

队伍编号	205745
题号	D

新零售目标产品精准需求预测问题的方案评估

摘要

近年来，伴随着互联网技术的应用，一场重塑传统商业模式的“新零售”时代浪潮在日益刷新我们的生产与生活的方式。在新零售行业下，商场内零售店铺里的产品种类变得更加琳琅满目。本文从“区域层级”、“小类层级”及“门店 skc 层级”三个方向，做出双重相关分析，并建立预测模型，从而能够给出精准的需求预测。

针对问题一：本文通过处理附件中提供的数据，首先对数据进行预处理，将售价作为销售特征，选取了库存量、售价及折扣作为影响因素。然后本文通过对影响因素与目标 skc 的销售做了相关性分析与偏相关性分析。首先对数据进行相关性系数检验和正态分布检验，根据检验结果，首先对数据进行斯皮尔曼相关系数计算，得出库存量、折扣和售价的 ρ 均小于 0.05，商品销售量与库存量存在较强的正向相关，商品销售量与折扣存在较弱的正相关，商品销售量与售价存在一定的负向相关；与此同时，对数据进行偏相关性分析，在剔除了其他变量的影响后，得出商品销售量与库存量存在较强的正向相关，商品销售量与折扣存在较弱的正相关，商品销售量与售价存在一定的负向相关。

针对问题二：结合问题一中影响目标 skc 的销售量因素的分析结果，本文建立基于粒子群优化的 BP 神经网络预测模型，输入数据为目标小类的售价、库存、折扣，输出数据为目标小类的销售量。通过 MATLAB 对模型求解，得到模型预测给定区域内目标小类 3 个月中每月销售量的结果，进而通过计算得到对于 10 个小类的预测值平均 MPAE 值为 0.22257。同时，本文采用了基于蚁群算法优化神经网络进行预测得到的平均 MPAE 值为 1.03517。得出粒子群优化的神经网络模型效果更优，可以满足一定的实际应用。

针对问题三：为满足企业更加精准的营销需求，结合上述问题，本文通过粒子群优化的神经网络预测目标小类内所有 skc 在 2019 年 10 月 1 日后 12 周内每周的周销量。经初步数据统计，共有 11725 个 skc，由于计算 MAPE 要求，故剔除不合格数据，最终剩余 264 个 skc 销售量的数据属于合格数据。因此，本文利用 MATLAB 对符合要求的数据进行模型预测，得到的总体 MAPE 平均值为 0.5491。经过对数据预测值和真实值的图像对比，可知此方案预测结果与真实值之间存在较大误差，因此，本文对此模型进行改进，根据历史数据得出 skc 占该小类的销售量比例，算出每一个 skc 的销售量，从而得到预测结果的平均 MAPE 值为 0.6002。

针对问题四：本文通过对影响因素做出双重相关分析及建立 BP 神经网络模型，对结果进行预测。根据结果的分析，我们向企业写一份推荐信，推荐我们的预测结果和方法，阐述我们的方案，并提供后续的优化方向作为参考意见。

关键词：相关性分析；偏相关性分析；BP 神经网络模型；粒子群优化算法；MATLAB

目录

一、问题的重述	1
1.1 问题的背景及研究意义	1
1.2 问题的相关信息	1
1.3 问题的重述	1
二、问题的分析	1
2.1 问题一的分析	1
2.2 问题二的分析	2
2.3 问题三的分析	2
2.4 问题四的分析	2
三、模型的假设	2
四、符号说明	3
五、模型的建立与求解	3
5.1 问题一模型的建立与求解	3
5.1.1 数据的预处理	3
5.1.2 相关系数检验	3
5.1.3 偏相关分析	6
5.1.4 双重相关分析确定影响因素对销售量影响	8
5.2 问题二模型的建立与求解	8
5.2.1 模型的建立	8
5.2.2 模型的求解	11
5.3 问题三模型的建立与求解	12
5.3.1 模型的建立与求解	12
5.3.2 模型的改进	13
5.3.3 改进后的结果	13
5.4 致企业领导的一封信	14
六、模型的评价	15
6.1 模型的优点	15
6.2 模型的缺点	15
参考文献	15
附录	17

一、问题的重述

1.1 问题的背景及研究意义

随着消费市场的不断发展，市场上的消费模式已经逐步转变为“以客为主”。在新零售行业，人们的需求也把注意力放在了“个性化、时尚、美观”等方面。因此，在这类特殊需求的推动下，新零售企业的生产模式逐步向多品种、小批量迈进，这让商场内零售店铺里的饰品和玩具等种类变得更加琳琅满目，同时也给零售行业的库存管理增加了很大的难度。这就使得根据历史销售数据，以不同层级给出精准的需求预测，成为了当前大多数新零售企业需要重点关注并思考的问题。

1.2 问题的相关信息

在附件中给出了某零售企业生产的 N 款产品，在华东区内的相关数据。其中附件一给出了销售流水数据即产品与实际花费的对应数据；附件二给出了产品的信息表即产品的编码、所处年份、小类编码和标签价；附件三给出了产品与区域库存数据；附件四给了节假日的信息表；附件五是上述附件中字段的说明。

1.3 问题的重述

现需要对一以下问题进行分析与预测：

问题 1：根据附件中的信息，分析节假日内各种相关因素对目标 skc 的销售量的影响，同时也可以考虑进其他因素的影响；

问题 2：根据问题一分析结果，预测未来给定区域内目标小类在 2019 年 10 月 1 日后 3 个月中每个月的销售量，并给出每个月预测的 MAPE；

问题 3：为了满足企业更加精准的营销需求，需要建立相关数学模型，并且考虑到小类预测结果的同时，预测目标小类内所有 skc 在 2019 年 10 月 1 日后 12 周内每周的周销量，并给出每周预测值的 MAPE。

问题 4：根据预测结果和方法，从而给出企业一份推荐信，并且要说明方案的合理性以及优化方向。

二、问题的分析

2.1 问题一的分析

本文通过附件中数据，对影响目标 skc 的销售量的因素数据进行分析，首先对数据进行预处理，将售价作为销售特征，选取了库存量、售价（销售特征）及折扣作为影响因素。然后本文将通过对影响因素与目标 skc 的销售做相关性分析与偏相关性分析。首先对数据进行相关性分析：需要判断影响因素与目标 skc 的销售量之间的线性或非线性关系，若影响因素与目标 skc 的销售量之间存在非线性关系，则选择建立斯皮尔曼相关系数模型；若影响因素与目标 skc 的销售量之间存在线性关系，则建立皮尔逊相关系数模型。随后，再对数据进行正态性检验，若两组变量之间服从正态分布，则通过效率较

高的皮尔逊相关系数法求得其相关系数，反正，若两组变量之间不服从正态分布，则通过斯皮尔曼相关系数法求得其相关系数。与此同时，对数据进行偏相关性分析：剔除其他变量的影响，分别得出商品销售量与库存量、商品销售量与折扣、商品销售量与售价（销售特征）的相关性，并对结果进行分析。

2.2 问题二的分析

本文结合问题一的分析结果，根据影响因素与目标小类在 2019 年 10 月 1 日后 3 个月中每个月的销售量的关系，建立基于粒子群优化的 BP 神经网络预测模型，首先采用改进粒子群优化算法反复优化 BP 神经网络模型的权值参数组合，再用 BP 算法对得到的网络参数进一步精确优化，最后用得到精确的最优参数组合进行预测，将库存、折扣和售价作为输入信号，通过在输入层输入信号，经过隐层点作用于输出层，利用非线性变换，最终输出信号，同时，不断地调整网络权值和阈值使误差函数 E 达到极小。最后通过 MATLAB 对模型求解，能够得到对于 10 个小类的预测值平均 MPAE 值。同时本文预采用基于蚁群算法优化神经网络进行预测，并将两个算法的预测结果进行比较，得出更加优化的神经网络模型，能够满足一定的实际应用。

2.3 问题三的分析

为满足企业更加精准的营销需求，结合上述问题的分析结果，建立基于粒子群优化的 BP 神经网络模型，预测目标小类内所有 skc 在 2019 年 10 月 1 日后 12 周内每周的周销量。首先对数据统计，得到总体数据，根据计算 MAPE 具有 $y_i > 0$ 的要求，故再剔除不合格数据，最后，本文利用 MATLAB 对符合要求的数据进行模型预测，并结合预测结果与真实数据，计算得出每周预测值的 MAPE 平均值。

2.4 问题四的分析

根据对基于粒子群优化的 BP 神经网络预测模型结果的分析，我们向企业写一份推荐信，推荐我们的预测结果和方法，阐述我们的方案，并提供后续的优化方向作为参考意见。

三、模型的假设

1. 假设本文所提供的数据具有真实性、可靠性、普遍性；
2. BP 神经网络模型的输入量只有库存、售价、折扣的数据，其他因素不作为输入量，即假设其他因素不影响目标 skc 的销售量；
3. 在现实生活中，一件商品不可能每一天都卖出，因此我们假设销售量缺失日期数据且当天有库存时，当天无货物卖出，即销售量为 0；

4. 假设产品的销售特征只由 skc 的售价表示. 即售价越低产品销售的越好; 反之, 售价越高, 产品销售的越差;
5. 假设问题三中预测的 skc 均需要满足的条件.

四、符号说明

符号	符号说明
d	表示两组数据的等级之差
r_{sp}	斯皮尔曼相关系数
w_{ij}	连接权值
X_{max}	样本最大值
X_{min}	样本最小值
H_0	原假设
H	备择假设
$x = (x_1, x_2, \dots, x_n)$	BP 神经网络模型的输入向量
$d = (d_1, d_2, \dots, d_m)$	BP 神经网络模型的期望输出向量
$hi = (hi_1, hi_2, \dots, hi_p)$	BP 神经网络模型的隐含层输入向量
$ho = (ho_1, ho_2, \dots, ho_p)$	BP 神经网络模型的隐含层输出向量
$yi = (yi_1, yi_2, \dots, yi_m)$	BP 神经网络模型的输出层输入向量
$yo = (yo_1, yo_2, \dots, yo_m)$	BP 神经网络模型的输出层输出向量

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 数据的预处理

根据题目要求, 本文选取了销售时间处于 2018 年 7 月 1 日至 2018 年 10 月 1 日内且累计销售额排名前 50 的 skc, 对于销售额的计算, 本文通过表格 sale_info.csv 和 prod_info.csv 的对比和实际情况推测出 sale_info.csv 的 real_cost 字段为当天的销售额. 在实际生活中, 因为商品不可能每一天都有销售额, 所以对于缺失日期的销售额, 在有库存的情况下, 本文假设当天无商品卖出, 即销售额为 0. 本文假设表格 prod_info.csv 的 tag_price 字段为单个商品的原价, 表格 sale_info.csv 的 real_cost/s 为单个商品的售价. 同时剔除了 skc 无库存数据的情况.

