**涉密论文** □ **公开论文** □



**本 科 生 毕 业 论 文（设计）**

**题目** **垃圾焚烧中的烟气预测研究**

**姓名与学号 周天昱 3140102448**

**指导教师 邓水光**

**年级与专业 软件工程1402**

**所在学院 计算机科学与技术**

**提交日期 2018年5月30日**

**浙江大学本科生毕业论文（设计）承诺书**

1. 本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2. 本人在毕业论文（设计）中除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 **浙江大学** 或其他教育机构的学位或证书而使用过的材料。

3. 与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

4. 本人承诺在毕业论文（设计）工作过程中没有伪造数据等行为。

5. 若在本毕业论文（设计）中有侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

6. 本人完全了解 **浙江大学** 有权保留并向有关部门或机构送交本论文（设计）的复印件和磁盘，允许本论文（设计）被查阅和借阅。本人授权 **浙江大学** 可以将本论文（设计）的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编本论文（设计）。

作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

# 致 谢

感谢我的毕业论文指导老师邓水光教授，从选题、开题、工作开展以及难题解决上面为我提供了许多充满建设性的建议和帮助，邓老师的悉心教导令我获益良多，一方面获得了关于论文开展的工作方法和严谨态度，另一方面激起了对于科研工作的兴趣和激情。

感谢我的毕业设计指导学长蒋芳乔学士，蒋学长在机器学习、时间序列方面为我答疑解惑，提供许多新颖的思路和方案，让我在面对论文工作展开的困难时能够披荆斩棘，勇往直前。同时，蒋学长用自己严谨认真的科研态度，在激励我不断的前进。

感谢我的毕业设计指导学长柳洋学士，柳学长和我一起探讨了工作的开展方向和可能的方案思路，并在论文逐渐完成的过程中一直给予我鼓励和帮助。

感谢我的室友蒋晨书，在和蒋同学的探讨当中我产生了关于数据划分以及模型选用的思路，最终决定了论文的走向。

感谢我的同学吴郁文，吴同学在我困惑、失落、难过、徘徊不前的时候给我了无微不至的关怀和帮助，让我鼓起勇气解决难题，让我有信心一直将论文完成。

最后要感谢我的母亲，在我的成长路上给我的谆谆教诲，给我的呵护和鼓励，感谢我的父亲，为我的学业和生活提供了有力的支持。

# 摘 要

现阶段的垃圾处理工业当中，焚烧是一种比较普遍的处理手段。出于环保需要，垃圾焚烧产生的有害烟气需要及时的进行处理。由于缺乏有效的烟气预测手段，且传统模型依赖垃圾成分分析，因此在实际的垃圾焚烧工业当中，往往是根据当前监测到的烟气数值，人工调节垃圾入炉焚烧速率和处理烟气的环保耗材投放量，具有一定的滞后性。本文在此背景下，针对以炉排炉为焚烧技术的垃圾焚烧烟气预测问题，提出了不依赖垃圾成分分析的基于机器学习理论的预测模型。

本文通过对垃圾焚烧工况数据的特性研究，合理筛选可用数据、确定时间间隔、设计样本形式，形成了可供模型训练的，较为准确且理想的数据集，同时形成了一套完整的工况数据筛选流程，并为后续的工况数据标签加入提供了相应的方式，最后在针对数据清理和变换上提供了较为合理的方法。

本文结合了机器学习相关理论，将普通最小二乘法、岭回归法、普通循环神经网络、GRU神经网络分别具体实现，按照数据集的特性，分不同的预测烟气探究了最适合的的时间窗口大小、网络节点个数等各种模型参数。在神经网络模型当中进行了激活函数的测试，选取了较优的一种。

本文对上述四种模型在不同烟气的预测环境下进行了测试，在短期预测上线性模型的预测效果相对较好，且训练速度快；而在较长的时间间隔上，神经网络的预测较为理想，尽管该模型的训练时间较长。不同烟气的预测效果，由于烟气分布规律的不同，会产生较大的差异。

本文最终通过数据处理、模型设计、效果对比，得到了不依赖垃圾成分分析和垃圾入炉速率，仅参考历史工况数据的垃圾焚烧烟气预测模型。该模型在短期预测上具有较大的应用价值。未来的研究，基于更多的高精准度的工业传感器和封闭环境下的焚烧实验，将集中在如何提高长期预测的准确性和分析烟气排放历史规律等问题上。

**关键词：**烟气预测；回归模型；监督学习；数据处理；循环神经网络

Abstract

In the waste treatment industry at this stage, incineration is a relatively common treatment method. For environmental protection needs, hazardous flue gas generated from waste incineration needs to be processed in a timely manner. The current lack of effective smoke forecasting methods. Traditional models rely on garbage composition analysis. Therefore, the amount of waste and chemicals are usually adjusted manually based on the current smoke value. This has some lag. Under this background, this paper proposes a prediction model based on machine learning theory, which does not depend on garbage analysis, for the problem of waste incineration of flue gas with grate furnace as incineration technology.

In this paper, through the study of the characteristics of the garbage incineration data, the available data are selected reasonably, the time interval is determined, and the sample form is designed to form a more accurate and ideal data set that can be used for model training. At the same time, a complete set of working conditions is formed. The data screening process provides a corresponding way for the subsequent labeling of working data, and finally provides a more reasonable method for data cleansing and transformation.

In this paper, the theory of machine learning is combined, and ordinary least square method, ridge regression method, common circulatory neural network, and GRU neural network are implemented respectively. According to the characteristics of the data set, under the different smoke, find the most suitable time window, various model parameters such as size and number of network nodes. In the neural network model, the activation function was tested and the better one was selected.

In this paper, the above four models are tested in different smoke prediction environments. In the short-term prediction, the linear model has a better forecasting performance, and the training speed is faster; but in a longer time interval, the neural network prediction is more ideally, although the model has a long training time. The prediction performance of different smoke will produce a large difference due to the different distribution of smoke.

This article discusses about the data processing, model design, the performance comparison. Without the garbage composition analysis and the speed of garbage pushed into the furnace, only referencing to historical operating conditions of the waste incineration, the flue gas forecasting model is created. This model has great application value in short-term prediction. Future research, based on more high-precision industrial sensors and incineration experiments in closed environments, will focus on issues such as how to improve the accuracy of long-term predictions and analyze historical patterns of smoke emissions.

**Key words**: Flue gas prediction; Regression model; Supervised learning; Data processing; Recurrent neural network

# 目 录

**第一部分 毕业论文**

1 绪论 1

1.1 课题背景及意义 1

1.1.1 课题背景 1

1.1.2 本研究的意义 2

1.2 研究现状及本文贡献 2

1.2.1 研究现状 2

1.2.2 本文贡献 3

1.3本文结构 4

2 线性模型和循环神经网络概述 5

2.1 线性模型 5

2.1.1 普通最小二乘法 5

2.1.2 岭回归 7

2.2 循环神经网络 9

2.2.1 神经网络 9

2.2.2 SimpleRNN 12

2.2.3 LSTM 13

2.2.4 GRU 15

2.3 激活函数和损失函数 16

2.3.1 激活函数 16

2.3.2 损失函数 18

2.4 优化器 18

3 垃圾焚烧工况数据处理 20

3.1 线性模型 22

3.1.1 接口测试 22

3.1.2 数据初步采集和可视化 23

3.1.3 数据特性研究和标签筛选 25

3.1.4 数据库存储设计 27

3.2 数据集成 27

3.3数据清理 28

3.4 数据变换 30

3.4.1 数据归一化 30

3.4.2 数据重组 31

4 烟气预测模型设计 32

4.1 模型概述 32

4.1.1 输入样本 33

4.1.2 输出样本 36

4.1.3 模型内容 37

4.1.4 测试及可视化 41

4.2 线性模型 41

4.2.1 最小二乘法可变参数设计 41

4.2.2 岭回归可变参数设计 47

4.3 神经网络模型 49

4.3.1 时间窗口设计 51

4.3.2 激活函数选择 52

4.3.3 节点个数设计 54

5 测试与分析 56

5.1 测试目标 56

5.2 测试策略 57

5.3 测试结果分析 59

5.3.1 模型测试结果与分析 59

5.3.2 模型对比与分析 62

6 总结与展望 65

6.1 全文成果总结 65

6.2 研究展望 66

参考文献 67

作者简历 68

《浙江大学本科生毕业论文（设计）任务书》

《浙江大学本科生毕业论文（设计）考核表》

**第二部分 文献综述和开题报告**

文献综述和开题报告封面

指导教师对文献综述和开题报告具体内容要求

目录 I

一、文献综述 1

二、开题报告 6

三、外文翻译 13

四、外文原文 19

《浙江大学本科生文献综述和开题报告考核表》

第一部分

**毕业论文**

# 1 绪论

## 1.1 课题背景及意义

### 1.1.1 课题背景

人们的日常生活当中每时每刻都在产生垃圾，如外卖吃完之后剩下的餐盒，瓜皮果核等。在中国，绝大部分的垃圾都会统一处理。

改革开放至今，人民的生活水平在不断的提高，相应的，生活当中产生的垃圾数量也在不断增加。据统计，我国每年产生的城市生活垃圾达到了1.5亿吨，并正在以每年10%的速度不断增长，历年产生的垃圾堆积量达到了60多亿吨，直接占据土地面积达5000多平方公里[1]。

现阶段的垃圾处理工业当中，焚烧是一种比较普遍的处理手段。通过将垃圾进行高温焚烧处理，消灭垃圾当中的有害生物病菌；同时焚烧之后的残渣只有原垃圾体积的20%左右[2]；而焚烧当中产生的热能则转化为电能投入电网。因此垃圾焚烧技术不仅使垃圾减量、生物无害，同时还可以转化能源发电。

当前国内外的垃圾焚烧技术主要有三类：层状燃烧技术、流化床式燃烧技术和旋转燃烧技术。其中层状燃烧技术最为成熟，欧洲、美国和日本都有基于该技术的垃圾焚烧厂，这种技术的优点是处理容量大、运行管理容易，缺点是燃烧效率低。该技术在我国也有广泛的应用。

由于燃烧效率低，该方法产生的有害气体如、等较多。原先国家对于环保排放的要求不高，企业普遍对于烟气的处理重视度不够，现在国家加强了环保监管的力度，所以如何在环保达标的情况下进行垃圾焚烧，就成为了垃圾焚烧企业关注的课题。

垃圾焚烧当中产生的烟气，主要有5种，二氧化硫、氮氧化物、一氧化碳、烟尘和氯化氢。由于缺乏有效的烟气预测手段，因此在实际的垃圾焚烧工厂当中，往往是根据当前监测的到烟气数值进行人工调节垃圾入炉焚烧速率和处理烟气的环保耗材投放量，具有一定的滞后性。

因此，在实际的生产工业当中，为了能够及时的控制烟气的排放，需要更早的了解烟气的排放趋势，以便尽早做出反应，防止烟气排放超标，所以需要能够预测烟气排放指标的模型。

### 1.1.2 本研究的意义

垃圾焚烧当中的烟气处理，主要有前后两个环节——先是预测烟气产生量的变化，然后根据产生烟气的量动态调节环保耗材（NaOH、熟石灰等）的投放量。

由于现在缺乏有效的预测手段，无法给出烟气指标的变动趋势，所以在实际工业场景当中，往往是直接监控当前烟气的排放量，在发现烟气指标超出环保标准之后，手动减少垃圾焚烧量，由于焚烧垃圾量减少，因此相应的，废气排放自然也就减少了。显然在这种情况下，烟气的排放减少距离采取措施有相当的一段时间间隔。

这样带来的结果就是较大的经济损失（由于垃圾焚烧量降低，发电量和处理效率都会降低）和烟气排放指数的短时间频繁超标。为了改变这种现状，烟气预测就成为了较为重要的课题。

