中国研究生创新实践系列大赛

“华为杯”第十八届中国研究生数学建模竞赛

学　　校	贵州大学
参赛队号	**21106570100**
队员姓名	**1.** 漆浪浪
	2. 赵继露
	3. 赵坤

中国研究生创新实践系列大赛
“华为杯”第十八届中国研究生
数学建模竞赛

题 目　　空气质量预报二次建模

摘　　要：

大气污染是指大气中污染物质的浓度达到有害程度，以至破坏生态系统和人类正常生存和发展的条件，对人和物造成危害的现象。运用空气质量预报模型可在大气污染发生之前采取相应的控制措施，是减少大气污染对人体健康和生态环境造成影响的有效方法之一。目前常用的 WRF-CMAQ 模拟体系由于 WRF 模拟的气象场数据与大气污染中部分污染物生成机理的不确定性，导致 WRF-CMAQ 模拟体系的预报结果不理想，因此，如何在 WRF-CMAQ 模拟体系的基础上，结合实测数据对一次预报模型进行二次建模，在 WRF-CMAQ 模拟体系的一次预报数据优化方面具有重要意义。

针对问题一，结合附录的 AQI 计算方法，计算出监测点 A 从 2020 年 8 月 25 日至 8 月 28 日每天实测的 AQI 和首要污染物。主要通过计算各污染物的空气质量分指数 IAQI，并以各污染物中 IAQI 最大值作为当天的首要污染物。

针对问题二，首先分析得出一次预报数据和实测数据具有明显的差异性，之后针对其差异采用不同的方式完成气象条件的分类。对于实测数据，因其污染物浓度较为准确，故通过搭建 AQI 预测模型的方式计算出实测数据的 AQI 值。这样就将求气象条件对于各类污染物浓度影响问题简化，这样做是因为每个污染物浓度受到所有气象条件以及其他各种因素的共同影响，并且这种影响难以消除，既然所有的污染物都被多种因素影响，那么相比于多次求解各类污染物和气象条件的影响，求解 AQI 和气象条件的关系要简便很多。对于 AQI 预测模型本文采用青悦公开数据提供的气象数据集完成。最后使用 3 天为一个时间周期，采用周期做差比较的方式来完成数据分类。对于一次预测数据，因其计算 AQI 的精度不能保证，所以本文采用灰色关联分析来计算每类污染物的主要气象影响因素完成分类。

针对问题三，首先对一次预报数据处理，删除准确性较差的后两天预测数据，保留预测准确率较高的第一天预测数据，其次将一次预报数据与逐小时实测数据按时间对于拼接起来，并对两者污染物浓度做差，以差值与各气象条件作为新的数据集，建立实测数据与一次预测数据的联系。在对数据分析处理之后，利用灰色关联度分析找出影响污染物浓度的主要气象条件，以此为输入训练神经网络，并通过混沌反向初始化 CS 算法优化神经网络，最终进行 AQI 预测和首要污染物判断。

针对问题四，在问题三解题思路的基础上，分别对 A、A1、A2、A3 四个监测点进行建模，采用问题三相同的方法完成数据预处理与建立数据间的关联，之后使用灰色关联度找出关联度较大的因素作为神经网络的输入，再使用集成学习 Adaboost 算法将四个监测点得到的弱学习器构建为最终的强学习器，使用该强学习器对 A、A1、A2、A3 四个监测点在 2021 年 7 月 13 日至 7 月 15 日的 6 种污染物浓度值进行预测。为保证预测的准确性，将一天的 24 条数据进行负值判断、去最大最小操作，剩余数据取均值得到最终预测结果。

关键词：灰色关联度分析；　混沌反向初始化 CS 算法；　Adaboost 算法；

目 录

一. 问题重述

1.1 问题背景

大气污染系指由于人类活动或自然过程引起某些物质进入大气中，呈现足够的浓度，达到了足够的时间，并因此危害了人体的舒适、健康和福利或危害了生态环境。根据《环境空气质量标准》（GB3095-2012），用于衡量空气质量的常规大气污染物共有六种，分别为二氧化硫（SO_2）、二氧化氮（NO_2）、粒径小于 10μm 的颗粒物（PM10）、粒径小于 2.5μm 的颗粒物（PM2.5）、臭氧（O_3）、一氧化碳（CO）。其中臭氧并不来自于污染源的直接排放，而是在大气中经过一系列的化学及光化学反应生成，因此臭氧污染也是最难预报的一种大气污染。

大气污染防治是指在一个特定区域内，把大气环境看作一个整体，统一规划能源结构、工业发展、城市建设布局等，综合运用各种防治污染的技术措施，充分利用环境的自净能力，以改善大气质量。大气污染防治实践表明，通过空气质量预报模型预测可能发生的大气污染过程并采取相应的措施，是降低大气污染危害，提高环境空气质量的有效方法之一。

空气质量模型是基于人类对大气物理和化学过程科学认识的基础上，运用气象学原理及数学方法，从水平和垂直方向在大尺度范围内对空气质量进行仿真模拟，再现污染物在大气中输送、反应、清除等过程的数学工具[2]。

第一代空气质量模型主要包括了基于质量守恒定律的箱式模型、基于湍流扩散统计理论的高斯模型和拉格朗日轨迹模式。以 Pasquill 和 Gifford 等研究者得出的离散不同稳定度条件下的大气扩散参数曲线[3]和 Pasquill 方法确定的扩散参数为基础，采用简单的、参数化的线性机制描述复杂的大气物理过程，适用于模拟惰性污染物的长期平均浓度。

第二代空气质量模式在设计上仅考虑了单一的大气污染问题，对于各污染物间的相互转化和相互影响考虑不全面，而实际大气中各种污染物之间存在着复杂的物理、化学反应过程。因此，20 世纪 90 年代末美国环保局基于“一个大气”理念，设计研发了第三代空气质量模式系统 Medels-3/CMAQ[4]，CMAQ 是一个多模块集成、多尺度网格嵌套的三维欧拉模型，突破了传统模式针对单一物种或单相物种的模拟，考虑了实际大气中不同物种之间的相互转换和互相影响，开创了模式发展的新理念。

目前常用的空气质量预报模型为 WRF-CMAQ 模型，该模型主要包括 WRF 和 CMAQ 两个系统。WRF 是一种中尺度数值天气预报系统，用于为 CMAQ 提供所需的气象场数据，CMAQ 根据 WRF 提供的气象场数据及该场域内的污染排放清单得到具体时间地点的空气质量预报结果。但由于模拟的气象场数据和排放清单的不确定性，以及对包括臭氧在内的污染物生成机理的不完全明晰，WRF-CMAQ 模型的预测结果并不理想。因此，如何在 WRF-CMAQ 模型的一次预报模型的基础上再次建模，结合更多的数据提高预报的准确性具有重要意义。

1.2 问题提出

问题一：按照 AQI 计算公式及空气质量分指数（IAQI）及对应的污染物项目浓度限值表计算附件 1 中监测点 A 从 2020 年 8 月 25 日到 8 月 28 日每天实测的 AQI 和首要污染物。

问题二：在污染物排放情况不变的条件下，根据对污染物浓度的影响程度，对气象条件进行合理分类。

问题三：建立一个同时适用于 A、B、C 三个监测点的二次预报数学模型，用来预测未来三天 6 种常规污染物单日浓度值，要求二次预报模型预测结果中 AQI 预报值的最大相对误差

应尽量小，且首要污染物预测准确度尽量高。

问题四：建立包含 A、A1、A2、A3 四个监测点的协同预报模型，要求二次模型预测结果中 AQI 预报值的最大相对误差应尽量小，且首要污染物预测准确度尽量高。使用该模型预测监测点 A、A1、A2、A3 在 2021 年 7 月 13 日至 7 月 15 日 6 种常规污染物的单日浓度值，计算相应的 AQI 和首要污染物。

1.3 问题分析

1.3.1 问题一的分析

根据附录提供的空气质量指数（AQI）的计算方法，使用附录 1 中“监测点 A 逐日污染物浓度实测数据”提供的数据计算 2020 年 8 月 25 日到 8 月 28 日每天实测的 AQI 和首要污染物。

1.3.2 问题二的分析

在污染物排放情况不变的条件下，某一地区的气象条件有利于污染物扩散或沉降时，该地区的 AQI 会下降，反之会上升。在不考虑外部污染物排放的情况下，仅考虑气象条件对该地区的 AQI 的影响。

针对本问，在数据处理上，由于只考虑单一的污染物浓度与气象条件之间的关系无法做到合理的分类，需要结合问题一的 AQI 值进行建模分类，同时，由于数据量的庞大，为简化模型，需先对 AQI 只进行预测处理，在形成新的数据集用以分析各气象条件与 AQI 之间的关系。

1.3.3 问题三的分析

由题意可知，WRF-CMAQ 一次预报模型预测的数据由于 WRF 模型提供的气象场以及污染物的生成机理的不确定性，导致效果不理想。因此在 WRF-CMAQ 等一次预报模型模拟结果的基础上，结合更多的数据源进行再建模，以提高预报的准确性。二次模型优化的 WRF-CMAQ 空气质量预报过程如图 1 所示。

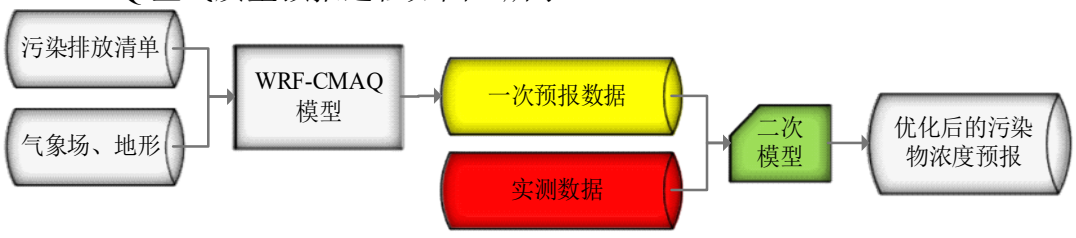


图 1 二次模型优化的 WRF-CMAQ 空气质量预报过程

一般来说，一次预报数据与实测数据相关性不高，但预报过程中常会使用实测数据对一次预报数据进行修正以达到更好的效果。需使用实测数据对一次预报的数据进行优化，如何建立一次预报数据与实测数据之间的联系是本问的重点。

针对本问，因先考虑如何合理的将一次预报数据与实测数据之间建立联系，这里本文采取用实测数据的各个污染物浓度减去一次预报数据的污染物浓度的方式，这样最后模型预测得到的结果就为实测数据减一次预报数据，得到结果后将其加上一次预报数据就得到了预测的实测数据，之后对缺失值和异常值进行分析处理，值得一提的是本文在分别处理了一次预报数据和每小时实测数据的缺失值和异常值后，还在一次预报数据减去实测数据得到模型预测值后，对该相减数据进行了异常值分析和处理，因为我们发现一些数据可能

在单独的数据中因其数据范围正常无法被判别为异常数据，但在进行实测数据减去一次预报数据之后却变为了异常值的情况，这种情况发生的原因是因为一次预报数据与实测数据差距过大，完全不符合正常情况，故对该类一次预测数据和实测数据差距过大的数据进行筛选去除，最终得到可以进行建模的数据集，以保证模型的合理性与科学性。同时，一次预报数据中的气象条件繁杂，为简化模型，应从中挑选关联度较高的若干项属性作为预测的依据，这里文本以第二问的方式为基础，采用灰色关联度分析的方式完成各类污染物的主要影响因素判断。最后通过处理后的数据训练神经网络进行预测，从而达到优化一次预报数据的目的。在经过对一次预报数据和实测数据对比观察后发现一次预报数据误差过大，神经网络得到的效果很可能会收到很大的影响，故考虑采用布谷鸟算法对神经网络进行优化，提升模型效果。对于布谷鸟算法采用的随机初始化生成初始值的方式也进行了优化，通过混沌初始化结合反向初始化的方式来生成在解空间内表达能力更强，分布更为均为的初始值，以此提升布谷鸟算法的寻优效率。

1.3.4 问题四的分析

建立包含 A、A1、A2、A3 四个监测点的协同预报模型。以问题三为基础，在完成了数据的处理之后使用改进的 CS_BP 算法分别对各个监测点进行建模，之后为了达到区域协同预报的效果，采用了 Adaboost 集成学习方式来将各个监测点模型结合起来，共同完成预测污染物浓度的工作。该方式的弱学习器即为各个改进的 CS-BP 模型，根据各个弱学习器的学习误差率得到来迭代的得到训练样本的权重以及弱分类器的权重，在这个过程中将使得之前弱学习器学习误差率高的训练样本权重变高，若后续的弱分类器在同一个样本上再次错误判断，那么将会得到更大的误差，后续的弱分类器则会在错误样本上的权值变低，以此来修正错误预测的样本。最终将这四个监测点的弱学习器通过集合策略进行整合，得到最终的强学习器。并且通过观察强学习器的预测结果发现仍然存在一些预测误差较大的情况，故本文将强分类器得到的各个监测点每天的 24 条预测数据进行处理，首先进行负值筛选，将预测浓度可能存在的预测负值删除，之后将每天 24 条预测数据中剩余的数据进行异常值处理，即去掉 3 个最大值和 3 个最小值的操作，最后剩余的所有预测较为合理的数据取平均得到最终的有效数据，问题四解题具体流程图如图 2 所示。