5.1.2 相关系数检验

在解决相关性问题时, 常用的是皮尔逊相关系数和斯皮尔曼相关系数. 其中, 皮尔逊相关系数是研究变量之间线性相关程度的量, 运用皮尔逊相关系数系数, 如果满足数

据连续、正态分布、线性关系的条件，则运行效率高且效果较好。为此，首先需要对数据进行预处理，判断数是否成线性关系，之后检验数据是否符合正态分布，进而确定采用的相关系数，最终得出结论。数据流程图如下：

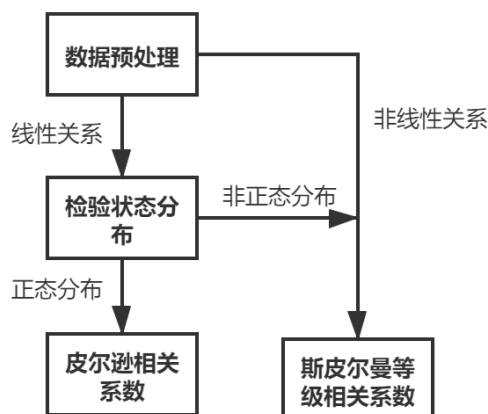


图 1 相关性检验流程图

(1) 描述统计

为了检验每个指标下的数据之间是否存在线性关系，因此画出了各影响因素与销售量之间的散点图。

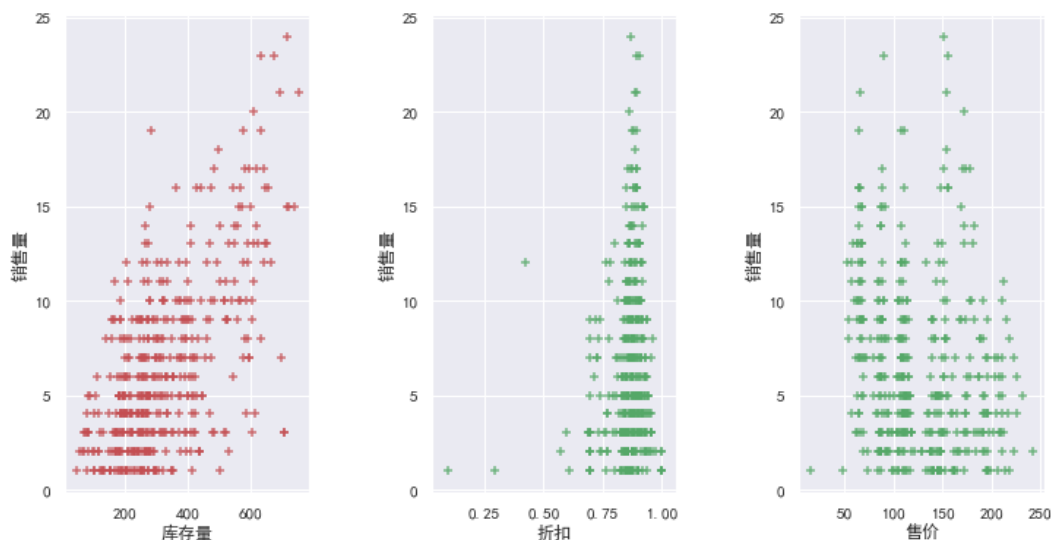


图 2 各影响因素与销售量之间的散点图

根据上图，我们发现仅仅与 skc 的销售量和库存呈较明显的线性关系，而与其他影响因素不存在明显的线性关系。

(2) 正态分布检验

采用皮尔逊相关系数，需要检验数据是否是正态分布，且我们搜集的数据数量为 467 条，属于大样本。因此采用了雅克-贝拉检验 (Jarque-Beratest) 进行正太分布检验。

下面进行假设检验：

- 原假设 H_0 : 各指标都服从正态分布；
- 备择假设 H: 各指标都不服从正态分布。

接着，对数据进行雅克-贝拉检验 (JarqueBeratest)，在 95% 的置信水平 (即显著性水平 $\alpha=0.05$) 下，结果如表所示：

影响因素	销售量	售价	折扣	库存量
h	1	1	1	1
p	0.00010	0.00010	0.00010	0.00010

表 1 雅克-贝拉检验结果表

$h = 0$ 表示接受原假设， $h = 1$ 表示拒绝原假设，由计算结果可以得：均拒绝原假设，表明数据不符合状态分布。

通过对数据的预处理，销售量与部分影响因素不存在明显的线性关系，使用皮尔逊相关系数模型效果不佳，且有部分指标数据不服从正态分布，这意味着即便计算出皮尔逊相关系数，也无法进行假设检验。综上所述，本文选择斯皮尔曼相关系数来衡量数据的相关性。

(3) 斯皮尔曼相关系数

斯皮尔曼相关性系数，通常也叫斯皮尔曼秩相关系数，它是用来描述两个变量之间是县管辖的，且该检验不需要进行正态分布检验且可以衡量非线性关系，而是需要确定变量在每个点上的等级即可，这样使其具有较好的性质，便于更好的分析。若有两组观测值且都不重复，则其斯皮尔曼等级相关系数的公式为：

$$r_{sp} = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (1)$$

其中,d 表示两组数据的等级之差，n 为样本量。

(4) 斯皮尔曼等级相关系数的检验

因为样本是随机性的，虽然小样本数据之间有存在相关时，但是总体数据之间不一定存在相关。因此需要再进行假设检验。在此，设定原假设 H_0 ：研究的总体之间无相关（即 $\rho = 0$ ），备责假设即为 H 研究的总体之间有相关。

检验的样本估计量为样本的相关系数 r_{sp} ，在小样本的情况下通常临界值的 r 可直接查表，但在大样本的情况下可以通过变换。

$$t = r_{sp} \sqrt{\frac{n-2}{1-r_{sp}^2}} \quad (2)$$

服从 $t(n-2)$ 的 t 分布，采用 t 检验。得到影响因素与价格的斯皮尔曼检验结果如下表：

影响因素	库存量	折扣	售价
r_{sp}	0.535**	0.113*	-0.242**
p	0.000	0.015	0.000

表 2 斯皮尔曼检验结果表

** 在 0.01 级别（双尾），相关性显著。

** 在 0.05 级别（双尾），相关性显著

由以上斯皮尔曼检验的结果可以看出，对于的 $p < 0.05$ 的影响因素，商品销售量与库存量存在较强的正向相关，商品销售量与折扣存在较弱的正相关，商品销售量与售价存在一定的负向相关。

5.1.3 偏相关分析

简单的相关系数不能完全真实的反映两个变量之间的相关性，因为往往有第三变量的影响或作用。为了进一步准确的确定影响因素，我们团队利用控制变量的思想，分析两变量之间的相关性，即剔除其他的变量的影响，采用偏相关分析对分析销售量与影响因素的关系。

- 一阶偏相关系数。

在 3 个变量中，任意两个变量的偏相关系数是在排除其余一个变量影响后计算得到的，称为一阶偏相关系数，公式为：

$$r_{ij \cdot h} = \frac{r_{ij} - r_{ih}r_{jh}}{\sqrt{(1 - r_{ih}^2)(1 - r_{jh}^2)}} \quad (3)$$

式中， r_{ij} 是变量 x_i 与 x_j 的简单相关系数， r_{ih} 是变量 x_i 与 x_h 的简单相关系数， r_{jh} 与 x_h 是变量 x_j 与 x_h 的简单相关系数。

- 二阶偏相关系数。

在 4 个变量中，任意两个变量的偏相关系数是在排除其他两个量影响后计算得到的，称为二阶偏相关系数，公式为：

$$r_{ij \cdot hm} = \frac{r_{ijh} - r_{im \cdot h}r_{jm \cdot h}}{\sqrt{(1 - r_{im \cdot h}^2)(1 - r_{jm \cdot h}^2)}} \quad (4)$$

式中， i, j, h, m 分别取 1、2、3、4 的组合。显然，二阶偏相关系数是由一阶偏相关系数求得。

• 高价偏相关系数.

一般地, 假设有 $k(k > 2)$ 个变量 x_1, x_2, \dots, x_k , 则任意两个变量 x_i 和 x_j 的 $g(g \leq k - 2)$ 阶样本偏相关系数公式为:

$$r_{ij \cdot l_1 l_2 \dots l_g} = \frac{r_{ij \cdot l_1 l_2 \dots l_{g-1}} - r_{il_g \cdot l_1 l_2 \dots l_{g-1}} r_{jl_g \cdot l_1 l_2 \dots l_{g-1}}}{\sqrt{\left(1 - r_{il_g \cdot l_1 l_2 \dots l_{g-1}}^2\right) \left(1 - r_{jl_g \cdot l_1 l_2 \dots l_{g-1}}^2\right)}} \quad (5)$$

式中右边均为 $g - 1$ 阶的偏相关系数.

(1) 偏相关系数的假设检验

偏相关系数检验的零假设为: 总体中两个变量间的偏相关系数为 0. 使用 t 检验方法, 公式如下:

$$t = \frac{\sqrt{n - k - 2} \cdot r}{\sqrt{1 - r^2}} \quad (6)$$

式中, r 是相应的偏相关系数, n 是样本观测数, k 是可控制变量的数目, $n - k - 2$ 是自由度. 当 $t > t_{0.005}(n - k - 2)$ 或 $p < 0.05$ 时, 拒绝原假设.

(2) 结果

利用上述思想, 可以得出 skc 的销售量与售价、库存、折扣的偏相关性.

控制变量			售价
库存量 & 折扣	销售量	相关性	-0.103
		显著性 (双尾)	0.027

表 3 skc 的销售量与售价的偏相关性

控制变量			库存量
折扣 & 售价	销售量	相关性	0.589
		显著性 (双尾)	0.000

表 4 skc 的销售量与库存量的偏相关性

控制变量			折扣
库存量 & 售价	销售量	相关性	0.132
		显著性 (双尾)	0.0045

表 5 skc 的销售量与折扣的偏相关性

从上表可以看出, 所有的 p 值均小于 0.05, 拒绝原假设, 说明商品销售量与售价, 库存和折扣均有关系. 商品销售量与库存量存在较强的正向相关, 商品销售量与折扣存在较弱的正相关, 商品销售量与售价存在一定的负向相关.

