赛区评阅编号（由赛区组委会填写）：

C:\Users\Lenovo\AppData\Local\Temp\ksohtml\wpsC72B.tmp.png

**2016年高教社杯全国大学生数学建模竞赛**

**承 诺 书**

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称为“竞赛章程和参赛规则”，可从全国大学生数学建模竞赛网站下载）。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛章程和参赛规则的，如果引用别人的成果或资料（包括网上资料），必须按照规定的参考文献的表述方式列出，并在正文引用处予以标注。在网上交流和下载他人的论文是严重违规违纪行为。

**我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。**

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从A/B/C/D中选择一项填写）： B

我们的报名参赛队号（12位数字全国统一编号）： 20王富帅\_赵振进\_冯璟\_

参赛学校（完整的学校全称，不含院系名）： 燕山大学

参赛队员 (打印并签名) ：1. 王富帅

2. 赵振进

3. 冯璟

指导教师或指导教师组负责人 (打印并签名)：

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期： 2018 年 8 月 16 日

赛区评阅编号（由赛区组委会填写）：

C:\Users\Lenovo\AppData\Local\Temp\ksohtml\wps7FEA.tmp.png

**2016年高教社杯全国大学生数学建模竞赛**

**编 号 专 用 页**

赛区评阅记录（可供赛区评阅时使用）：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 评  阅  人 |  |  |  |  |  |  |
| 备  注 |  |  |  |  |  |  |

送全国评阅统一编号（赛区组委会填写）：

全国评阅随机编号（全国组委会填写）：

基于主成分分析和神经网络预测的航食配菜规划问题模型求解

摘要

本文建立了遗传算法改进的BP神经网络模型，通过对航空公司历史数据的分析利用来预测某航班在某条件下的航食需求，使航空公司能够对飞机航班的航食进行节约、高效、合理的配置。

针对问题一，基于聚类去噪的原理，建立了均值聚类模型。将附件数据进行多次不同条件和不同范围的聚类分析，找到聚类后的孤立点，根据其孤立的原因，确定是将数据进行剔除还是进行数据的修复，我们将空缺的数据行和不符合逻辑的数据行进行剔除，将颠倒顺序的数据行进行数据修复。我们使用matlab和Excel软件进行聚类分析等操作。最终将附件中的70多万数据清洗为33823条数据。

针对问题二，基于灰色关联分析与主成分分析等原理，将各自变量间进行相关性分析，把相关性极强的数据认为数据重叠，对其进行删减。然后再用spss软件对自变量进行主成分分析，减少变量维度。进行灰色关联分析，将关联性不强的主成分剔除。我们最终分析得到因变量与日期、航班号、天气状况、风、空气、星期及温度有关，并将其整合为5个主成分，最后通过熵值法确定每个主成分的权重，分析得出和航班订单数（配餐数）主要相关的为季节温度、天气情况以及节假日情况。

针对问题三，基于遗传算法优化的BP神经网络模型建立求解。根据问题二所求结果，对五个主成分以及人数进行分析。使用BP神经网络模型进行求解，误差较大，通过查阅资料，考虑使用遗传算法优化的BP神经网络，得到了最优染色体以及最优适应度为25.0727，寻找出最优的权值和阈值，增加了神经网络预测的精准性，将误差精确的缩小到了8.94\*10^-5以内，经过大量数据的训练，精确的预测出了航班配食量。

针对问题四整合前三问模型后使用熵权值法来确定各主成分所占比重进行合理规划。

关键词：聚类去噪 主成分分析 遗传算法 BP神经网络 灰色关联分析

1. 问题重述

随着经济社会的发展，选择乘坐飞机出行的旅客越来越多，在飞机上进餐几乎成了一个必不可少的环节，因而如何让飞机上配备的食物和乘客消费的数量能最接近，进而最大限度的节约成本，是一个有待解决的问题。这是一个考虑多因素的资源配置问题。本文研究分析和处理历史航班和所提供的餐饮量以及其它一些数据信息， 问题要求分析历史航班和影响航班相关因素的数据进行处理和分析，完成以下问题：

1. 分析历史航班和所提供的餐饮量以及其它一些数据信息，确认并踢出不符合实际逻辑的噪声数据。
2. 结合提供的数据，分析天气季度与航班订单数量（配餐数量）相关的因素，并提供科学地分析方法。
3. 结合相关数据集进行建模，建立合理的模型预测配餐数量，并对构建的模型的可信度和可靠性给出检验和分析
4. 请结合分析的数据关系、建模结果以及研究结论。提出对航食配餐合理的优化策略。
5. 模型假设

1、假设准备的食材满足所有乘客的订餐需求，不考虑乘客乘坐飞机但不订餐的情况。

2、假设提供的航班信息大部分真实可靠。

3、假设无重大自然灾害影响。

1. 符号说明

**3.1 名词解释**

1.子订单：不同时段该航班的订单数量统计。

2．其他：非头等舱、公务舱、经济舱的机舱。

3.2 变量说明

|  |  |  |
| --- | --- | --- |
| **变量名称** | **含义** | **单位** |
|  | **第i个子订单的订单情况** | **人** |
|  | **P机型总订单数** | **人** |
| **i** | **子订单编号** | **个** |
| **M** | **总仓位数** | **个** |
|  | **第q种气象条件** |  |
| **q** | **气象条件种类** |  |

1. 问题分析

问题一：本题的求解主要是对庞大的原始数据进行处理，通过观察数据找出该数据库的主要的错误，建立逻辑错误模型，通过聚类去噪方法找出错误数据，并修正会对结果产生干扰的异常值；对于超售问题通过处理的数据将累计的子订单进行加和，通过matlab遍历，显示异常数据并将其剔除和修正。将噪声数据对结果的影响降到最小。

问题二：本题采用主成分分析的方法，将多个影响变量转换为少数几个综合指标表进行分析，建立优化模型。基于灰色关联分析与主成分分析等原理，将各自变量间进行相关性分析，把相关性极强的数据认为数据重叠，对其进行删减。然后再用spss软件对自变量进行主成分分析，减少变量维度。进行灰色关联分析，将关联性不强的主成分剔除。分析得到于因变量相关的因素。

问题三：初步使用BP神经网络模型进行求解，并且求得结果误差，当误差较大时摒弃初步的想法。因为BP神经网络的权值与阈值都是随机赋值的，这就导致了误差的存在并且可能只能求得局部最优解 。通过查阅资料，根据遗传算法的全局性、不受约束性及灵活性等特点，考虑使用遗传算法优化的BP神经网络。通过遗传算法的“适者生存”法则，得到了最优染色体以及最优适应度，寻找出最优的权值和阈值，从而提高的增加了神经网络预测的精准性。

问题四：本题的求解即是通过对第二问产生的主成分的权重进行分析，以确定贡献最大者，以期航空公司制定配餐战略时能更加注重权重大的条件的影响，随后再用第三问解得的预测配置数量模型进行求解。