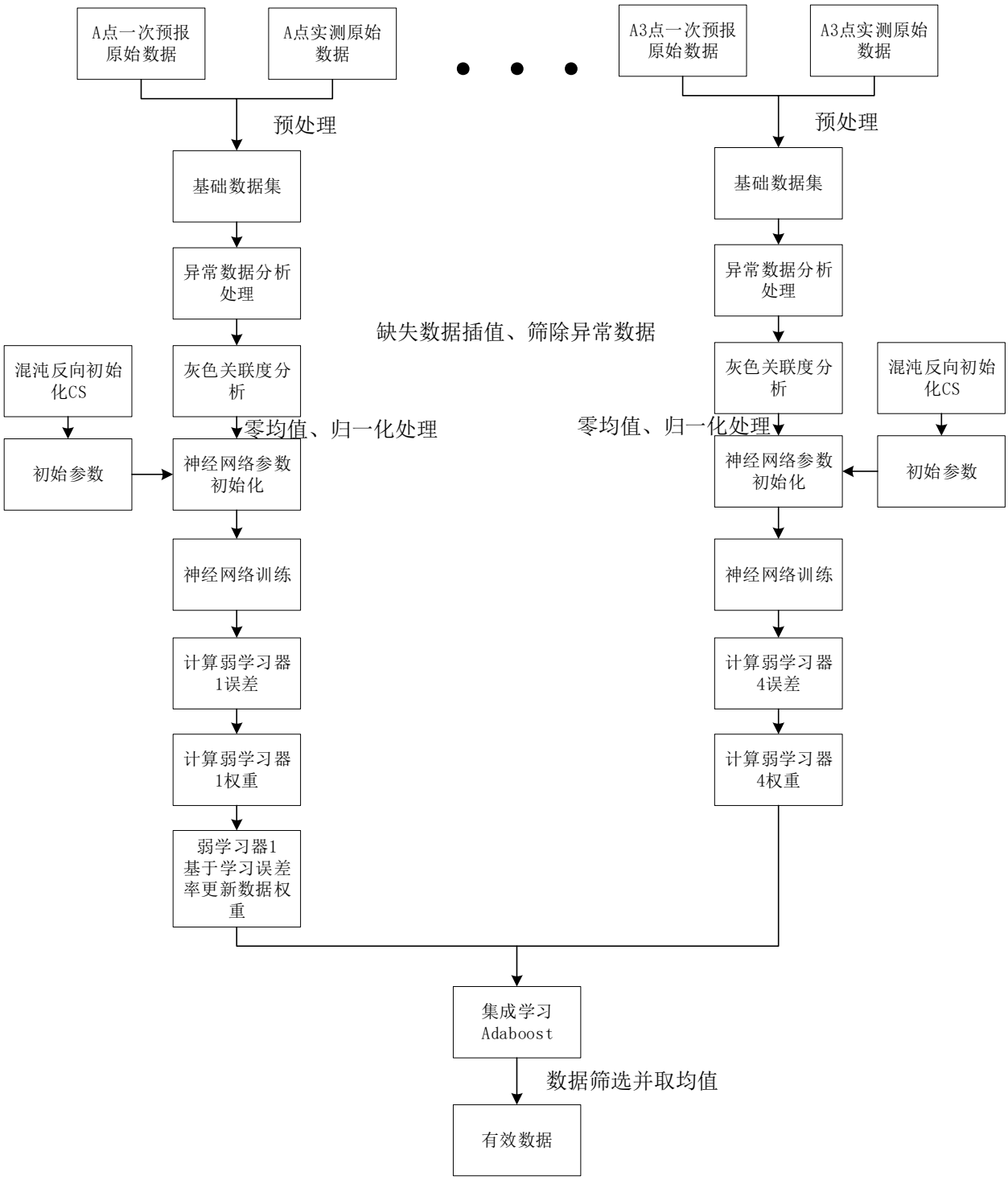

A点一次预报原始数据
A点实测原始数据
A3点一次预报原始数据
A3点实测原始数据
预处理
预处理
基础数据集
基础数据集
异常数据分析处理
异常数据分析处理
缺失数据插值、筛除异常数据
混沌反向初始化CS
灰色关联度分析
灰色关联度分析
混沌反向初始化CS
零均值、归一化处理
零均值、归一化处理
初始参数
神经网络参数初始化
神经网络参数初始化
初始参数
神经网络训练
神经网络训练
计算弱学习器1误差
计算弱学习器4误差
计算弱学习器1权重
计算弱学习器4权重
弱学习器1基于学习误差率更新数据权重
集成学习Adaboost
数据筛选并取均值
有效数据

图 2 问题四解题流程图

二. 基本假设及符号说明

2.1 基本假设

- 假设问题二中，监测点 A 周围的污染物排放情况不变，只考虑气象条件与监测点 A 的 AQI 之间的关系；
- 假设问题三中 A、B、C 三个监测点之间相互没有影响；
- 假设污染物浓度的影响因素仅包含所给数据中的气象条件；

2.2 符号说明

符号	含义
$IAQI_P$	污染物 P 的空气质量分指数
C_P	污染物 P 的质量浓度值
BP_{Hi}	与C_P相近的污染物浓度限值的高位值
BP_{Lo}	与C_P相近的污染物浓度限值的低位值
$IAQI_{Hi}$	与BP_{Hi}对应的空气质量分指数
$IAQI_{Lo}$	与BP_{Lo}对应的空气质量分指数
temp_2m	近地 2 米温度（℃）
temp_0	地表温度（K）
BS	比湿（kg/kg）
humi	湿度（%）
w_speed	近地 10 米风速（m/s）
w_dir	近地 10 米风向（°）
rain	雨量（mm）
cloud	云量
high	边界层高度（m）
press	大气压（Kpa）
GRTL	感热通量（W/m²）
QRTL	潜热通量（W/m²）
CBFS	长波辐射（W/m²）
DBFS	短波辐射（W/m²）
sunFS	地面太阳能辐射（W/m²）

三. 问题一求解

根据《环境空气质量指数（AQI）技术规定（试行）》（HJ633-2012），空气质量指数（AQI）可用于判别空气质量等级。

(1) 第一步：得到各项污染物的空气质量分指数，各项污染物项目浓度限值及对应的空气质量分指数级别见表 1。以附件 1 中细颗粒物（PM2.5）、可吸入颗粒物（PM10）、二氧化硫（SO_2）、二氧化氮（NO_2）、臭氧（O_3）、一氧化碳（CO）等各项污染物的每日实测浓度值分别计算得出空气质量分指数（Individual Air Quality Index，简称 IAQI），计算公式如公式（1）所示。

$$IAQI_p = \frac{IAQI_{Hi} - IAQI_{Lo}}{BP_{Hi} - BP_{Lo}} \cdot (C_P - BP_{Lo}) + IAQI_{Lo} \qquad (1)$$

表 1 空气质量分指数（IAQI）及对应的污染物项目浓度限值

序号	指数或污染物项目	空气质量分指数 及对应污染物浓度限值								单位
0	空气质量分指数（IAQI）	0	50	100	150	200	300	400	500	-
1	一氧化碳（CO）24 小时平均	0	2	4	14	24	36	48	60	mg/m^3
2	二氧化硫（SO_2）24 小时平均	0	50	150	475	800	1600	2100	2620	$\mu g/m^3$
3	二氧化氮（NO_2）24 小时平均	0	40	80	180	280	565	750	940	
4	臭氧（O_3）最大 8 小时滑动平均	0	100	160	215	265	800	-	-	
5	粒径小于等于 10 μm 颗粒物（PM10）24 小时平均	0	50	150	250	350	420	500	600	
6	粒径小于等于 2.5 μm 颗粒物（$PM_{2.5}$）24 小时平均	0	35	75	115	150	250	350	500	

(2) 第二步是从各项污染物的 IAQI 中选择最大值确定为 AQI。在本题中，对于 AQI 的计算仅涉及表 1 提供的六种污染物，因此计算公式如公式（2）所示。

$$AQI = \max\left\{IAQI_{SO_2}, IAQI_{\mathrm{NO}_2}, IAQI_{PM10}, IAQI_{PM2.5}, IAQI_{O_3}, IAQI_{CO}\right\} \quad (2)$$

当 AQI 小于或等于 50（即空气质量评价为“优”）时，称当天无首要污染物；当 AQI 大于 50 时，IAQI 最大的污染物为首要污染物。若 IAQI 最大的污染物为两项或两项以上时，并列为首要污染物。将附件 1 中的数据带入公式 1 中进行计算得到对应的 AQI，计算结果如表 2 所示。

表 2　AQI 计算结果表

监测日期	地点	AQI 计算	
		AQI	首要污染物
2020/8/25	监测点 A	60	O_3
2020/8/26	监测点 A	46	无
2020/8/27	监测点 A	109	O_3
2020/8/28	监测点 A	138	O_3

四. 问题二模型建立与求解

4.1 数据处理

附件 1 中共包含三种数据，分别为“监测点 A 逐小时污染物浓度与气象一次预报数据”、“监测点 A 逐小时污染物浓度与气象实测数据”和“监测点 A 逐日污染物浓度实测数据”。因需要根据对污染浓度的影响程度，对气象条件进行分类，因此采用实测数据作为基础数据集，同时为保证数据集的数量以及气象对污染物影响程度的合理分类，最终选择了对“监测点 A 逐小时污染物浓度与气象实测数据”与“监测点 A 逐小时污染物浓度与气象一次预报数据”分别采用不同的分类方式处理。

首先对于“监测点 A 逐小时污染物浓度与气象实测数据”，考虑到每种气象对于每种污染物均有可能存在影响，仅单一的对每种污染物浓度进行气象分类无法准确对气象进行合理的分类，因为即使单独的列出某种污染物数据和某种气象数据，也无法消除该污染物浓度受到其他气象数据影响。并且虽然该问是建立在排放的污染物浓度不变的条件下，但实际上数据的每日污染物排放浓度是切实存在差异的，因此结合问题一，引入了 AQI 的概念，通过判断 AQI 与各个气象之间的关系，进而对气象进行合理的分类，以此来将问题简化为环境因素对 AQI 的影响，该简化是建立在单一污染物浓度受到各种环境情况的影响无法去除的情况下，并且该数据为实测数据，计算得到的 AQI 较为准确。同时为了简化计算，并未逐条计算“监测点 A 逐小时污染物浓度与气象实测数据”中每个小时该监测点处的 AQI，而是采用了青悦公开数据（其数据来源为环保部实时空气质量发布系统）进行了神经网络模型的训练，以训练的模型预测附件 1 中的“监测点 A 逐小时污染物浓度与气象实测数据”，模型训练完成后 AQI 预测结果如图 3 所示。

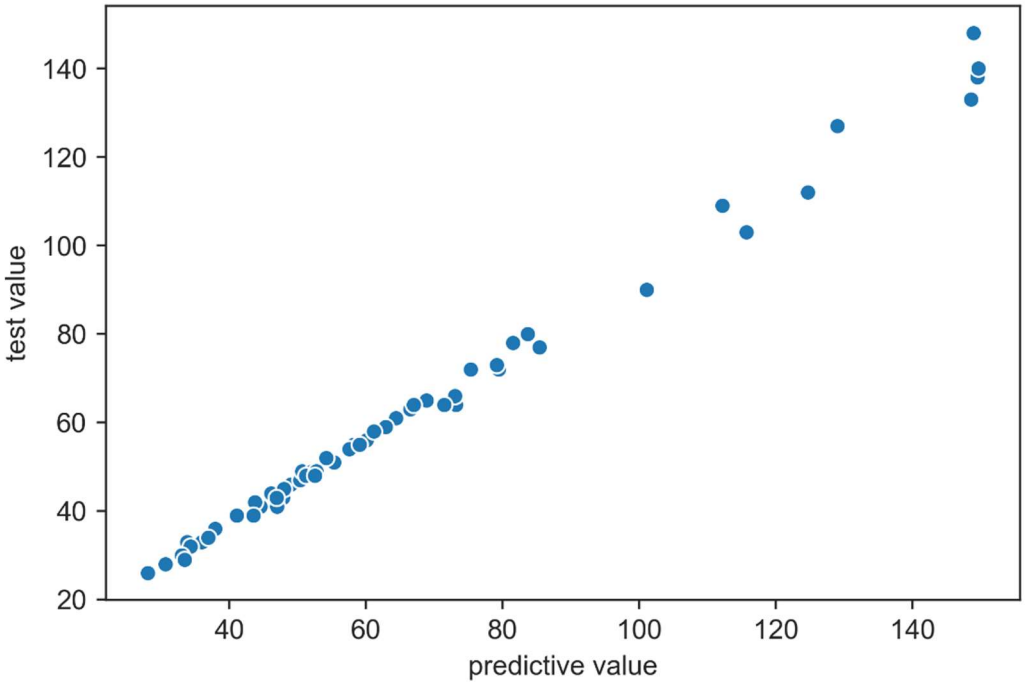


图 3　AQI 预测模型预测结果图

由图可知，该模型具有一定效果，以此模型对附件 1 中“监测点 A 逐小时污染物浓度与气象实测数据”进行预测，为该数据添加一个 AQI 的属性列，AQI 预测模型的部分数据如表 3 所示。