本研究的意义是通过提出烟气预测的模型，来解决在烟气处理当中的烟气指标的变动趋势预测问题，从而提升垃圾焚烧工厂的烟气处理能力，在环保达标的前提下，提升垃圾焚烧企业的收益。

通过大幅推广基于烟气预测的烟气处理技术，推进工业烟气处理自动化，带来更多的收益和环保价值。

## 1.2 研究现状及本文贡献

### 1.2.1 研究现状

垃圾焚烧当中产生的有害气体，需要进行及时的处理。为了达到符合国家环保要求的指标，一方面需要完善焚烧工艺，提高焚烧效率，另一方面要监测排放烟气浓度，通过添加化学材料的方法中和酸性气体，减小有害气体的排放量。

由于垃圾产量的不同（国内人口密度远远大于国外），在国外相关研究当中，有害气体的关注度并不大，研究相对集中在重金属和温室气体（甲烷）的排放上。国内的研究进展也相对较慢。

在20世纪末期21世纪初期，有针对垃圾焚烧排放气体的研究。当时对于气体的预测模型是基于焚烧的化学原理，在焚烧物成分固定的情况下，建立成套的化学反应方程式模型[3]。

近年来，更多的研究则是集中在温室气体或是重金属的排放问题上。其中对于生活垃圾的焚烧过程中的重金属排放，就有一系列的动力学模型，例如， Local CFD kinetic model[4]，还有关于Cd，Pb和Zn排放的动力学规律[5]。在韩国，有对于，和排放量的检测，在不同燃烧物、不同工艺、不同气体浓度的情况下，温室气体的排放变化[6]；也有估算垃圾焚烧厂的排放因子，并基于这些因素计算排放量[7]。在瑞士，研究了固体废物燃烧当中一氧化二氮和甲烷的排放问题[8]。也有基于地面的遥感方法对于难以监测的气态副产物进行图像测量的研究[9]。

我国垃圾焚烧的相关研究也以温室气体和重金属排放居多。在研究气体排放方面，多是通过改善工业流程来实现减排的目标[10]。改善条件诸如焚烧温度等工业环节，同时根据季节性因素来进行调节。针对单个垃圾焚烧锅炉组的气体排放往往不是关注的重点，研究多是围绕宏观的区域排放量来展开的。

值得一提的是，浙江大学的张东平博士提出了基于BP神经网络的酸性气体预测模型，该模型针对的是流化床焚烧技术。基于燃烧垃圾成分分析建立了酸性气体的BP神经网络模型，并考察了模型的性能，总体上来看尽管有所误差，但是结果较为准确，误差在10%左右[11]。

针对焚烧垃圾时的气体排放问题，国内外的关注点往往集中在宏观上的排放和对于大气环境的影响上，只有少数研究着眼于具体锅炉技术的烟气排放。这是因为在实际的工业当中，环保指标较低，对于排放的重视度也不高。所以关于垃圾焚烧的烟气预测模型很有限。涉及到垃圾焚烧烟气排放时，也往往考虑优化工业流程，增加设备或是改进工艺等等，这对于已经投入使用的垃圾焚烧厂来说，这样的改进方法会使得成本大量增加，存在经济效益上的困难。

### 1.2.2 本文贡献

本文作者的主要贡献是结合机器学习的相关理论，建立了基于垃圾焚烧设备实时工况数据的烟气预测模型。

本文提出的模型不依赖垃圾成分的分析，仅根据垃圾焚烧工厂在进行垃圾焚烧处理时的各种工业监控器采集的历史数据进行预测。通过将工况数据进行处理，划分时间窗口、确定模型参数，利用历史数据进行训练，可以得到一个在短期预测上面表现良好的模型。

工况数据的处理包括传感器采样频率的获取，时间间隔的划分，时间窗口的设计，特征矩阵的建立以及数据标准化等处理流程。模型依赖的机器学习方法主要是线性回归和循环神经网络。本文根据具体的工况数据特性，进行模型参数的选择和结构优化，得到了相应的具有较好表现力的模型。

本文在提出模型的基础上，展望了后续预测模型的改进思路和方向，同时分析了不同技术的优缺点。

## 1.3 本文结构

本文第1章介绍了与课题有关的背景和意义，阐述了垃圾焚烧烟气预测领域的研究现状和本文在该领域的贡献，并介绍了本文的结构。

第2章介绍了本文提出模型所用到的机器学习理论，包括线性回归和循环神经网络。线性回归包括普通最小二乘法和对系数的大小施加惩罚的岭回归法。循环神经网络包括SimpleRNN、LSTM和GRU。本章同时还介绍了在设计神经网络模型时候用到的激活函数和损失函数，并且简要说明了加速模型训练的优化器的作用。

本文作者的主要工作在第3、4、5章。

第3章详细介绍了垃圾焚烧工况数据处理的过程，包括从数据接口获取数据，通过数据可视化人工筛选有用数据标签，按照一定的时间间隔划分数据，填补空缺值和处理异常值。最后按照时间窗口集成数据再进行标准化。

第4章详细描述了模型的设计过程，包括线性回归模型——普通最小二乘法模型和岭回归模型，以及循环神经网络模型——SimpleRNN、GRU。每种模型的调整和设计都有不同的地方，该章当中详细描述了相关的调整工作以及最终选择的方案。

第5章是介绍模型的测试和分析。由于应用了多种机器学习的理论模型，为了客观评价以及选择最优的模型，在该章节当中进行了多种模型性能的对比，包括预测能力的分数、模型预测的误差大小等。

第6章总结了全文的研究成果，并对未来的研究方向和思路进行了展望。

# 2 线性模型和循环神经网络概述

## 2.1 线性模型

假定有一组向量，将 看成是一组多维属性值，其中， 是 在第维度上面的属性值。我们希望得到一个模型，该模型以 作为输入，输出的值 与输入有如下对应

其中 都是常数。我们称该模型为线性模型，这样的关系为线性组合。一般用向量形是写成

其中。在通过训练得到和之后，线性模型就确定下来了。

由上面的公式可以看出，线性模型的形式简单，较为容易建立模型。由于每个输入维度 都有相应的系数 ，在训练结束之后，可以通过系数来判断每个属性在预测当中的重要程度。

我们称这样的模型具有良好的可解释性(comprehensibility)[12]。

### 2.1.1 普通最小二乘法

最小二乘法是一种优化技术，利用数学的方法，通过最小化残差的平方来寻找与数据集最佳匹配的函数。此处有一个较为重要的概念——残差，这是观测值(数据集当中的数据)与拟合值(输入值经过模型得到的值)的差值。通过最小化残差的平方来寻找最优拟合的函数，称之为最小二乘法。

最小二乘法通常被认为是高斯提出的(Carl Friedrich Gauss，1795)[13]，但事实上是阿德里安-马里·勒让德(Adrien-Marie Legendre)率先发表的[14]。

考虑最简单的一种情况，输入的维度只有一维，即训练数据是一组如的序列，其中是输入观测值，是输出观测值。我们希望得到一个线性模型，有如下的表达式

运用残差平方和来衡量模型的拟合程度，即求得一组使得残差平方和最小

记为当系数为时模型的残差平方和，为了使得最小，分别对和求偏导，得到

函数的极值点即为偏导为的点，令式和为零，可以得到使得最小的解

其中，，是的平均值。

上述考虑的情况是一元线性模型的最小二乘法，更为普遍的情况是多元的线性模型，即一开始提到的多维向量，以及对应的。这时候模型为

这时候的回归即为多元线性回归，同样我们可以利用最小二乘法求得最适合的。为了更方便的表示，将组合成向量形式，同样的，我们将数据集表示为一个矩阵，其中为数据的个数，为数据的维度，每行相当于一个观测值，前个数据为属性值，最后一个元素恒为，同理将组合成维度为的向量。效果如下

于是模型就变为了

其中

那么现在就需要求得最优的，类似于之前的式，有

此时的残差平方和，对进行求导，得到

和一元线性回归的方法一样，此时令式等于，可以得到的最优解，但由于上述公式中有矩阵逆的计算，所以比一元变量的情况复杂，此时需要分两种情况讨论。

第一种情况是为满秩矩阵(full-rank matrix)或者是正定矩阵(positive definite matrix)。此时令式等于，得到

其中是矩阵的逆矩阵。

第二种情况则是更为普遍的一般情况，此时不是满秩矩阵。这个时候可以解出多个 使得残差平方和最小，这个时候选择哪个将由具体的算法决定，常见的作法是引入正则化(regularization)[15]。

### 2.1.2 岭回归

岭回归，也叫做吉洪诺夫正则化(Tikhonov regularization)，在统计学当中通常称为岭回归(ridge regression)，在机器学习当中，称之为权重衰减(weight decay)[16]。

在上文介绍最小二乘法的时候，讲到求解多元线性模型的最优参数矩阵，表达式如下所示

当式当中的不是列满秩或者某些列的属性值相关性比较大(即原来的观测值向量并不是每个属性之间相互独立)的时候，的行列式值接近于，这个时候，计算它的逆矩阵时会有较大的误差。因此在这种情况下，上文描述的普通最小二乘法，缺乏准确性和可依赖性。

为了解决这种问题，引入了新的方法——岭回归。

将原先的残差平方和函数

稍作修改，改成残差平方和加上一个正则化项，形式如下

此处的正则化项当中的，是单位矩阵，是控制系数 收缩量的复杂性参数， 越大，收缩量越大，这样系数对共线性的鲁棒性也更强。

从而最优参数矩阵的表达式变为

由式可以看出， 越大，当中的各个元素也在向零靠近，它们和正确值的误差绝对值也越大。当时，当中的元素趋向于0。我们可以绘制出关于的曲线，称之为岭轨迹。当曲线在一个值附近趋于稳定，那么这个值就是合适的取值。

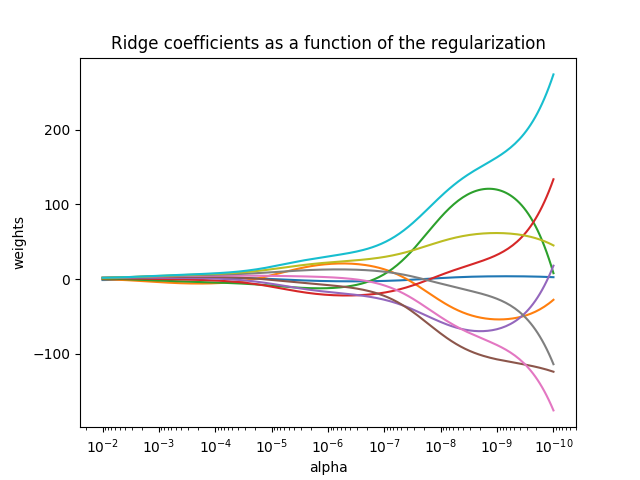


图 2.1 系数矩阵随的变化图

上图显示了岭回归在高度病态数据集当中的应用，所对应的岭轨迹图。所谓“病态”的矩阵，即数据矩阵元素的轻微变动会导致计算得出的权重矩阵产生巨大差异。在这种情况下，岭回归通过设置正则化参数来增强数值的稳定性，进而得到较为准确的线性模型。

可以看到，当很大时，岭回归误差计算函数当中的正则化项占绝对主导，因而求得的系数趋于零。在岭轨迹的末尾，由于趋于零，得到的解类似于普通最小二乘，求得的系数表现出很大的振荡。在实践中，需要调整以使两者之间保持平衡。

## 2.2 循环神经网络

### 2.2.1 神经网络

本文讲到的神经网络，不涉及生物领域，即非生理构造，而是完全由人类设计的数学模型，成为人工神经网络(Artificial Neural Network，简称ANN)。人工神经网络并非是凭空产生，是人类仿造生物神经网络的构造构建出的数学模型，其中每个运算节点类似于生物的神经元。