5.1.4 双重相关分析确定影响因素对销售量影响

通过相关性分析与偏相关分析的双重分析，我们团队对两种结果进行对比，审查，得出商品销售量与库存量存在较强的正向相关，商品销售量与折扣存在较弱的正相关，商品销售量与售价存在一定的负向相关。

5.2 问题二模型的建立与求解

5.2.1 模型的建立

根据分析各种相关因素对目标 skc 的销售量的影响，本文建立基于粒子群优化的 BP 神经网络预测模型。首先采用改进粒子群优化算法反复优化 BP 神经网络模型的权值参数组合，其次再用 BP 算法对得到的网络参数进一步精确优化，最后用得到精确的最优参数组合进行预测。本文需要预测的数据为：（1）预测给定区域内目标小类在 2019 年 10 月 1 日后 3 个月中每个月的销售量；（2）预测目标小类内所有 skc 在 2019 年 10 月 1 日后 12 周内每周的周销量。

(1) BP 神经网络模型

BP 神经网络基本结构如图所示：

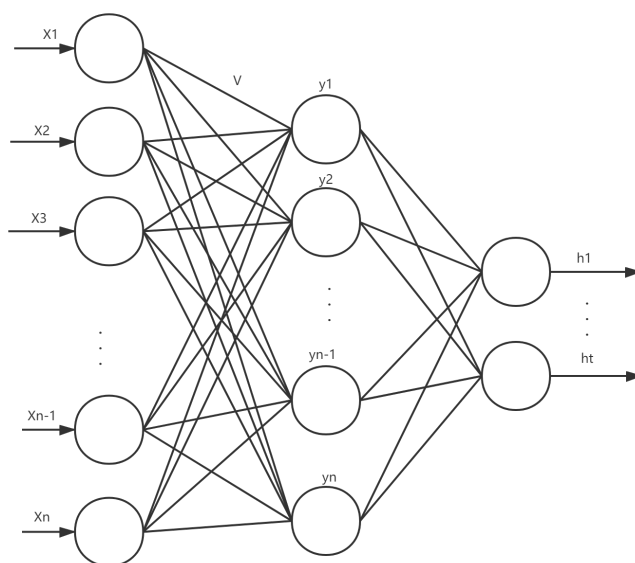


图 3 BP 神经网络基本结构

在建立 BP 神经网络模型之前，为了提升 BP 神经网络的训练速度和估计准确程度，要对输入层的数据进行标准化处理，剔除各因素不同量纲的影响后，建立 BP 神经网络模型。BP 神经网络的预测流程如图所示：

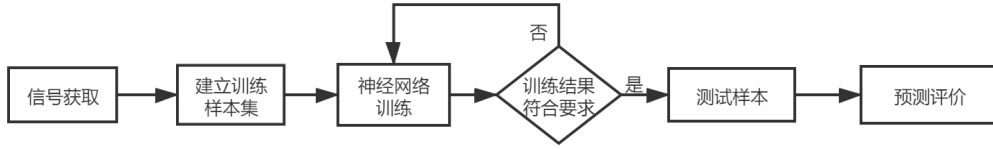


图 4 BP 神经网络的预测流程

在输入层输入信号 x_i , 通过隐层点作用于输出层, 经过非线性变换, 最终输出信号 y_k , 神经网络中的每组样本应含有输入向量 x 和期望输出值 t , 神经网络输出信号 y 与期望输出值 d 之间的误差 E , 通过不断地调整网络权值 (输入层信号和隐层点的关联强度 w_{ij} 和隐层点与输出信号之间的关联强度 T_{jk} 和阈值使误差函数 E 达到极小. 第 i 层第 j 个神经元的输入用 $in_j^{(i)}$, 第 i 层第 j 个神经元的输出用 $out_j^{(i)}$.

网络各层的输入输出的关系具体可以描述为:

输入层:

$$in_j^{(1)} = out_j^{(1)} = x \quad j = 1, 2, \dots, n \quad (7)$$

隐含层:

神经元的输入:

$$hi_h(k) = \sum_{i=1}^n w_{ih} x_i(k) - b_h (h = 1 \dots p) \quad (8)$$

神经元的输出:

$$ho_h(k) = f(hi_h(k)) (h = 1 \dots p) \quad (9)$$

输出层:

神经元的输入:

$$yi_o(k) = \sum_{h=1}^n w_{ho} ho_h(k) - b_o (o = 1 \dots m) \quad (10)$$

神经元的输出:

$$yo_o(k) = f(yi_o(k)) (o = 1 \dots p) \quad (11)$$

这里 $f(x)$ 为 sigmoid 型传递函数 $f(x) = \frac{1}{1+e^{-x}}$.

设神经网络输入层到隐含层之间的连接权值为 w_{ij} , 设输出层神经元 k 的最终输出信号为 y_k , 设数据的实际结果为 d_k , 设隐含层神经元 j 的输入和为 net_j , 设输入的神经元为 x_i , 那么 w_{ij} 的调整表达式为:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \quad (12)$$

$$\Delta w_{ij} = -\vartheta \frac{\partial E}{\partial w_{ij}} = \vartheta \sum_{i=1}^m (d_k - y_k) f_1'(net_k) v_{jk} f_2'(net_j) x_i$$

设神经网络隐含层与输出层之间的权值为, 设其学习效率为, 设输出层神经元 k 的最终输出信号为, 设数据的实际结果为, 设隐含层神经元 j 的输入和为, 则 BP 神经网络隐含层到输出层之间的权值调整的表达式为:

$$v_{jk}(t+1) = v_{jk}(t) + \Delta v_{jk} \quad (13)$$

$$\Delta v_{jk} = -\vartheta \frac{\partial E}{\partial v_{jk}} = \vartheta (d_k - y_k) f_1'(net_k) y_k$$

针对以 D 为维度的优化问题, 本文采用平方和来测度神经网络的误差, 其表达式为:

$$E_D = \frac{1}{2} \sum_{d=1}^D \sum_{i=1}^m (d_k - y_k)^2 \quad (14)$$

(2) 改进的粒子群神经网络

由于神经网络具有易陷入局部最优解、预测偏差较大的缺点, 且此问题提供的数据庞大, 因此本文用搜索能力强、速度快、效率高的粒子群算法对 BP 神经网络模型进行训练.

在进行 BP 神经网络模型训练前, 先将样本数据进行归一化处理:

$$X_t = \frac{X_t - X_{\min}}{X_{\max} - X_{\min}} \quad (t = 1, 2, \dots, 24) \quad (15)$$

式子中, x 为 t 时刻样本数值; X_{\min} 为样本最小值; X_{\max} 为样本最大值.

基于粒子群优化的 BP 神经网络模型训练的主要步骤如下:

- Setp1: 对单层前馈神经网络结构的权值和阈值编码, 从而能够使其与改进粒子群优化算法中的个体相对应;
- Setp2: 针对粒子群优化问题设置 BP 神经网络, 确定参数, 从而生成优化后的 BP 神经网络模型;
- Setp3: 神经网络模型的权值和阈值的参数取自粒子群中维度信息解码;
- Setp4: 采用精英学习策略对寄生群粒子的更新速度进行更新, 采用压缩因子粒子群优化算法的方法对宿主群粒子的速度进行更新;
- Setp5: 重新计算粒子的适应度, 将其与现有个体历史最优值进行比较;
- Setp6: 当迭代的次数满足条件时, 发生寄生行为, 宿主群粒子通过高频变异获得免疫能力;
- Setp7: 判断条件是否满足, 若满足, 则输出结果, 否则转为 step2.

5.2.2 模型的求解

• 模型结果及模型的检验

依据题目条件，我们使用基于粒子群优化的神经网络模型对历史销售时间处于 2019 年 6 月 1 日至 2019 年 10 月 1 日内且累计销售额排名前 10 的小类进行预测，输入数据为目标小类的售价、库存、折扣，输出数据为目标小类的销售量。其预测结果和实际值的比较如下表所示：

编号		2019-10	2019-11	2019-12	MAPE
27050401	真实值	15136	16976	17899	0.1369869643144925
	预测值	16578	16575	16655	
27217089	真实值	4756	4913	3312	0.756066183256427
	预测值	9091	7311	5160	
27164944	真实值	7002	7623	7042	0.17488053671369683
	预测值	6678	13054	11426	
27196225	真实值	3928	4030	3950	0.5969028683899551
	预测值	3712	7840	7135	
27060804	真实值	7143	7723	8387	0.3287128438647797
	预测值	8018	8238	8019	
27112849	真实值	5953	6012	4775	0.2077123398698504
	预测值	11442	8638	4397	
27092025	真实值	2721	2439	1445	1.3215295064547106
	预测值	2432	2618	2739	
27206656	真实值	1341	1536	750	0.9773891947299734
	预测值	2299	3766	640	
27071209	真实值	2200	2100	1288	0.7907191436618899
	预测值	4830	3623	1310	
27102436	真实值	1148	1061	652	1.386381233367651
	预测值	2030	407	971	

表 6 预测结果和实际值的比较

由上述预测结果可得：10 个小类的预测值平均 MP AE 值为 0.2226。另外，本文还采用了基于蚁群算法优化神经网络进行预测，得到的平均 MP AE 值为 1.0352。将两种算法对于 BP 神经网络模型的优化成果进行比较，从而得出：粒子群优化的神经网络模型效果更优，可以满足一定的实际应用。