1. 模型的建立和求解

**5.1问题一**

**5.1.1模型的建立**

本模型结合生活中的实际逻辑关系，对于数据进行处理和逻辑分析，对于不符合客观事实的数据予以剔除和修正。

逻辑关系一：航班的子订单退票数不能超出初始订单数

（1）

逻辑关系二：航班的总订单数不定超过航班相应座位的荷载人数

（2）

对于以上逻辑关系混乱的数据，应当予以处理。

对于数据错乱问题进行聚类分析，数据间的距离为

（3）

通过比较距离大小确定类归属于哪个簇。

**5.1.2 模型的求解**

通过大致浏览数据可以发现数据的错误大致有着几个方面：数据错乱、数据缺失、数据错误等。我们对其分别进行了处理。

我们首先对经济舱总数、公务舱总数这些数据进行小范围内的多次均值聚类分析，采用数据间的距离作为评价指标，将每个数据区域聚为两类，这样数目少的一簇我们就认为是由数据错乱项构成，通过使其值修正为大簇的数据来进行数据恢复。

具体做法为：

1：计算数据之间的距离（采用欧氏距离），记为距离矩阵

2：合并距离最小的两类作为新类。

其中m为选定数据域的行数，n为选定数据域的列数。

重复此过程，直到将选定数据域聚合为两类为止。

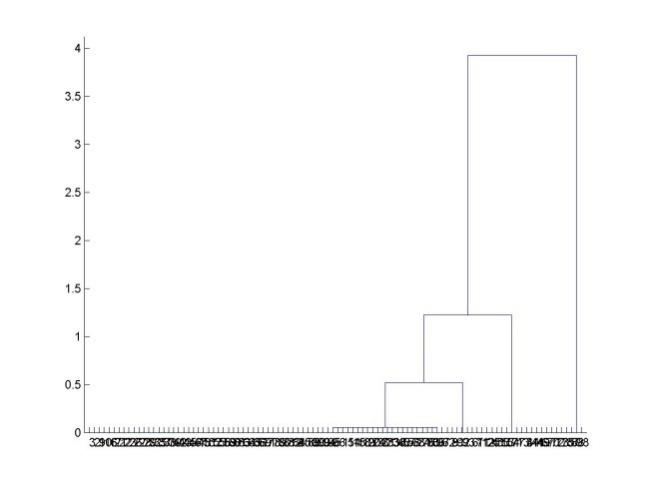
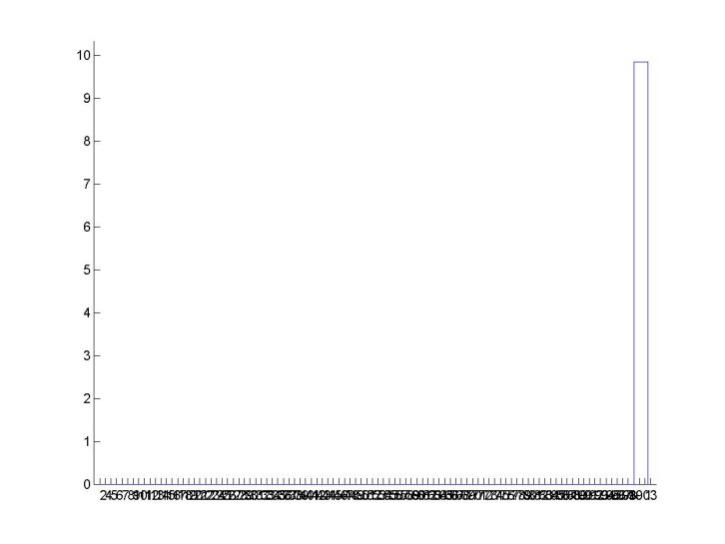


图1：部分聚类结果图

聚类完成后将较小的一簇的数据值修改为大簇中相应位置的值。

其次进行对数据缺失项的处理，我们通过遍历整合文件的每个数据域来判断其是否为空，若为空则直接删除其所在的数据行。即遍历数据时如果发现为空，则删除

最后是对数据错误地数据行进行剔除，先将同一航班同一天的数据的机舱人数及订单状况等信息通过其本身的逻辑关系整合到一个数据行 ，再依据公式

若不满足其中任意一项，则删除。

在数据库中有关于航班总座位数存在大量错误数据，例如同一天同一航班仓位数出现混乱等情况，为减小后期处理逻辑问题的工作量，我们通过matlab聚类分析识别噪声数据，并考察它们值与其期望之间的距离；然后，对于能够判定噪声的记录，寻找它所属的分类，并利用它所属分类中噪声期望上的值进行矫正；通过此步操作修正航班座位数据。

**数据剔除** 利用预处理数据，利用公式（1）（2）通过matlab遍历和迭代求解，将不符合模型逻辑的数据进行剔除。详见见附件2.

**5.1.3结果分析**

经过数据清洗，原本七十万的航班数据缩减成33823条。通过再次对数据中的逻辑进行分析，绝大部分数据符合模型中的逻辑公式，并再次对数据域进行遍历及组内距离分析，发现数据中已经没有空数据及数据错乱情况，说明针对这三个方面进行的数据清洗较为正确、合理。

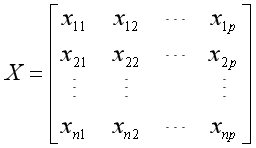
但是数据剔除时遍历到空数据域时直接删除掉其所在数据行有可能导致数据损失过多，因为有些数据列会产生信息重叠，即使某些数据域信息丢失，其他列的数据也有可能对数据进行描述，关于信息之间具有强相关性导致信息重叠在后面会进行理。

**5.2问题二**

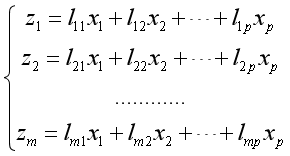
**5.2.1模型的建立**

当前，对于航食配餐问题的预测应当建立在影响航班订单数的因素的相关性的基础上进行分析因此需要确定各个影响因素对于航班订单数的相关性程度。本题研究主要采用主成分分析的方法进行分析。通过数据分析发现，由于提供的影响因素较多，各个因素对于航班的订单人数的影响结果并不显著。因此采用主成分分析的方法，将多个影响变量转换为少数几个综合指标表进行分析。

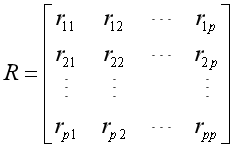
1. 有n个航班订单，每个样本共有p个影响因素，构成一个n×p阶的数据矩阵，

 (4)

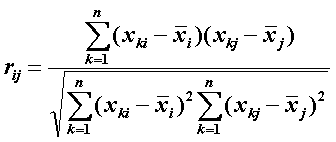
记原变量指标为x1，x2，…，xp，设它们降维处理后的综合指标，即新变量为 z1，z2，z3，… ，zm(m≤p)，则

 (5)

1. 计算相关系数矩阵

 (6)

（i，j=1，2，…，p）为原变量与的相关系数，，其计算公式为

 (7)

1. 计算特征值与特征向量

解特征方程=0，常用雅可比法（Jacobi）求出特征值，并使其按大小顺序排列; 分别求出对应于特征值的特征向量，要求，即，其中表示向量的第j个分量。

1. 计算主成分贡献率及累计贡献率

贡献率：

（i=1,2，…,p） (8)

累计贡献率：

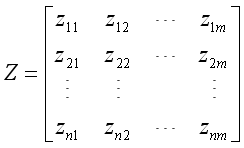
(i=1,2…,p) (9)

取累计贡献率达85%-95%的特征值，所对应的第1、第2、…、第m（m≤p）个主成分。

1. 计算主成分

(i，j=1,2…,p) (10)

1. 各主成分得分

 (11)

从而求解出如公式（4）de各主成分的线性表达式。

1. 根据各主成分对于各变量的相关性系数对各主成分进行分析分类。通过熵权值法对各主成分所占权重进行分析求解得出航班订单数的主要影响因素

**5.2.2 模型的求解**

对非数值型数据进行数值化处理，对天气状况、风力、污染情况按等级赋值（见附件1）同时对数据进行标准化处理。通过spss对自变量进行相关性分析，如表1.