本文由于篇幅限制，仅仅介绍神经网络的基本要素和循环神经网络的具体模型。

**1） 神经元**

ANN当中的神经元，也叫做神经节点，类比人类神经元特性，是由多个输入、处理函数、多个输出组成的计算节点。如下图2.2所示

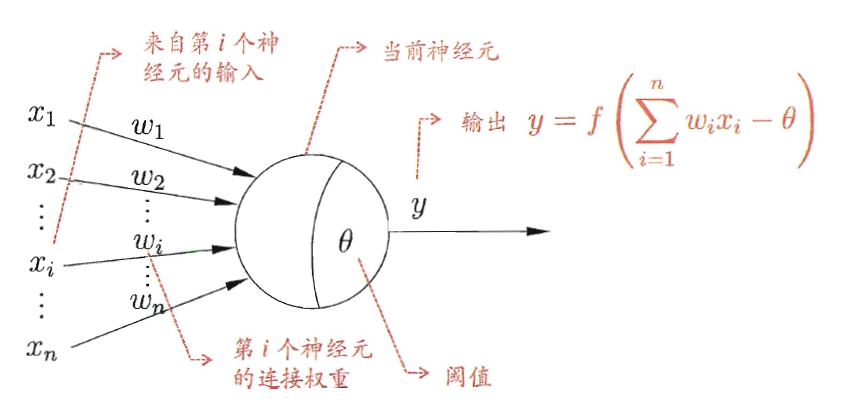


图 2.2 人工神经元

该神经节点接收到的多个输入可能是来自之前的神经节点的输出，也有可能是来自数据集的输入，这里的关于的函数 就是上文提到的处理函数，最后得到的可能是一维的，也有可能是多维的，主要看之后连接到该节点的神经节点个数以及该节点在网络当中所处的位置。

**2） 感知机与多层神经网络**

由两层神经节点组成的网络成为感知机(Proceptron)，如下图2.3。输入层接受数据集的输入，传递给输出层的神经节点，经过处理函数之后输出。

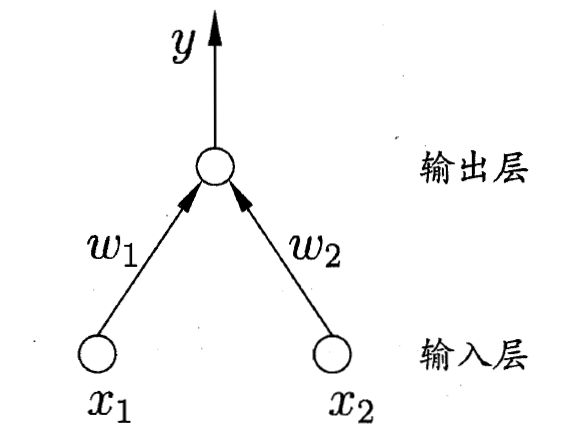
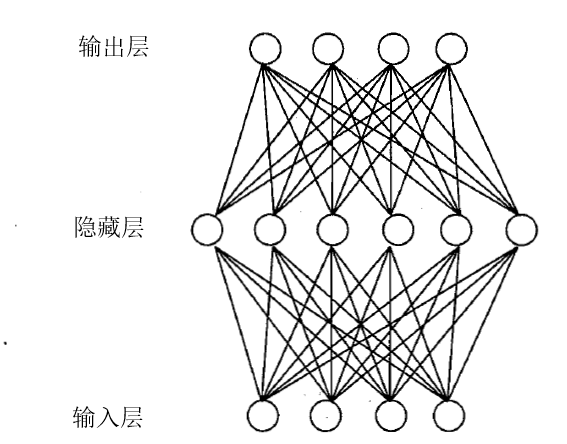
 

图 2.3 输入神经元为2的感知机 图 2.4 隐藏层数为1的多层神经网络

感知机的结构相当的简单，且只有输出层的神经节点有处理函数，所以能够解决的问题很有限，基本只能解决线性可分的问题。

更加常见的神经网络大多都含有多个隐藏层(Hidden Layer)， 如上图2.4。每个隐藏层当中含有多个神经节点，这样的神经网络能够较好的进行数据集的训练和拟合，但是相对的由于节点数的增加，训练的收敛速度会减慢。

**3） 前向传播与****反向传播**

前向传播和反向传播是神经网络运作的两个主要的算法，分别对应了由输入计算得到模型的输出和由数据集的数据修正模型的各个参数矩阵。

**（一）前向传播**

上文在介绍神经元的相关信息的时候讲到，神经元的输入和输出的对应关系可以由以下表达式来表示

这里的是单个神经元的输出，由于一层当中会由多个神经元，那么每个神经元都有各自的和，所以对于第层来讲，所有神经元的系数可以合并为矩阵和。有如下的式子

其中为当前层的输入，为这一层在处理函数之后的值，通过激活函数之后成为下一层神经元的输入

以上的运算过程在每一层进行迭代，最后可以得到输出层的结果，即模型接受输入之后计算所得的预测值。

前向传播算法就是上述过程，即利用每一层的系数矩阵和偏移量进行线性计算，再通过激活函数进行激活运算，从输入层一直到输出层，层层计算最后输出结果的过程。

**（二）反向传播**

反向传播算法的目的是通过训练集当中的观测值输出和模型输出来进行参数修正。在建立模型的时候，往往是根据一定的随机算法初始化参数，反向传播算法能够根据训练集当中的数据进行参数修正，最终找到合适的系数。

反向传播需要一个合适的损失函数，损失函数的输入为训练样本的输出和模型的输出，通过对该函数进行优化求最小化的极值，对应的系数矩阵和偏移量就是我们需要的最终结果。

对神经网络的损失函数运用“梯度下降法”进行迭代求极小值的过程就是反向传播算法。

以均方误差(Mean Squared Error)为损失函数举例。此时损失函数的表达式为

其中为输出层的输出，||S||2为S的L2范数，L2范数是指对向量各元素的平方和求平方根。

有了损失函数之后，就可以用梯度下降法(Gradient descent)来求解每一层的系数矩阵和偏移量。通过每一层的损失函数，分别对系数矩阵和偏移量求偏导，具体步骤和线性回归的方法类似。

之所以称之为反向传播算法，是因为算法是从最后一层的系数开始，逐渐往前计算损失函数的偏导，从后往前进行参数的修正。在进行修正的时候，需要设置一个学习率，每次更新参数的时候都要乘以该学习率，合适的能够较好的提升模型的训练质量。

反向传播算法就是从最后一层开始计算每层的损失函数，通过对系数矩阵和偏移量求偏导，令偏导为0来最小化损失函数，从而更新系数矩阵的过程。

### 2.2.2 SimpleRNN

RNN是循环神经网络(Recurrent Neural Network)[17]的缩写，RNN的精髓在于“循环”两个字。之所以称之为循环，是因为RNN的预测是和数据的时间先后有关系的，和传统的神经网络不同，RNN能够对输入序列产生一定的记忆性，因此RNN的输出和输入的先后有潜在的联系。

假定有一组输入数据，对应的输出为。

将这组输入输出带进神经网络的应用场景，如下图2.5所示。

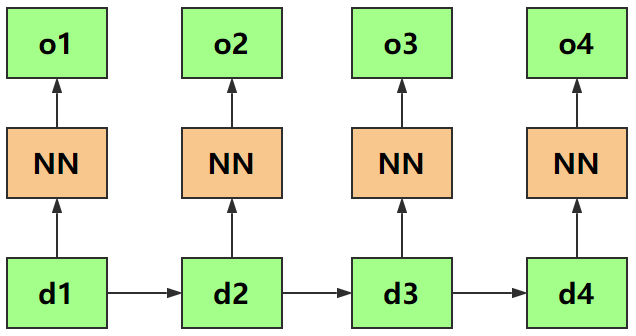


图 2.5 普通神经网络应用示例

可以看到，在进行预测的时候，普通神经网络模型仅仅基于当前的对应的数据，在不同数据之间毫无关系的时候，比如温度和气压的关系这种问题上面，能够较好的进行预测，但是在数据集是有顺序的时候，比如烧饭做菜，掌控调料和火候的顺序是固定的，此时普通的神经网络并不能较好的理解隐含在数据集当中的这种关联。这个时候就需要RNN的帮助。

RNN的基本思路就是记住之前发生事情，比如在预测的时候，不仅仅参考还参考了上一次的数据集和 。因而利用这种思想，设计出了如下图2.6的RNN模型。

图2.6中左边是RNN模型没有展开的状态，右侧是RNN模型在时刻展开的状态。其中为训练样本的输入，为隐藏状态，分别代表了模型输出、损失函数和训练样本的真实输出。为线性关系的参数，在整个网络当中是共享的。

这里比较重要的是RNN引入了隐藏状态的概念，隐藏状态是由隐藏状态和训练样本共同决定的，这样一来RNN就能够对潜在的时间先后关系进行记忆。同时，共享的线性关系参数也体现了RNN模型“循环”的思想。

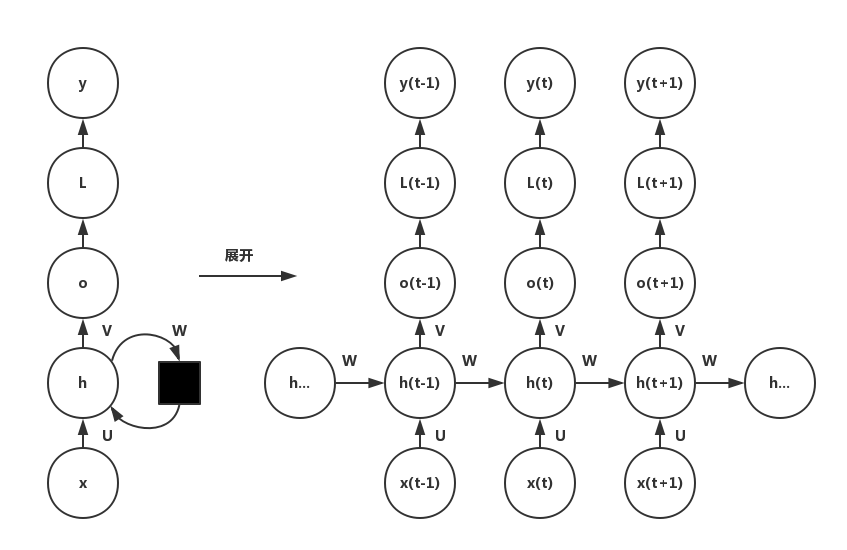


图 2.6 主流RNN模型

在理论上，RNN可以解决很多问题，类似分类、回归，只要由先后时序关系，就可以进行模拟和预测，然而不可避免的是简单的RNN只能记住较为近的信息，无法进行中长期的信息记忆，所以后续引入了LSTM和GRU。

### 2.2.3 LSTM

上文讲到，RNN尽管可以记忆短期信息，但是对于中长期的信息以来做的并不好。LSTM(Long Short Term Memory)是一种特殊的RNN模型，可以解决长期依赖(Long-Term Dependencies)问题。比如当进行文本信息预测的时候，较长的具有复杂语义的句子，对于RNN来说，关键信息和预测位置的距离非常的远，在如此大的间隔之下，RNN的预测能力非常的弱。理论上RNN可以处理这样的问题，但是实际的训练当中，不可避免的会产生梯度消失等问题，具有很大的实现难度。