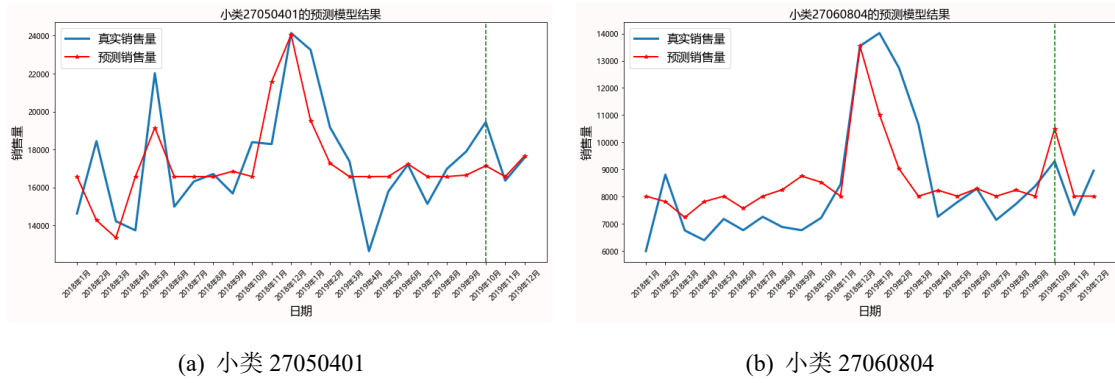


图 5 预测模型结果

从上面两图可以看成，基于粒子群优化的神经网络模型拟合效果较好，预测的准确率较高，进一步验证了模型的实际可用性。

5.3 问题三模型的建立与求解

5.3.1 模型的建立与求解

为了满足企业更加精准的营销需求，在考虑小类预测结果的同时，本文使用粒子群优化的神经网络预测目标小类内所有 skc 在 2019 年 10 月 1 日后 12 周内每周的周销量。经数据统计， skc 的总数为 11725。由于部分 skc 数据缺少严重，且若要计算 $MAPE$ ，必须要求 $y_i > 0$ 故本文剔除不合格数据，最终剩余 264 个 skc 销售量的数据属于合格数据，因此为方便计算，本文只对符合要求的 264 个数据进行模型预测。根据预测结果，经过计算可得：总体 $MAPE$ 平均值为 0.5491，预测效果良好。最后，本文问题三的预测结果详见附件。

另外，本文随机选取了两种满足条件的 skc 预测值与真实值的图像：

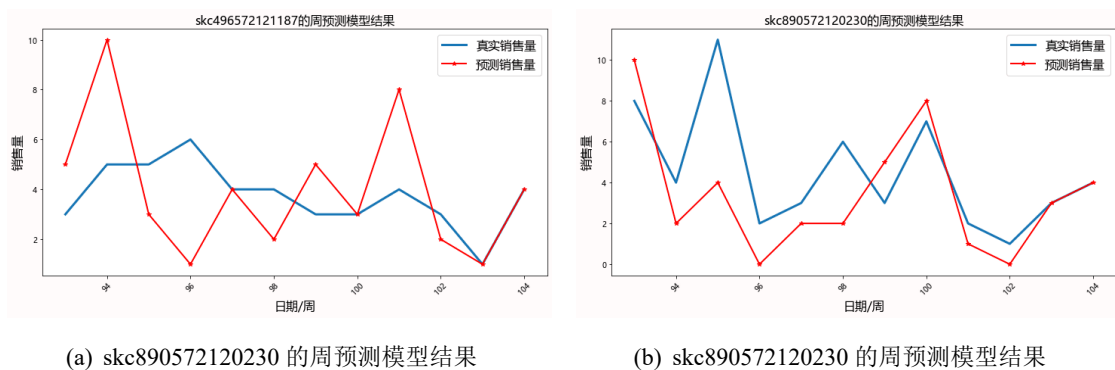


图 6 周预测模型结果

通过上述两张图片可以看出模型的预测值与真实值在某些时间点的误差较大，说明此方法计算的预测结果一般，下面本文给出改进的模型预测。

5.3.2 模型的改进

本文对目标小类的所有 skc 进行粒子群优化的神经网络进行预测, 时间复杂度过高, 难以实现实际运用, 因此, 我们团队提出以下解决方案:

首先, 使用粒子群优化的神经网络预测目标小类的 2019 年 10 月 1 日后 12 周内每周的周销量. 然后, 我们根据历史数据得出 skc 占该小类的销售量比例, 算出每一个 skc 的销售量. 具体算法如下所示:

$$\begin{bmatrix} \hat{y}_{skc_1} \\ \hat{y}_{skc_2} \\ \vdots \\ \hat{y}_{skc_n} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \hat{Y}_{\text{小类}} \quad (16)$$

w_i 为 y_{skc_i} 的销售量占其所属类别销售量的比例, n 是该类中 skc 的数量.

5.3.3 改进后的结果

依据上述思想, 我们首先得对目标小类的 2019 年 10 月 1 日后 12 周内每周的周销量预测结果.

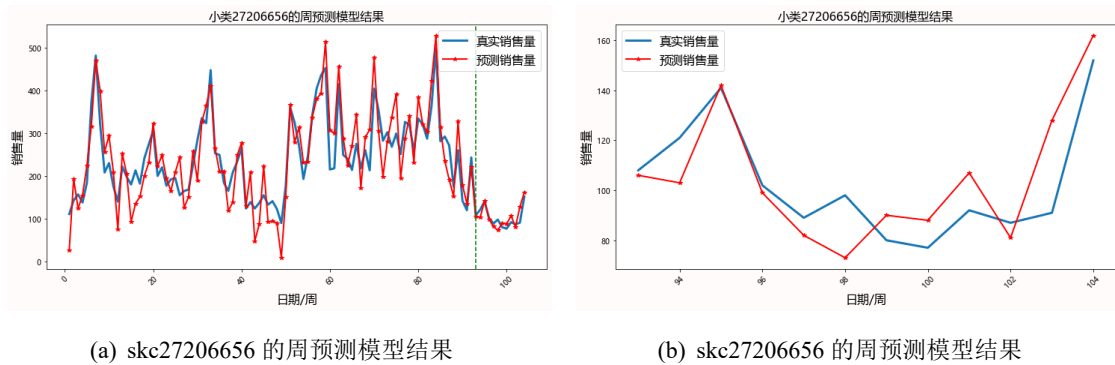
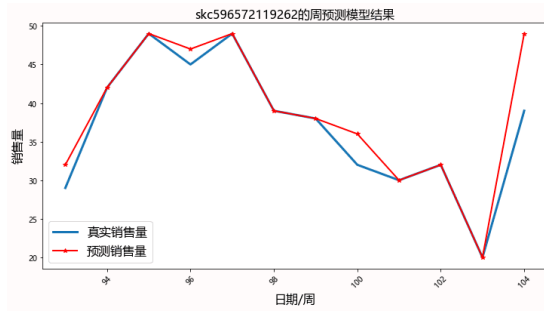


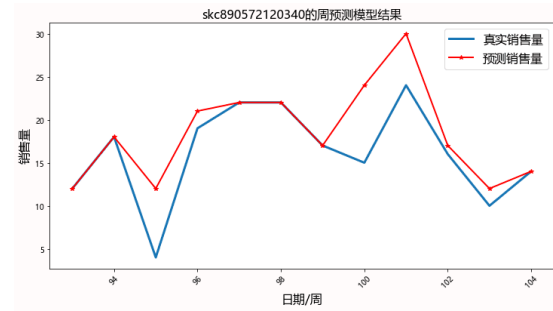
图 7 周预测模型结果

我们从图中看出整体拟合程度程度较优, 尤其在 2019 年 10 月 1 日后 12 周的预测效果比较显著.

我们依据上述问题三的公式可以得到每一个小类内的所有 skc 的销售量, 由于数据结果过大, 本文以附件形式提交.



(a) skc596572119262 的周预测模型结果



(b) skc890572120340 的周预测模型结果

图 8 周预测模型结果

预测结果的平均 MAPE 值为 0.6002，且上图可以看成预测结果较优，虽然平均 MAPE 值比第一种模型稍微低一点，但在运行时间上大大提升，在要求精度不是很高的场合下，完全可以满足对实际的应用，证明了本次改进的成功性。

5.4 致企业领导的一封信

尊敬的领导：

您好，感谢您对我们团队的信任。通过分析总结我们团队建立的模型预测结果，向领导介绍我们团队的方案，希望我们的成果能对新零售行业的发展有所帮助。

（一）预测方法和预测结果

1. 预测方法

我们根据影响因素与目标小类的销售量的关系，建立基于粒子群优化的 BP 神经网络预测模型，首先采用改进粒子群优化算法反复优化 BP 神经网络模型的权值参数组合，再用 BP 算法对得到的网络参数进一步精确优化，最后用得到精确的最优参数组合进行预测将库存、折扣和售价作为输入信号，通过在输入层输入信号，经过隐层点作用于输出层，利用非线性变换，最终输出信号。同时，不断地调整网络权值和阈值使误差函数 E 达到极小，利用 MATLAB 求解模型预测结果。根据分析各种相关因素对目标 skc 的销售量的影响，分别对月销售量和周销售量进行预测：（1）预测给定区域内目标小类在 2019 年 10 月 1 日后 3 个月中每个月的销售量；（2）预测目标小类内所有 skc 在 2019 年 10 月 1 日后 12 周内每周的周销量。

2. 预测结果

由于预测结果数据过多，因此对于问题二和问题三的预测值详见附件，如有不便，还请领导谅解。对于模型的预测结果，我们团队将其与真实值进行了比较，在改进了方案后，可以看出预测值与真实值的误差缩小了。

（二）方案的合理性

BP 神经网络突出优点就是具有很强的非线性映射能力和柔性的网络结构，它能够进行数据压缩、数据分类，且能够用一个待定的输出向量将它与输入向量联系起来。而基于粒子群优化的神经网络预测模型针对原 BP 神经网络模型的易陷入局部极小、收敛速度慢和容易引起震荡的固有缺点进行改进。由于本文要解决的问题提供的数据庞大，因此我们团队采用改进的 BP 神经网络模型建模，其优势在于它可将输入向量所定义的合适方式进行分类，也能够减少输出向量维数，以便于传输和存储，同时还能改善收敛速度等。我们团队还采用了基于蚁群算法优化神经网络进行预测，并将两种算法对于 BP 神经网络模型的优化成果进行比较，从而得出：粒子群优化的神经网络模型效果更优，可以满足一定的实际应用。

（三）后续的优化方向

1. 加强阿里巴巴、京东等大型零售产业的线上线下融合；
2. 通过监管后台顾客信息及购买数据，为不同人提供不同产品，做到高度定制化；
3. 加强产品与供应链的监管流程；
4. 增强购买方式的可行性、普遍性与便利性。