通过对自变量之间的分析可知，对于自变量之间的相关性高的变量进行剔除，因此保留日期、航班号、天气状况、风速、空气数、星期、温度七个相互独立性较高的自变量，日期与季、月、年呈现很强的相关性，尤其日期和年，相关性为1，认为信息完全重叠，因此将季、月、年变量去除，只用日期来衡量时间变换，而高温、低温之间也有极强的相关性0.962，认为信息重叠率足够高，因此选择高温与低温的均值来表述一天的温度。

去除信息重叠的变量之后的相关性矩阵为：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 日期 | 航班号 | 天气 | 高温 | 低温 | 风数 | 空气数 | 季 | 月 | 年 | 星期 |
| 相关性 | 日期 | 1.000 | -.011 | -.260 | .624 | .722 | -.245 | .078 | .955 | .995 | 1.000 | -.001 |
| 航班号 | -.011 | 1.000 | .008 | -.003 | -.007 | .002 | .004 | -.011 | -.012 | -.011 | -.010 |
| 天气 | -.260 | .008 | 1.000 | -.125 | -.249 | -.046 | .212 | -.276 | -.260 | -.261 | .052 |
| 高温 | .624 | -.003 | -.125 | 1.000 | .962 | -.303 | .025 | .547 | .629 | .624 | .001 |
| 低温 | .722 | -.007 | -.249 | .962 | 1.000 | -.299 | .038 | .646 | .726 | .721 | .013 |
| 风数 | -.245 | .002 | -.046 | -.303 | -.299 | 1.000 | -.214 | -.241 | -.250 | -.245 | .018 |
| 空气数 | .078 | .004 | .212 | .025 | .038 | -.214 | 1.000 | .045 | .070 | .080 | -.074 |
| 季 | .955 | -.011 | -.276 | .547 | .646 | -.241 | .045 | 1.000 | .960 | .955 | -.001 |
| 月 | .995 | -.012 | -.260 | .629 | .726 | -.250 | .070 | .960 | 1.000 | .995 | .001 |
| 年 | 1.000 | -.011 | -.261 | .624 | .721 | -.245 | .080 | .955 | .995 | 1.000 | -.024 |
| 星期 | -.001 | -.010 | .052 | .001 | .013 | .018 | -.074 | -.001 | .001 | -.024 | 1.000 |

表1.相关性矩阵

采用主成分分析的方法对数据进行降维处理。如表2分析可知，因此选取5个主成分时就可以解释总方差的99.173%，因此我们选择5个主成分来解释变量。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 成分 | 初始特征值 | | | 提取载荷平方和 | | |
| 总计 | 方差百分比 | 累积 % | 总计 | 方差百分比 | 累积 % |
| 1 | 1.942 | 27.737 | 27.737 | 1.942 | 27.737 | 27.737 |
| 2 | 1.320 | 18.851 | 46.588 | 1.320 | 18.851 | 46.588 |
| 3 | 1.024 | 14.632 | 61.221 | 1.024 | 14.632 | 61.221 |
| 4 | .996 | 14.234 | 75.454 | .996 | 14.234 | 75.454 |
| 5 | .726 | 10.370 | 85.825 | .726 | 10.370 | 85.825 |
| 6 | .682 | 9.738 | 95.563 |  |  |  |
| 7 | .311 | 4.437 | 100.000 |  |  |  |

表2.总方差解释

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 日期 | 0.71 | -0.08 | 0.03 | 0.03 | 0.03 |
| 航班号 | -0.01 | 0.03 | -0.34 | -0.35 | -0.41 |
| 天气状况 | -0.29 | 0.59 | 0.22 | 0.23 | 0.27 |
| 风速 | -0.43 | -0.43 | -0.06 | -0.07 | -0.08 |
| 空气数 | 0.12 | 0.67 | -0.1 | -0.1 | -0.12 |
| 星期 | -0.03 | -0.08 | 0.9 | 0.91 | 1.07 |
| 温度 | 0.71 | -0.05 | 0.07 | 0.07 | 0.08 |

表3.主成分成分系数表

由上表，三个主成分的线性组合如下：

通过分析五个主成分的线性组合可知，第一主成分可以表示为季节温度；第二各主成分可以表示为天气情况；第三、四、五主成分表示为节假日情况； 因此我们可以分析得出和航班订单数（配餐数）主要相关的为季节温度、天气情况以及节假日情况。

**5.3问题三**

**5.3.1模型的建立**

在第二问主成分分析的基础上，通过五个主成分变量对数据进行BP神经网络训练从而完成航食配餐的预测。由于BP神经网络与其他的网络相比具有很优越的特点，即分布式存储方式、大规模并行处理、自学习和自适应能力，能够对数据进行很好的预测，所以我们初步打算使用BP神经网络模型对问题三进行分析。

1. BP神经网络模型

BP神经网络模型是一个具有一个输入层、若干个隐含层以及一个输出层，它的本质就是一种多层前馈神经网络。BP神经网络相当于一个黑盒子，通过大量的数据输入进行训练，不断的调整权值与阈值，从而达到所要的目的。BP神经网络中各层之间是通过全连接的方式构成一定的网络结构，因此各层神经元只能接受下层神经元送来的信号，并向多层映射网传递修正误差及反馈，也正是这一结构特点，使得BP神经网络是目前应用最广的一种神经网络。

这里我们采用两层BP神经网络(如图 ？？？)进行求解，为使得结果更加的精确，我们采用动量BP算法(momentum backpropagation,MOBP)进行计算。

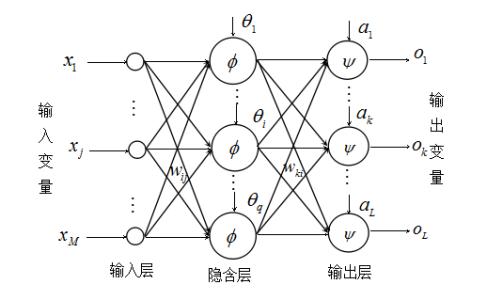


图2. 两层BP神经网络分析图

* 动量BP算法(momentum backpropagation,MOBP)

动量BP算法是在梯度下降算法的基础上引入动量因子(0<):

­

（12）

该算法是以前一次的修正结果来影响本次修正量，当前一次的修正量过大时，式第二项的符号将与前一次修正量的符号相反，从而使本次的修正量减小，起到减小振荡的作用；当前一次的修正量过小时，第二项的符号将与前一次的修正量的符号相同，从而使本次的修正量增大，起到迅速加速修正的作用。可以看出，动量BP算法，总是力图使在同一梯度方向上的修正量增加。动量因子越大，同一梯度方向上的“动量”也越大。

