text{Ref}(\$volume, 1)), \%d) + 1e-12)$
量价相关性	CORD	[5, 10, 20, 30, 60]	价格变化率与交易量变化率之间的相关性。	$\text{Corr}(\$close / \text{Ref}(\$close, 1), \text{Log}(\$volume / \text{Ref}(\$volume, 1) + 1), \%d)$
	CORR	[5, 10, 20, 30, 60]	绝对收盘价与对数交易量之间的相关性。	$\text{Corr}(\$close, \text{Log}(\$volume + 1), \%d)$

资料：Qlib、信达证券研发中心

备注：算式中的“%d”代表回看窗口期

2.2、基本面因子数据集

除了 Alpha158 量价因子外，我们还加入了常规的基本面因子，包含了估值、质量、成长和分析师一致预期等维度，共计 38 个因子，详细列表如下：

表 2：基本面因子列表

因子大类	因子名称	算式
估值	EP_TTM	归属于母公司所有者的净利润_TTM（财报+预告+快报） / 总市值
	BP_LF	归属母公司所有者权益(或股东权益)合计（财报+快报+发行/分配/吸收合并调整） / 总市值
	SP_TTM	营业收入_TTM（财报+预告+快报） / 总市值
	CFP_TTM	经营活动现金净流量_TTM（财报） / 总市值
	DIVIDEND_YIELD_TTM	股息率（近 12 个月）

	EPS_QR	归属于母公司所有者的净利润_单季度【财报+预告+快报】 / 最新总股本
	EPS_QR_R	(当前季度 EPS - 过去 8 个季度 EPS 均值) / 过去 8 个季度 EPS 标准差【财报+预告+快报】
	EPS_TTM	归属于母公司所有者的净利润_TTM【财报+预告+快报】 / 最新总股本
	EPS_TTM_R	(当前季度 EPS_TTM - 过去 8 个季度 EPS_TTM 均值) / 过去 8 个季度 EPS_TTM 标准差【财报+预告+快报】
质量	ROE_QR	2 * 归属于母公司所有者的净利润_单季度【财报+预告+快报】 / (股东权益合计【最新财报+快报+调整】 + 股东权益合计【上期财报+快报+调整】)
	ROE_QR_R	(当前季度 ROE - 过去 8 个季度 ROE 均值) / 过去 8 个季度 ROE 标准差【财报+预告+快报】
	ROE_TTM	归属于母公司所有者的净利润_TTM【财报+预告+快报】 / 股东权益合计【最新财报+快报+调整】
	ROE_TTM_R	(当前 ROETTM - 过去 8 个季度 ROETTM 均值) / 过去 8 个季度 ROETTM 标准差【财报+预告+快报】
成长	NETPROFIT_QR_YOY	归属于母公司所有者的净利润_单季度_同比增速【财报+预告+快报】
	NETPROFIT_QR_YOY_R	(当前季度归母净利润同比 - 过去 8 个季度归母净利润同比均值) / 过去 8 个季度归母净利润同比标准差【财报+预告+快报】
	NETPROFIT_QR_YOY_DIFF	归属于母公司所有者的净利润_单季度_同比增速_环比差分【财报+预告+快报】
	NETPROFIT_QR_YOY_DIFF_R	(当前季度归母净利润同比的环比 - 过去 8 个季度归母净利润同比的环比均值) / 过去 8 个季度归母净利润同比的环比标准差【财报+预告+快报】
	NETPROFIT_TTM_YOY	归属于母公司所有者的净利润_TTM_同比增速【财报+预告+快报】
	NETPROFIT_TTM_YOY_R	(当前季度归母净利润 TTM 同比 - 过去 8 个季度归母净利润 TTM 同比均值) / 过去 8 个季度归母净利润 TTM 同比标准差【财报+预告+快报】
	NETPROFIT_TTM_YOY_DIFF	归属于母公司所有者的净利润_TTM_同比增速_环比差分【财报+预告+快报】
	NETPROFIT_TTM_YOY_DIFF_R	(当前季度归母净利润 TTM 同比的环比 - 过去 8 个季度归母净利润 TTM 同比的环比均值) / 过去 8 个季度归母净利润 TTM 同比的环比标准差【财报+预告+快报】
分析师	CON_BP_ROLL	一致预期净市率 (未来 12 个月)
	CON_BP_ROLL_PCT_60	(一致预期净市率 - 一致预期净市率 (60 天前)) / abs (一致预期净市率 (60 天前)) (未来 12 个月)
	CON_SP_ROLL	一致预期销市率 (未来 12 个月)
	CON_SP_ROLL_PCT_60	(一致预期销市率 - 一致预期销市率 (60 天前)) / abs (一致预期销市率 (60 天前)) (未来 12 个月)
	CON_EP_ROLL	一致预期盈市率 (未来 12 个月)
	CON_EP_ROLL_PCT_60	(一致预期盈市率 - 一致预期盈市率 (60 天前)) / abs (一致预期盈市率 (60 天前)) (未来 12 个月)
	CON_EPS_ROLL	一致预期 EPS (未来 12 个月)
	CON_EPS_ROLL_PCT_60	(一致预期 EPS - 一致预期 EPS (60 天前)) / abs (一致预期 EPS (60 天前)) (未来 12 个月)
	CON_ROE_ROLL	一致预期 ROE (未来 12 个月)
	CON_ROE_ROLL_PCT_60	(一致预期 ROE - 一致预期 ROE (60 天前)) / abs (一致预期 ROE (60 天前)) (未来 12 个月)
	CON_OR_ROLL_PCT_60	(一致预期营业收入 - 一致预期营业收入 (60 天前)) / abs (一致预期营业收入 (60 天前)) (未来 12 个月)
	CON_NP_ROLL_PCT_60	(一致预期净利润 - 一致预期净利润 (60 天前)) / abs (一致预期净利润 (60 天前)) (未来 12 个月)
	CON_OR_YOY_ROLL	一致预期营业收入同比 (未来 12 个月)