LSTM由Hochreiter和Schmidhuber在1997年提出，并经过许多人的改进和推广，现在在各种领域被广泛的使用。

LSTM和RNN最大的不同在于，LSTM引入了遗忘的概念，通过增加交互层来使得信息进行进一步的筛选和组合，同时保存最重要的信息。

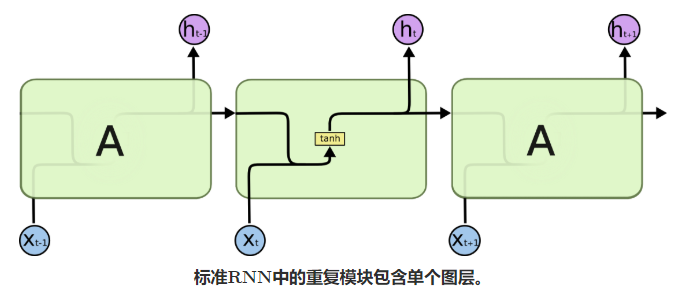


图 2.7 标准RNN的重复模块内部

如图2.7所示，在标准RNN当中，重复模块的结构非常的简单，如单个的tanh层。但在LSTM的重复模块当中(图2.8)，有四个不同的交互层。

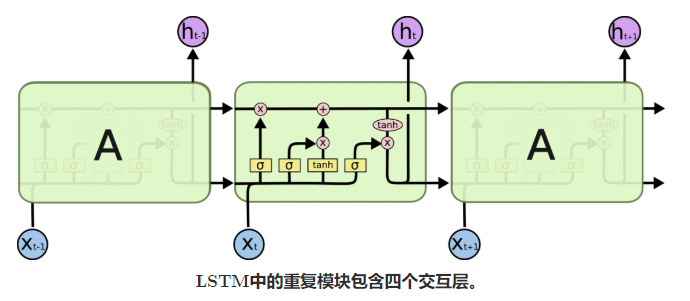


图 2.8 LSTM的重复模块内部

与RNN不同，LSTM有贯穿整个展开链条的主线状态，只有一些次要的线性交互作用，新的训练集输入进入的时候，通过交互层之后可以修正主线状态，如下图2.9的，主线状态和当前输入决定了当前状态。

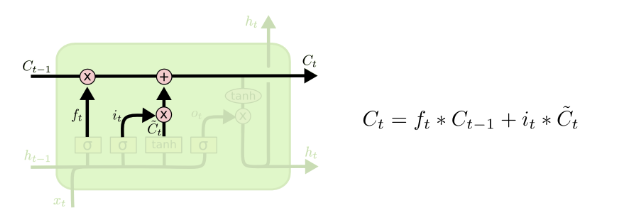


图 2.9 LSTM重复模块——改变主线内容

如果当前的输入非常重要，所占比重大，那么就会按照比重写入主线进行分析，如果此时分线(隐藏状态)和主线冲突了，那么就应该按照一定方法忘记主线的某些冲突内容，如图2.9的，最后根据主线和分线共同决定输出，如下图2.10当中的。

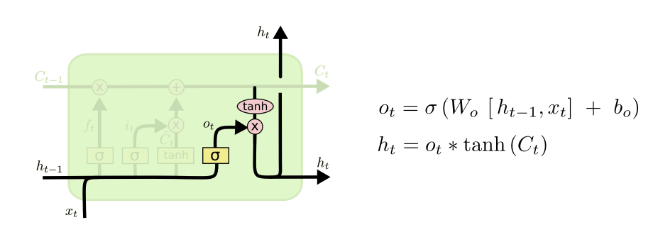


图 2.10 LSTM重复模块——输出当前状态

通过增加交互层，LSTM实现了“记忆衰退”的延缓，同时能够更好的掌握重要信息，由此在长期依赖问题上面有更好的表现力。

### 2.2.4 GRU

GRU是LSTM的一个变体，它将“遗忘门”和“输入门”组合成了一个“更新门”，同时主线状态和隐藏状态也被合并了。这样产生的模型，如下图2.11，比标准的LSTM模型更加的简单，而且训练速度也有相应的提升。

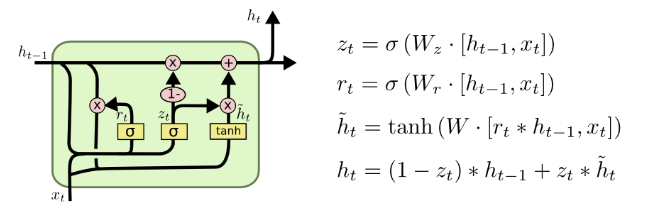


图 2.11 GRU重复模块内部结构

有研究LSTM和GRU的研究资料表明[18]，LSTM和GRU在预测能力上面相差不大，但是速度明显GRU要更快一筹，二者的能力在不同的问题上有不同的高低，需要以实验为基础。

## 2.3 激活函数和损失函数

### 2.3.1 激活函数

神经元内输入经过系数矩阵线性运算之后，需要经过激活函数再输出，不同的激活函数可以表现不同的输入输出关系，由于篇幅限制，此处仅仅简要介绍本文后续用到的几种激活函数。

1. **sigmoid函数**

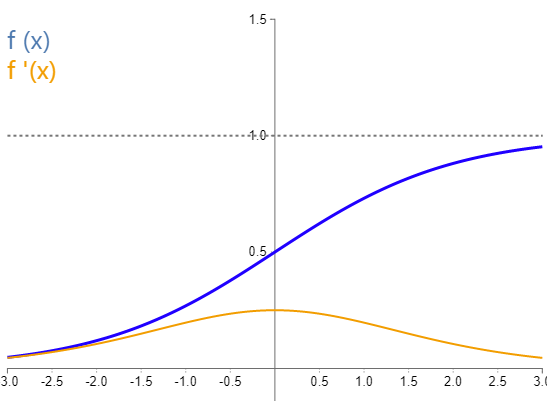


图 2.12 Sigmoid函数图像和导数图像

Sigmoid函数的特性是输出在(0, 1)之间，且斜率在靠近0的时候最大，在远离0的时候非常的小，如图2.12。该函数有几个缺点：

1. 由于神经网络修正参数需要进行反向传播，但是由于该函数在输入远离坐标原点的时候梯度会变得很小，就会导致梯度消失的问题，不利于权重的优化。
2. 函数的输出中心为0.5而不是0，这样会再度降低权重的更新效率。
3. 函数的指数运算会增加训练的时间。

尽管如此，sigmoid函数在很多问题上的表现还是很不错的。

1. **tanh函数**

tanh函数和sigmoid函数的曲线形状很相似，如下图2.13，都是在输入远离远点的时候梯度很小，权重更新的效率较低，但是由于tanh函数的输出区间以0为中心，所以在这一点上权重更新效率要比sigmoid要好。

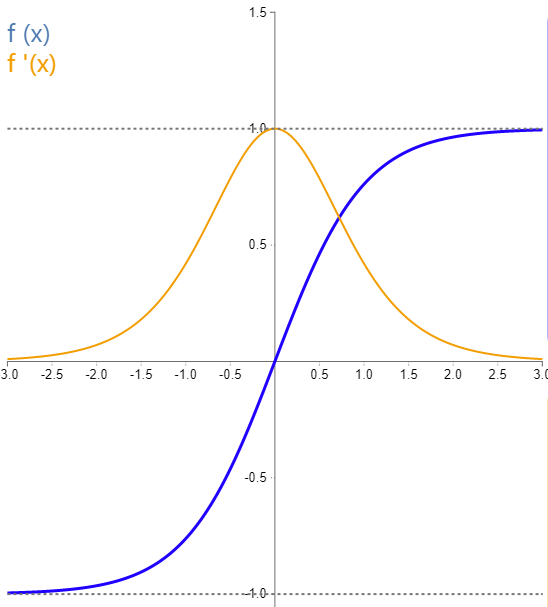


图 2.13 tanh函数图像和导数图像

在通常的二分类问题当中，隐藏层使用tanh函数而输出层采用sigmoid函数，但具体情况需要通过实验来决定选择。

1. **ReLU函数**

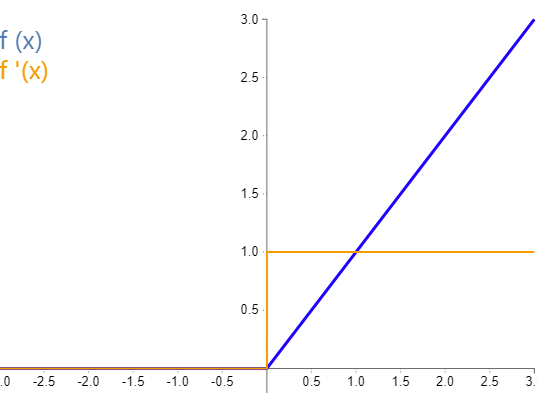


图 2.14 ReLU函数图像和导数图像

和上述的两个函数比较来看，ReLU在输入为正数的时候不会有梯度消失的情况，同时由于仅仅是线性关系，它的计算速度会快很多。然而ReLU对于负数的梯度为0，所以一旦输入集有负数，参数就不会进行修正，这在本文的实际操作当中已经多次验证。而且和Sigmoid一样，ReLU也不是以0为输出中心的函数。

针对ReLU的表现问题，后续还有LReLU、PReLU、CReLU、ELU、SELU等改进方案，此处就不再赘述。

### 2.3.2 损失函数

由于损失函数的作用环节是在反向传播的参数修正当中，在不同的具体问题当中的表现也会不尽相同，通常根据数据的特点来选择损失函数，实际场景当中可以通过训练时间、预测效果等等来评估。此处仅罗列本文可能使用的损失函数。

1. 均方误差 mean square error (MSE)
2. 平均绝对误差mean absolute error (MAE)
3. 平均绝对百分误差mean absolute percentage error (MAPE)

## 2.4 优化器

在训练神经网络的时候，需要花费很多时间，一方面是因为数据量过于庞大，另一方面是因为计算量太大，因为涉及到求偏导、指数运算等等。但为了解决较为复杂的问题，庞大的数据量是不可或缺的，因为数据量越大，模型的训练就越准确，同时也需要更加复杂的神经网络结构。

这个时候，就需要一些方法加速神经网络的训练。优化器(Optimizer)就是因此而诞生的。

这一节将简要介绍当前学术界提出的几种优化器模式，并比较它们的优劣。

**Stochastic Gradient Descent (SGD)**

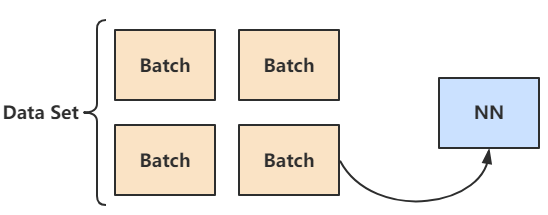


图 2.15 SGD示意图

SGD是最基础的方法，我们将数据集分成一小批一小批，分批次不断地放入神经网络当中进行训练，虽然每批数据不能够完整的反应数据集的情况，但是这样的速度会加快许多。

其余的优化器都是通过更改系数矩阵的更新方法来实现加速。

系数矩阵W的原有更新方法为

**Momentum 更新方法**

这种方法在修正的时候加上了一个惯性，越往下速度越快，能够避免修正过程走弯路。

**AdaGrad 更新方法**

这种方法修改了学习率, 它让每个参数都有与之对应的, 类似于给修正过程提供了阻力，防止修正方向错误。

**RMSProp 更新方法**

合并了前两种方法的优势，但是缺少了项

**Adam 更新方法**

考虑的前两种方法的所有项。

这几种加速方法当中，最后一种Adam更新方法效果最好，不论是在拟合程度还是训练时间上面，都优于前几种优化器，但是实际场景中可能会根据问题的不同产生差异，所以要是要基于实验来选择优化器。

# 3 垃圾焚烧工况数据处理

本文针对垃圾焚烧当中的烟气预测问题建立了若干预测模型，模型的输入是历史焚烧工况数据，训练样本也是来自历史焚烧工况数据和烟气排放数据，因此针对模型训练和预测需要的数据，进行了一系列的数据处理操作。

之所以要进行数据预处理的操作，有以下几个原因：

1. 数据并非直接给出，需要联网获取
2. 数据来源可靠性一般，需要在和企业协商沟通之后，人工筛选有效数据的时间段
3. 并非所有的数据都有参考价价值，比如图3.1当中的数据，含义是反应塔进水流量，在可视化之后发现该数据基本不变动，或者说由于测量误差的关系在非常微小的范围波动，这个时候认为该数据对于烟气数值的变动没有参考价值，应该做人为的剔除