从理论上来说，多层BP网络可以实现任意可实现的线性和非线性函数的映射，克服了感知器和线性神经网络的局限性。但在实际应用中，BP网络往往在训练过程中，也可能找不到某个具体问题的解，比如在训练过程中陷入局部最小的情况。尝试过各种训练函数以及变换隐含层神经元数目后，其精度依旧不能达到我们的要求，甚至相差很大。得出结果如图3。

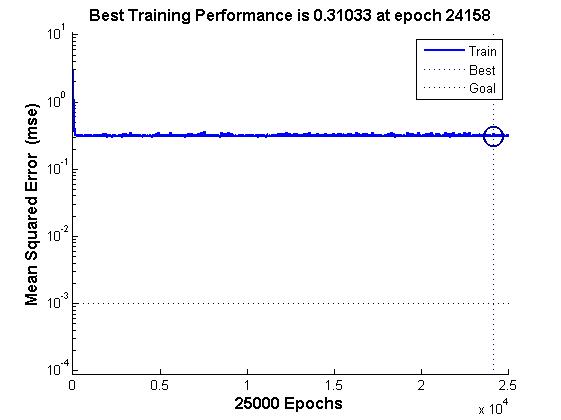


图3.误差分析图

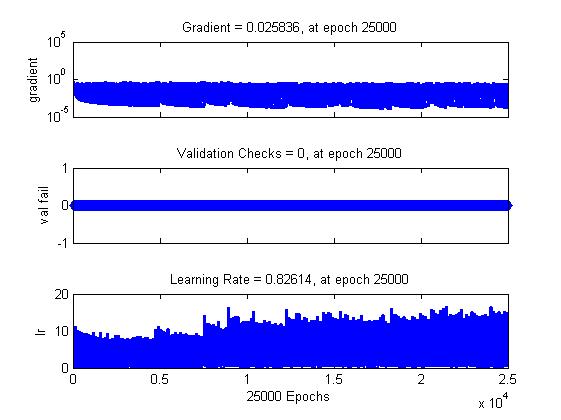


图4.训练过程图

由图4可知，训练达到最大步长时，均方误差依旧稳定在0.930左右，可知训练结果是不够精确地。

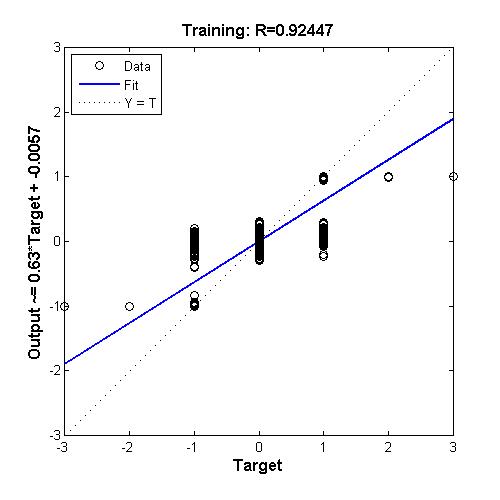


图5.训练过程图

得到原始样本数据与神经网络预测之后的数据进行对比，如图5，由上图可以看出，相差较大。

1. 遗传算法优化的BP神经网络模型

由于单纯的使用BP神经网络误差较大，经过查阅资料，我们采用遗传算法优化的BP神经网络。它是一种全局概率搜索模型，此模型很好的解决了BP神经网络存在局部最优解的缺点。此模型通过遗传算法来寻找BP神经网络初始的权值和阈值，然后再利用BP神经网络进行迭代求解。显而易见，此模型分为两部分，遗传算法和BP神经网络，BP神经网络在上文已经叙述的很详细了，在此不过多叙述，主要对遗传算法进行介绍。

①遗传算法基本操作

在第二问主成分分析的基础上，通过七个主成分变量对数据进行神经网络训练从而完成航食配餐的预测。这里我们采用遗传算法优化bp神经网络，优化BP神经网络的初始权值和阈值，使用优化后的网络更好的预测。

* 种群初始化

个体编码方法为实数编码，每隔个体均为一个实数串，由输入层和隐含层连接权值、隐含层阈值、隐含层与输出层连接权值以及输出层阈值4个部分组成。个体包含了神经网络全部的权值和阈值，在网路结构一直的情况下，就可以构成一个结构、权值、阈值确定的神经网络。

* 适应度函数

根据个体得到BP神经网络的初始权值和阈值，用训练数据训练BP神经网络后预测系统输出，把预测输出个期望的输出之间的误差绝对值和E作为个体适应度值F计算公式为

（13）

式中，n为网络输出节点数；yi为BP神经网络第i个节点的期望输出oi为i个节点的实际输出；k为系数。

* 选择操作

遗传算法选择操作有轮盘赌法、锦标赛法等多种方法。选择轮盘赌法时， 即基于适应度比例的选择策略，每个个体i的选择概率

（14）

式中，Fi为个体i的适应度值，由于适应度值越小越好，所以在个体选择前对适应度求倒数；k为系数；N为种群个体数目。

* 交叉操作

由于个体采用实数编码，所以交叉操作方法采用实数交叉法，第k个染色体和第l个染色体在j为的交叉操作方法如下：

（15）

式中，b是[0,1]之间的随机数。

* 变异操作

选取第i个体的第j个基因进行变异变异操作方法如下：

（16）

式中，为基因的上界；为基因的下界；是 一个随机数；g为当前迭代次数；是最大的进化次数；r为[0,1]间的随机数。

总结以上模型得出基于遗传算法优化的BP神经网络算法流程图，如图2.

数据预处理

Bp神经网络训练得误差作为适应度值

对初始值编码

选择操作

交叉操作

变异操作

满足结束条件

确定网络拓扑结构结构

初始BP神经网络权值阈值

获取最优权值阈值

权值阈值更新

计算误差

满足结束条件

遗传算法部分

Bp神经网络部分

输入数据

图2.遗传算法优化BP神经网络流程图

**5.3.2模型的求解**

利用问题二中由主成分分析所得的五个主成分数据进行遗传算法优化的神经网络训练，其中由遗传算法的选择、交叉、变异所得的部分最优染色体如下表(表4)所示以及遗传算法图。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 1 | 2 | 3 | 4 | 5 | 6 |
| 系数 | -0.39861 | -0.61879 | 0.43604 | -0.30226 | 0.42923 | 0.46188 |
| 序号 | 7 | 8 | 9 | 10 | 11 | 12 |
| 系数 | -2.8656 | -3 | -0.34586 | 1.1117 | 0.24596 | 1.6379 |
| 序号 | 13 | 14 | 15 | 16 | 17 | 18 |
| 系数 | 0.08470 | -1.251 | 0.42061 | -0.56588 | 0.6186 | -0.1998 |
| 序号 | 19 | 20 | 21 | 22 | 23 | 24 |
| 系数 | 0.049775 | 3 | -1.3591 | 0.56947 | 1.4755 | 1.3229 |
| 序号 | 25 | 26 | 27 | 28 | 29 | 30 |
| 系数 | 1.3435 | -0.12045 | -0.14228 | -0.06916 | -1.0737 | -1.8486 |