CON_OR_YOY_ROLL_PCT_60	$(\text{一致预期营业收入同比} - \text{一致预期营业收入同比}(60\text{天前})) / \text{abs}(\text{一致预期营业收入同比}(60\text{天前}))$ (未来12个月)
CON_NP_YOY_ROLL	一致预期净利润同比 (未来12个月)
CON_NP_YOY_ROLL_PCT_60	$(\text{一致预期净利润同比} - \text{一致预期净利润同比}(60\text{天前})) / \text{abs}(\text{一致预期净利润同比}(60\text{天前}))$ (未来12个月)
SUE_MINUS_CON_QR	$(\text{单季度归母净利}(未调整财报+调整后财报+快报+预告) - \text{上个交易日一致预期单季度归母净利}) / \text{历史8期标差}$

资料 : 信达证券研发中心

2.3、数据预处理与网络参数设置

特征与标签:

- 特征 X: 158 个 Alpha158 因子与 38 个基本面因子, 去极值、截面 ZSCORE 标准化、市值行业中性化, 分开测试。
- 标签 Y: 个股从下一个交易日的 VWAP 算起, 未来 5 个交易日与 20 个交易日的 VWAP 收益率, 市值行业中性化后, 计算截面排序百分位。分开测试预测 5 日收益率与预测 20 日收益率。
- 数据采样: 以 2013 年 12 月 31 日为第一次预测点, 回看过去 5 年 (252*5 个交易日) 数据作为样本, 数据集按顺序划分, 前 90% 的交易日作为训练集, 后 10% 的交易日作为验证集, 用于拟合模型, 接下来半年 (126 个交易日) 用该模型预测每个股票的收益率, 每半年 (126 个交易日) 训练一次模型。

神经网络参数:

- Batch: 按交易日做 Batch 拆分, 因此训练集共有 1134 (252*5*0.9) 个 Batch, 每个 Batch 大小为当前截面股票数量。
- 损失函数: 预测值与真实值 Pearson 相关性的相反数。
- 优化器: Adam; 学习速率: 0.001。
- 最大 Epoch: 100; 早停 Epoch: 10 (验证集 Loss 连续 10 轮没有创新低则停止训练)。

全 A 选股组合回测参数与方式:

- 回测区间: 2013 年 12 月 31 日至 2024 年 5 月 31 日。
- 剔除: 剔除上市不满 365 个自然日的新股, 剔除 ST 股。
- 组合构建: 取因子值前 10% 为多头组合, 后 10% 为空头组合, 所有股票等权为基准组合。
- 交易方式: 每 5 日 / 20 日调仓, 以下一个交易日的 VWAP 价格成交, 交易费率为单边千分之一。
- 交易限制: 一字涨停不能买入, 一字跌停不能卖出, 停牌不可交易, 多头 / 空头组合中的多余权重分给其余股票。