以上是我们团队的一些想法和建议，如果有不妥之处，还请领导多多谅解。

六、模型的评价

6.1 模型的优点

1. 采用相关性分析与偏相关分析的双重相关性分析销售量的影响因素，使得分析结果更加准确、客观与合理；
2. 采用相关性分析与偏相关分析的双重相关性分析销售量的影响因素，使得分析结果更加准确、客观与合理；
3. 基于粒子群优化的神经网络模型可以较好的预测类的月销售，可以满足实际工程运用；
4. 为了满足企业的“精准需求预测”，我们采用了两种方法对目标小类的所有 skc 进行预测，第一种方法准确度高但时间复杂度过大，第二种方法与第一种方法的准确度相差不大的情况下，时间复杂度大大的减小，可应用于实践工程。

6.2 模型的缺点

1. 数据部分缺失, 需要收集更多、更精确的数据；
2. 销售量的考虑的影响因素过少，可以考虑季节，宣传程度等摘要影响因素；
3. 双重相关分析只定性分析影响因素与销售量的关系, 而不能定量分析；
4. 粒子群优化算法缺乏速度的动态调节，容易陷入局部最优，导致收敛精度低和不易收敛, 可以进行适当的调参或者融合其他优化算法进行改进；
5. 确定目标小类与目标小类内 skc 之间的关系过于简单，可以尝试引进相关机器学习算法进行改进。

参考文献

- [1] 程军, 李荣钧. 基于粒子群优化的神经网络预测模型 [J]. 数学的实践与认识, 45(03):176-180, 2015.
- [2] 段立峰. 粒子群算法优化神经网络的旅游热门景点预测模型 [J]. 微型电脑应用, 36(03):154-156, 2020.
- [3] 宋颂, 董宝力. 基于改进 BP 神经网络的饮料销售预测模型研究 [J]. 成组技术与生产现代化, 35(03):9-15, 2018.
- [4] 李丙春, 孙蕾. 基于粒子群优化的神经网络预测模型 [J]. 新疆大学学报: 自然科学版 (4):481-485, 2007.
- [5] 杨娟丽, 徐梅, 王福林, 等. 基于 BP 神经网络的时间序列预测问题研究 [J]. 数学的实践与认识, (4).158-164, 2013.
- [6] 曾晓青. BP 神经网络在建模中的参数优化问题研究 [J]. 气象, 39(3):333-339, 2013.
- [7] 王世耆. 偏相关系数的计算机算法 [J]. 农业气象, (4): 32-36, 1981.
- [8] 史峰, 王小川, 郁磊, 李洋, MATLAB 神经网络 30 个案例分析, 北京航空航天大学出版, 2010-4-1.

附 录

附录中有问题 1 数据处理与筛选的代码、问题 2 与问题 3 数据处理与筛选的代码、问题 3 权重系数数据处理与筛选的代码、JB 正太检验的代码、粒子群算法优化神经网络的代码以及遗传算法优化神经网络的代码。

附录 1: 问题 1 数据处理与筛选的代码

```
#!/usr/bin/env python
# coding: utf-8
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
get_ipython().run_line_magic('matplotlib', 'inline')

sale=pd.read_csv("data/附件1: sale_info.csv")
sale.head()

from datetime import datetime
# 时间转换函数
def cal(time):
    format = '%Y-%m-%d'
    time=str(time)[0:10]
    return datetime.strptime(time, format)

#获取符合题目要求的数据, 目标 skc 为销售时间处于 2018 年 7 月 1日至 2018 年 10 月 1
    日
skc_dict={}
for ind,row in sale.iterrows():
    if( cal(row["date_rcd"])<cal("2018-07-01") or cal(row["date_rcd"])>cal("
        2018-10-01")):
        continue
    if row["skc"] not in skc_dict:
        skc_dict[row["skc"]]=[[row["date_rcd"],row["s"],row["real_cost"]]]
    else:
        skc_dict[row["skc"]].append([row["date_rcd"],row["s"],row["real_cost"]])

#累计销售额排名前 50 的 skc
data_list=[]
for key,val in skc_dict.items():
    summ=0
    for it in val:
        summ+=it[2]
    data_list.append([str(key),summ])
```

```

data_list.sort(key=lambda x:x[1],reverse=True)

prod=pd.read_csv("data/附件2: prod_info.csv")
inv=pd.read_csv("data/附件3: inv_info.csv")
holiday=pd.read_csv("data/附件4: holiday_info.csv")

prod.head()

#二维字典的添加
def addtwodimdict(thedict, key_a, key_b, val):
    if key_a in thedict:
        thedict[key_a].update({key_b: val})
    else:
        thedict.update({key_a:{key_b: val}})

from collections import defaultdict
prod_dict={}
#获取skc的类信息与原价
for ind,row in prod.iterrows():
    prod_dict[row["skc"]]=[row["tiny_class_code"],row["tag_price"]]

inv.head()
#获取skc的库存信息
inv_dict={}
temp_dict=defaultdict(int)
for ind,row in inv.iterrows():
    addtwodimdict(inv_dict,row["skc"],row["date_rcd"],row["ie"])
    temp_dict[row["skc"]]=1

#累计销售额排名前 50 的 skc
item=set()
i=0
for it in data_list:
    if temp_dict[int(it[0])]==1:
        i+=1
        item.add(it[0])
    if i>=50:break

```

```

print(len(item))

dict=defaultdict(int)
for it in item:
    dict[str(it)]=1

print(inv.dtypes)
print(prod.dtypes)
print(sale.dtypes)

#2018 年国庆节，双十一，双十二和元旦这四个节假日销售记录
data=[]
for ind,row in sale.iterrows():
    if dict[str(row["skc"])] ==1:
        try:
            if( cal(row["date_rcd"])<=cal("2018-10-07") and cal(row["date_rcd"]>=
                cal("2018-10-01")):#国庆

                data.append([row["skc"],row["date_rcd"],row["s"],row["real_cost"],
                    prod_dict[row["skc"]][1],inv_dict[row["skc"]][row["date_rcd"]]])

            if( cal(row["date_rcd"])==cal("2018-11-11")):#双十一
                data.append([row["skc"],row["date_rcd"],row["s"],row["real_cost"],
                    prod_dict[row["skc"]][1],inv_dict[row["skc"]][row["date_rcd"]]])

            if( cal(row["date_rcd"])==cal("2018-12-12")):#双十二
                data.append([row["skc"],row["date_rcd"],row["s"],row["real_cost"],
                    prod_dict[row["skc"]][1],inv_dict[row["skc"]][row["date_rcd"]]])

            if( cal(row["date_rcd"])<=cal("2019-01-01") and cal(row["date_rcd"]>=
                cal("2018-12-30")):#元旦
                data.append([row["skc"],row["date_rcd"],row["s"],row["real_cost"],
                    prod_dict[row["skc"]][1],inv_dict[row["skc"]][row["date_rcd"]]])
        except KeyError:
            pass

name=["skc","date_rcd","s","real_cost","tag_price","ie"]
all=pd.DataFrame(columns=name,data=data)
all.to_csv("问题1.csv")

name=["skc","总销售额"]
all=pd.DataFrame(columns=name,data=data_list[:50])
all.to_csv("50.csv")

```

```

all=[]
for it in data:
    all.append([it[0],it[1],it[2],it[3],it[4],it[5],it[3]/it[2]/it[4],it[3]/it
                [2]])
name=["skc","date_rcd","s","real_cost","tag_price","ie","discount","price"]
all=pd.DataFrame(columns=name,data=all)
all.to_csv("问题1_all.csv")

#条形图
# 指定默认字体
plt.rcParams['font.sans-serif'] = ['SimHei']

# 解决保存图像是负号 '-' 显示为方块的问题
plt.rcParams['axes.unicode_minus'] = False
sns.set(font='SimHei') # 解决Seaborn中文显示问题

data = pd.read_csv("all.csv",encoding="GBK")
data.head()

#散点图绘制
plt.figure(figsize=(30, 10))
fig,axes=plt.subplots(1,3,figsize=(12,6))
plt.subplots_adjust(, wspace =1, hspace =1)
plt.subplots_adjust(wspace =0.5,hspace =0.5)
sns.regplot(x='库存量',y='销售量',data=data,
            color='r',marker='+',ax=axes[0], fit_reg=False)

sns.regplot(x='折扣',y='销售量',data=data,
            color='g',marker='+',ax=axes[1],fit_reg=False)

sns.regplot(x='售价',y='销售量',data=data,
            color='g',marker='+',ax=axes[2],fit_reg=False)

```

附录 2：问题 2 与问题 3 数据处理与筛选的代码

```

#!/usr/bin/env python
# coding: utf-8
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

```

```

get_ipython().run_line_magic('matplotlib', 'inline')

prod=pd.read_csv("data/附件2: prod_info.csv")
inv=pd.read_csv("data/附件3: inv_info.csv")
prod.head()
#二维字典的添加
def addtwodimdict(thedict, key_a, key_b, val):
    if key_a in thedict:
        thedict[key_a].update({key_b: val})
    else:
        thedict.update({key_a:{key_b: val}})

from collections import defaultdict
prod_dict={}

for ind,row in prod.iterrows():
    prod_dict[row["skc"]]=[row["tiny_class_code"],row["tag_price"]]

sale=pd.read_csv("data/附件1: sale_info.csv")
sale.head()

from datetime import datetime

def cal(time):
    format = '%Y-%m-%d'
    time=str(time)[0:10]
    return datetime.strptime(time, format)

class_dict={}
for ind,row in sale.iterrows():
    if( cal(row["date_rcd"])>cal("2019-10-01") or cal(row["date_rcd"])<cal("2019-06-01")):
        continue
    #print(row)
    try:
        if prod_dict[row["skc"]][0] not in class_dict:
            class_dict[prod_dict[row["skc"]][0]]=[[row["skc"],row["date_rcd"],row["s"],row["real_cost"]]]
        else:
            class_dict[prod_dict[row["skc"]][0]].append([row["skc"],row["date_rcd"],row["s"],row["real_cost"]])
    except KeyError:

```

```

        pass

data_list=[]
for key,val in class_dict.items():
    summ=0
    for it in val:
        summ+=it[3]
    data_list.append([key,summ])
data_list.sort(key=lambda x:x[1],reverse=True)
data_list[0:10]
inv=pd.read_csv("data/附件3: inv_info.csv")
inv.head()

inv_dict=defaultdict(int)
temp_dict=defaultdict(int)
for ind,row in inv.iterrows():
    inv_dict[str(row["skc"])+","+str(row["date_rcd"][0:7])]+=row["ie"]
    temp_dict[row["skc"]]=1

from datetime import datetime
#获取月时间
def call(time):
    format = '%Y-%m'
    return datetime.strptime(time, format)

class_dict={}
for ind,row in sale.iterrows():
    try:
        if prod_dict[row["skc"]][0] not in class_dict:
            class_dict[prod_dict[row["skc"]][0]]=[[row["skc"],row["date_rcd"],row["s"],row["real_cost"]]]
        else:
            class_dict[prod_dict[row["skc"]][0]].append([row["skc"],row["date_rcd"],row["s"],row["real_cost"]])
    except KeyError:
        pass

#历史销售时间处于 2019 年 6 月 1 日至 2019 年 10 月 1 日内的小类销售记录
data=[] for i in range(0,11)]
item=set()
i=0
for it in data_list:

```



```

i+=1
if i>=11:break
class_dict[it[0]].sort(key=lambda x:cal(x[1]))
s=0
price=0
pre=class_dict[it[0]][0][1][:7]
for row in class_dict[it[0]]:
    #print(row[1][:7])
    item.add(row[0])
    if cal1(row[1][:7])==cal1(pre):
        s+=row[2]
        price+=row[3]
    else:

        kucun=0
        tag=0
        for id in item:
            tag+=prod_dict[id][1]
            kucun+=inv_dict[str(id)+", "+str(pre)]
        tag/=len(item)

        data[i].append([it[0],pre,s,price,price/s,tag,kucun,price/s/tag])
        s=row[2]
        price=row[3]
        pre=row[1][:7]
        item=set()
kucun=0
tag=0
for id in item:
    tag+=prod_dict[id][1]
    kucun+=inv_dict[str(id)+", "+str(pre)]
tag/=len(item)
data[i].append([it[0],pre,s,price,price/s,tag,kucun,price/s/tag])
s=row[2]
price=row[3]
pre=row[1][:7]
item=set()

name=["classid","date_rcd","s","real_cost","price","tag_price","ie","discount"]
for i in range(1,11):
    all=pd.DataFrame(columns=name,data=data[i])
    all.to_csv("问题2/"+str(i)+"_new.csv")

```

```

name=["classid","date_rcd","s","real_cost","price","tag_price","ie","discount"]
all_list=[]
for i in range(1,11):
    for j in data[i]:
        all_list.append(j)
all=pd.DataFrame(columns=name,data=all_list)
all.to_csv("问题2/all.csv")

```

#判断时间是第几周

```

import datetime
def cal_week(y,m,d):
    y=int(y)
    m=int(m)
    d=int(d)
    be = int(datetime.datetime(2018, 12, 31).strftime("%W"))
    if y<2019:
        return int(datetime.datetime(y, m, d).strftime("%W"))
    return be+int(datetime.datetime(y, m, d).strftime("%W"))

```

#根据周获取时间

```

def week2date(n,w):
    be = int(datetime.datetime(2018, 12, 31).strftime("%W"))
    if n<=be:
        y=2018
    else:
        y=2019
        n-=be
    wk = str(y)+'-W'+str(n)+'-'+str(w)
    return str(datetime.datetime.strptime(wk, '%Y-W%W-%w'))[0:10]

```

```

inv_dict=defaultdict(int)
for ind,row in inv.iterrows():
    inv_dict[str(row["skc"])+","+str(row["date_rcd"])]+=row["ie"]

```

```

def tran(time):
    format = '%Y-%m-%d'
    time=str(time)[0:10]
    return datetime.datetime.strptime(time, format)
tran("2018-01-01")

```

#获取目标小类内所有 skc 的周数据

```

data=[] for i in range(0,11)
item=set()
i=0
for it in data_list:
    i+=1
    if i>=11:break
    class_dict[it[0]].sort(key=lambda x:tran(x[1]))
    item=set()
    for row in class_dict[it[0]]:
        item.add(row[0])

    for id in item:
        s=0
        price=0
        num=1

        for row in class_dict[it[0]]:
            if row[0]!=id:continue
            if cal_week(row[1][:4],row[1][5:7],row[1][8:10])==num:
                s+=row[2]
                price+=row[3]
            elif cal_week(row[1][:4],row[1][5:7],row[1][8:10])==num+1:
                kucun=0
                tag=0
                tag=prod_dict[id][1]
                for t in range(1,7+1):
                    x=t
                    if t==7:x=0
                    kucun+=inv_dict[str(id)+' ',''+str(week2date(num,x))]
                single=0.0
                if s==0:
                    single=0
                else:
                    single=price/s
                data[i].append([it[0],row[0],num,s,price,single,tag,kucun,single/tag
                    ])
                s=row[2]
                price=row[3]
                num+=1
            else:
                while(num<cal_week(row[1][:4],row[1][5:7],row[1][8:10])):
                    kucun=0
                    tag=0
                    tag=prod_dict[id][1]

```

```

        for t in range(1,7+1):
            x=t
            if t==7:x=0
            kucun+=inv_dict[str(id)+'_'+str(week2date(num,x))]
        single=0.0
        if s==0:
            single=0
        else:
            single=price/s
        data[i].append([it[0],row[0],num,s,price,single,tag,kucun,single/
            tag])
        s=0
        price=0
        num+=1
    s=row[2]
    price=row[3]

while num<=104:
    kucun=0
    tag=0
    tag=prod_dict[id][1]
    for t in range(1,7+1):
        x=t
        if t==7:x=0
        kucun+=inv_dict[str(id)+'_'+str(week2date(num,x))]
    single=0.0
    if s==0:
        single=0
    else:
        single=price/s
    data[i].append([it[0],id,num,s,price,single,tag,kucun,single/tag])
    s=0
    price=0
    num+=1

name=["classid","ksc","date_rcd","s","real_cost","price","tag_price","ie","
    discount"]
all_list=[]
for i in range(1,11):
    for j in data[i]:
        all_list.append(j)
all=pd.DataFrame(columns=name,data=all_list)
all.to_csv("问题3/all.csv")

```

```

print(len(all_list))

data=[[] for i in range(0,11)]
item=set()
i=0
summ=0
for it in data_list:

    i+=1
    if i>=11:break
    print(it[0])
    print(it[0])
    class_dict[it[0]].sort(key=lambda x:tran(x[1]))
    item=set()
    for row in class_dict[it[0]]:
        item.add(row[0])
    summ+=len(item)
print(summ)

import scipy.io as sio
import matplotlib.pyplot as plt
import numpy as np
import scipy.io as scio
skc = [i[1] for i in all_list]
data=[[i[5],i[7],i[8]] for i in all_list]
s = [i[3] for i in all_list]
sio.savemat('all.mat', {'input':data,'output':s , 'skc': skc})

```

附录 3： 问题 3 权重系数数据处理与筛选的代码

```

#!/usr/bin/env python
# coding: utf-8

import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
get_ipython().run_line_magic('matplotlib', 'inline')
data=pd.read_csv("all.csv")
data.head()

```

```

list=[]
for index,row in data.iterrows():
    list.append([row[1],row[2],row[3],row[4],row[6],row[8],row[9],0])

name=["classid","skc","date_rcd","s","price","ie","discount","predict"]
all=pd.DataFrame(columns=name,data=list)
all.to_csv("all_list.csv")

item=set()
for it in list:
    item.add(it[0])
#筛选小类周数据
from collections import defaultdict
item_dict=defaultdict(int)
class_list=[]
T=104
l=1
r=104
num=0
MAPE=0
name=["0classid","1skc","2date_rcd","3s","4price","5ie","6discount","7predict"]
for classid in item:
    l=0
    for i in range(0,len(list)):
        if int(list[i][0])==int(classid):
            l=i
            break
    temp=l
    for i in range(1,104+1):

        l=temp
        l+=i
        s=0
        price=0
        kucun=0
        tag=0
        cnt=0

        while(l<=len(list) and int(list[l-1][0])==int(classid)):
            cnt+=1
            if i<92:
                item_dict[list[l-1][1]]+=s
            s+=list[l-1][3]

```

```

        price+=list[l-1][4]
        kucun+=list[l-1][5]
        tag+=list[l-1][4]/list[l-1][6]
        l+=104
    if price==0:
        discount=0
    else:
        discount=tag/price

    class_list.append([classid,i,price/cnt,kucun,discount,s])

name=["classid","date_rcd","price","ie","discount","s"]
all=pd.DataFrame(columns=name,data=class_list)
all.to_csv("BP_class_week.csv")

#读取目标小类周数据的预测结果
data=pd.read_csv("PS0_BP_class_week_result.csv")
data.head()

class_dict=defaultdict(int)
class_d=defaultdict(int)
for index,row in data.iterrows():
    class_dict[str(int(row[1]))+","+str(int(row[2]))]=row[7]

    if row[2]<92:
        class_d[str(int(row[1]))]+=row[6]

#依旧权重系数计算目标小类的所有skc
BP_list=[]
T=104
l=1
r=104
num=0
MAPE=0
name=["0classid","1skc","2date_rcd","3s","4price","5ie","6discount","7predict"]
while l<=len(list):
    cnt=0
    mape=0.0
    for i in range(r-12+1,r+1):
        w=item_dict[list[i-1][1]]/class_d[str(int(list[i-1][0]))]
        list[i-1][7]=w*class_dict[str(int(list[i-1][0]))+","+str(i-1)]
        mape+=abs(list[i-1][3]-list[i-1][7])/list[i-1][3]

```

```

        if list[i-1][3]==0:
            cnt+=1
    if cnt>=1:
        l+=T
        r+=T
        continue

    cnt=0
    for i in range(1,r-12+1):
        if list[i-1][5]!=0:
            cnt+=1

    if cnt>20:
        MAPE+=mape/12
        for i in range(1,r+1):
            BP_list.append([list[i-1][0],list[i-1][1],list[i-1][2],list[i-1][4],
                            list[i-1][5],list[i-1][6],list[i-1][3],list[i-1][7],mape/12])
            num+=1
        l+=T
        r+=T
print(num)
print(MAPE/264)

name=["classid","skc","date_rcd","price","ie","discount","s","predict","MAPE"]
all=pd.DataFrame(columns=name,data=BP_list)
all.to_csv("PSO_BP_calss_skc_result.csv")

data=pd.read_csv("PSO_BP_calss_skc_result.csv")
data.head()
#画图可视化
plt.figure(figsize=(12, 6),facecolor='snow')

plt.rcParams['font.sans-serif']=['Microsoft YaHei']
T=104
x = data["date_rcd"][T-12:T]
y1 = data["s"][T-12:T]
y2 = data["predict"][T-12:T]

plt.plot(x,y1,label='真实销售量',linewidth=3,mfc='w',markersize=12,mfcalt='b')
plt.plot(x,y2,label='预测销售量',color='r',marker='*',linewidth=2)
#plt.vlines(0, 0, 0.5, colors = "r", linestyle = "dashed")
#plt.axvline(x=x[92],ls="--",c="green")#添加垂直直线