表4. 最优染色体部分数据表

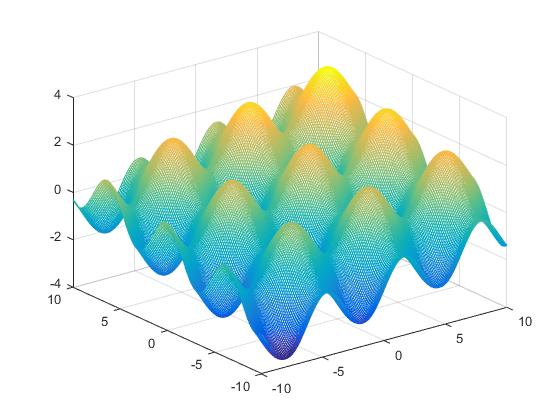


图3.遗传算法图

经过大量的数据训练后，训练出的神经网络误差已经极大的减小(图4)，其误差已经小于0.0005。

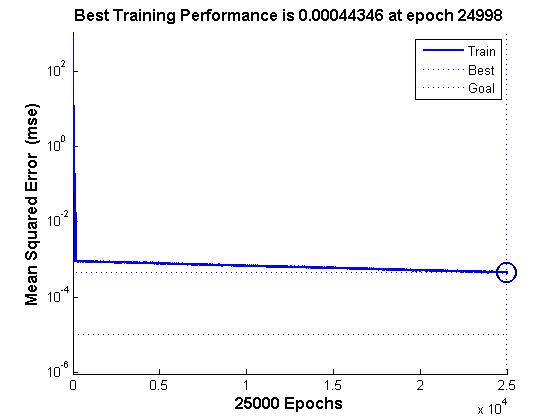


图4. 神经网络误差

**5.3.3模型的可行性分析与评价**

经过大量的数据训练后，对训练出的BP神经网络进行测试，得到如下结果(图5)，从图中可以清晰的观察出蓝线与红线基本吻合(蓝线为实际测试结果，红线为神经网络预测后的结果)。

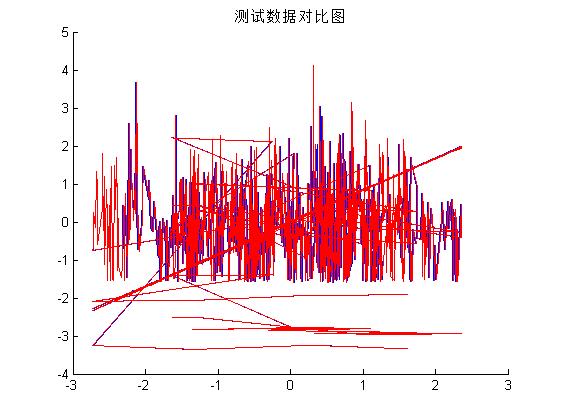
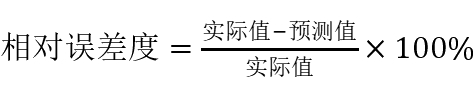


图5. BP神经网络测试对比图

为了更好的描述实际配餐量与预测配餐量相对的误差变化，将实际值与预测值的差在实际值中所占的百分比命名为相对误差度，该值越低代表误差越小。



我们使用训练出来的神经网络对随机地10组值进行了预测，并求得相对误差度，结果如下表(表5)所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 航班号 | 实际人数 | 预测人数 | 相对误差度(%) |
| 2016/1/2 | 8160 | 130 | 129.998378 | 8.94\*10^-3 |
| 2016/1/5 | CA1801 | 84 | 93.9924905 | 8.92\*10^-3 |
| 2016/1/28 | MU5720 | 110 | 109.98966 | 8.94\*10^-3 |
| 2016/3/2 | CA1657 | 6 | 5.9997 | 5.12\*10^-2 |
| 2016/3/30 | VN513 | 146 | 145.98686 | 9.45\*10^-3 |
| 2016/4/25 | CA1591 | 134 | 135.00808 | 6.03\*10^-3 |
| 2016/6/6 | CA1827 | 130 | 131.00001 | 7.69\*10^-2 |
| 2016/8/12 | PK852 | 84 | 83.98032 | 9.33\*10^-3 |
| 2016/10/7 | MU5102 | 270 | 270.67121 | 6.02\*10^-3 |
| 2016/10/17 | CA1461 | 151 | 151.00089 | 1.08\*10^-3 |

表5. 神经网络训练相对误差

通过上表可知，相对误差度<0.0001,由此可以得出我们所创建的基于遗传算法的BP神经网络模型可行并且有相对较高的准确度。

**5.4问题四**

**5.4.1模型的建立**

1、数据标准化

将各个指标数据进行标准化处理假设给定k个指标，其中，假设对各指标数据标准化后的值为，那么

（17）

2、求各指标的信息熵

根据信息论中信息熵的定义，一组数据的信息熵,其中

（18）

3、确定各指标权重

根据信息熵的计算公式，计算出各个指标信息熵为，通过信息熵计算各指标的权重

（19）

**5.4.2模型的求解**

将主成分数据按照求解其其信息熵，因为在选定主成分时我们已经对原变量做了标准化处理，因此不需要在主成分数据中再多做一次。

接下来计算信息熵冗余度

（20）

最后进行各项指标权值的计算

（21）

最终得到各主成分的权重为

www1 = 0.196099626096782

www2 = 0.177209729910743

www3 = 0.237233536969838

www4 = 0.158673134149467

www5 = 0.230783972873169

由结果可知第三个和第五个主成分对人数影响的贡献比较大，而又可以由成分的得矩阵来分析各原变量对当前主成分的得分。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.71 | -0.08 | 0.03 | 0.03 | 0.03 |
| -0.01 | 0.03 | -0.34 | -0.35 | -0.41 |
| -0.29 | 0.59 | 0.22 | 0.23 | 0.27 |
| -0.43 | -0.43 | -0.06 | -0.07 | -0.08 |
| 0.12 | 0.67 | -0.1 | -0.1 | -0.12 |
| -0.03 | -0.08 | 0.9 | 0.91 | 1.07 |
| 0.71 | -0.05 | 0.07 | 0.07 | 0.08 |

表6. 主成分得分矩阵

由图6可知，影响第三个和第五个主成分的主要因变量是星期，因此我们认为在进行航食配餐时，先将星期作为首要考虑因素，再结合遗传算法改进的BP神经网络进行预测。

**5.4.3结果分析**

由熵权值法确定了各主成分的贡献比，又据此分析出原变量对配餐影响较大的因素，由变量的权重确定考虑的优先级，结果准确可信。

但是我们发现，熵权值法在使用过程中，如果指标值得变动很小或者很突然的变大变小，熵权值法求解就有一定的局限性。

1. 模型的优缺点与推广

**6.1模型的优缺点**

**6.1.1模型的优点**

本题在数据的预处理阶段，采用聚类分析对数据异常的情况进行识别和改正，避免了一概而论剔除数据导致的数据丢失而引起的结果偏差。对于问题二，我们采用灰色关联分析，将关联性较强的自变量进行整合，减小了主成分分析的数据处理偏差，利用熵值法对各主成分加权，从而得出科学的关联因素分析的方法。有效的总结了和航班订单因素相关的整体性因素。对于第三问模型，在bp神经网络的基础上，我们采用遗传算法优化，给定bp神经网络的初始值和阈值，提高了神经网络预测的精准性。