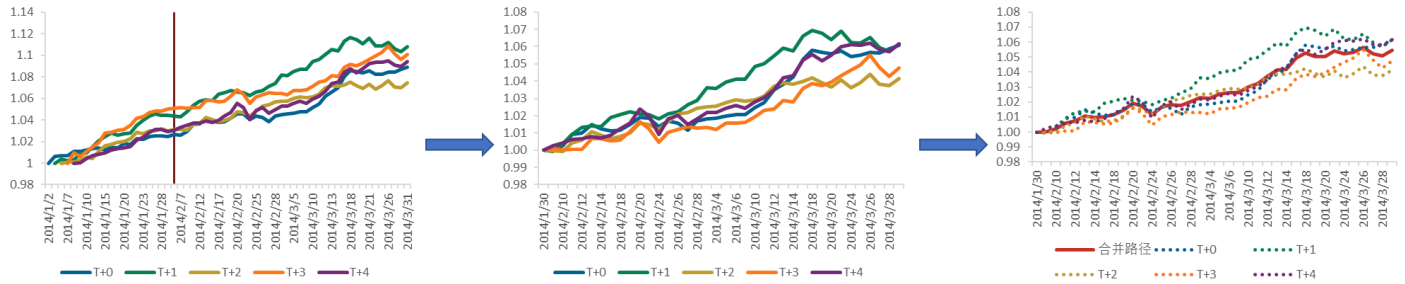
路径依赖与路径合并:

日频因子, 每一天皆可预测未来 5 日 (20 日) 的收益率, 这意味着我们可以利用这些因子在每个交易日对未来一段时间内的股票收益率进行预测。然而, 如果我们选定某一个特定的起点并每隔 5 日 (20 日) 进行一次

调仓操作，那么由于起点选择的不同，可能会导致我们得到不同的净值曲线，这种现象被称为路径依赖问题，若任取一条净值曲线计算收益与风险都会有失公允。

为规避此问题，我们在 2014 年 1 月 30 日将不同调仓起点的净值曲线做截断，并将所有净值曲线归一化处理，使得它们的初始值相同，然后计算所有净值曲线在每一天的收益率的均值。通过这种方法，我们可以得到一条综合的净值曲线，代表了在不同起点下调仓策略的平均表现。这样一来，我们能够更准确地评估投资策略的收益与风险，避免了单一调仓起点带来的路径依赖问题。

图 5：回测路径合并方式示意图



资料：信达证券研发中心

2.4、Lasso Vs MLP —— 量价因子

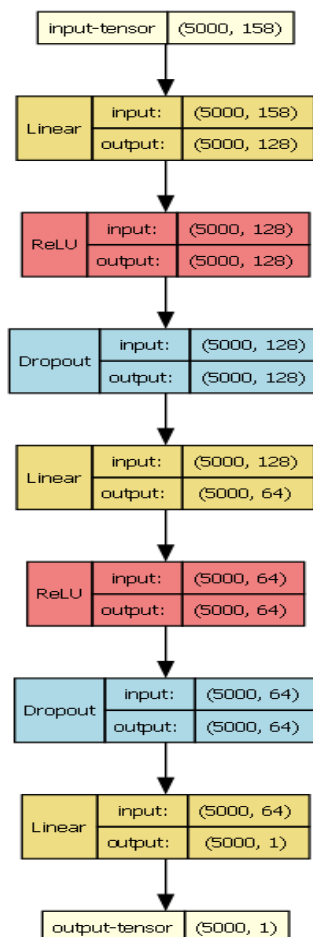
本篇报告作为系列第一篇报告，从简单的线性与非线性模型入手，先分别测试量价因子与基本面因子在线性与非线性结合框架下的差异。为了使得模型结论尽可能可比，我们确保线性模型与非线性模型的特征与标签一致，训练集范围相同，滚动训练时点相同。

对于 Alpha158 量价因子，我们首先采用线性模型 Lasso 作为基准模型，正则化参数 λ 取 0.001。

$$\text{Lasso} = \min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n (y_i - X_i \beta)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}$$

在非线性模型端，对于 158 个量价因子，我们采用 2 层的全连接神经网络（MLP）。隐藏层神经元数量分别为 128 和 64，每层皆采用 RELU 激活函数，并在激活函数后衔接参数为 0.2 的 Dropout 层。图 6 中的输入(5000, 158)代表 5000 个示例股票和 158 个因子。