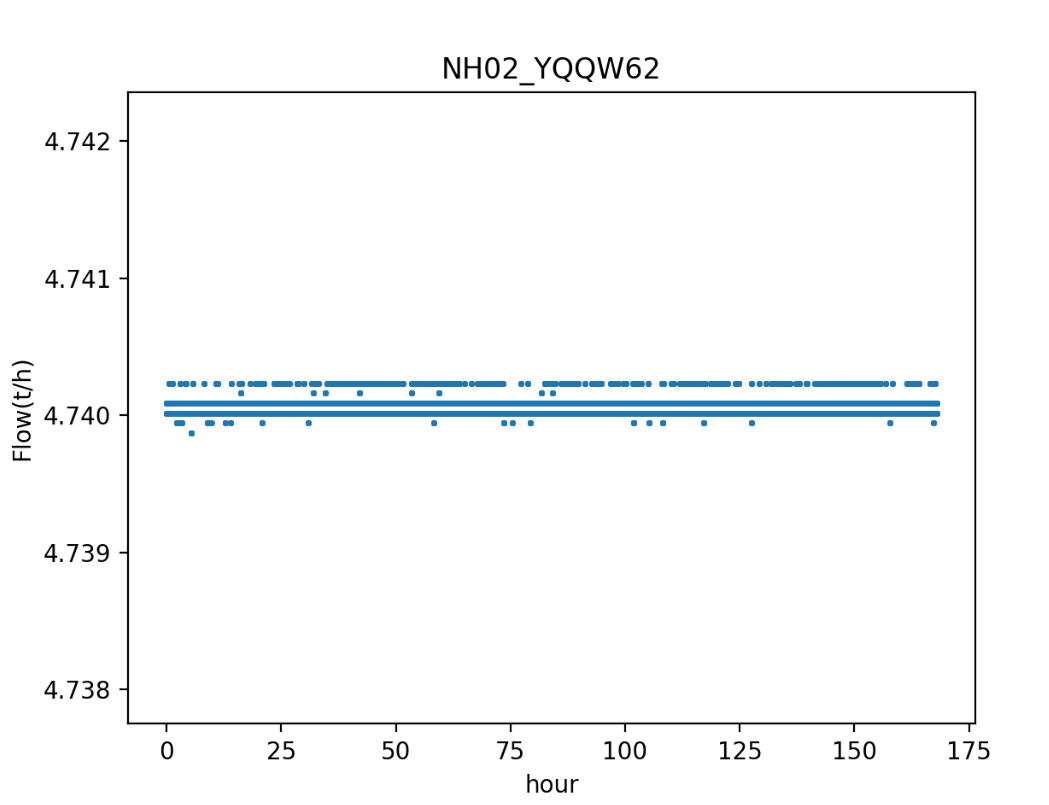
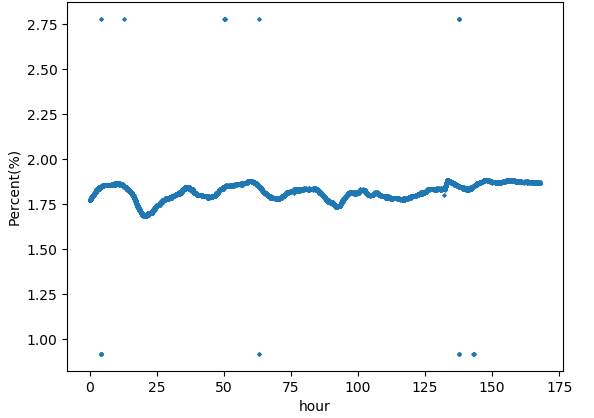


图 3.1 反应塔进水流量变化图

1. 由于数据来源是实际工业场景当中的传感器，会受到传感器灵敏度以及系统记录的客观环境因素影响，因此数据当中会出现异常值、不连贯的零点等等影响模型训练的或者是不合理的数据点，如下图3.2所示，所以需要进行筛选和处理



异常值

零点

图 3.2 进NaOH阀门开度变化图

1. 不同的数据量纲不同，在进行模型训练的时候需要消除因为量纲带来的参数差异
2. 数据的变化方式和采样方式比较特殊，需要根据具体情况进行整理和筛选，同时划分出合理的时间间隔，重新排列数据。

数据处理环节并非是和模型建立和训练严格分开的，往往在模型训练当中发现训练效果不够好，就会考虑数据质量的问题，这个时候就会返回来继续修改数据处理的环节，这样可以进一步加速模型的训练，同时提高模型预测的精准度。

数据的处理过程如下图3.3所示。

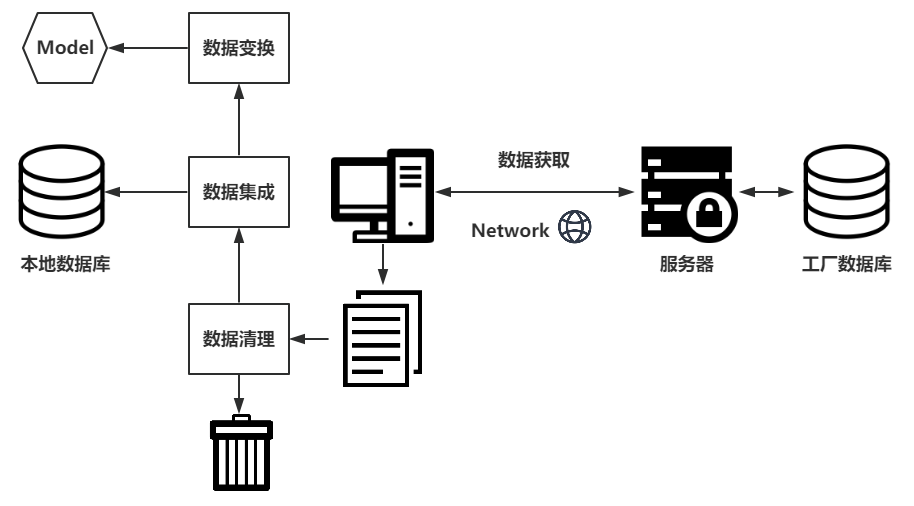


图 3.3 工况数据处理过程示意图

## 3.1 数据获取和筛选

### 3.1.1 接口测试

由于是从互联网开放接口获取数据，需要对接口的可靠性和稳定性进行测试。

从接口获取数据的时候会有一定的误差，工厂给出的数据标签需要进行人为的筛选，对于缺少的数据标签，和工厂进行协商和获取。

本文采用的是基于Python的获取方法。通过引入Python当中的request模块，使用http GET方法进行数据获取，在接口测试阶段一共分成两步：

1、测试标签可用性，通过在数据库当中查找标签，并验证返回结果是否正确。如果标签正确的话，会返回该数据对应的设备号和处理号还有系统编号，如果不正确，会返回空值，具体操作如下图3.4所示

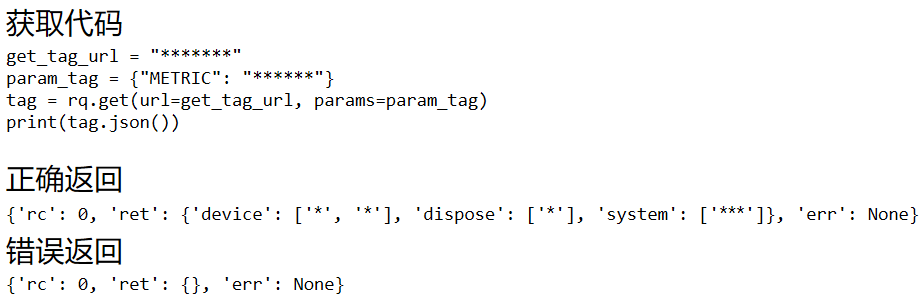


图 3.4 测试标签可用性代码和返回结果

2、测试数据真实性，设备号和标签进行数据获取，只需要获取少量数据观察该数据是否具有参考价值(设置请求当中的limit参数)，部分数据由于缺少传感器或者传感器损坏，数值没有参考价值，需要剔除。具体操作如下图3.5所示

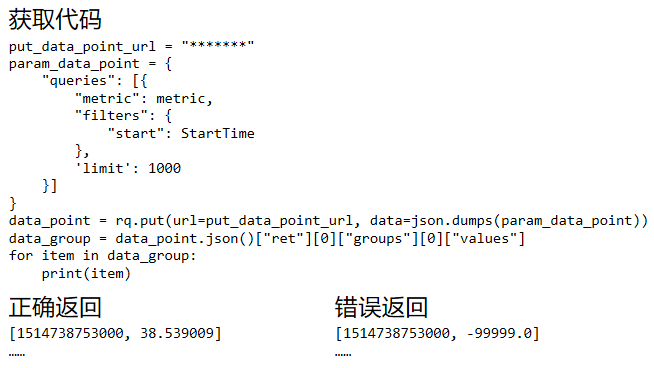


图 3.5 测试数据参考价值代码和返回结果

在进行了接口测试之后，可以得到具有参考价值的数据，这些数据的标签准确，能够正常的进行数据获取。在结束这个环节之后，形成了一份关于数据可用性的文档，文档当中准确的标明了数据标签、数据含义、数据单位、数据对应的设备等信息，部分信息如下图3.6所示。



图 3.6 数据整理文档示意图

### 3.1.2 数据初步采集和可视化

上一环节确定了数据标签的可用性，同时剔除了明显不具备参考价值的数据，但是需要注意到，有些数据尽管一直有合理的数值，但是浮动不明显，或者是根本没有波动，这样的数据对于模型训练和预测来讲没有太大参考价值。

同时，对于需要预测的烟气指标，需要进行一定的统计和可视化、观察总体变化、划分不同状态的时间段并和厂家进行沟通。

此阶段的数据采集数量比上一阶段多，但由于上一阶段已经剔除了一部分数据，所以需要进行可视化的标签减少了很多。主要的步骤也是两个：

1、和上一环节的第二步一样，按照筛选过后的标签进行数据获取，这次需要设定一个较大的时间范围，比如一天、一周或者一个月，将获取的数据整理到数组当中，为下一步的可视化做准备，具体代码和上一环节类似，就不再赘述。

2、可视化使用的是Python的matplotlib.pyplot库，以时间为横坐标，第一步获取的数据为纵坐标，绘制曲线。根据曲线的变动幅度和异常值的出现频率，对数据进行评估，特别是涉及到烟气相关的数据，还需要和环保指标进行对比，查看其按照时间变化的特性，决定是否按照月份分开预测。