表 3　AQI 预测模型部分数据

测试组序号	测试值	预测值
2	32	30.33776
33	41	33.767803
257	38	38.12275
392	65	66.43906
522	83	85.1019

附件 1 中的实测数据因监测站点设备调试、维护等原因，实测数据在连续时间内存在部分或全部缺失的情况；受监测站点及其附近某些偶然因素的影响，实测数据在某个小时（某天）的数值偏离数据正常分布，为了确保建模分析结果的合理性与科学性，需对异常数据进行识别和剔除，并对缺失值进行插值处理，填补的方式为该缺失值的上一条数据对应值和下一条数据对应值的平均值。

为“监测点 A 逐小时污染物浓度与气象实测数据”添加新属性 AQI 后，对整个将要用的数据集进行了异常值的分析，通过均值、标准差、四分位数等统计学数据对数据集进行评估。将“监测点 A 逐小时污染物浓度与气象实测数据”全部数据分别按照大小顺序排列，并求出相应的均值、标准差以及四分位数。六种污染物的对应数据统计结果如表 4 所示。

表 4　六种污染物对应数据统计结果表

	SO_2 监测浓度（μg/m³）	**NO_2 监测浓度（μg/m³）**	**PM10 监测浓度(μg/m³)**	**PM2.5 监测浓度(μg/m³)**	**O_3 监测浓度(μg/m³)**	**CO 监测浓度(mg/m³)**
样本总数	17044.000	17044.000	17044.000	17044.000	17044.000	17044.000
平均值	7.078679	31.672847	44.023586	23.794825	57.997242	0.707991
标准差	3.707199	23.077521	26.649829	17.957707	50.773687	0.228777
最小值	1.000000	2.000000	1.000000	1.000000	1.000000	0.100000
第一四分位数 (25%)	5.000000	16.000000	25.000000	10.000000	22.000000	0.500000
第二四分位数(50%)	6.000000	26.000000	38.000000	20.000000	44.000000	0.700000
第三四分位数 (75%)	9.000000	40.000000	57.000000	33.000000	80.000000	0.800000
最大值	47.000000	211.00000	217.00000	163.00000	405.00000	2.500000

从表中可以发现，NO_2、PM10 和 O_3 的标准差相对较大，四分位数之间差距大，数据离散程度较大，相较之下，SO_2 和 CO 标准差相对较小，四分位数差距小，数据离散程度较小。整体来说六种污染物的数值离散程度大部分较高。同时也对各项气象指标以及 AQI 值进行了基础数据统计分析，分析结果如表 5 所示。

表 5　各项气象指标及 AQI 值数据统计结果表

	温度(℃)	湿度(%)	气　压(MBar)	风速(m/s)	风向(°)	AQI 值
样本总数	17044.000	17044.000	17044.000	17044.000	17044.000	17044.000
平均值	25.129782	68.193323	1010.6068	1.408214	155.64521	51.028211
标准差	5.746649	15.934193	6.533802	0.659817	117.33617	19.664140
最小值	5.800000	14.000000	993.50000	0.100000	0.100000	23.619308
第一四分位数 (25%)	21.200000	59.000000	1005.3000	0.900000	49.900000	37.452616
第二四分位数(50%)	26.200000	70.000000	1010.2000	1.400000	110.60000	45.781918
第三四分位数 (75%)	29.500000	80.000000	1015.6000	1.800000	256.00000	58.730485
最大值	38.200000	98.000000	1029.2000	5.800000	360.00000	204.26495

表中可以看出，湿度与风向的标准差相对较高，四分位数差距大，数据离散程度较大。其余气象条件的方差相对较小，但从整体来看，气象条件和 AQI 数据的离散程度较大。结合表 4 和表 5 不难发现，整体使用的数据集离散程度相对较大。

考虑到实际环境具有一定的周期性，例如温度数据在一天当中的变化存在明显的周期性。在实际分析过程中，需要去除其周期性的影响，并且考虑到数据的离散程度较大，故采用变化差值的方式来观察环境对污染物的影响。在使用数据时，将对应的环境数据以三天时间为间隔做差（如：2020/1/3/24:00　- 2020/1/1/24:00，即三天的 72 条数据通过最后一条数据减去第一条数据的方式合并为 1 条数据，以此得到环境的变化情况），之所以选择三天是保证做差之后数据仍具有足够的差异性，并且最终的数据量不至于太少，最后的实验发现三天的时间虽然环境变化并不是太过明显，但已经足够观察变化了。完成处理后使用做差之后的各项环境数据和对应的 AQI 变化差值进行分析，对气象条件进行合理分类。

对于“监测点 A 逐小时污染物浓度与气象一次预报数据”，在考虑到其数据偏差较大，计算 AQI 值不能保证其准确性，故采取了灰色关联度分析的方式对 6 类污染物浓度进行分析，通过分析得到各个污染物的主要影响因素，以此来对环境数据进行合理分类。由于污染物共有 6 类，所以最终的分类也为 6 类，即 SO_2 的主要气象影响因素为一类气象条件，NO_2 的主要气象影响因素为一类气象条件，PM10 的主要气象影响因素为一类气象条件，PM2.5 的主要气象影响因素为一类气象条件，O_3 的主要气象影响因素为一类气象条件，CO 的主要气象影响因素为一类气象条件。灰色关联度分析的具体在后续进行说明，这里给出

最终分析得到 6 类污染物的主要气象影响因素。

(1) SO_2 的主要气象影响因素：temp_2m、temp_0、humi、w_speed、w_dir、rain、high、CBFS、GRTL。

(2) NO_2 的主要气象影响因素：temp_2m、temp_0、humi、w_speed、w_dir、rain、high、CBFS。

(3) PM10 的主要气象影响因素：temp_2m、temp_0、w_speed、rain、high、GRTL、CBFS。

(4) PM2.5 的主要气象影响因素：temp_2m、temp_0、humi、w_speed、w_dir、rain、CBFS。

(5) O_3 的主要气象影响因素：temp_2m、temp_0、w_speed、rain、high、GRTL、CBFS。

(6) CO 的主要气象影响因素：temp_2m、temp_0、humi、w_speed、rain、high。

(7) 至此完成对“监测点 A 逐小时污染物浓度与气象一次预报数据”的气象条件进行分类。

4.2 模型的建立与求解

- 边际分布线性回归分析

回归分析是研究自变量与因变量之间数量变化关系的一种分析方法，它主要是通过因变量Y与影响它的自变量 X_i (i=1,2,3…)之间的回归模型，衡量自变量X_i对因变量Y的影响能力的。在本题中，边际分布线性回归分析具体步骤如下：

(1) 确定自变量 X_i 和因变量 Y 。因变量 Y 则为数据处理部分得到的新属性列 AQI，自变量 X_i 为附件 1 中“监测点 A 逐小时污染物浓度与气象实测数据”中的 5 钟气象条件

(2) 绘制散点图。在本题中为一元线性回归，回归模型中只含有一个自变量，用来处理一个自变量与一个因变量之间的线性关系，如温度与 AQI 之间的线性关系。

(3) 估计模型参数，建立回归模型。使用最小二乘法进行模型参数的估计，最小二乘法在回归模型上的应用，就是要使得观测点和估计点的距离的平方和达到最小。使得尽可能多的到(X_i ， Y_i)数据点落在或者更靠近这条拟合出来的直线上，如图 4 所示。

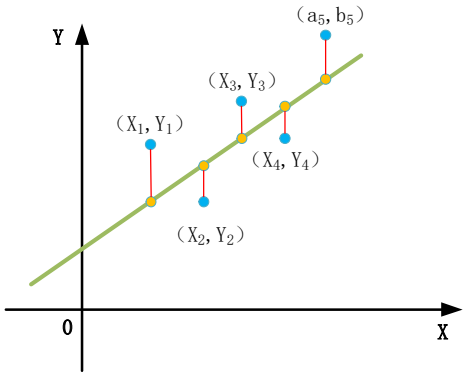
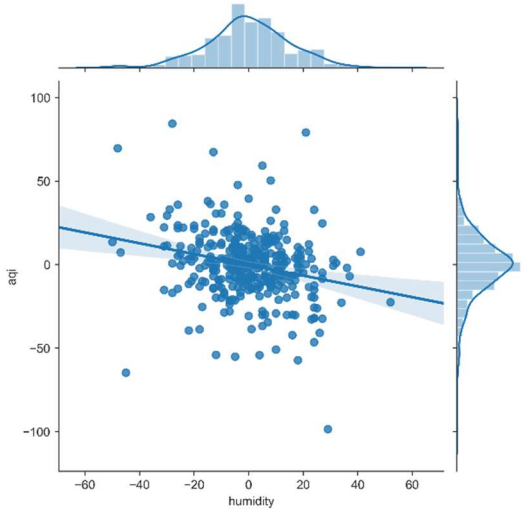


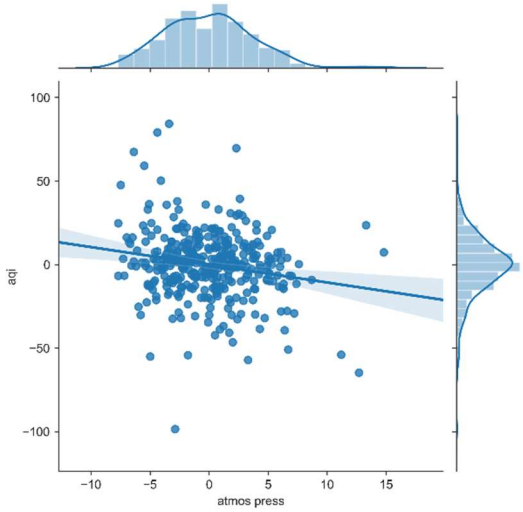
图 4　最小二乘法原理示意图

经过最小二乘法对模型参数进行估计并作图，各环境 X_i 变量与 AQI 之间的关系如图 5 所示.由图可知，湿度、气压和风速的升高均会使得 AQI 下降，温度和风向的升高使得 AQI 上升。但由于环境气象条件的复杂性以及相互影响性，不能简单地认为每种气象条件对 AQI 的影响是固定的，例如气温升高时，环境湿度有可能会下降，温度升高和湿度下降均会造成 AQI 升高，无法判断 AQI 的升高，温度和湿度哪个条件因素起的作用更大，因此需要将温度和湿度并且来看，同时气压在水平方向上，温度越高，气压越低，且温度升高和气压变低均会使得 AQI 升高。温度升高导致气压变化，气压变化又会引起风速变化，因此，各气象与 AQI 之间的关系并不是简单的一对一或一对多的关系，存在相互影响的情况，只能相对的看待问题，相对于其他条件来说，湿度、气压和风速的升高均会使得 AQI

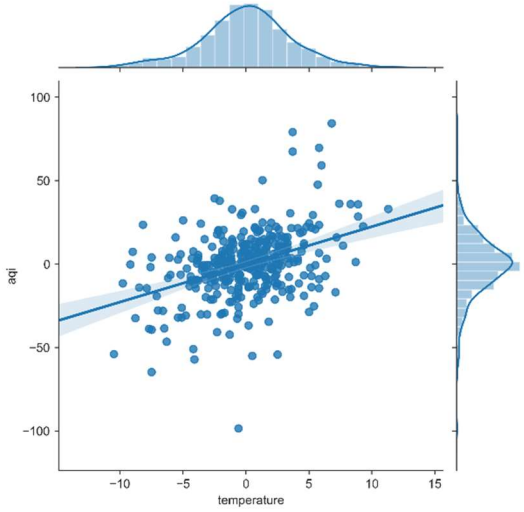
下降，温度和风向的升高使得 AQI 上升。



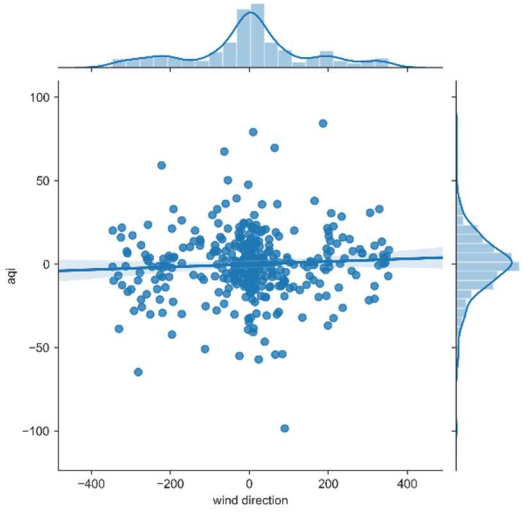
(a) 湿度与 AQI 线性回归图



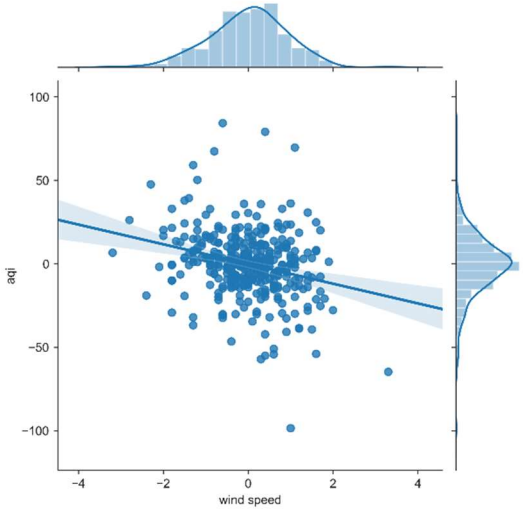
(b) 气压与 AQI 线性回归图



(c) 温度与 AQI 线性回归图



(d)风向与 AQI 线性回归图



(e)风速与 AQI 线性回归图

图 5 各环境变量与 AQI 的线性回归图

五. 问题三模型建立与求解

5.1 数据处理

5.1.1 数据预处理

(1) 首先需要对数据进行预处理。由题可知，一次预报数据包含当天、第二天和第三天的预测数据，且题中提到一次预报数据中对临近日期的预测较为准确，因此在二次建模时，考虑到后两天预测数据的准确率低的问题，将一次预测数据中对第二天和第三天的预测数据删除，只保留第一天的预测数据。