**6.1.2模型的缺点**

对于问题一数据错误的识别存在盲区，不能全面的识别数据中的错误，只能简要的找出大部分的逻辑错误，之后对其进行修正和剔除。对于问题二，我们采用主成分分析，此方法会导致部分数据的丢失，对结果会造成影响；再者在分析的过程中第三第四第五主成分都是和温度极大相关，此处存在偏差。对于问题三，我们采用bp神经网络的方法进行预测，其中训练的变量为第二问分析出来的五个主成分，对于此自变量本身存在一定误差，座位自变量训练会加大误差影响训练数据的误差。

**6.2模型的推广**

该题所建立的四个模型可以针对各个交通工具的配餐问题的解决，同时基于遗传算法的bp神经网络模型可以解决预测问题。四个模型环环相扣，可以被推广到各行业的配餐问题研究中。

参考文献

[1] [刀客123](https://blog.csdn.net/dingyahui123),遗传算法优化BP神经网络,https://blog.csdn.net/dingyahui123/article/details/72831102

[2]百度文库，基于遗传算法的BP神经网络的应用，https://wenku.baidu.com/view/ab7987eeb8d528ea81c758f5f61fb7360b4c2bcd.html

[3]王崇骏，于汶，陈兆乾，谢俊元，一种基于遗传算法的BP神经网络算法及其应用，3、4页，2002

附录

附件一：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 雷阵雨 | 20 | | 小雨~中雨 | 10 | 多云~小雨 | | 30 | 晴 | | | 39 |
| 阵雨~多云 | 19 | | 小雨~大雨 | 9 | 多云~中雨 | | 29 | 晴~多云 | | | 38 |
| 阵雨~阴 | 18 | | 中雨~小雨 | 8 | 多云~霾 | | 28 | 晴~阴 | | | 37 |
| 雷阵雨~阵雨 | 17 | | 大暴雨~中雨 | 7 | 阴 | | 27 | 晴~小雨 | | | 36 |
| 小雨~晴 | 16 | | 霾 | 6 | 阴~晴 | | 26 | 晴~霾 | | | 35 |
| 小雨~阴 | 15 | | 霾~多云 | 5 | 阴~多云 | | 25 | 多云 | | | 34 |
| 小雨 | 14 | | 霾~晴 | 4 | 阴~小雨 | | 24 | 多云~晴 | | | 33 |
| 大雨~多云 | 13 | | 霾~阵雨 | 3 | 阴~阵雨 | | 23 | 多云~阴 | | | 32 |
| 小雨~多云 | 12 | | 霾~雷阵雨 | 2 | 阴~雷阵雨 | | 22 | 多云~雷阵雨 | | | 31 |
| 小雨~小到中雨 | 11 | | 霾~雾 | 1 | 阵雨 | | 21 |
| 东南风1-2级 | | 3 | | | | 东北风微风 | | | | 4 | | |
| 东风1-2级 | | 3 | | | | 东南风微风 | | | | 4 | | |
| 北风1-2级 | | 3 | | | | 东风微风 | | | | 4 | | |
| 南风1-2级 | | 3 | | | | 北风微风 | | | | 4 | | |
| 西北风1-2级 | | 3 | | | | 南风微风 | | | | 4 | | |
| 西南风1-2级 | | 3 | | | | 无持续风向微风 | | | | 4 | | |
| 西风1-2级 | | 3 | | | | 西北风微风 | | | | 4 | | |
| 东风3-4级 | | 2 | | | | 西南风微风 | | | | 4 | | |
| 北风3-4级 | | 2 | | | | 西风微风 | | | | 4 | | |
| 南风3-4级 | | 2 | | | | 东北风1-2级 | | | | 3 | | |
| 北风4-5级 | | 1 | | | | 良 | | | 5 | | | | |
| 严重污染 | | 2 | | | | 轻度污染 | | | 4 | | | | |
| 中度污染 | | 3 | | | | 重度污染 | | | 1 | | | | |
| 优 | | 6 | | | |

附件二：第二问代码

function [ ] = shujuzhuanhuan( )

clear;

clc;

tic

fileID='C:\Users\logan\Desktop\B题 航食数据\test\airplane\_order\_test.csv';

file=fopen(fileID,'r');

text=textscan(file,'%s',1);

data=textscan(file,'%d %s %d %s %d %d %d %d %d %d %d', 'delimiter', ',');

fclose(file);

% datenum(data{4}{1})%日期转换成数字

% datenum('2016/11/2')

dateStart=datenum('2016/11/1');

dateEnd=datenum('2016/12/31');

date=dateStart:dateEnd;

% datestr(736695,26);%数值转换成日期

HangBan=data{2};

HangBanDeal=unique(HangBan);

[m,n]=size(HangBanDeal);

out=zeros(m,3);

x=1;

y=1;

d=-1;

for j=2:m

x=1;

d=d+1;

for k=dateStart:dateEnd

a=find(strcmp(HangBan,HangBanDeal{j}));

data1=data{4}(a);

b=find(strcmp(data1,datestr(k,29)));

outpre1=sum(data{5}(a(b)));

outpre2=sum(data{6}(a(b)));

outpre3=sum(data{7}(a(b)));

y=1+d;

out(x,y)=outpre1;

y=y+1+d;

out(x,y)=outpre2;

y=y+1+d;

out(x,y)=outpre3;

x=x+1;

% data{5}(a(b))

xlswrite('C:\Users\logan\Desktop\1.xls',out);

end

end

toc

end

function [  ] = julei(  )  
%针对数据中总数写反进行聚类分析  
clear;  
clc;  
tic  
fileID='C:\Users\lenovo\Desktop\airplane\_order\_train.csv';  
file=fopen(fileID,'r');  
text=textscan(file,'%s',1);  
data=textscan(file,'%d %s %d %s %d %d %d %d %d %d %d', 'delimiter', ',');  
fclose(file);  
HangBan=data{2};  
HangBanDeal=unique(HangBan);  
[m,n]=size(HangBanDeal);  
% for i=2:size(HangBanDeal)  
n=2000;  
set(0,'recursionlimit',n);  
for i=2:m

a=find(strcmp(HangBan,HangBanDeal(i)));  
    for k=1:a  
        if ~data{8}(k)    
        data{8}(k)=0;    
        end  
         if ~data{9}(k)  
        data{9}(k)=0;  
         end  
         if ~data{10}(k)  
              data{10}(k)=0;  
          end  
         if ~data{11}(k)  
            data{11}(k)=0;  
         end  
    end  
    Tou=data{8}(a);  
     Gong=data{9}(a);  
    Jing=data{10}(a);  
    Qi=data{11}(a);  
  