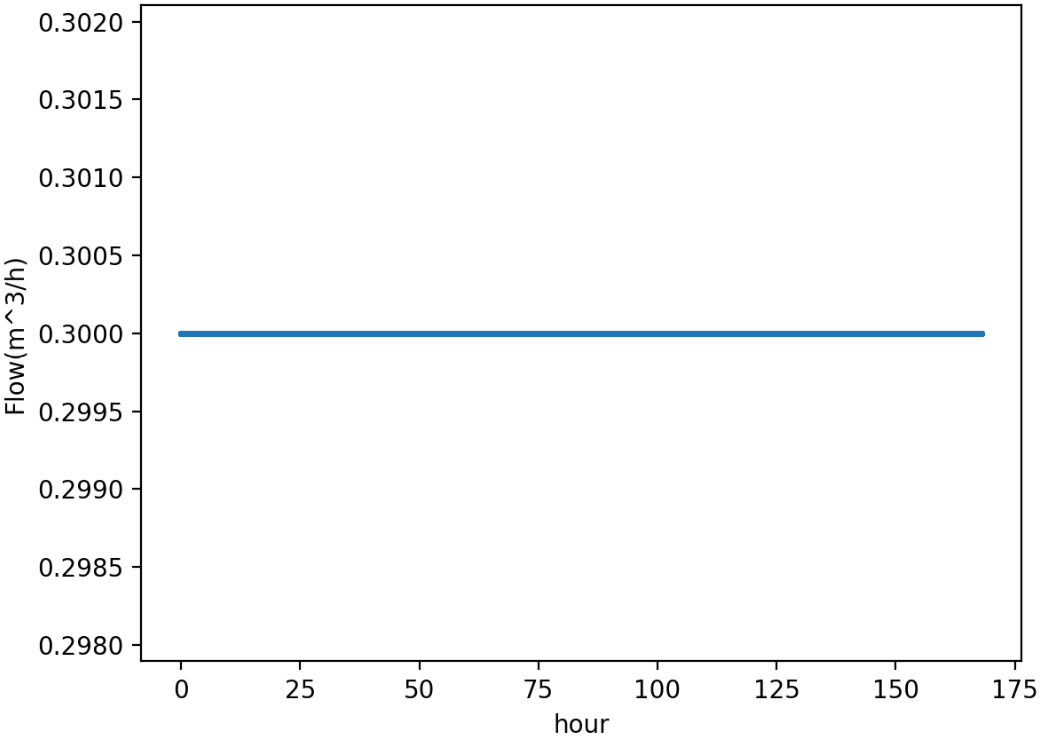
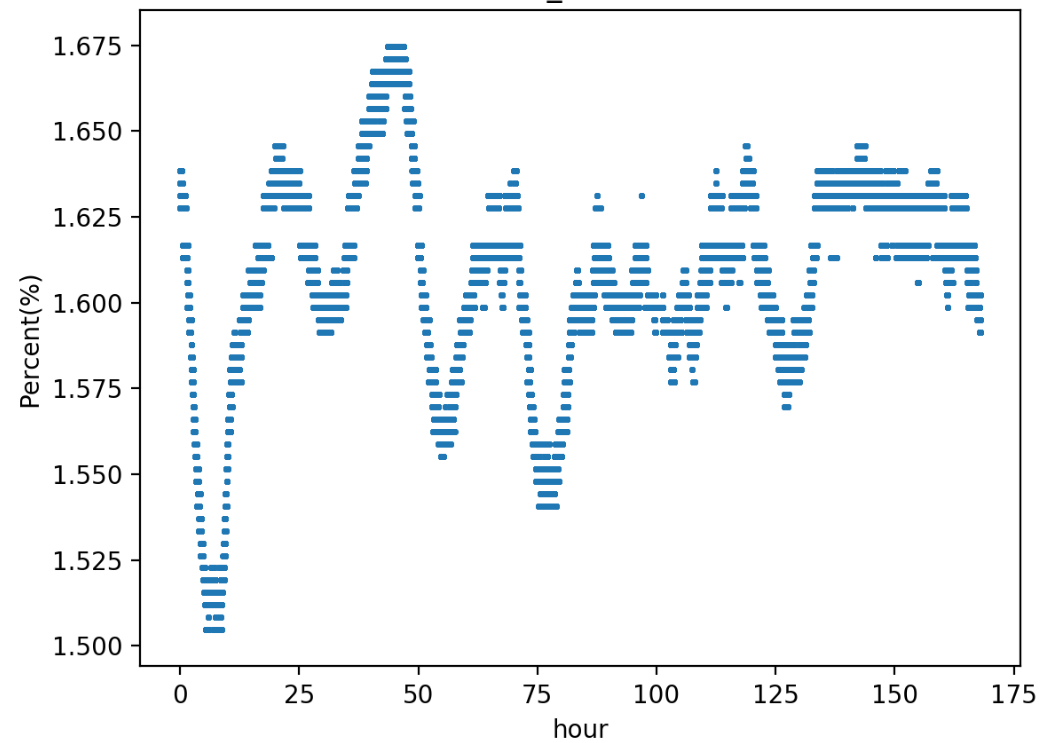
 

图 3.7 氢氧化钠流量设置 图 3.8 反应塔进水阀位

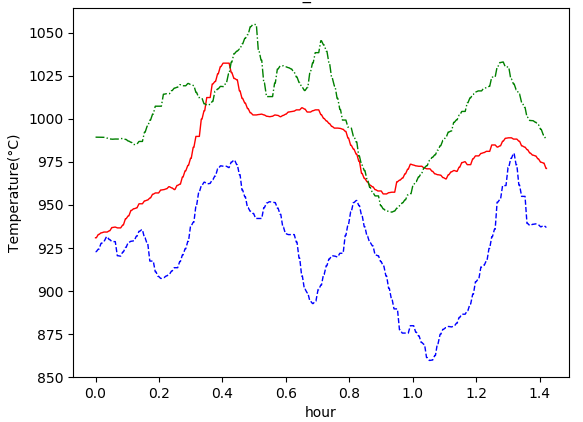
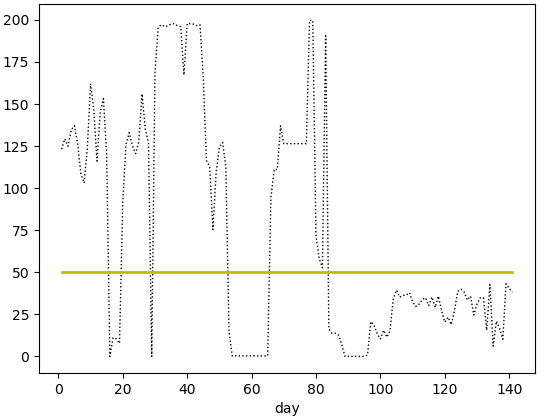


图 3.9 HCl每小时排放速率 图 3.10 下部床温

在进行一系列的可视化之后，观察到数据基本分成四类：

1. 完全没有变动，如上图3.7所示的氢氧化钠流量设置标签；
2. 尽管图像有变动，但是实际上变动的尺度并不大，可以考虑在数据维度过高时剔除，如上图3.8所示的反应塔进水阀位标签；
3. 随时间变化呈现两种不同的变化趋势，如上图3.9所示的HCl排放图像，环保标准是50，在前80天和之后的排放量有明显的差异，和工厂沟通之后了解到前80天处在试验阶段，数据不是很准确，所以确定了以80天之后的数据作为参考；
4. 如上图3.10，有较为明显的波动幅度，且异常值较少，能够为模型训练提供很大的帮助(图中不同颜色的曲线代表不同的设备)。

在对数据标签进行可视化之后，和企业进一步进行了沟通，确定以某号锅炉的烟气排放作为预测对象，因为该锅炉的传感器较为齐全，且准确率高，能够提供较为完备的、良好的数据。

在整理出可用标签之后，需要对数据的采样频率和分布特征做一个较为详细的研究，由于模型的输入是按照一定的时间步长排列的，必须确保数据的粒度一致，这在下一节有相应的说明。

### 3.1.3 数据特性研究和标签筛选

在经过上一步的可视化之后，可以筛除掉没有变动的数据标签。在这一节需要考虑的问题，就是如何形成适合模型输入的数据。首先对预测的对象——烟气数据进行更为细小时间粒度的可视化，如下图3.11，可以发现，数据是按照一定的时间间隔进行采样的，而且在放大之后，如下图3.12，发现每次采样都会有10个相同的数据点。

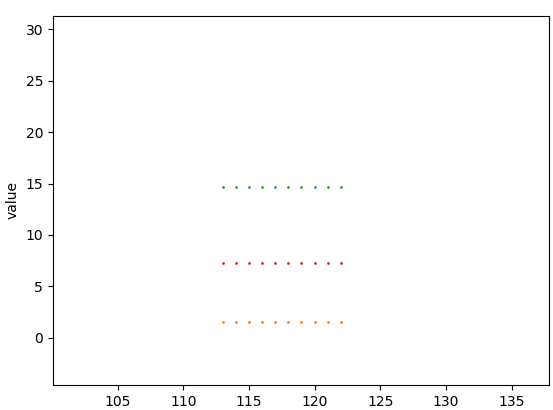
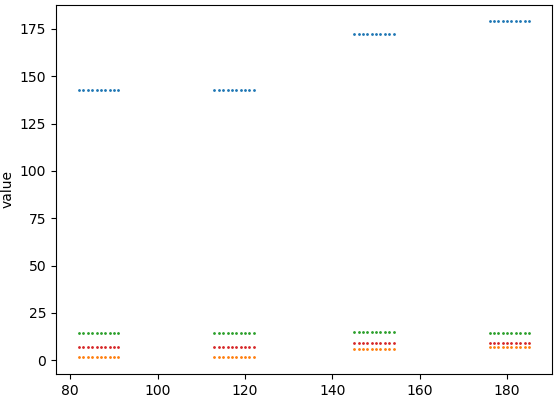


图 3.11 烟气数据 图 3.12 烟气数据（放大）

由于训练数据集当中的烟气数据是按照这个频率和特性进行采样的，所以在设计训练数据的时候，最小的时间粒度就应该是采样的时间间隔。进一步进行可视化发现，不仅仅是烟气数据，其他的工况数据，都是以该特性进行采样的，而且所有数据的采样时刻都是一致的，推测是传感系统统一控制的采样操作。上一节当中整理的数据，有部分是的采样频率是天，这类数据对于模型训练没有太大的帮助，所以在这个环节就剔除出去。

现在的重点就是找到数据采样的时间间隔，然后根据时间间隔来划分输入数据集的时间步长。但是经过计算发现，时间间隔不是稳定的常数。尽管每次采样的个数都是10个相同的数据，但是每组数据点的间隔在29s，30s，31s，32s之间浮动，平均值在30.7s左右。所以现在的问题转变成如何设置时间间隔。

本文以30s作为数据的时间间隔，即每30s有一个数据样本，这个时候会产生问题，有些30s之间是没有数据采样的，有些30s间隔内可能有多个采样点(真实间隔29s的情况)。所以针对真实采样间隔不稳定的情况，本文提供了三种较为合理的处理办法：

1. 折线法，首先确定间隔为30s的时刻点，如果在该时刻点附近(s)有采样数据，那么选择该数据作为该时刻的数据，下图3.13的蓝色圆圈，如果有两个数据点，选择时间上更近的，如果距离也一样，那么就谁的时刻大选谁。也会有没有数据点的情况，这个时候就将该时刻的数据点记为0，如下图3.13的绿色圆圈。

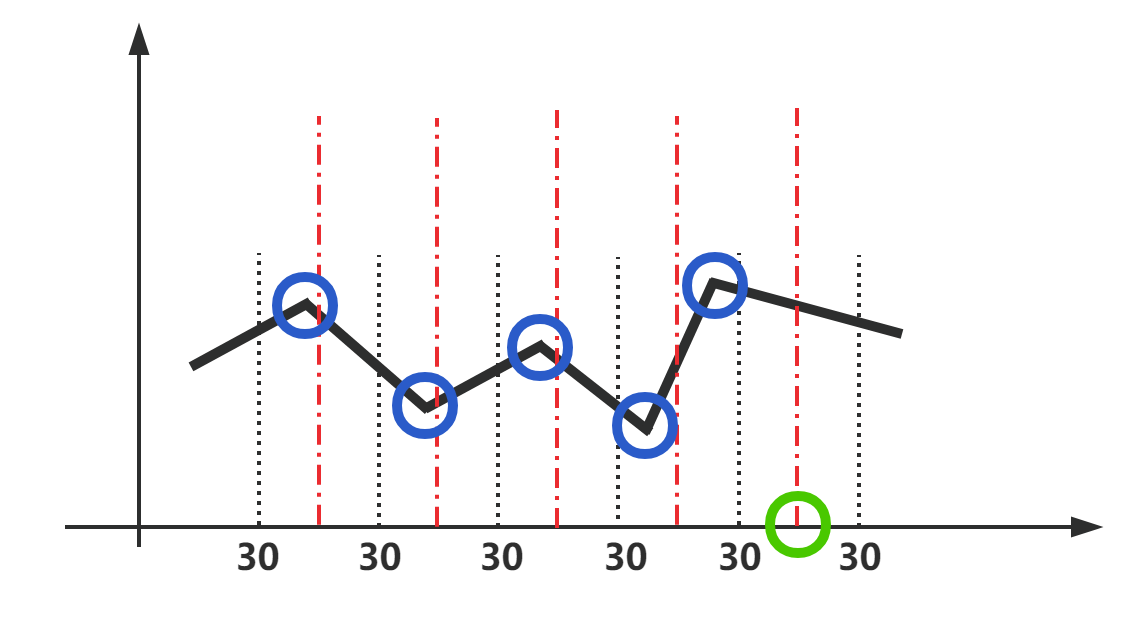


图 3.13 折线法处理数据

1. 多边形面积法。首先确定间隔为30s的时刻点，将所有的数据点连线，以每个时刻点的左右15s(s)为边界，曲线和x轴以及边界围成的多边形的面积，就是该时刻点的数据值。