(2) 其次，建立一次预报数据和实测数据之间的联系，将“监测点 A 逐小时污染物浓度与气象实测数据”中的六种污染物值和一次预报数据拼接到一起，且因为预报工作中，服务器受外接电源长时间停电等情况影响，导致部分运行日期的一次预报数据缺失，为保证建立模型的鲁棒性，则需要对缺失的数据进行修补。在本题中使用“监测点 A 逐日污染物浓度实测数据”来对“监测点 A 逐小时污染物浓度与气象实测数据”的缺失数据进行补充，保证“监测点 A 逐小时污染物浓度与气象实测数据”和“监测点 A 逐小时污染物浓度与气象一次预报数据”在日期上匹配。两份数据拼接完成后需要对仍缺失的值以及异常值进行处理，缺失值及异常值（如负数的浓度）使用其前一条数据和后一条数据的均值来填补，对缺失数据较多的数据则直接筛除，对于某个表中存在一整天的数据丢失情况也将另一个表中的对应日期删除。

(3) 最后将“监测点 A 逐小时污染物浓度与气象实测数据”中的污染物浓度与“监测点 A 逐小时污染物浓度与气象一次预报数据”中的污染物浓度做差，保留其差值和环境因素作为新的数据集。使用该数据集合进行再次建模，模型预测结果为实测数据与一次预报数据的差值，将模型预测结果与上一次预测值相加就能得到二次预测模型预测的实测值，以此，建立了一次预测数据和实测数据中的关系，达到了实测数据修正一次预测数据的目的。

(4) 在完成“监测点 A 逐小时污染物浓度与气象实测数据”中的污染物浓度与“监测点 A 逐小时污染物浓度与气象一次预报数据”中的污染物浓度做差之后，还需对新生成的数据进行异常值分析，这是由于一次预报数据和实测数据偏差较大，做差之后有相当一部分数据明显不符合正常情况，如一次预报数据和实测数据的温差达到 50 度以上等情况。在保证数据量以及处理后的数据偏差在可接受的范围的情况下，对异常值筛选的范围进行合理的选取即可。

(5) 最后完成训练数据和测试数据的划分，本题对污染物浓度的预测是建立 6 个改进的 CS-BP 模型，每个模型分别预测一个污染物浓度，所以划分的数据也对应有 6 个，不仅是 Y 值分为 SO_2、NO_2、PM10、PM2.5、O_3、CO，每个模型的 X 也为不一样，X 的选择为灰色关联分析之后的该污染物浓度主要影响因素。

5.1.2 灰色关联度分析

灰色关联分析是对一个系统发展变化态势的定量描述和比较的方法，其基本思想是通过确定参考数据列和若干个比较数据列的几何形状相似程度来判断其联系是否紧密，反映了曲线间的关联程度。灰色关联分析的具体计算步骤如下：

(1) 确定反映系统行为特征的参考数列和影响系统行为的比较数列。

在本题中，参考数列即为各污染物的，比较数列则为数据处理中得到的各项一次预报得到的气象指标以及实测的气象指标。

- 参考数列为 $Y = Y(k) \mid k = 1, 2...n;$
- 比较数列为 $Xi = Xi(k) \mid k = 1, 2...n, i = 1, 2...m.$

(2) 无量纲化处理

由于各因素列中的数据因量纲不同，不便于比较，因此在进行灰色关联度分析时，需要先进行数据的无量纲化处理，使用的无量纲化公式如公式（3）所示。

$$f(x(k)) = \frac{x(k) - \min\limits_{k} x(k)}{\max\limits_{k} x(k) - \min\limits_{k} x(k)} = y(k) \qquad (3)$$

(3) 计算关联系数

计算关联度需先计算关联系数，关联系数计算公式如公式所示。其中$\rho\epsilon(0,\infty)$，称为分辨系数。ρ越小，分辨力越大，当$\rho \le 0.5463$时，分辨力最好，本题中取$\rho = 0.5$。关联系数计算公式如公式（4）所示。

$$\zeta_i(k) = \frac{\min\limits_{i}\min\limits_{\&} \mid y(k) - x_i(k) \mid + \rho \max\limits_{i}\max\limits_{\&} \mid y(k) - x_i(k) \mid}{\mid y(k) - x_i(k) \mid + \rho \max\limits_{i}\max\limits_{\&} \mid y(k) - x_i(k) \mid} \qquad (4)$$

记 $\Delta_i(k) = \mid y(k) - x_i(k) \mid$，则

$$\zeta i(k) = \frac{\min\limits_{i}\min\limits_{\&} \Delta_i(k) + \rho \max\limits_{i}\max\limits_{\&} \Delta_i(k)}{\Delta_i(k) + \rho \max\limits_{i}\max\limits_{i} \Delta_i(k)} \qquad (4\text{-}1)$$

(4) 计算关联度

关联系数是比较数列与参考数列在各个时刻的关联程度值，因此关联系数并不唯一，而信息过于分散不便于进行整体性比较。因此需要将将各个时刻的关联系数集中为一个值，即求其平均值，计算公式如公式（5）所示。

$$r_i = \frac{1}{n}\sum_{k=1}^{n} \zeta i(k), k = 1, 2, \Lambda, n \qquad (5)$$

r_i 为比较数列 Xi 对参考数列 Y 的灰色关联度，r_i 越接近 1，说明相关性越好。

(5) 关联度排序

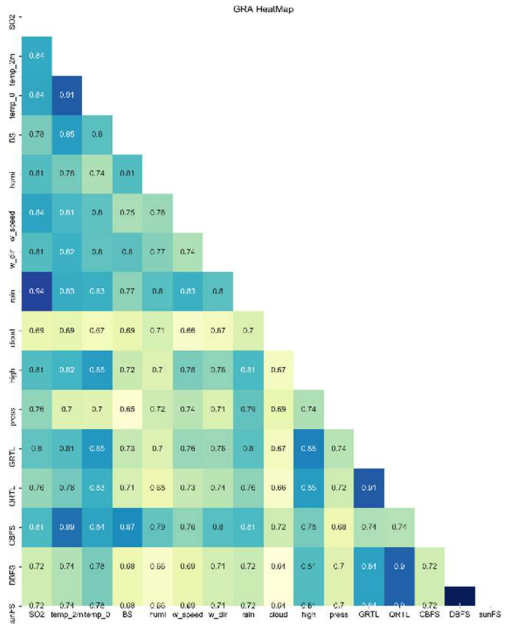
因素间的关联程度，主要通过关联度的大小次序描述，而不仅是关联度的大小。将若干个比较数列对同一参考数列的关联度按大小顺序排列起来，便组成了关联序，反映了对于参考数列来说各比较数列的“优劣”关系。灰色关联度分析实现代码如表 6 所示。

表 6　灰色关联度分析实现代码

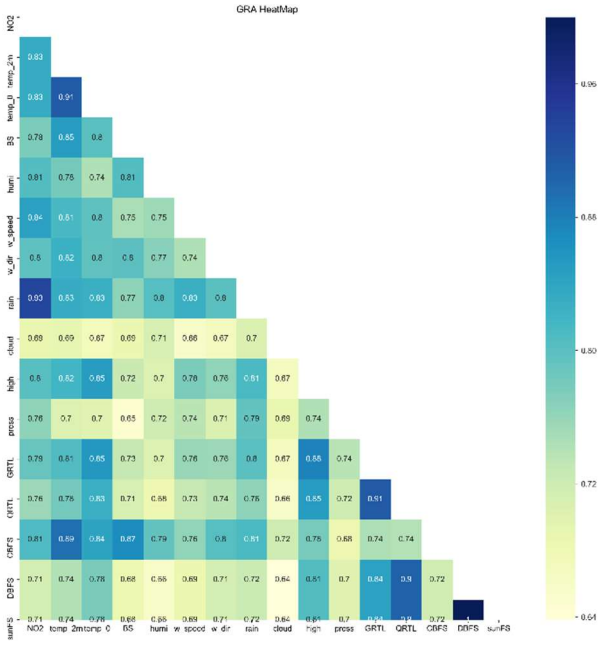
```
1  def GreyCD_sig(gray, m=0):
2      gray = dimensionlessProcessing(gray)
3      std = gray.iloc[:, m]   # 标准要素
4      gray.drop(str(m),axis=1,inplace=True)
5      ce = gray.iloc[:, 0:]   # 比较要素
6      shape_n, shape_m = ce.shape[0], ce.shape[1]   # 计算行列
7      # 与标准要素比较
8      a = np.zeros([shape_m, shape_n])
9      for i in range(shape_m):
10         for j in range(shape_n):
11             a[i, j] = abs(ce.iloc[j, i] - std[j])
12     c, d = np.max(a), np.min(a)
13     # 计算结果
14
```

```
        result = np.zeros([shape_m, shape_n])
        for i in range(shape_m):
            for j in range(shape_n):
                result[i, j] = (d + 0.5 * c) / (a[i, j] + 0.5 * c)
        # 求均值，取得灰色关联值
        result_list = [np.mean(result[i, :]) for i in range(shape_m)]
        result_list.insert(m,1)
        return pd.DataFrame(result_list)
```