图 6：全连接神经网络 —— Alpha158

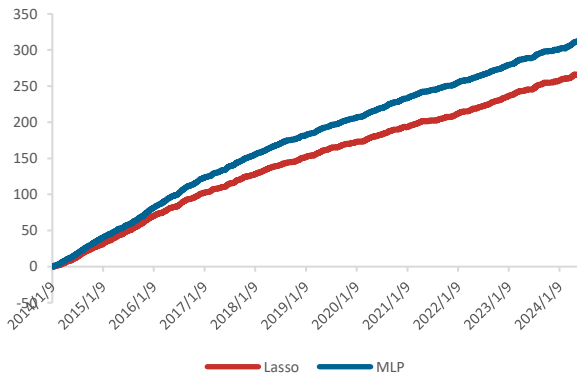


资料：信达证券研发中心

将上述线性 Lasso 模型与非线性全连接神经网络 MLP 模型用于 Alpha158 量价因子，用于预测未来 5 个交易日的收益率。通过实验可以发现，非线性的全连接神经网络相较于线性模型 Lasso，展现出了更出色的预测能力。MLP 模型的 RankIC 高出 Lasso 模型近 2pct，费后的多头超额收益提升近 10pct，多空超额收益提升约 40pct。然而，尽管 MLP 模型在收益率预测上表现更优，但 Lasso 模型在多头超额收益的最大回撤方面表现更好，显示了其在某些风险控制方面的优势。

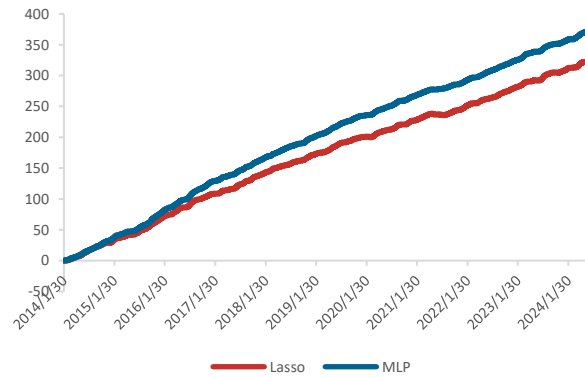
将上述两个模型用于预测未来 20 个交易日的收益率。结果表明，非线性的 MLP 相较于线性 Lasso，同样展现出了更出色的选股能力。MLP 模型在 RankIC、多头超额收益、多空超额收益等指标上皆有较为显著的提高。这表明非线性模型在拟合截面价量数据时，能够更有效地捕捉数据中的非线性关系，从而提供相较于线性结合更准确的收益率预测。

图 7：各模型累计 RankIC — 量价因子预测未来 5 日



资料：Wind、信达证券研发中心
 数据日期：2014 年 1 月 9 日至 2024 年 5 月 31 日

图 8：各模型累计 RankIC — 量价因子预测未来 20 日



资料：Wind、信达证券研发中心
 数据日期：2014 年 1 月 30 日至 2024 年 5 月 31 日

表 3：各模型全区间 IC 统计 — 量价因子预测未来 5 日与 20 日

预测区间	模型名称	RankIC	ICIR	胜率
预测未来 5 日	Lasso	10.57%	1.09	86.82%
	MLP	12.39%	1.40	92.72%
预测未来 20 日	Lasso	12.81%	1.32	90.53%
	MLP	14.75%	1.78	96.46%