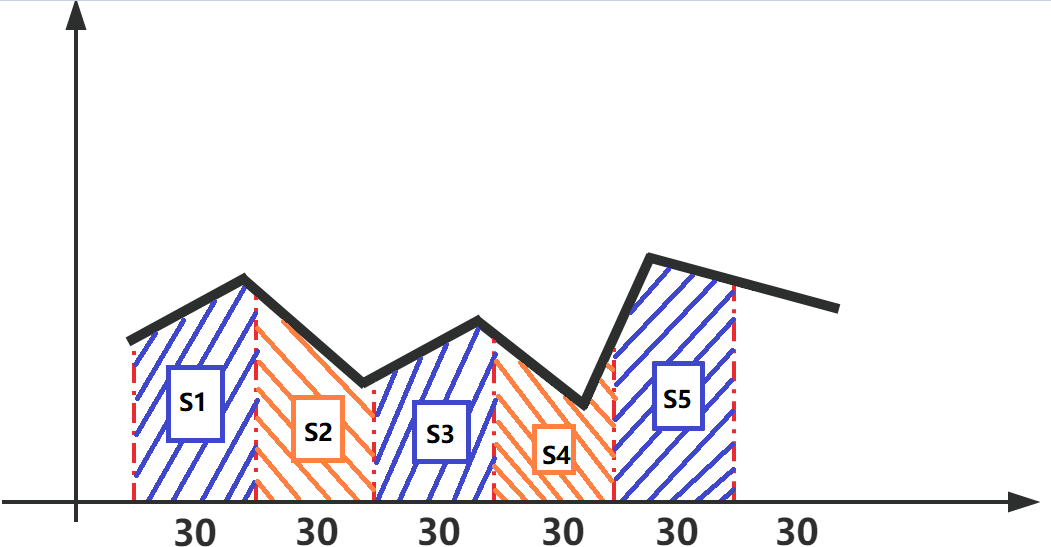


图 3.14 多边形面积法处理数据

1. 拟合曲线面积法。首先确定间隔为30s的时刻点，对数据点进行曲线拟合，以每个时刻点的左右15s(s)为边界，曲线和x轴以及边界围成的多边形的面积，就是该时刻点的数据值。

为了更好的显示出数据的变化差异，本文选用了第一种方法进行数据处理，这样带来的好处是每组数据点的差异较为明显，不会因为求面积的原因使得数据之间的差异减小。同时，在预测的时候不需要进行数据变换，预测的结果就是某时刻点的数值。

这样的做的缺点就是会有一部分的0值，这在后续的空缺值处理当中会进行填补或者删除。

### 3.1.4 数据库存储设计

本文的数据量大，但是依赖性不强，为了追求较好的数据存储和读取效率，本文采用了MongoDB数据库，该数据库的特点就是操作灵活、存储和读取性能优越，而且可以快速的导入导出，为后续的数据处理提供了很大的便利。

由于同时获取多个数据会造成服务器的瘫痪，所以本地数据库的存储也是按照不同数据标签进行获取和存储的，考虑到不同月份之间可能存在一些差异，所以每个月的数据也分开来存储。数据的属性如下表3.1。

表 3.1 数据存储格式

|  |  |  |
| --- | --- | --- |
| 属性名 | 含义 | |
| **\_id** | 数据库的标识特征码 | |
| **TimeStamp** | 时间戳，可以转化为具体的时刻 | |
| **value** | 数据值 |

注意到该表的属性当中并没有包含数据标签和月份信息，本文以数据标签为表名，以月份名称为数据库名，方便操作和记忆。存储和读取数据的方法是利用Python当中的pymongo模块，可以方便的进行存储、查询等操作。

## 3.2 数据集成

由于训练模型的时候是按照一组一组向量输入训练的，所以需要将分标签存储的数据，按照时刻点进行合并，公式如下

因此这一步的工作就是合并之前存储的每个标签的数据，合并的原则就是相同的时刻，这样每个时刻会有一组向量，最终的形式就如同，是一个矩阵，在Python当中即为一个二维数组。由于处理的数据量大，且需要进行一些操作变化，本文采用Python的numpy库，将数组存储为ndarry的形式，方便后续的进一步操作。

具体的操作如下图3.15所示。利用Python中Pandas库进行数据读取和处理，将之前存入数据库的所有标签数据全部整合进一个矩阵，在整合的时候，如果在某个时刻没有相应的标签数值，那么就填入0。最后将整个矩阵保存为数据文件的格式，进行下一步的数据清理操作。

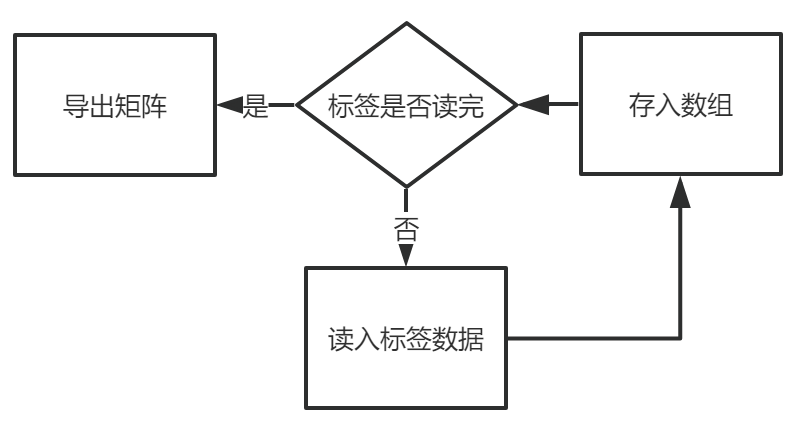


图 3.15 数据集成示意图

## 3.3 数据清理

数据集当中存在了许多异常数据点和0数据点，由于需要较为准确的数据样本进行模型训练，所以需要对上文处理过后的数据矩阵进行数据清理操作。数据清理主要分为两个部分——异常值处理和空缺值处理。

在该系统当中，由于锅炉当中的传感器精确度高，所以一般不会产生离群点，如果没有测量到某时刻的数据，会用-999999替代，所以数据集当中需要处理的就是这样的异常值。另外，由于之前设计的数据处理方法会产生0值，需要在这个环节对0值进行填补，考虑到短时间内的变动幅度不大，所以空缺值采用线性插值法来进行填补。

具体的数据清理过程如下：

1. 异常值处理。异常值一般出现在两种情况，一种是传感器失灵，这个时候的异常值往往只是一次采样的数据偏差，反应在矩阵当中就是一个点异常；另一种情况是锅炉因为某种原因暂停处理垃圾，或者是系统因为某种原因停止采样，这时候会出现大片的异常值，反应在矩阵当中就是连续几行全部都为异常值。第一种情况只需要按照空缺值填补的方法进行插值就行，但第二种情况比较普遍。在某月份的数据集当中，有1天左右的数据出现异常，如图3.16所示。

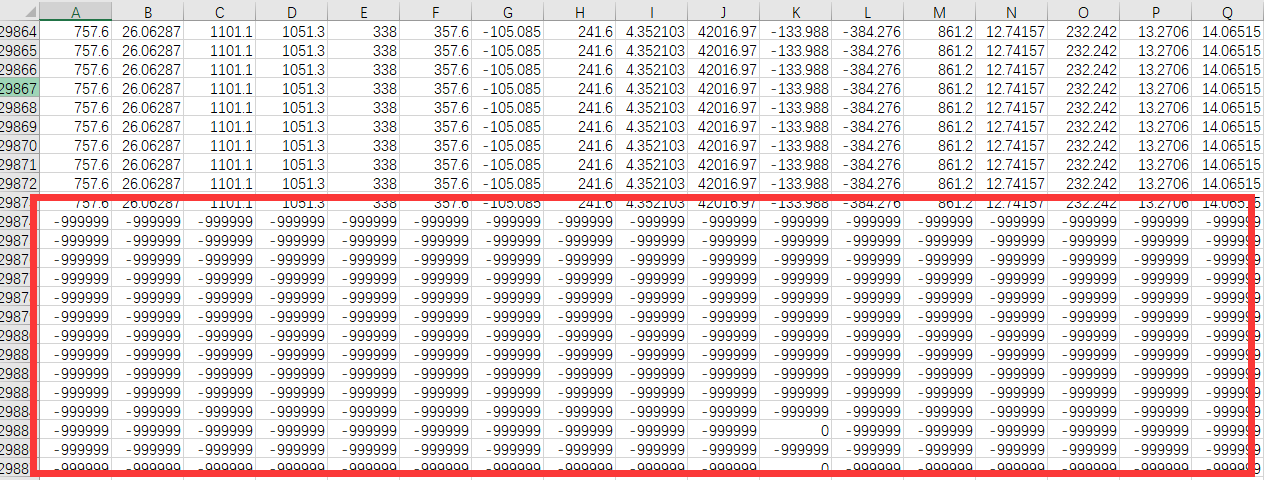


图 3.16 异常数据示意图

这个时候就需要人为的进行剔除，通过找到相应的开始位置和结束位置，将原有的数据矩阵分为两个新的矩阵，这时候异常值部分已经被抛弃了，简要的过程如下图3.17所示。

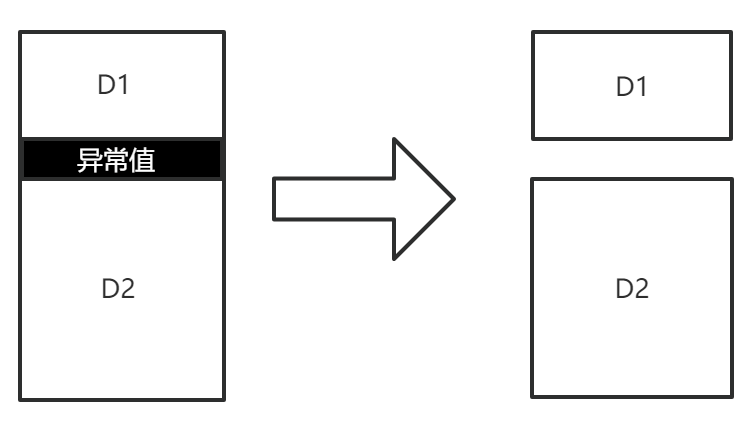


图 3.17异常数据处理过程示意图

2. 空缺值填补。空缺值的存在，也分两种情况，一种是由之前异常值转化的空缺值，另一种是由于数据处理方式，产生了没有对应数据的时刻点。空缺值填补的方法，一般采用插值法，主流的插值方法有0值填充、平均数填充、众数填充、线性插值和拉格朗日插值法。本文在此处采用的是线性插值，原因是由于采样时间较短，线性插值足够反应曲线在此时刻的对应点，如果时间间隔过长，应该采用拉格朗日等基于多个数据的插值法，这样带来的缺点就是处理时间会有较大的增长。处理过程的示意图如下图3.18所示。红色圆点就是经过线性插值法之后填充的数据点，本文运用的是Pandas库当中的interpolate函数。