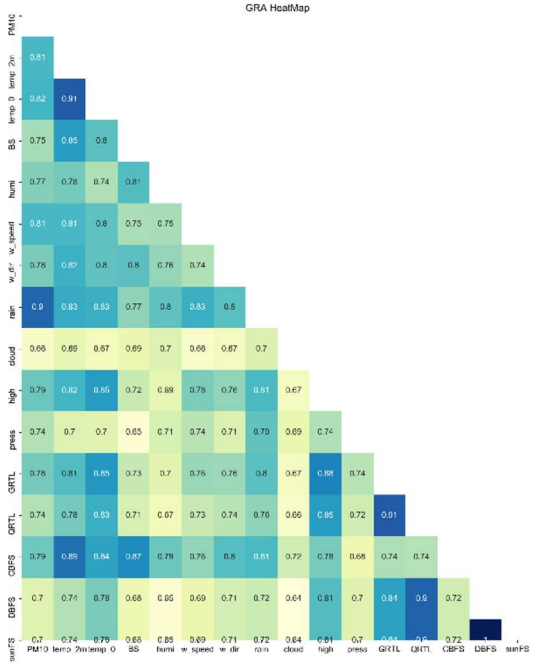
各污染物与各气象条件的关联度排序如图 6 所示



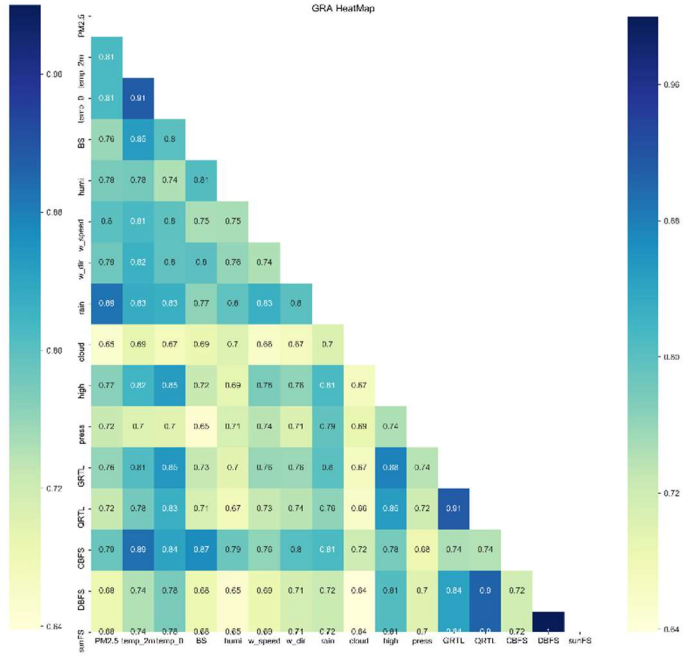
(a) SO_2 关联度大小排序



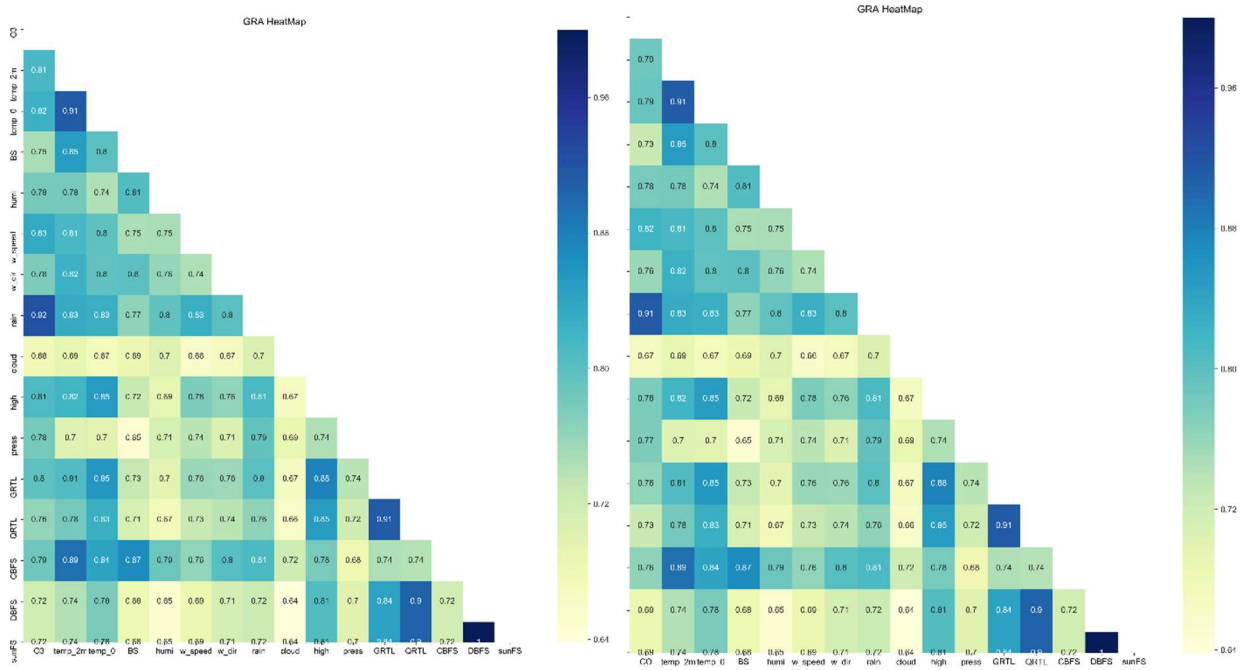
(b) NO_2 关联度大小排序



(c)PM10 关联度大小排序



(d)PM2.5 关联度大小排序



(e) O_3 关联度大小排序　　(f)CO 关联度大小排序

图 6 各污染物与各气象条件的关联度大小排序图

由图可以得出各污染物与各气象条件之间的关联度大小，并选取与各污染物关联度较大的气象条件作为其特征因素进行神经网络模型训练。各污染物与各气象条件的关联度表与各污染物所选取的特征因素如表 6 所示。

表 6 各污染物与各气象条件的关联度表与各污染物所选取的特征因素表

污染物 / 气象条件	SO2	NO2	PM10	PM2.5	O3	CO
近地 2 米温度（℃）	0.84609	0.842457	0.840696	0.843437	0.824173	0.814707
地表温度（K）	0.838104	0.833101	0.836026	0.833292	0.825393	0.80844
比湿（kg/kg）	0.778308	0.778639	0.773095	0.784562	0.749712	0.748723
湿度（%）	0.805711	0.806201	0.783135	0.794842	0.773073	0.788967
近地 10 米风速（m/s）	0.823488	0.827298	0.811356	0.802852	0.815651	0.81654
近地 10 米风向（°）	0.817252	0.810565	0.808565	0.820177	0.789336	0.789455
雨量（mm）	0.942198	0.939122	0.916444	0.909306	0.926823	0.926524
云量	0.695081	0.698116	0.694034	0.694318	0.687761	0.700649
边界层高度（m）	0.815305	0.81226	0.815749	0.803141	0.819713	0.803638
大气压（Kpa）	0.768031	0.770004	0.768943	0.757599	0.784803	0.796426
感热通量（W/m^2）	0.783	0.780251	0.786923	0.773241	0.785524	0.768674
潜热通量（W/m^2）	0.764189	0.760976	0.768144	0.756575	0.765668	0.751345
长波辐射（W/m^2）	0.823778	0.822039	0.817516	0.825239	0.798667	0.796246
短波辐射（W/m^2）	0.727191	0.725051	0.731083	0.722316	0.72773	0.717087
地面太阳能辐射（W/m^2）	0.727191	0.725051	0.731083	0.722316	0.72773	0.717087

5.2 模型的建立与求解

5.2.1 CS 算法

谷鸟搜索（Cuckoo Search，CS）是由 Xin-She Yang 和 Suash Deb 于 2009 年开发的自然启发式算法[5]。布谷鸟算法是群体智能算法的一种，算法思想来源于布谷鸟特殊的生活习性。这种鸟会将生下来的蛋放到其他鸟的鸟巢去，这在算法的寻优当中就是全局最优化，该算法采用莱维飞行来模拟全局随机寻优。但鸟蛋有概率被寄养的鸟发现，发现之后该蛋会被丢弃，布谷鸟则会在蛋被丢弃之后在附加随机选择一个位置重新安置一个新蛋，这里则对应算法的局部寻优部分，采用局部随机行走算法来完成，布谷鸟算法对比与遗传算法以及粒子群算法等具有结构简单、容易实现、效果较好的优点。在对上述算法思想的总结之后，具体算法的步骤如下：

(1) 在最初可行域内随机生成一组解（布谷鸟）。

(2) 记录这些点的适应度值，并单独记录最优解的位置及其适应度值。

(3) 通过莱维飞行更新这些解的位置，并计算新解的位置，与之前的解对比适应度值，留下更优的解。更新位置的计算公式如公式（6）所示

$$x_i^{t+1} = x_i^t + \alpha \otimes S, (i = 1, 2, 3, ..., n) \qquad (6)$$

式（6）中 x_i^{t+1} 为更新的位置，x_i^t 为之前的位置，α 为步长，$\otimes$ 为点乘运算。

$$S = \frac{u}{|v|^{\frac{1}{\beta}}} \qquad (7)$$

式（7）中 u 服从 $N(0, \sigma_u)$ 正态分布，v 服从 $N(0, \sigma_v)$ 正态分布。

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\frac{\pi\beta}{2})}{\Gamma[\frac{1+\beta}{2}\beta * 2\frac{(\beta-1)}{2}]} \right\}^{\frac{1}{\beta}} \qquad (8)$$

$$\sigma_v = 1 \qquad (9)$$

式（8）中 Γ 表示标准伽玛函数，$\Gamma(z) = \int_0^\infty \frac{t^{z-1}}{e^t} dt$

(4) 更新后的解有一定的概率被抛弃，被抛弃的解将在附近寻找一个新的位置，没被抛弃的保存原样。这里采用局部随机行走来完成这个过程，更新公式如下：

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(pa - \varepsilon) \otimes (x_j^t - x_k^t) \qquad (10)$$

式（10）中 x_i^{t+1} 为更新的位置，x_i^t 为之前的位置，α 为步长系数，H 为跃迁函数（x 大于等于 1 时数值为 1，否则为 0），x_j^t 和 x_k^t 为 t 时刻随机选取的两个解。

(5) 重新计算适应度值，并保存这一代最好的解和适应度。

(6) 得到新的一组解从步骤（3）重新开始迭代，直到达到适应度要求或是迭代次数完成。布谷鸟算法的实现核心代码如表 7 所示.

表 7　布谷鸟算法的实现核心代码

```
#n：布谷鸟数　m：布谷鸟的维度
def cuckoo_search(n, m, lower_boundary, upper_boundary, iter_num = 10,pa = 0.25, beta
= 1.5, step_size = 0.1,alpha=0.77,xn=0.33):

```

```
    num=1
    # get initial nests' locations
    nests,best_nest,best_fitness,lossness = generate_nests(n, m,alpha,xn,
lower_boundary, upper_boundary) #alpha,xn 用于 tent 初始化阈值和初值设置

    # get the best nest and record it
    LossArr.append(best_fitness) #全局变量，保存每代最佳适应度值


    print('\r\n BEST_LOSSNESS IS %.2f : \r\n',best_fitness)

    for _ in range(iter_num):

        print('\r\n*******第 %d 代开始迭代优化**********\r\n'%num)
        nests = update_nests(lower_boundary, upper_boundary, nests, best_nest,
lossness, step_size,best_fitness)
        nests = abandon_nests(nests, lower_boundary, upper_boundary, pa)

        print('\r\n********第 %d 次迭代，计算适应度*********\r\n'%num)
        lossness,_ = calc_fitness( nests)
        print('\r\n************第 %d 次迭代结束********\r\n'%num)

        min_loss_index = np.argmin(lossness)
        min_loss = lossness[min_loss_index]
        min_nestloss = nests[min_loss_index]
        LossArr.append(min_loss)

        if min_loss < best_fitness    : #and    min_loss_fit > best_two_fitness
            best_nest = min_nestloss
            best_fitness = min_loss
            print('\r\n******')
            print('\r\n 第 %d 次迭代最优 Loss 是 %.2f : \r\n'%(num,best_fitness))
            print('\r\n******\r\n')
        num+=1

    return (best_nest, best_fitness)
```

5.2.2 混沌反向初始化改进 CS 算法

使用混沌反向初始化的方式取代其原本随机初始化的方式，随机初始化的方式生成的数据在可行域中表达能力并不强，不能充分均匀布满整个可行域区间。混沌初始化本文选择 Tent 混沌初始化，其初始值经测试 选择 0.77， 初值选择 0.33。Tent 初始化公式如公式（11）所示。

$$x_{n+1}=f(x_n)=\begin{cases}\dfrac{x_n}{\alpha} & ,x_n\in[0,\alpha)\\ \dfrac{1-x_n}{1-\alpha} & ,x_n\in[\alpha,1]\end{cases}\qquad \text{其中}0<\alpha<1 \qquad (11)$$

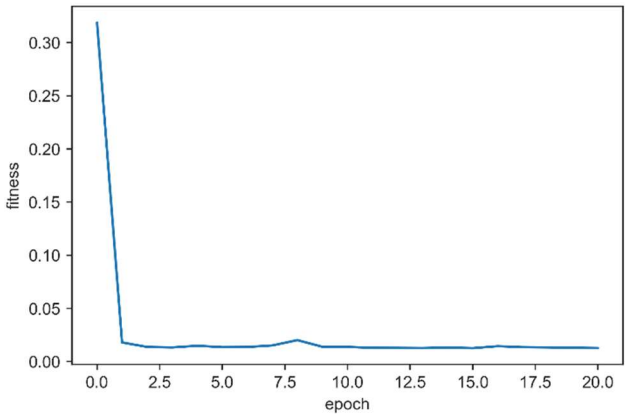
完成 Tent 混沌初始化之后，在 Tent 算法产生的初始解的基础上取其反向解，并计算

这两组解的适应度，依据贪婪法则留下其中的最优，以此构成初始解。这样将能在解空间中生成更加均匀、更具遍历性、表达能力更强的初始解。混沌反向初始化 CS 算法实现代码如表 8 所示。

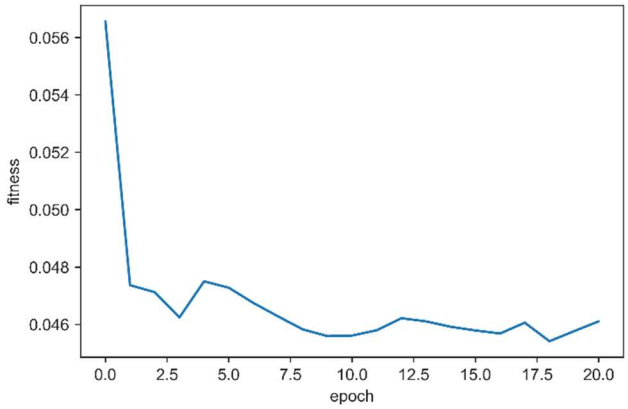
表 8　混沌反向初始化 CS 算法实现代码

```
def generate_nests(n, m,alpha,xn):
#Tent 混沌反向初始化
# #混沌初始化
    nests = np.empty((n, m))
    sig_nest = np.empty(m)
    alpha = alpha
    xn = xn
    for i in range(0,n):     #*2  值域为[-1,1] *6  值域为[-3,3]
        for j in range(0,m):
            if 0<=xn<alpha:
                xn = xn/alpha
                sig_nest[j]=(xn-0.5)*6
            elif alpha <= xn <= 1:
                xn = (1-xn)/(1-alpha)
                sig_nest[j] = (xn-0.5)*6
            nests[i] = sig_nest
# #反向初始化
    renests = -1 * nests     #定义：Pi = ai + bi - pi    生成反向 nests
# #拼接两个初始化 nests
    nests = np.vstack((nests,renests))   #拼接 nests 和 renests  准备计算适应度选择最优的 n 个 nest
#计算适应度
    lossness,_ = calc_fitness( nests)
# #根据 loss 值排序
    arrIndex = np.argsort(lossness)     #获得排序数组    从小到大
    lossness = lossness[arrIndex]        #将 lossness 数组按照从小到大排序
    nests = nests[arrIndex] #将 nests 也按照相同序列进行排序，保证和 lossness 对齐
#删除多余的 n 组 nest，这里从最底下开始一个个删，因为已经排好序了，所以删除的为效果最差的
    for i in range(n):
        nests = np.delete(nests,-1,0)
        lossness = np.delete(lossness,-1,0)
#现在的 nests 是按照 loss 排序的，第一个 loss 最小
    return nests,nests[0],lossness[0],lossness
```