```



```

# x轴标签
plt.xlabel('日期/周',fontdict={'size' : 16})
# y轴标签
plt.ylabel('销售量',fontdict={'size' : 16})

# 可视化图标题
plt.title('skc596572119262的周预测模型结果',fontdict={'size' : 16})

# 显示图例
plt.legend(prop={'size' : 16})
plt.xticks(rotation=45)
plt.show()

```

附录 4: JB 正太检验的代码

```

n_c = size(all,2); % number of column    %
H = zeros(1,4); %  $H_0: \mu = \frac{1}{2}, \sigma^2 = \frac{1}{4}$ 
P = zeros(1,4);
for i = 1:n_c
    [h,p] = jbtest(all(:,i),0.05);
    H(i)=h;
    P(i)=p;
end
H
P

```

附录 5: 粒子群算法优化神经网络的代码

```

function error = fun(x,inputnum,hiddennum,outputnum,net,inputn,outputn)
% 输入 4x1 的向量
%x      input      ,
%inputnum input
%outputnum input     $\sigma^{-2}$ 
%net      input
%inputn    input    w      %
%outputn    input    w      %

%error      output      ,      %

%
w1=x(1:inputnum*hiddennum);
B1=x(inputnum*hiddennum+1:inputnum*hiddennum+hiddennum);

w2=x(inputnum*hiddennum+hiddennum+1:inputnum*hiddennum+hiddennum*

```

```

        outputnum);
B2=x(inputnum*hiddennum+hiddennum+hiddennum*outputnum+1:inputnum*hiddennum+
    hiddennum+hiddennum*outputnum+outputnum);

net=newff(inputn,outputn,hiddennum);
%      >> ^2
net.trainParam.epochs=20;
net.trainParam.lr=0.1;
net.trainParam.goal=0.00001;
net.trainParam.show=100;
net.trainParam.showWindow=0;

%      > ^3
net.iw{1,1}=reshape(w1,hiddennum,inputnum);
net.lw{2,1}=reshape(w2,outputnum,hiddennum);
net.b{1}=reshape(B1,hiddennum,1);
net.b{2}=B2;

%      w
net=train(net,inputn,outputn);

an=sim(net,inputn);

error=sum(abs(an-outputn));

```

```

%%      • 3/4^3
clc
clear

%%      3/4
T=24;
data=[];
for t=1:10
    load 'G:\jupyter_notebook\1_4\mathorcup\ 2\all.mat'
    % w 3/4^2 3/4
    l=1+T*(t-1);
    mid=l+20;
    r=T*t;
    l
    mid
    r
    input_train=input(l:mid,:);
    input_test=input(mid+1:r,:);

```

```

output_train=output(1:mid)';
output_test=output(mid+1:r)';

%% ½
inputnum=3;
hiddennum=4;
outputnum=1;

% 1 ±%      % »-
[inputn,inputps]=mapminmax(input_train);
[outputn,outputps]=mapminmax(output_train);

% 1 1½''
net=newff(inputn,outputn,hiddennum);

%% 2 3 »-
%c 烽 }、
c1 = 1.49445;
c2 = 1.49445;

mgen=50; % ½ »-
sizepop=100; % 1 g

Vmax=1;
Vmin=-1;
popmax=5;
popmin=-5;

for i=1:sizepop
    pop(i,:)=5*rands(1,21);
    V(i,:)=rands(1,21);
    fitness(i)=fun(pop(i,:),inputnum,hiddennum,outputnum,net,inputn,outputn);
end

% 、 强° 强
[bestfitness bestindex]=min(fitness);
zbest=pop(bestindex,:); %ō%
gbest=pop; %、
fitnessgbest=fitness; %、 ¶
fitnesszbest=bestfitness; %ō% ¶

%% μ´

```

```

for i=1:mgen
    for j=1:sizepop

        %
        V(j,:) = V(j,:) + c1*rand*(gbest(j,:) - pop(j,:)) + c2*rand*(zbest -
            pop(j,:));
        V(j,find(V(j,*)>Vmax))=Vmax;
        V(j,find(V(j,*)<Vmin))=Vmin;

        % 速度
        pop(j,:)=pop(j,*)+0.2*V(j,);
        pop(j,find(pop(j,*)>popmax))=popmax;
        pop(j,find(pop(j,*)<popmin))=popmin;

        % 位置
        pos=unidrnd(21);
        if rand>0.95
            pop(j,pos)=5*rands(1,1);
        end

        % 适应度
        fitness(j)=fun(pop(j,:),inputnum,hiddennum,outputnum,net,inputn,outputn
            );
    end

    for j=1:sizepop
        % 更新个体最优
        if fitness(j) < fitnessgbest(j)
            gbest(j,:) = pop(j,);
            fitnessgbest(j) = fitness(j);
        end

        % 更新全局最优
        if fitness(j) < fitnesszbest
            zbest = pop(j,);
            fitnesszbest = fitness(j);
        end
    end

    yy(i)=fitnesszbest;

end

```

```

%% ½ •
plot(yy)
title(['  ' ' ' ' ' num2str(mgen)]);
xlabel('½ »⁻¹ ');ylabel(' ');

%% ° ¼ • §,³ ²
% % Ẃ« 烽 Ž⁻ᵖ BP ²

x=yy;
w1=x(1:inputnum*hiddennum);
B1=x(inputnum*hiddennum+1:inputnum*hiddennum+hiddennum);
w2=x(inputnum*hiddennum+hiddennum+1:inputnum*hiddennum+hiddennum+hiddennum*
      outputnum);
B2=x(inputnum*hiddennum+hiddennum+hiddennum*outputnum+1:inputnum*hiddennum+
      hiddennum+hiddennum*outputnum+outputnum);

net.iw{1,1}=reshape(w1,hiddennum,inputnum);
net.lw{2,1}=reshape(w2,outputnum,hiddennum);
net.b{1}=reshape(B1,hiddennum,1);
net.b{2}=B2;

%% BP w
% »⁻²
net.trainParam.epochs=100;
net.trainParam.lr=0.1;
%net.trainParam.goal=0.00001;

% w
[net,per2]=train(net,inputn,outputn);

%% BP ²
% ¼ »⁻
inputn_test=mapminmax('apply',input(1:r,:),inputps);
an=sim(net,inputn_test);
test_simu=mapminmax('reverse',an,outputps);
%error=test_simu-output_test;
for i=1:T
    test_simu(i)=round(test_simu(i))
end
MAPE=0;
for i=22:T
    test_simu(i)=round(test_simu(i))
    MAPE=MAPE+(abs(test_simu(i)-output_test(i-21)))/output_test(i-21);
end

```

```

    data=[data;output_train,output_test,0;test_simu,MAPE]
    MAPE=MAPE/3;
    break;
end

```

附录 6：遗传算法优化神经网络的代码

```

% 读取数据
data=xlsread('data.xls');

% 划分训练和测试数据
data_train=data(1:113,:);
data_test=data(118:123,:);

input_train=data_train(:,1:9)';
output_train=data_train(:,10)';

input_test=data_test(:,1:9)';
output_test=data_test(:,10)';

% 创建神经网络
[inputn,mininput,maxinput,outputn,minoutput,maxoutput]=premnmx(input_train,
    output_train); % 归一化数据
net=newff(minmax(inputn),[10,1],{'tansig','purelin'},'trainlm');

net.trainParam.epochs=100;
net.trainParam.lr=0.1;
net.trainParam.goal=0.00001;
%net.trainParam.show=NaN

% 训练神经网络
net=train(net,inputn,outputn);

% 测试神经网络
inputn_test = trmnmx(input_test,mininput,maxinput);

an=sim(net,inputn);

test_simu=postmnmx(an,minoutput,maxoutput);

error=test_simu-output_train;

plot(error)

```

```
k=error./output_train
```

```
function ret=Code(lenchrom,bound)
%±¼² ½«± -± 壬 ³ »²,
% lenchrom input : 峴¶
% bound input : ±-µ • ¶X
% ret output: 1
flag=0;
while flag==0
    pick=rand(1,length(lenchrom));
    ret=bound(:,1)'+(bound(:,2)-bound(:,1))'*pick; % ±-± -´ ret
    flag=test(lenchrom,bound,ret); %¼ L
end
```

```
function ret=Cross(pcross,lenchrom,chrom,sizepop,bound)
%±¼² »²
% pcross input : ½»²
% lenchrom input : ¶¶
% chrom input :
% sizepop input : ¹ ġ
% ret output : ½»²
for i=1:sizepop %Ÿ for » • ² »² ġ-½»² ġ-½µ«, for » •
    »² »² ³ "continue; ©
    % } , »²
    pick=rand(1,2);
    while prod(pick)==0
        pick=rand(1,2);
    end
    index=ceil(pick.*sizepop);
    % ½»² ³ »²
    pick=rand;
    while pick==0
        pick=rand;
    end
    if pick>pcross
        continue;
    end
    flag=0;
    while flag==0
        %
        pick=rand;
```

```

while pick==0
    pick=rand;
end
pos=ceil(pick.*sum(lenchrom)); % 2^24 - 1/2^24 捲 } , 彻^2

pick=rand; %2^24 漈
v1=chrom(index(1),pos);
v2=chrom(index(2),pos);
chrom(index(1),pos)=pick*v2+(1-pick)*v1;
chrom(index(2),pos)=pick*v1+(1-pick)*v2; %2^24
flag1=test(lenchrom,bound,chrom(index(1),:)); %4 1μL
flag2=test(lenchrom,bound,chrom(index(2),:)); %4 2μL
if flag1*flag2==0
    flag=0;
else flag=1;
end % } , 岷 1/2 1/2^2
end
end
ret=chrom;