%%  
x=[Tou Gong Jing Qi];    %聚类修改总数中异常数据  
x=double(x);  
bx=zscore(x);  
y=pdist(bx);  
d=squareform(y);  
z=linkage(y,'average');  
ZY=cophenet(z,y)  
t=cluster(z,2);  
%t=clusterdata(x,2);  
aaaa=find(t==2);  
bbbb=find(t==1);  
for k=1:11                %找到并赋值  
    data{k}(aaaa)=data{k}(bbbb(1));  
end  
[h,t]=dendrogram(z,100);  
aaa=strcat('C:\Users\lenovo\Desktop\111\',num2str(i));  
saveas(gcf,strcat(aaa,'.jpg'));  
%  
    b=0:max(Tou);  
    c=histc(Tou,b);%数据中数的出现次数  
    [max\_num,max\_index]=max(c);  
    data{8}(a)=max\_index;  
      
    b=0:max(Gong);  
    c=histc(Gong,b);%数据中数的出现次数  
    [max\_num,max\_index]=max(c);  
    data{9}(a)=max\_index;  
      
    b=0:max(Jing);  
    c=histc(Jing,b);%数据中数的出现次数  
    [max\_num,max\_index]=max(c);  
    data{10}(a)=max\_index;  
      
    b=0:max(Qi);  
    c=histc(Qi,b);%数据中数的出现次数  
    [max\_num,max\_index]=max(c);  
    data{11}(a)=max\_index;  
end  
table={'A1:A768343','B1:B768343','C1:C768343','D1:D768343','E1:E768343','F1:F768343','G1:G768343','H1:H768343','I1:I768343','J1:J768343','K1:K768343'};  
% xlswrite('C:\Users\logan\Desktop\julei.txt',data);  
% for i=1:11  
% xlswrite('C:\Users\logan\Desktop\julei.xlsx',data{i},table{i});  
% end  
toc  
end

%%熵值法

function [ ] = shangzhifa( )

% 函数shang.m, 实现用熵值法求各指标(列）的权重及各数据行的得分

% x为原始数据矩阵, 一行代表一个国家, 每列对应一个指标

% s返回各行得分, w返回各列权重

fileID='C:\Users\lenovo\Desktop\lalala.csv';

file=fopen(fileID,'r');

text=textscan(file,'%s',1);

data=textscan(file,'%d %d %d %d %d %d %d %d %d %d %d %d %d %d', 'delimiter', ',');

fclose(file);

%%

format rational;

x=[data{9} data{10} data{11} data{12} data{13}];

[n,m]=size(x); % n=23个国家, m=5个指标

%% 数据的归一化处理

% Matlab2010b,2011a,b版本都有bug,需如下处理. 其它版本直接用[X,ps]=mapminmax(x',0,1);即可

[X,ps]=mapminmax(x');

ps.ymin=0.002; % 归一化后的最小值

ps.ymax=0.996; % 归一化后的最大值

ps.yrange=ps.ymax-ps.ymin; % 归一化后的极差,若不调整该值, 则逆运算会出错

X=mapminmax(x',ps);

% mapminmax('reverse',xx,ps); % 反归一化, 回到原数据

X=X'; % X为归一化后的数据, 23行(国家), 5列(指标)

%% 计算第j个指标下，第i个记录占该指标的比重p(i,j)

for i=1:n

for j=1:m

p(i,j)=X(i,j)/sum(X(:,j));

end

end

%% 计算第j个指标的熵值e(j)

k=1/log(n);

for j=1:m

e(j)=-k\*sum(p(:,j).\*log(p(:,j)));

end

d=ones(1,m)-e; % 计算信息熵冗余度

w=d./sum(d) % 求权值w

s=w\*p'; % 求综合得分[\code]

end

附件三：第二问

%%  打开文件  
fileID='C:\Users\lenovo\Desktop\duoyiisufenxi\不去除变量的主成分.csv';  
file=fopen(fileID,'r');  
text=textscan(file,'%s',1);  
data=textscan(file,'%d %d %d %d %d %d %d %d', 'delimiter', ',');  
fclose(file);  
%%  灰色关联分析  
for i=1:7        %总循环，每个自变量均比较一次  
    x0=[data{8} data{i}];  
    for j=2:length(x0(:,1))         %斜率序列  
        x1(j,:)=x0(j,:)-x0(j-1,:);  
    end  
    x1=double(x1);  
    m=length(x1(1,:));  
    for j=1:m                      %标准化  
        x2(:,j)=x1(:,j)/std(x1(:,j));  
    end  
    %a排序  
    [y,pos]=sort(x2(:,1));  
    x2\_sorted=x2(pos,:);  
    %判定关联性质  
    n=length(x1(:,1));  
    k=[1:n]';  
    for j=1:m  
        sig\_j(j)=sum(k.\*x2\_sorted(:,j))-sum(x2\_sorted(:,j))\*sum(k)/n;  
    end  
    for j=2:m  
        dist\_0i(:,j)=abs(sign(sig\_j(:,j)./sig\_j(:,1)).\*x2\_sorted(:,j)-x2\_sorted(:,1));  
    end  
    for j=1:n  
        for k=1:m  
            coef\_rela(j,k)=(min(dist\_0i)+0.5\*max(dist\_0i))/(dist\_0i(j,k)+0.5\*max(dist\_0i));  
        end  
    end  
    for j=1:m  
        output(j)=sum(coef\_rela(:,j))/n  
    end  
end

附件四

①神经网络

fileID='C:\Users\lenovo\Desktop\lalala.csv';  
file=fopen(fileID,'r');  
text=textscan(file,'%s',1);  
data=textscan(file,'%d %d %d %d %d %d %d %d %d %d %d %d %d %d', 'delimiter', ',');  
fclose(file);  
for i=9:13  
    data{i}=double(data{i});  
    data{i}=zscore(data{i});  
end  
%%  
p=[data{9} data{10} data{11} data{12} data{13}]';  
t=[data{14}]';  
q=[-1 1;-1 1;-1 1;-1 1;-1 1];  
net=newff(q,[20,1],{'tansig','tansig'},'traingdx');  
net.trainParam.goal=0.001;  
net.trainParam.epochs=25000;  
[net,tr]=train(net,p,t);

②基于遗传算法的BP神经网络

clc

clear

tic

%% 网络结构建立

%读取数据

% load data input output

%节点个数

inputnum=5;

hiddennum=10;

outputnum=1;

%训练数据和预测数据

fileID='C:\Users\logan\Desktop\lalala.csv';

file=fopen(fileID,'r');

text=textscan(file,'%s %s %s %s %s %s %s %s %s %s %s %s %s',1);

data=textscan(file,'%f %f %f %f %f %f %f %f %f %f %f %f %f %f', 'delimiter', ',');

fclose(file);

trainnum=1000;

testnum=trainnum+10;

input\_train=[data{9}(1:trainnum) data{10}(1:trainnum) data{11}(1:trainnum) data{12}(1:trainnum) data{13}(1:trainnum)]';

input\_test=[data{9}(trainnum+1:testnum) data{10}(trainnum+1:testnum) data{11}(trainnum+1:testnum) data{12}(trainnum+1:testnum) data{13}(trainnum+1:testnum)]';

output\_train=data{14}(1:trainnum)';

output\_test=data{14}(trainnum+1:testnum)';

%选连样本输入输出数据归一化

inputn=input\_train;

outputn=output\_train;

%构建网络

net=newff([-5 5;-5 5;-5 5;-5 5;-5 5;],[hiddennum 1],{'purelin','purelin'},'traingdx');

x=yichuansuanfa(inputnum,hiddennum,outputnum,net,inputn,outputn);