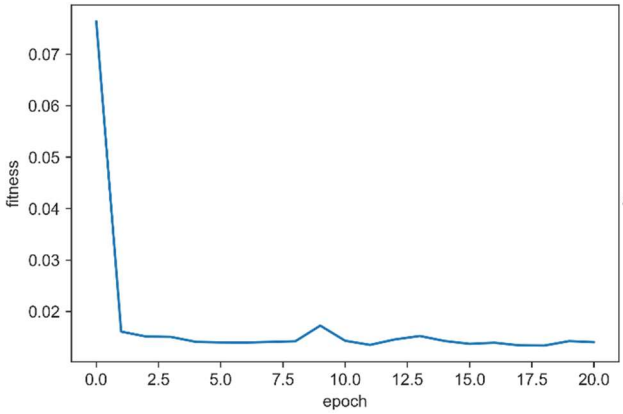
由于反向的方式在算法进行局部寻优的时候也能够起到优化寻优效率的效果，故，本文在布谷鸟蛋被抛弃，在局部重新寻找新的新的位置放置鸟蛋的时候也在其找到的位置的反向位置也生成一个新的鸟蛋，之后评估这两个位置的 Loss，选择一个最优的留下，以此减少算法的寻优代数。CS 算法具体的寻优效果如图 7 所示。



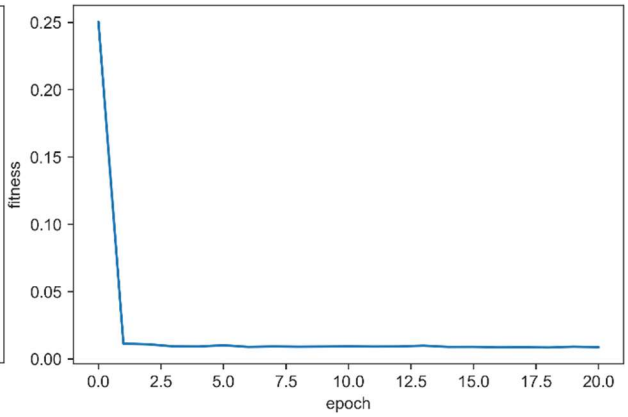
(a) SO_2 损失函数



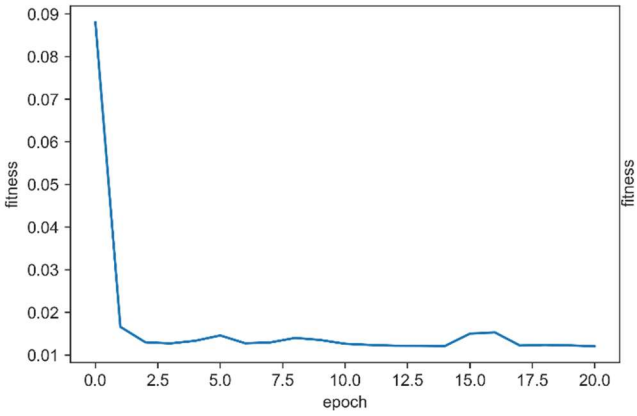
(b) NO_2 损失函数



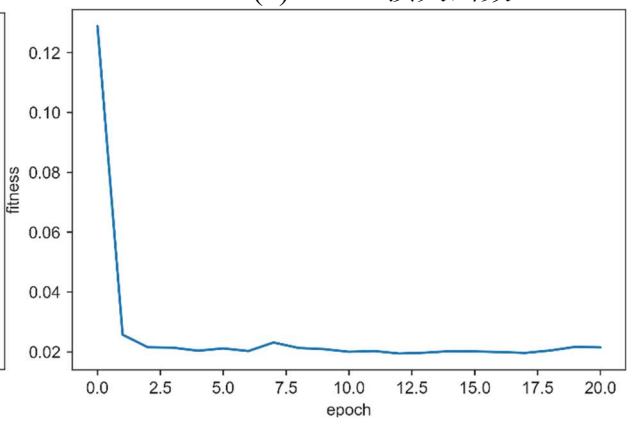
(c) PM10 损失函数



(d)PM2.5 损失函数



(e) O_3 损失函数



(f) CO 损失函数

图 7　各项污染物模型训练损失函数

由图 7 可以看到算法改进之后的寻优效果明显，能够很快找到一个较小的损失，并且没有丢失算法的随机性。对于智能算法来说随机性是很重要的，如果随机性不足就容易陷入局部最优无法跳出，例如遗传算法就容易在多次迭代后所有的染色体都趋于一致的情况，哪怕增加变异的程度也难以改变这种趋势，而 CS 算法就能看到在达到一个较小的 Loss 之后算法仍然会随机丢弃掉一些鸟蛋，重新随机的找一个位置，以期望找到一个 Loss 更小的位置，所以该算法虽然会在达到较小值之后 Loss 突然增大，但其找到全局最优的可能性更高。这对于优化神经网络来说是很关键的，因为找到的参数如果是一个局部最优参数，那么神经网络利用该参数进一步训练时也很容易陷入局部最优。

5.2.3 模型求解

在 CS 算法完成神经网络参数寻优之后就能够以一个较好的初始权值和阈值去训练神经网络，在进过更少的训练周期之后得到更好的效果以及更低的 Loss。这里本文将搭建 6 个改进的 CS-BP 模型去完成预测，每个模型预测一个污染物浓度，每个污染物浓度的训练参数也是由灰色关联度分析得到的各个污染物浓度对应的主要气象影响因素条件。对于模型的结构由于每个神经网络的输入层神经元个数不同，间接的也将导致其隐层神经元个数不同，本文采用的神经网络为 3 层结构，其大致结构如图 8 所示。

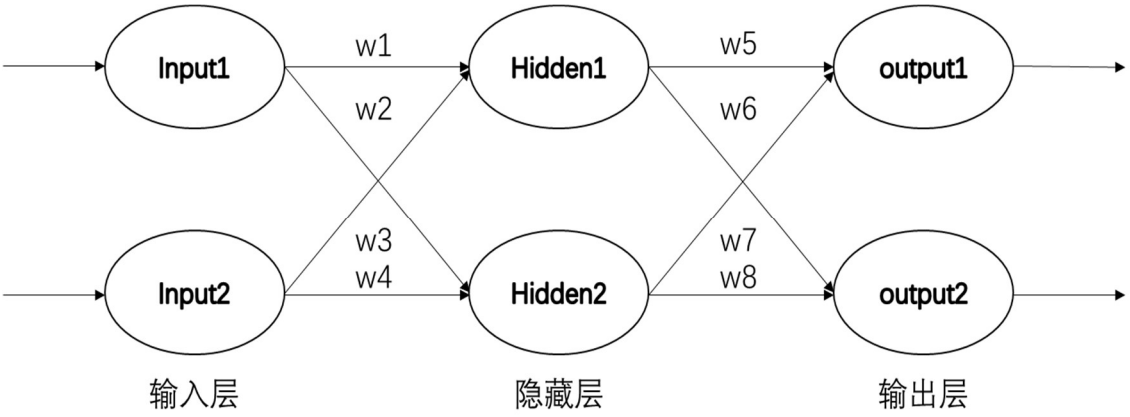


图 8 神经网络结构图

其中输入层神经元个数由各污染物浓度的主要气象影响因素条件决定，隐藏层神经元数量满足如下两个条件：

- 隐藏层神经元个数小于 n-1，n 为训练样本的数量，这是为了防止最后造成模型的泛化能力过低，最终模型效果不好。
- 隐藏层神经元个数大于输入层神经元个数，这是为了增加模型的拟合效果。

在此基础上隐藏层个数在通过多次实验之后对比效果，最后选择较优的哪一种情况。

模型的预测数据选择 A、B、C 三个检测点的“监测点 X 逐小时污染物浓度与气象一次预报数据”的 2021/7/13 预测的 2021/7/13、2021/7/14、2021/7/15 数据，每个监测点共 72 条，之所以不选择其他日期预测的数据是因为题目中提到间隔日期越长则预测效果越不准确，故选择间隔最短的数据来完成预测。

首先对预测的数据 X 作和训练数据相同的零均值和归一化处理，之后将 X 输入训练好的改进 CS-BP 神经网络中，得到的预测结果为实测数据减去一次预测数据，最后得到的最终值（预测的实测值）为二次预测数据+一次预测数据，即改进 CS-BP 神经网络模型输出加上一次预测数据的对应污染物浓度值。

在得到 72 组最终结果后需要对其做最后的处理，首先将其分为 3 组，即 2021/7/13、2021/7/14、2021/7/15 的数据，之后检测数据中是否存在负值，污染物浓度为负就将其删除，最后去除 3 组最大值和 3 组最小值，这些值很有可能预测错误，造成最后的结果偏差过大，完成所有的处理之后将各个监测点每天剩余的数据计算其平均值，得到最终预测的 2021/7/13、2021/7/14、2021/7/15 单污染物浓度值。

在将其余的污染物也做出相同的处理以后得到的 A、B、C 三个检测点的 2021/7/13、2021/7/14、2021/7/15 预测数据分别如表 9、表 10、表 11 所示.

表 9 监测点 A 污染物浓度及 AQI 预测结果表

预报日期	地点	二次模型日值预测							
		SO2 (μg/m³)	NO2 (μg/m³)	PM10 (μg/m³)	PM2.5 (μg/m³)	O3 最大八小时滑动平均 (μg/m³)	CO (mg/m³)	AQI	首要污染物
2021/7/13	监测点 A	5.5	14	24	17.2	53.1	0.22	27	空气质量评价为“优”，当天无首要污染物
2021/7/14	监测点 A	6	10	44	18	82.7	0.3	44	空气质量评价为“优”，当天无首要污染物
2021/7/15	监测点 A	12	13	22	19.7	64.5	0.27	33	空气质量评价为“优”，当天无首要污染物

表 10 监测点 B 污染物浓度及 AQI 预测结果表

预报日期	地点	二次模型日值预测							
		SO2 (μg/m³)	NO2 (μg/m³)	PM10 (μg/m³)	PM2.5 (μg/m³)	O3 最大八小时滑动平均 (μg/m³)	CO (mg/m³)	AQI	首要污染物
2021/7/13	监测点 B	7.76	6.87	25.95	9.6	78.8	0.45	40	空气质量评价为“优”，当天无首要污染物
2021/7/14	监测点 B	7.1	6.77	23.86	7.3	75.8	0.48	38	空气质量评价为“优”，当天无首要污染物
2021/7/15	监测点 B	6.5	12.3	19.9	8.14	108.1	0.5	57	O3

表 11 监测点 C 污染物浓度及 AQI 预测结果表

预报日期	地点	二次模型日值预测							
		SO_2 ($\mu g/m^3$)	NO_2 ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	O_3 最大八小时滑动平均 ($\mu g/m^3$)	CO (mg/m³)	AQI	首要污染物
2021/7/13	监测点 C	7.78	19.19	27	14.4	138.5	0.72	83	O_3
2021/7/14	监测点 C	8.1	22.95	30	17.8	99.6	0.735	50	空气质量评价为“优”，当天无首要污染物
2021/7/15	监测点 C	9.34	32.82	28.7	29.6	169	0.84	109	O_3

六. 问题四模型建立与求解

6.1 数据处理

与问题三中的数据预处理方式类似，首先将 A、A1、A2、A3 四个监测点的一次预报数据中第二天和第三天的预测数据删除，只保留第一天的预测数据。

其次，建立一次预报数据和实测数据之间的联系，将各监测点数据中的“逐小时污染物浓度与气象实测数据”中的六种污染物值和一次预报数据拼接到一起，首先保证其在时间上完全对应，具体处理方式和问题三一致，且为保证建立模型的鲁棒性，使用各监测点“逐日污染物浓度实测数据”来对“逐小时污染物浓度与气象实测数据”的缺失数据进行补充。两份数据拼接完成后需要对仍缺失的值以及异常值进行处理，缺失值及异常值（如负数的浓度）使用其前一条数据和后一条数据的均值来填补，对缺失数据较多的数据则直接筛除。

最后将各监测点“逐小时污染物浓度与气象实测数据”中的污染物浓度与各监测点“逐小时污染物浓度与气象一次预报数据”中的污染物浓度做差，保留其差值和环境因素作为新的数据集。