```

```

function ret=Decode(lenchrom,bound,code,opts)
% ±¼° ¼
% lenchrom input : 岷¼
% bound input : ± - • ¼X
% code input ¼±
% opts input : 1/2 • "± K
% ret output: L
switch opts
case 'binary' % binary coding
    for i=length(lenchrom):-1:1
        data(i)=bitand(code,2^lenchrom(i)-1); %2μ ¼° 3« 1/2 1 data(i)
        code=(code-data(i))/(2^lenchrom(i)); %μ 得°
    end
    ret=bound(:,1)'+data./(2.^lenchrom-1).*(bound(:,2)-bound(:,1))'; %•
        -μ 1 ret

case 'grey' % grey coding
    for i=sum(lenchrom):-1:2
        code=bitset(code,i-1,bitxor(bitget(code,i),bitget(code,i-1)));
    end
    for i=length(lenchrom):-1:1
        data(i)=bitand(code,2^lenchrom(i)-1);
        code=(code-data(i))/(2^lenchrom(i));
    end
end

```



```

end
ret=bound(:,1)'+data./(2.^lencrom-1).*(bound(:,2)-bound(:,1))'; % •
    -µ    ret

case 'float' % float coding
    ret=code; %½    %    %    - 200- ret
end

```

```

function error = fun(x,inputnum,hiddennum,outputnum,net,inputn,outputn)
% ,ú- 4%    ¶
%x        input    ,
%inputnum input
%outputnum input    0-2
%net        input
%inputn    input    w    %
%outputn    input    w    %

%error    output    ,    ¶

%
w1=x(1:inputnum*hiddennum);
B1=x(inputnum*hiddennum+1:inputnum*hiddennum+hiddennum);

w2=x(inputnum*hiddennum+hiddennum+1:inputnum*hiddennum+hiddennum*
    outputnum);
B2=x(inputnum*hiddennum+hiddennum+hiddennum*outputnum+1:inputnum*hiddennum+
    hiddennum+hiddennum*outputnum+outputnum);

net=newff(inputn,outputn,hiddennum);
%    »-2
net.trainParam.epochs=20;
net.trainParam.lr=0.1;
net.trainParam.goal=0.00001;
net.trainParam.show=100;
net.trainParam.showWindow=0;

%    ,3
net.iw{1,1}=reshape(w1,hiddennum,inputnum);
net.lw{2,1}=reshape(w2,outputnum,hiddennum);
net.b{1}=reshape(B1,hiddennum,1);
net.b{2}=B2;

%    w

```

```
net=train(net,inputn,outputn);
```

```
an=sim(net,inputn);
```

```
error=sum(abs(an-outputn));
```

```
%% 计算误差
```

```
clc
```

```
clear
```

```
%% 初始化
```

```
T=24;
```

```
data=[];
```

```
for t=1:10
```

```
    load 'G:\jupyter_notebook\mathorcup\2\all.mat'
```

```
    % 初始化
```

```
    inputnum=3;
```

```
    hiddennum=4;
```

```
    outputnum=1;
```

```
    % 生成训练和测试数据
```

```
    l=1+24*(t-1);
```

```
    mid=l+20;
```

```
    r=24*t;
```

```
    l
```

```
    mid
```

```
    r
```

```
    input_train=input(l:mid,:);
```

```
    input_test=input(mid+1:r,:);
```

```
    output_train=output(l:mid);
```

```
    output_test=output(mid+1:r);
```

```
    % 归一化数据
```

```
    [inputn,inputps]=mapminmax(input_train);
```

```
    [outputn,outputps]=mapminmax(output_train);
```

```
    % 创建神经网络
```

```
    net=newff(inputn,outputn,hiddennum);
```

```
    % 训练神经网络
```

```

maxgen=20; % 1/2 10^2 10^2 10^2
sizepop=100; % 10^1 10^1 10^1
pcross=[0.3]; % 1/2 10^2 0 1 1/4
pmutation=[0.1]; % 1/2 0 1 1/4

% 1/2
numsum=inputnum*hiddennum+hiddennum+hiddennum*outputnum+outputnum;

lenchrom=ones(1,numsum);
bound=[-3*ones(numsum,1) 3*ones(numsum,1)]; % X 1/4

%----- 10^3 10^2 -----
individuals=struct('fitness',zeros(1,sizepop), 'chrom',[]); % 1/2 10^1 10^1
avgfitness=[]; % 10^1 10^1 10^1
bestfitness=[]; % 10^1 10^1 10^1
bestchrom=[]; % 10^1 10^1 10^1
% 10^3 10^2
for i=1:sizepop
    % 10^2 10^1
    individuals.chrom(i,:)=Code(lenchrom,bound); % 10^1 10^1 10^1 10^1
    % float 10^1 10^1 10^1
    x=individuals.chrom(i,:);
    % 10^1 10^1
    individuals.fitness(i)=fun(x,inputnum,hiddennum,outputnum,net,inputn,
        outputn); % 10^1 10^1
end

% 10^1 10^1
[bestfitness bestindex]=min(individuals.fitness);
bestchrom=individuals.chrom(bestindex,:); % 10^1 10^1
avgfitness=sum(individuals.fitness)/sizepop; % 10^1 10^1 10^1
% 10^1 10^1 10^1 10^1 10^1 10^1
trace=[avgfitness bestfitness];

% 10^1 10^1 10^1 10^1
% 10^1 10^1 10^1 10^1
for i=1:maxgen
    i
    %
    individuals=Select(individuals,sizepop);
    avgfitness=sum(individuals.fitness)/sizepop;
    % 10^1 10^2
    individuals.chrom=Cross(pcross,lenchrom,individuals.chrom,sizepop,bound);

```

```

% ±
individuals.chrom=Mutation(pmutation,lenchrom,individuals.chrom,sizepop,i,
    maxgen,bound);

% ¼ ¶
for j=1:sizepop
    x=individuals.chrom(j,:); %½
    individuals.fitness(j)=fun(x,inputnum,hiddennum,outputnum,net,inputn,
        outputn);
end

% ½ ° ¶ 弼
[newbestfitness,newbestindex]=min(individuals.fitness);
[worestfitness,worestindex]=max(individuals.fitness);
% ´ ´ »- õ
if bestfitness>newbestfitness
    bestfitness=newbestfitness;
    bestchrom=individuals.chrom(newbestindex,:);
end
individuals.chrom(worestindex,:)=bestchrom;
individuals.fitness(worestindex)=bestfitness;

avgfitness=sum(individuals.fitness)/sizepop;

trace=[trace;avgfitness bestfitness]; %¼¼ÿ´½»- õ ¶ ¼ ¶

end
%% 燄½ •
figure(1)
[r c]=size(trace);
plot([1:r]',trace(:,2),'b--');
title(['¶ ´ ´ ´ £½' num2str(maxgen)]);
xlabel('½»- ');ylabel('¶ ');
legend('¼ ¶ ','¶ ');
disp('¶ ± -');
x=bestchrom;

%% ° ¶ • § ,3 2
% % 燄 Z~p BP 2
w1=x(1:inputnum*hiddennum);
B1=x(inputnum*hiddennum+1:inputnum*hiddennum+hiddennum);
w2=x(inputnum*hiddennum+hiddennum+1:inputnum*hiddennum+hiddennum*
    outputnum);
B2=x(inputnum*hiddennum+hiddennum+hiddennum*outputnum+1:inputnum*hiddennum+

```

```

        hiddennum+hiddennum*outputnum+outputnum);

net.iw{1,1}=reshape(w1,hiddennum,inputnum);
net.lw{2,1}=reshape(w2,outputnum,hiddennum);
net.b{1}=reshape(B1,hiddennum,1);
net.b{2}=B2;

%% BP      w
%          >>^-2
net.trainParam.epochs=100;
net.trainParam.lr=0.001;
%net.trainParam.goal=0.00001;

%          w
[net,per2]=train(net,inputn,outputn);

%% BP      2
%          >>^-
inputn_test=mapminmax('apply',input_test,inputtps);
an=sim(net,inputn_test);
test_simu=mapminmax('reverse',an,outputtps);
error=test_simu-output_test;

MAPE=0
for i=1:3
    test_simu(i)=round(test_simu(i))
    MAPE=MAPE+(abs(test_simu(i)-output_test(i)))/output_test(i);
end
data=[data;output_test,0;test_simu,MAPE]
MAPE=MAPE/3
end

```

```

function ret=Mutation(pmutation,lenchrom,chrom,sizepop,num,maxgen,bound)
%  ±¼°-
% pcorss          input : ±
% lenchrom        input : 峽¶
% chrom   input :
% sizepop         input : ¹ 
% opts           input : ±      • ``µ
% pop            input : µ±ĵ µL »^- °      L »^-
% bound          input : ŷ,      ½
% maxgen         input : £°      ^ ^
% num            input : µ±ĵµ ^ ^

```

```

% ret          output : ±

for i=1:sizepop %ÿ for » • ½        ½        ½
    %µ½, for » •         "continue½ ©
    %
    pick=rand;
    while pick==0
        pick=rand;
    end
    index=ceil(pick*sizepop);
    % ±         » •
    pick=rand;
    if pick>pmutation
        continue;
    end
    flag=0;
    while flag==0
        % ±
        pick=rand;
        while pick==0
            pick=rand;
        end
        pos=ceil(pick*sum(lenchrom)); %        ½        pos, -½

        pick=rand; %±
        fg=(rand*(1-num/maxgen))^2;
        if pick>0.5
            chrom(i,pos)=chrom(i,pos)+(bound(pos,2)-chrom(i,pos))*fg;
        else
            chrom(i,pos)=chrom(i,pos)-(chrom(i,pos)-bound(pos,1))*fg;
        end %±
        flag=test(lenchrom,bound,chrom(i,:)); %¼ L
    end
end
ret=chrom;

```

```

function ret=select(individuals,sizepop)
% µ½
% individuals input
% sizepop input        ½ 
% ret        output

% µ½ µ¹

```

```

fitness1=10./individuals.fitness; %individuals.fitnessI,
%
sumfitness=sum(fitness1);
sumf=fitness1./sumfitness;

%2 k''
index=[];
for i=1:sizepop %sizepopI
    pick=rand;
    while pick==0
        pick=rand;
    end
    for i=1:sizepop
        pick=pick-sumf(i);
        if pick<0
            index=[index i];
            break;
        end
    end
end

%
individuals.chrom=individuals.chrom(index,:); %individuals.chromI
individuals.fitness=individuals.fitness(index);
ret=individuals;

```