资料：Wind、信达证券研发中心
 数据日期：2014 年 1 月 9 日至 2024 年 5 月 31 日

表 4：各模型全区间多头超额与多空超额收益风险特征指标 — 量价因子

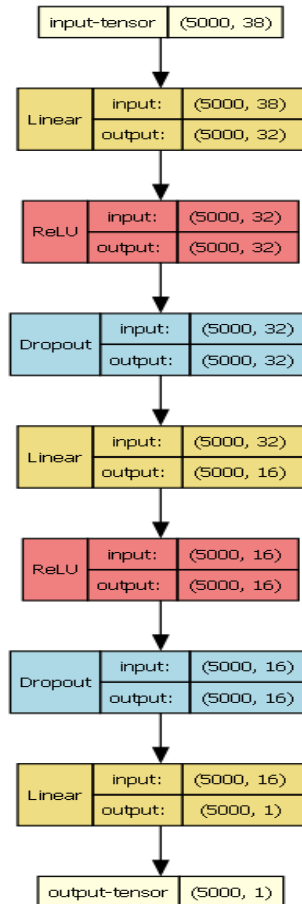
预测区间	模型名称	多头超额					多空超额				
		年化收益	年化波动	最大回撤	收益波动比	收益回撤比	年化收益	年化波动	最大回撤	收益波动比	收益回撤比
预测未来 5 日	Lasso	13.03%	4.31%	-6.47%	3.02	2.01	82.39%	12.09%	-9.96%	6.81	8.27
	MLP	22.45%	4.92%	-13.25%	4.57	1.69	122.44%	11.30%	-10.84%	10.84	11.30
预测未来 20 日	Lasso	10.08%	4.19%	-8.34%	2.41	1.21	43.47%	11.04%	-11.54%	3.94	3.77
	MLP	16.62%	4.27%	-7.84%	3.89	2.12	61.64%	9.98%	-8.71%	6.18	7.08

资料：Wind、信达证券研发中心
 数据日期：2014 年 1 月 30 日至 2024 年 5 月 31 日

2.5、Lasso Vs MLP —— 基本面因子

对于基本面因子，我们同样采用线性模型 Lasso 回归作为基准模型，正则化参数 λ 取 0.001。在非线性和端，同样采用 2 层的 MLP 模型，然而，由于基本面因子的数量为 38 个，因此我们将隐藏层神经元数量缩减为 32 和 16。

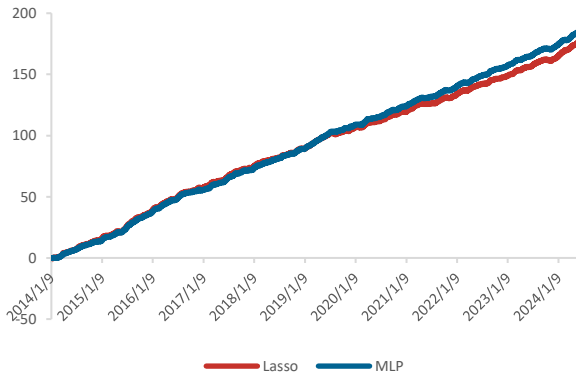
图 9：全连接神经网络 —— 基本面因子



资料 : 信达证券研发中心

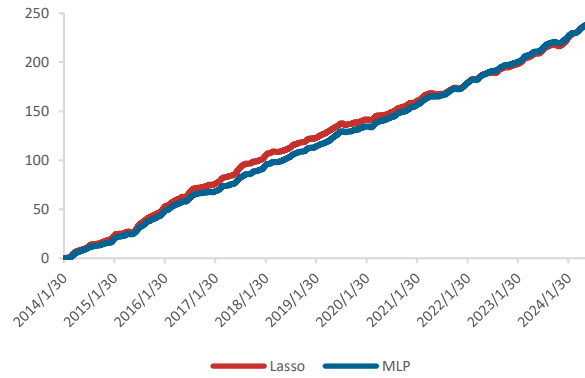
用 Lasso 与 MLP 拟合基本面因子预测未来 5 日与未来 20 日的收益率，可以发现 MLP 相较于 Lasso 的提升幅度，不如在量价因子上 MLP 相对 Lasso 的增强效果。预测未来 5 日收益率的 RankIC，MLP 相较于 Lasso 仅提升 0.35pct，多头超额与多空超额收益提升幅度微弱。当预测未来 20 日时，两个模型的差距亦不显著。

图 10：各模型累计 RankIC — 基本面因子预测未来 5 日



资料：Wind、信达证券研发中心
 数据日期：2014 年 1 月 9 日至 2024 年 5 月 31 日

图 11：各模型累计 RankIC — 基本面因子预测未来 20 日



资料：Wind、信达证券研发中心
 数据日期：2014 年 1 月 30 日至 2024 年 5 月 31 日

表 5：各模型全区间 IC 统计 — 基本面因子预测未来 5 日与 20 日

预测区间	模型名称	RankIC	ICIR	胜率
预测未来 5 日	Lasso	6.97%	0.82	78.83%
	MLP	7.32%	0.92	82.98%
预测未来 20 日	Lasso	9.46%	1.10	88.34%
	MLP	9.47%	1.27	90.84%