对各监测点“逐小时污染物浓度与气象实测数据”中的污染物浓度与各监测点“逐小时污染物浓度与气象一次预报数据”中的污染物浓度之差的新数据集进行误差分析，对应数据统计结果如表 12 所示。

表 12 六种污染物一次预报数据与实测数据之差的对应数据统计结果表

	SO_2 监测浓度（μg/m³）	**NO_2 监测浓度（μg/m³）**	**PM10 监测浓度(μg/m³)**	**PM2.5 监测浓度(μg/m³)**	**O_3 监测浓度(μg/m³)**	**CO 监测浓度(mg/m³)**
样本总数	32498	32498	32498	32498	32498	32498
平均值	0.094233	-15.9399	17.46539	-2.50433	17.7865	0.42186
标准差	7.479913	29.15113	37.53286	30.70726	35.55939	0.262027
最小值	-103.061	-429.778	-148.909	-124.171	-264.976	-2.03303
第一四分位数 (25%)	-3.11517	-30.8417	1.8457	-12.5365	-0.00053	0.283547
第二四分位数(50%)	1.401255	-12.3255	15.62682	0.38269	13.97374	0.397692
第三四分位数 (75%)	4.35628	1.052725	30.4907	8.47442	33.25792	0.547795
最大值	58.2618	172.8807	947.0921	975.5447	269.2301	11.43387

在表中可以看到最大值和最小值都出现的明显的误差，如 PM10 的一次预报数据和实

测数据之间的差值居然达到了 947.0921 等，哪怕是在四分为上也有较为明显的偏差，所以在完成一次预测数据表和实测数据表的单独异常值处理后，必须对差值数据进行进一步的异常值处理，否则最后将会极大的影响模型的性能。处理原则如下：

- 保证数据量不至于太少；
- 保证一次预测数据和实测数据的差值误差在可忍受的范围

本文中的处理范围如下表 13 所示，这样处理之后数据仍能保证在 1 万条以上，并且一次预测数据和实测数据的差值误差不再那么的夸张。

表 13　异常值处理实现代码

```
for i in range(0,len(data)):
    if   (data['SO2'][i] <= -15)|(data['SO2'][i]>=15):
        index.append(i)
    elif (data['NO2'][i] <= -25) | (data['NO2'][i] >=25):
        index.append(i)
    elif (data['PM10'][i] <= -25) | (data['PM10'][i] >= 25):
        index.append(i)
    elif (data['PM2.5'][i] <= -20) | (data['PM2.5'][i] >= 20):
        index.append(i)
    elif (data['O3'][i] <= -25) | (data['O3'][i] >=25):
        index.append(i)
    elif (data['CO'][i] >=3):
        index.append(i)
data.drop(index=index,inplace = True)
```

在完成基本的处理之后后续的各个监测点的各类污染物浓度的训练集和测试集划分与问题三保持一致，包括后续神经网络的隐藏层神经元个数选择、各模型的输入 X 与灰色关联度分析的主要气象影响因素来选择、输入数据的零均值处理、归一化处理、等操作都与问题三保持一致。

6.2 模型的建立与求解

该题要求通过区域协同预报来达到更加有效的预测效果，本文基于问题三的解题思路为基础，结合多地区协同为标准，在问题三的改进 CS-BP 算法基础上引入了集成学习 Adaboost 算法来实现区域协同，通过多模型共同决策以期达到比单模型更优的效果，具体实现步骤如下所示：

(1) 预处理数据。
(2) 确定神经网络结构。
(3) 依据神经网络结构确定 CS 算法生成鸟蛋的维度。
(4) 利用改进的 CS 算法去寻找神经网络的较优初始权值和阈值，这里针对 4 个地区 A、A1、A2、A3 寻找 4 组神经网络的较优初始权值和阈值。
(5) 将 CS 算法的输出赋值给神经网络的权值和阈值。
(6) 神经网络训练，得到 4 个改进的 CS-BP 神经网络。
(7) 将这 4 个神经网络按照 Loss 排序（因为 Adaboost 算法对第一个网络来说较为友好，所以需要保证最优的网络为 Adaboost 算法的第一个输入网络（弱分类器），经过实验，这种方式得到的效果较好）。
(8) 将 4 个神经网络模型（弱分类器）输入 Adaboost 算法，计算它们的权值。
(9) 弱分类器权值归一化，用 4 个弱分类器去预测污染物浓度，得到的预测结果维度为（4*288）。

(10)将得到的 4 组预测数据进行加权求和得到最终的强分类器预测结果，得到的预测结果维度为（288）。

(11)将 288 个预测数据拆分为 12 组 24 个预测数据，24 为一天的预测数据，12 组为 A、A1、A2、A3 这 4 个监测点的单污染物浓度 2021/7/13、2021/7/14、2021/7/15 这三天的预测数据。

(12)最后对每组数据进行负值判断、去除最大最小、取平均值之后得到单污染物浓度的 A、A1、A2、A3 这 4 个监测点的 2021/7/13、2021/7/14、2021/7/15 三天预测数据。

(13)回到第(2)步，进行下一个污染物浓度的预测。

该题基于第三问完成，CS 算法大部分都在第三问已经说明，CS 优化模型的最优损失值如表 14 所示。

表 14　CS 优化模型的最优损失值表

	SO2	NO2	PM10	PM2.5	O3	CO
CS 优化 A 模型	0.041550	0.107100	0.057550	0.081000	0.141000	0.009150
CS 优化 A1 模型	0.041770	0.108100	0.057200	0.077000	0.140000	0.009370
CS 优化 A2 模型	0.040000	0.107500	0.057510	0.081000	0.140000	0.009120
CS 优化 A3 模型	0.039360	0.107000	0.057710	0.081100	0.138000	0.009500

这里仅对集成学习 Adaboost 算法进行补充说明。

- Adaboost 算法

Adaboost（Adaptive Boosting）算法的核心思想是使用多个具有一定差异的弱分类器组合，来实现更加有效的效果。它会将前一个分类器分类错误的样本权重增大，分类正确的样本权重减少，这样当后面的分类器在同一个错误样本上再次分类错误的时候将会得到更大的误差，从而导致这次的分类器所占权重减少。后续的分类器将重复这个过程，最后得到的强分类器将具有修正弱分类器分类错误样本的能力。本文中弱分类器就是一个三层 BP 神经网络，强分类器则为它们的组合。Adaboost 算法的基本实现过程为：

(1) 初始化训练样本权值 D，最开始所有样本权值相同，总和为 1

$$D_1 = (w_{11},...,w_{1i},...,w_{1N}), w_{1i} = \frac{1}{N}, (i = 1,2,...,N) \qquad (12)$$

式（12）中 N 为样本总数。

(2) 训练一个弱分类器，并计算该分类器的误差

$$e_m = \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i) \qquad (13)$$

式（13）中，e_m 表示第 m 个弱分类器的误差，w_{mi} 表示第 m 个弱分类器的第 i 个样本权值，I 为指示函数，G_m 为第 m 个弱分类器。

(3) 计算该弱分类器权重

$$\alpha_m = 1/2 \log \frac{1-e_m}{e_m} \qquad (14)$$

式（14）中 α_m 表示第 m 个弱分类器的权重

(4) 更新样本权重分布

$$D_{m+1} = (w_{m+1,1}, ..., w_{m+1,i}, ..., w_{m+1,N}) \quad (15)$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} e^{(-\alpha_m y_i G_m(x_i))} \quad i = 1, 2, ..., N \quad (16)$$

式（16）中 Z_m 是规范化因子，目的是使 D_{m+1} 中所有元素和为 1。

(5) 用更新的权重分布区计算下一个弱分类器的权重

(6) 构建最终的强分类器

$$G(x) = sign(\sum_{i=1}^{M} \alpha_m G_m(x)) \quad (17)$$

式（17）中最终强分类器 G(x)为所有弱分类器的线性加权和。

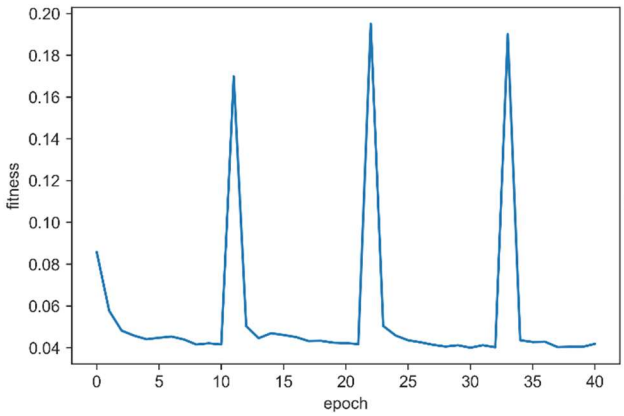
本文在 Adaboost 算法计算弱分类器误差的时候因为是预测模型，所以相应的做出了一些调整，具体改动在下表的 18-23 行，通过选择一个可接受的误差范围，在范围之内的定义为预测正确，范围之外的定义为预测错误。Adaboost 算法的具体实现代码如表 14 所示。

表 15　Adaboost 实现代码

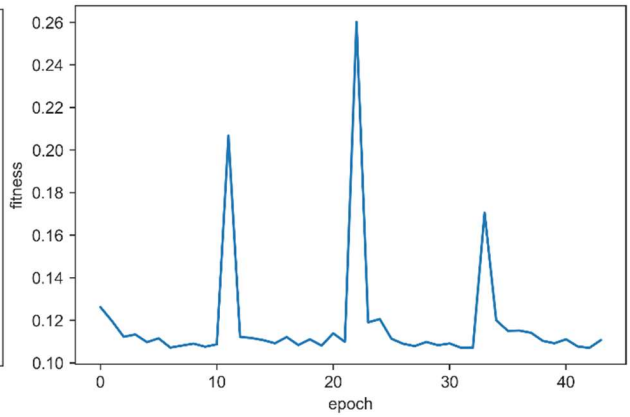
```
1  y_pre_train = []
2  # 初始化数据权重
3  w_data = np.ones(n_train) / n_train
4  # 初始化模型权重
5  w_model = np.zeros(M)
6  #train_y = tf.argmax(Y_onehot,axis=1)   #train_y 保存了原始训练集中标签经过 onehot
7  编码后的结果
8  #k = 3 #分类数量
9  for i in range(M):
10     miss = []
11 #求第 i 个弱分类器训练集结果
12 y_pred_train = Models[i].predict(PM10X)   #三元概率     !!!!!!!!!!!!!!!!!!!!
13 y_pred_train = np.squeeze(y_pred_train)
14 temp = np.array(abs(y_pred_train - PM10Y))   #!!!!!!!!!!!!!!!!!!!!!!!!!!
15 #y_pred_train = tf.argmax(y_pred_train,axis=1)#去最大概率的位置为预测值
16 #求第 i 个弱分类器误差
17    for k in range(0,len(y_pred_train)):
18     if temp[k] > 0.5:     #!!!!!!!!!!!!!!!!!!!!!!!!!!
19             miss.append(1)
20         else:
21             miss.append(0)
22     #miss = [int(x) for x in range(0,len(y_pred_train)) (y_pred_train != train_y)]   #不相
23 等则保存 1，相等则保存 0
24     error =np.dot(w_data, miss) #累加识别错误的样本权重，得到分类器误差
25     print('第 %d 个模型的误差是：%.2f'%(i,error))
26     #求第 i 个弱分类器权值，保存到 w_model 中
27     #a = 1/2 * log(1-e/e) + log(k-1),当 k = 2 时为二分类更新权值公式不用修改，否
28 则为多分类，算法准确率大于 1/k 即可
29     #该函数若准确率大于 0.5（1-error）则值为正，否则为负值，越大说明模型分类
30 越好
31     w_model[i] = 0.5 * np.log((1-error)/error) #+ np.log(k - 1)
32
33
```

```
        # 更新数据权重
        #分类结果和真实的结果一致，那么结果是-w_model[m]，是一个负值，
        #那么 exp(-w_model[m]*train_y[i]*y_pred_train[i]) 结果小于 1。也就是说该数据集的样本权重降低。否则该数据样本的权重增高。
        #通过这种计算就可以让那些容易分错的样本的权重升高，容易分对的样本权重降低。继续迭代就会导致对难分的样本能分对的模型的权重上涨。
        #最终，达到一个强分类器的目的。
        #注意，这里只适合二分类[1，-1]
        #多分类公式修改 wt = wt-1 * exp(at * (y_true!=y_pred))
        miss1 = np.array(miss)
        miss1 = w_model[i]*miss1
        for j in range(n_train):
        w_data[j] = w_data[j] * np.exp(miss1[j])  #*train_y[i]*y_pred_train[i] #二分类时用这个
        #正则化数据权值
        Z = np.sum(w_data)
        for j in range(n_train):
            w_data[j] /= Z
#结果这个模块以后将得到每个模型的权值，保存在 w_model 中
#弱分类器权值归一化
w_model = np.array(w_model / np.sum(w_model))
result1 = w_model[0]*models_pre_ds[0]
for i in range(1,M):
        result1 += w_model[i] * models_pre_ds[i]
```