%% 把最优初始阀值权值赋予网络预测

% %用遗传算法优化的BP网络进行值预测

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网络训练

%网络进化参数

net.trainParam.epochs=25000;

net.trainParam.lr=0.1;

net.trainParam.goal=0.00001;

net.trainParam.showWindow=1;

%网络训练

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

save E53net net;

toc

%% BP网络预测

%数据归一化

% inputn\_test=mapminmax('apply',input\_test,inputps);

an=sim(net,input\_test);

test\_simu=mapminmax('reverse',an,outputps);

error=test\_simu-output\_test;

function [chrom\_best] = yichuansuanfa(inputnum,hiddennum,outputnum,net,inputn,outputn )

%%基础参数

N = 100; %种群内个体数目

N\_chrom = inputnum\*hiddennum+hiddennum+hiddennum\*outputnum+outputnum; %染色体节点数

iter = 50; %迭代次数

mut = 0.2; %突变概率

acr = 0.2; %交叉概率

best = 1;

chrom\_range = [-3\*ones(N\_chrom,1) 3\*ones(N\_chrom,1)]'; %每个节点的值的区间

chrom = zeros(N, N\_chrom);%存放染色体的矩阵

fitness = zeros(N, 1);%存放染色体的适应度

fitness\_ave = zeros(1, iter);%存放每一代的平均适应度

fitness\_best = zeros(1, iter);%存放每一代的最优适应度

chrom\_best = zeros(1, N\_chrom+1);%存放当前代的最优染色体与适应度

%%初始化

chrom = Initialize(N, N\_chrom, chrom\_range); %初始化染色体

for i=1:N

fitness(i)= CalFitness(chrom,inputnum,hiddennum,outputnum,net,inputn,outputn); %计算适应度

end

chrom\_best = FindBest(chrom, fitness, N\_chrom); %寻找最优染色体

fitness\_best(1) = chrom\_best(end); %将当前最优存入矩阵当中

fitness\_ave(1) = CalAveFitness(fitness); %将当前平均适应度存入矩阵当中

'jinxing1'

for t = 2:iter

chrom = MutChrom(chrom, mut, N, N\_chrom, chrom\_range, t, iter); %变异

chrom = AcrChrom(chrom, acr, N, N\_chrom); %交叉

for i=1:N

fitness(i)= CalFitness(chrom,inputnum,hiddennum,outputnum,net,inputn,outputn); %计算适应度

end

chrom\_best\_temp = FindBest(chrom, fitness, N\_chrom); %寻找最优染色体

if chrom\_best\_temp(end)>chrom\_best(end) %替换掉当前储存的最优

chrom\_best = chrom\_best\_temp;

end

%%替换掉最劣

[chrom, fitness] = ReplaceWorse(chrom, chrom\_best, fitness);

fitness\_best(t) = chrom\_best(end); %将当前最优存入矩阵当中

fitness\_ave(t) = CalAveFitness(fitness); %将当前平均适应度存入矩阵当中

'jinxing2'

end

%%作图

figure(1)

plot(1:iter, fitness\_ave, 'r', 1:iter, fitness\_best, 'b')

grid on

legend('平均适应度', '最优适应度')

e = PlotModel(chrom\_best)

%%输出结果

disp(['最优染色体为', num2str(chrom\_best(1:end-1))])

disp(['最优适应度为', num2str(chrom\_best(end))])

end

function error = CalFitness(x,inputnum,hiddennum,outputnum,net,inputn,outputn)

%该函数用来计算适应度值

%x input 个体

%inputnum input 输入层节点数

%outputnum input 隐含层节点数

%net input 网络

%inputn input 训练输入数据

%outputn input 训练输出数据

%error output 个体适应度值

%提取

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=newff(inputn,outputn,hiddennum);

net=newff([-5 5;-5 5;-5 5;-5 5;-5 5;],[hiddennum 1],{'purelin','purelin'},'traingdx');

%网络进化参数

net.trainParam.epochs=100;

net.trainParam.lr=0.1;

net.trainParam.goal=0.00001;

net.trainParam.show=100;

net.trainParam.showWindow=0;

%网络权值赋值

%%

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;

%网络训练

net=train(net,inputn,outputn);

an=sim(net,inputn);

% an

% outputn

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

function chrom\_new = Initialize(N, N\_chrom, chrom\_range)

chrom\_new = rand(N, N\_chrom);

for i = 1:N\_chrom %每一列乘上范围

chrom\_new(:, i) = chrom\_new(:, i)\*(chrom\_range(2, i)-chrom\_range(1, i))+chrom\_range(1, i);

end

function chrom\_new = AcrChrom(chrom, acr, N, N\_chrom)

for i = 1:N

acr\_rand = rand;

if acr\_rand<acr %如果交叉

acr\_chrom = floor((N-1)\*rand+1); %要交叉的染色体

acr\_node = floor((N\_chrom-1)\*rand+1); %要交叉的节点

%交叉开始

temp = chrom(i, acr\_node);

chrom(i, acr\_node) = chrom(acr\_chrom, acr\_node);

chrom(acr\_chrom, acr\_node) = temp;

end

end

chrom\_new = chrom;

function fitness\_ave = CalAveFitness(fitness)

[N ,~] = size(fitness);

fitness\_ave = sum(fitness)/N;

function chrom\_best = FindBest(chrom, fitness, N\_chrom)

chrom\_best = zeros(1, N\_chrom+1);

[maxNum, maxCorr] = max(fitness);

chrom\_best(1:N\_chrom) =chrom(maxCorr, :);

chrom\_best(end) = maxNum;

function c\_new = IfOut(c, range)

if c<range(1) || c>range(2)

if abs(c-range(1))<abs(c-range(2))

c\_new = range(1);

else

c\_new = range(2);

end

else

c\_new = c;

end

function chrom\_new = MutChrom(chrom, mut, N, N\_chrom, chrom\_range, t, iter)

for i = 1:N

for j = 1:N\_chrom

mut\_rand = rand; %是否变异

if mut\_rand <=mut

mut\_pm = rand; %增加还是减少

mut\_num = rand\*(1-t/iter)^2;

if mut\_pm<=0.5

chrom(i, j)= chrom(i, j)\*(1-mut\_num);

else

chrom(i, j)= chrom(i, j)\*(1+mut\_num);

end

chrom(i, j) = IfOut(chrom(i, j), chrom\_range(:, j)); %检验是否越界

end

end

end

chrom\_new = chrom;

function y = PlotModel(chrom)

x = chrom(1);

y = chrom(2);

z = chrom(3);

figure(2)

scatter3(x, y, z, 'ko')

hold on

[X, Y] = meshgrid(-10:0.1:10);

Z =sin(X)+cos(Y)+0.1\*X+0.1\*Y;

mesh(X, Y, Z)

y=1;

function [chrom\_new, fitness\_new] = ReplaceWorse(chrom, chrom\_best, fitness)

max\_num = max(fitness);

min\_num = min(fitness);

limit = (max\_num-min\_num)\*0.2+min\_num;

replace\_corr = fitness<limit;

replace\_num = sum(replace\_corr);

chrom(replace\_corr, :) = ones(replace\_num, 1)\*chrom\_best(1:end-1);

fitness(replace\_corr) = ones(replace\_num, 1)\*chrom\_best(end);

chrom\_new = chrom;

fitness\_new = fitness;

end