资料：Wind、信达证券研发中心
 数据日期：2014 年 1 月 9 日至 2024 年 5 月 31 日

表 6：各模型全区间多头超额与多空超额收益风险特征指标 — 基本面因子

预测区间	模型名称	多头超额					多空超额				
		年化收益	年化波动	最大回撤	收益波动比	收益回撤比	年化收益	年化波动	最大回撤	收益波动比	收益回撤比
预测未来 5 日	Lasso	10.80%	5.39%	-8.51%	2.00	1.27	46.61%	11.36%	-14.11%	4.10	3.30
	MLP	10.85%	5.04%	-5.67%	2.15	1.91	47.17%	11.79%	-10.55%	4.00	4.47
预测未来 20 日	Lasso	8.47%	5.03%	-9.75%	1.69	0.87	31.10%	9.83%	-12.59%	3.16	2.47
	MLP	9.21%	4.48%	-5.09%	2.05	1.81	30.46%	9.96%	-12.67%	3.06	2.40

资料：Wind、信达证券研发中心
 数据日期：2014 年 1 月 30 日至 2024 年 5 月 31 日

2.6、全连接网络隐藏层神经元数量设置的敏感性分析与启示

前文中，我们对神经网络的隐藏层神经元数量设置是偏经验的设置方式，即设为 2 的次方数，且根据降维的思路，每一层神经元数量比上一层少。但是这种方式是否是最佳的设置方式？全连接神经网络在量价因子上相较于线性模型的优势在不同参数设置下是否稳定？自身效果是否对参数敏感？因此我们以 Alpha158 量价因子，预测未来 5 日收益率为例，对一层与两层的全连接神经网络的神经元数量进行遍历，以观察其中的规律。

表 7：一层的全连接神经网络不同神经元数量下合成因子效果

隐藏层神经元数量	和 Lasso 因子相关性	RankIC	ICIR
16	85.04%	11.91%	1.29
32	82.47%	12.23%	1.32
64	77.66%	12.24%	1.34
128	74.94%	12.39%	1.37
256	71.70%	12.39%	1.42
512	70.55%	12.35%	1.42

资料：Wind、信达证券研发中心

数据日期：2014 年 1 月 9 日至 2024 年 5 月 31 日

若是一个 1 层的全连接神经网络，我们观察到隐藏层的神经元数量越多，能够引入越多的非线性，因此和线性因子相关性越低。在 RankIC 评价角度，随着隐藏层神经元数量增加，RankIC 呈先增后减的趋势，因为若神经元数量较少，则引入的非线性不足，因此提升不够，而若神经元数量过多，则容易出现过拟合的问题。

在 2 层隐藏层的结构下，同样可以看到全连接神经网络合成因子与线性因子的相关性随着神经元数量越多（从表 8 的左上角至右下角），相关性越低。同时也观察到，下三角矩阵的相关性数值，总是小于其轴对称位置对应的上三角矩阵的相关性数值，简单来说就是在第一层单元数和第二层单元数的乘积一样的情况下，将第一层设置较大第二层设置较小，相较于第一层设置较小第二层设置较大能够引入更多的非线性。

表 8：两层的全连接神经网络不同神经元数量下合成因子与 Lasso 因子的相关性

	16	32	64	128	256	512
16	82.23%	80.87%	80.76%	80.08%	79.08%	78.60%
32	79.13%	78.47%	77.50%	75.39%	75.20%	74.34%
64	74.97%	75.99%	74.39%	73.62%	72.11%	71.45%
128	73.54%	71.37%	72.06%	69.89%	69.73%	68.08%
256	71.18%	69.02%	69.07%	68.43%	67.12%	65.32%
512	70.06%	68.87%	65.85%	66.41%	65.32%	63.19%

资料：Wind、信达证券研发中心

数据日期：2014 年 1 月 9 日至 2024 年 5 月 31 日

备注：第一列数字为第一层神经元数量，第一行数字为第二层神经元数量

观察两层全连接神经网络合成因子的 RankIC 矩阵，可见同样也符合一层全连接神经网络的结论，即隐藏层神经元越多（从表 9 左上角至右下角），合成因子的 RankIC 呈先增后减的趋势。除此之外，也能看到 RankIC 较高的位置基本集中在将第一层设置较大（64 至 512），第二层设置较小（16 至 64）的位置，说明先前经验值的设置方式在用全连接神经网络选股的场景下也同样较为适用。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/778061100127006124>