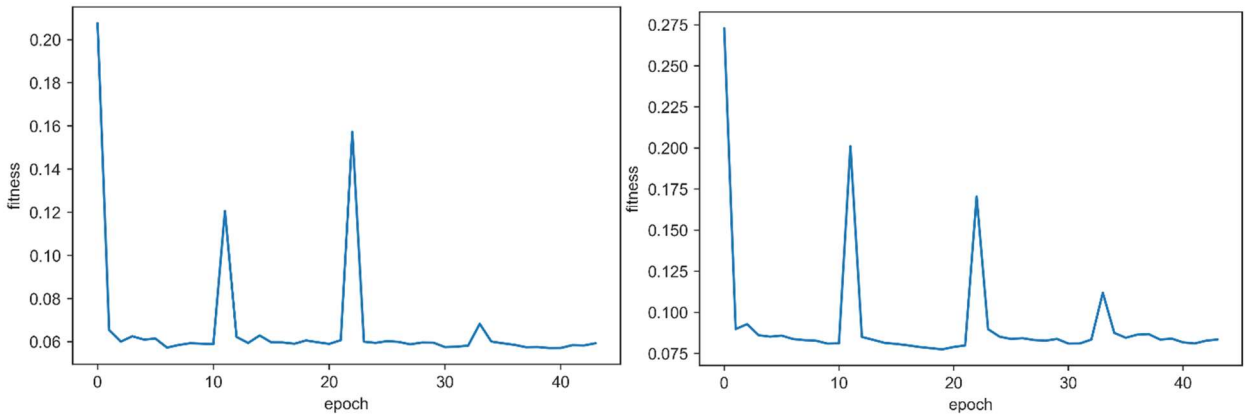
下图 8 为本问题的改进 CS 算法寻优效果，每个图为单污染物浓度的 A、A1、A2、A3 这 4 个监测点的效果总览，其中 epoch 每 10 个为 1 个检测点的参数寻优效果，通过观察可以发现算法寻优效果较为稳定且效果良好。



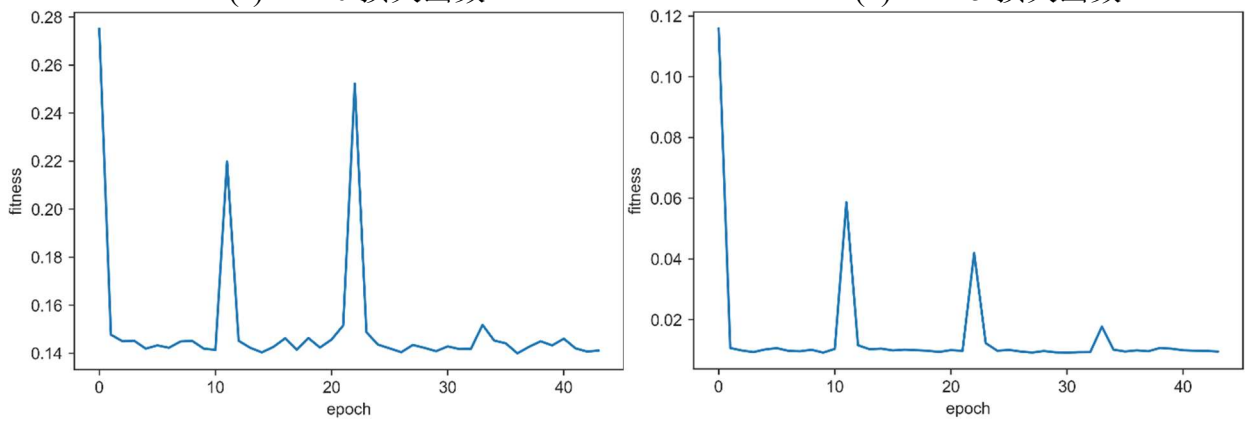
(a) SO_2 损失函数



(b) NO_2 损失函数



(c) PM10 损失函数　　(d) PM2.5 损失函数



(e) O_3 损失函数　　(f) CO 损失函数

图 8　各项污染物模型训练损失函数

在将 CS 优化得到的神经网络权值和阈值赋值给神经网络并完成 100 个 epoch 的训练以后，按照本题的实现步骤完成弱分类器的权值确定，以及强分类器的构建并预测得到最终的 288 条预测结果数据，模型的预测数据选择 A、A1、A2、A3 四个检测点的“监测点 X 逐小时污染物浓度与气象一次预报数据”的 2021/7/13 预测的 2021/7/13、2021/7/14、2021/7/15 号的数据，每个监测点共 96 条。

在得到 288 条最终结果后需要对其做最后的处理，首先将其分为 12 组，即单污染物 2021/7/13、2021/7/14、2021/7/15 号在监测点 A、A1、A2、A3 的数据，之后检测是否存在负值，去除 3 组最大值和 3 组最小值，完成处理之后将各个监测点每天剩余的数据计算其平均值，得到最终预测的 2021/7/13、2021/7/14、2021/7/15 号的 A、A1、A2、A3、A4 共 12 条单污染物浓度值。在将这个过程循环 5 次，预测出剩余 5 种污染物浓度的所有数据即可。最终神经网络训练结果如表 16 所示。

表 16　最终神经网络训练结果

	SO2	NO2	PM10	PM2.5	O3	CO
神经网络 A 模型	0.040300	0.102000	0.05630	0.078000	0.138000	0.008700
神经网络 A1 模型	0.039500	0.100000	0.05560	0.078300	0.137000	0.008900
神经网络 A2 模型	0.039000	0.098200	0.05420	0.080000	0.134000	0.008900
神经网络 A3 模型	0.037500	0.097500	0.05210	0.079000	0.135500	0.008900

使用该模型预测监测点 A、A1、A2、A3 在 2021 年 7 月 13 日至 7 月 15 日 6 种常规污染物的单日浓度值结果如表 17、表 18、表 19、表 20 所示。

表 17　监测点 A 在 2021 年 7 月 13 日至 7 月 15 日 6 种污染物的单日浓度值预测结果

预报日期	地点	二次模型日值预测							
		SO_2 (μg/m³)	NO_2 (μg/m³)	PM_{10} (μg/m³)	$PM_{2.5}$ (μg/m³)	O_3 最大八小时滑动平均 (μg/m³)	CO (mg/m³)	AQI	首要污染物
2021/7/13	监测点 A	8.13	27	17.15	7.6	32	0.26	34	空气质量评价为“优”，当天无首要污染物
2021/7/14	监测点 A	6.87	26	18.24	6.11	40	0.239	33	空气质量评价为“优”，当天无首要污染物
2021/7/15	监测点 A	6.45	30	18.44	8.95	23.3	0.3	38	空气质量评价为“优”，当天无首要污染物

表 18　监测点 A1 在 2021 年 7 月 13 日至 7 月 15 日 6 种污染物的单日浓度值预测结果

预报日期	地点	二次模型日值预测							
		SO_2 (μg/m³)	NO_2 (μg/m³)	PM_{10} (μg/m³)	$PM_{2.5}$ (μg/m³)	O_3 最大八小时滑动平均 (μg/m³)	CO (mg/m³)	AQI	首要污染物
2021/7/13	监测点 A1	7.6	17.3	19.6	10	27.3	0.156	22	空气质量评价为“优”，当天无首要污染物
2021/7/14	监测点 A1	6.13	18.3	20.16	7.67	34.8	0.152	23	空气质量评价为“优”，当天无首要污染物
2021/7/15	监测点 A1	6.91	29	18.1	8.9	41.5	0.222	37	空气质量评价为“优”，当天无首要污染物

表 19　监测点 A2 在 2021 年 7 月 13 日至 7 月 15 日 6 种污染物的单日浓度值预测结果

预报日期	地点	二次模型日值预测							
		SO_2 (μg/m³)	NO_2 (μg/m³)	PM_{10} (μg/m³)	$PM_{2.5}$ (μg/m³)	O_3 最大八小时滑动平均 (μg/m³)	CO (mg/m³)	AQI	首要污染物
2021/7/13	监测点 A2	8.9	19.8	17.1	6.86	29.7	0.176	25	空气质量评价为“优”，当天无首要污染物
2021/7/14	监测点 A2	6.82	20.5	18	6.3	46	0.186	26	空气质量评价为“优”，当天无首要污染物
2021/7/15	监测点 A2	5.52	32	17.07	8.2	35	0.25	40	空气质量评价为“优”，当天无首要污染物

表 20　监测点 A3 在 2021 年 7 月 13 日至 7 月 15 日 6 种污染物的单日浓度值预测结果

预报日期	地点	二次模型日值预测							
		SO_2 (μg/m³)	NO_2 (μg/m³)	PM_{10} (μg/m³)	$PM_{2.5}$ (μg/m³)	O_3 最大八小时滑动平均 (μg/m³)	CO (mg/m³)	AQI	首要污染物
2021/7/13	监测点 A3	6.5	23	17.15	7.69	33	0.25	29	空气质量评价为“优”，当天无首要污染物
2021/7/14	监测点 A3	4.73	22	18.04	6.41	30	0.25	28	空气质量评价为“优”，当天无首要污染物
2021/7/15	监测点 A3	6	34	17.82	10.8	27	0.33	43	空气质量评价为“优”，当天无首要污染物

七. 模型检验与评价

7.1 模型优点

在对数据进行充分了解之后完成足够的数据预处理是取得好的模型效果的关键所在，若是预处理做得不好模型是不可能有好的效果的，本文模型在经过了大量的预处理之后使用周期变化差值结合灰色关联度的方式完成气象条件的分类，并使用神经网络为基本框架，结合布谷鸟算法、Adaboost 算法共同完成了后续问题的解决。针对问题二，本文通过观察“监测点 A 逐小时污染物浓度与气象一次预报数据”和“监测点 A 逐小时污染物浓度与气象实测数据”的不同，分别采用了灰色关联度和周期变化差值来做出不同方式的气象条件分类。针对问题三，在问题二的灰色关联度分析得到的各污染物主要气象影响因素的前提下，本文使用了布谷鸟算法来优化神经网络的权值和阈值，使其在训练的时候能够快速的达到较优的效果以及较小的损失。另外，本文针对布谷鸟算法的随机初始化初始值做出了改进，采用了混沌初始化结合反向初始化的方式使得布谷鸟算法的初始值能够在解空间中具备更强的表达能力、分布更加均匀并且能够快速的收敛，并将反向的方式应用到了布谷鸟算法的局部寻优部分，使其寻优能力进一步提升，达到更好的效果。最后将布谷鸟算法优化的神经网络权值和阈值赋值给神经网络，取得了不错的效果。针对问题四，本文在问题三的基础上，为了实现区域协同预报的效果，引入了 Adaboost 算法，通过构建 A、A1、A2、A3 这四个区域的 CS-BP 模型，并将其使用 Adaboost 算法的方式结合起来共同完成污染物浓度的预测，使用其他相关区域的模型效果来修正预测地区可能出现的错误预测情况，并针对各个污染物的数据特征选择了不同的预测正确与错误分类界限，以此来计算适应各个污染物浓度的弱分类器误差。

最后，考虑到了一次预报数据预测精度不高的情况，以及题目中说明的间隔日期越长预测效果越差的情况，采用了最近日期的数据作为预测数据，以及对最终的单检测点单天单浓度预测的 24 条预测数据（每小时一条）进行了最后的预防处理，通过检测其是否存在预测负值以及去掉各 3 条最大最小数据，保证留下了预测效果不错的部分数据，最后计算这些数据的均值得到了最终的模型预测结果，具备一定的科学性和合理性。

7.2 模型缺点

本次大赛由于一开始走错了方向导致浪费了很多时间，后续模型的构建很多地方还不够成熟，对于模型的稳定性以及结构的合理性还有待优化，如于神经网络的结构以及 Adaboost 算法进行模型误差计算时的计算方式都显得有点随意等。除此之外，包括数据的处理也有很多地方没有经过验证，具有明显的主观性，如一次预测值和实测值之间的差值的异常值处理，筛选的范围就比较主观，没有经过太多的实验验证。

最后，感谢本次大赛给我们提供了这次机会，使我们得到了一次充分的锻炼和对自己学习成果的检验。经过这次比赛我们学到了很多，也发现了我们很多的不足，包括团队之间的配合以及磨合都是我们学习到的宝贵经验。希望以后能继续保持这种面对困难时不放弃的精神，通过大家的共同努力、配合才能解决掉种种的困难，最后将达到个人难以实现的结果。

八. 参考文献

[1] 郝吉明, 马广大, 王书肖. 大气污染控制工程 [M]. 北京: 高等教育出版社, 2010.

[2] 薛文博, 王金南, 杨金田,等. 国内外空气质量模型研究进展[J]. 环境与可持续发展, 2013, 038(003):14-20.

[3] Pasquill F, Michael P. Atmospheric Diffusion, 2nd edition[J]. Physics Today, 1977, 30(6):55-57.

[4] Byun D W , Ching J . Science algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System[J]. Nerl, 1999.

[5] X. Yang and Suash Deb, "Cuckoo Search via Lévy flights," 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009, pp. 210-214, doi: 10.1109/NABIC.2009.5393690.