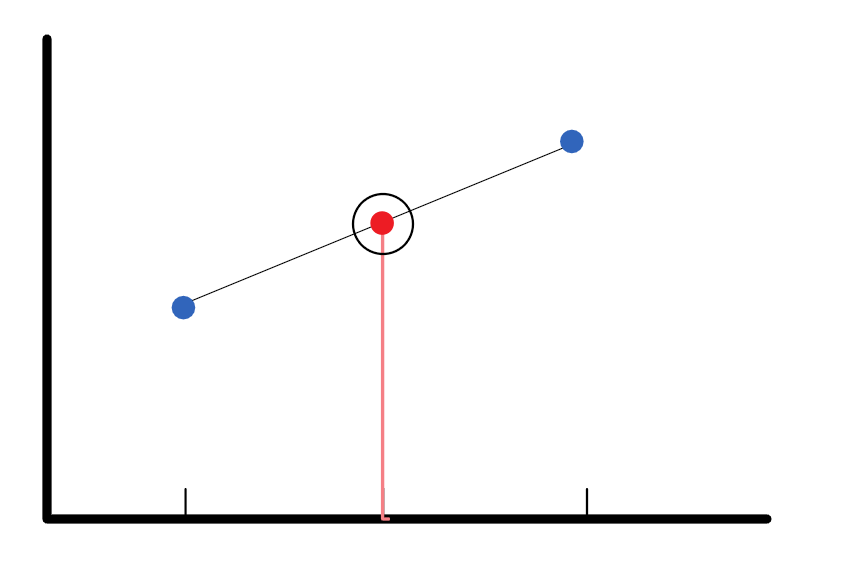


图 3.18 线性插值法示意图

## 3.4 数据变换

本节有两个目标，一个是消除不同类型数据之间的量纲差异，另一个是整理训练模型的样本数据。第一个目标对应的是数据归一化，需要按照每个标签分别进行归一化；第二个目标对应的是数据重组，将原来的二维矩阵组合成三维矩阵。具体过程将在下文详细说明。

### 3.4.1 数据归一化

按照数据预处理的方法，数据归一化有很多种类，比如标准化(Z-Score)，Min-Max规范化，正则化等等手段。本文采用的是Min-Max规范化，即将所有数据缩放到(0, 1)之间，消除量纲带来的差异。注意到标准化和正则化并不能有效的消除量纲，同时本次训练的样本并不需要归一化方差，唯一可以参考的就是去平均的操作，但是在实验过程当中发现Min-Max规范化的表现相对较好。Min-Max规范化的公式如下

在进行数据归一化操作之后，数据矩阵当中的所有数据都在(0,1)之间，消除了量纲的差异，极大的加速了模型训练的收敛速度和拟合准确性。本文利用Python的sklearn.proccesing模块，通过引入缩放器MinMaxScaler进行数据归一化。在进行结果比较的时候，可以去归一化操作，将模型输出和测试集放在原先的量纲上面比较。

### 3.4.2 数据重组

本文提出模型的基本思路，就是根据历史工况数据来预测未来的烟气指标，所以需要向模型输入不止一个时刻的数据，假设此时训练的输出是时刻的烟气指标，那么输入就应该是到为止的所有工况数据，此处的就是时间窗口。

经过一系列的训练测试，本文最后令，即有20组时刻数据同时作为输入样本输入到模型当中。数据的组合如下所示

,…

如果原本的数据集有数目为m\*n在时间上连贯的数据，那么最后组成的训练样本如下图3.19所示，共有(m-20-1)个输入输出组合，橙色矩形为输入，绿色方块为输出。

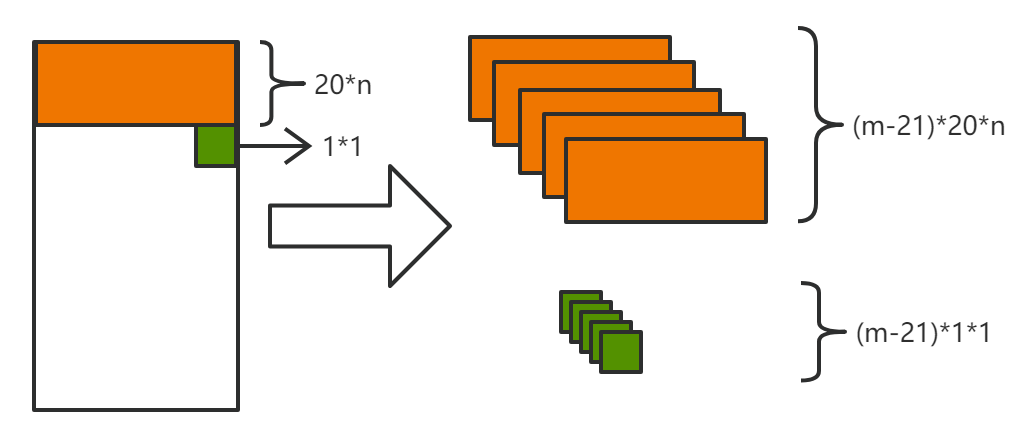


图 3.19 训练样本重组示意图

# 4 烟气预测模型设计

## 4.1 模型概述

本文提出的烟气预测模型从训练方式和输入输出关系上进行划分，可以分为两类：

1. 一类是线性模型，输入和输出之间是严格的线性关系，训练方式较为简单，只需要利用输入数据集和输出数据集进行矩阵运算即可，线性模型的训练过程如下图4.1所示。

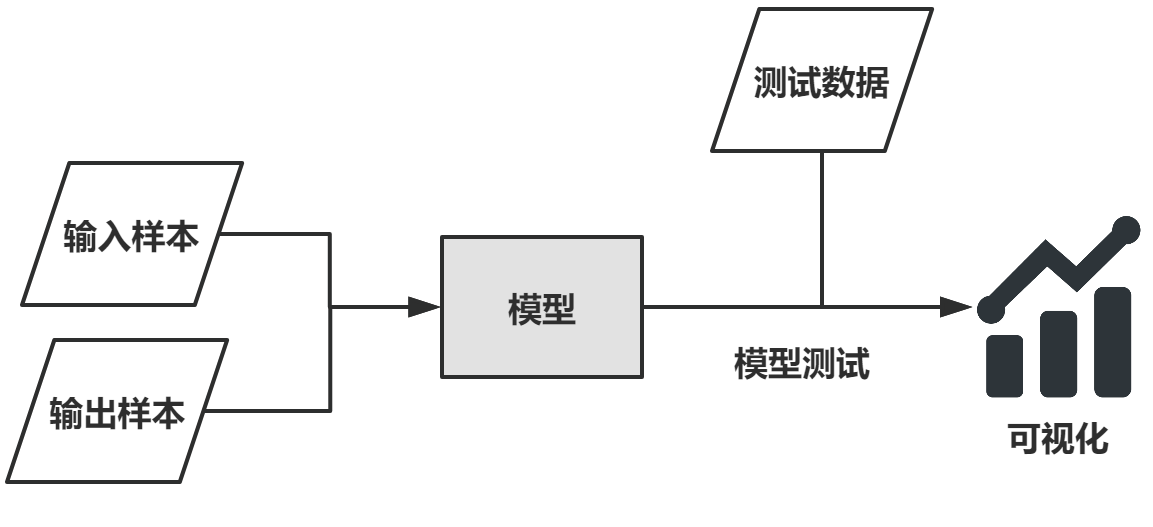


图 4.1 线性模型训练过程示意图

1. 另一类是神经网络模型，输入和输出之间的关系不一定是线性关系，并且该类模型的训练较为复杂，如下图4.2所示，需要设计输入样本的数目、修正参数的方法等等一系列的训练方法，同时需要迭代多次数据集时模型达到较好的收敛，因而训练时间要远远高于第一类模型，然而由于该类模型能够更好的拟合输入集和输出集的关系，所以预测的精度往往要高一些。

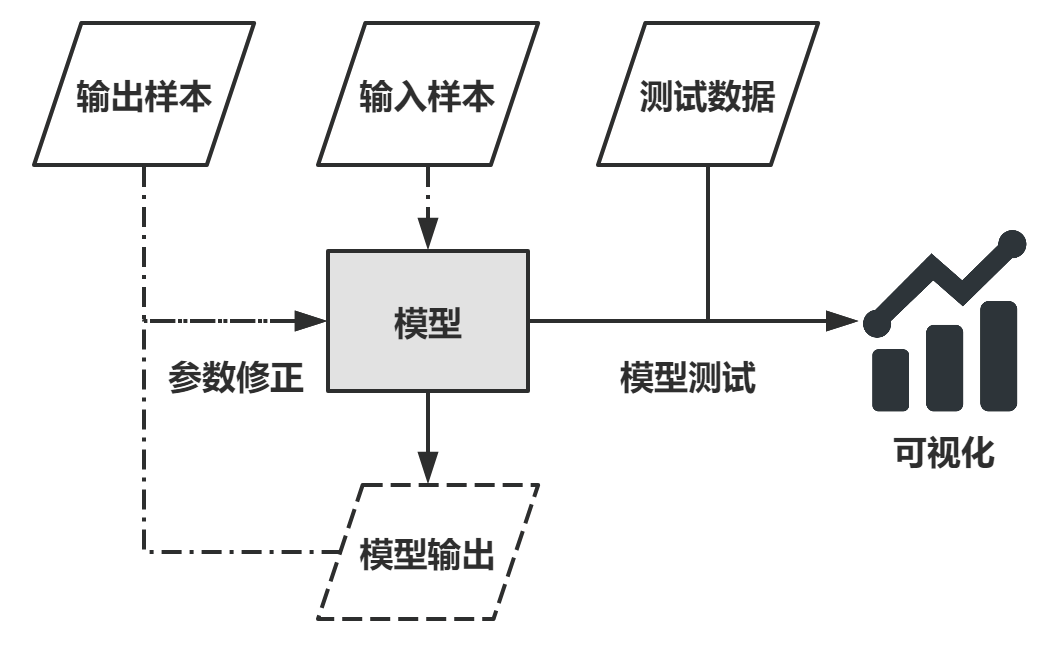


图 4.2 神经网络模型训练过程示意图

可以看到，两类模型的训练过程当中都涉及到了“输入样本”、“输出样本”、“模型”和“测试及可视化”这几个部分，这也是训练模型的核心部分，本小节将对这几个部分做相应的介绍，其中模型训练环节当中更为具体的部分将分别在第4.2节、第4.3节按照不同的模型类型分别介绍。

### 4.1.1 输入样本

输入样本是模型训练样本的一部分，由历史工况数据组成。输入样本从数据存储方式来讲是多个二维矩阵，如下所示

二维矩阵的个数就是输入样本的总个数，在机器学习当中称为Sample，Sample越大，那么数据集里面的数据也就越多，相应的，在训练模型的时候模型能够拟合的就越好，在预测时的表现就越好。

需要注意的是，在实际工业场景当中的数据，并不一定具有在宏观上统一分布的规律。具体来说，在一月份烟气和历史工况数据的对应关系，和在二月份的对应关系，未必就是同一种关系。具体的模型适应时间段，需要根据实际操作和实验来找到。

在本文研究的工业场景当中，由于垃圾焚烧锅炉的环境相对封闭，且焚烧对象——生活垃圾的成分相对稳定，所以不需要针对不同的时间段划分数据集，可以将连续的几个月数据用来训练。

需要指出的是，输入样本和最初的数据集是有差异的，需要经过上文介绍的数据重组阶段。原先的数据集形状如下，此处要指明的是，数据集已经经过了Min-Max规范化(见3.4.1数据归一化)。

要转化为形如的输入样本，需要拆分和重组。在拆分和重组的时候需要确定时间窗口，在本文的模型训练当中，称之为TimeStep。这个时间窗口是基于本文提出的一种假设——即未来的烟气数据，和历史工况数据有很大的关系，根据一定范围内的历史工况数据，能够推测出下一个时间步长内的烟气排放数据，如下图4.3所示。

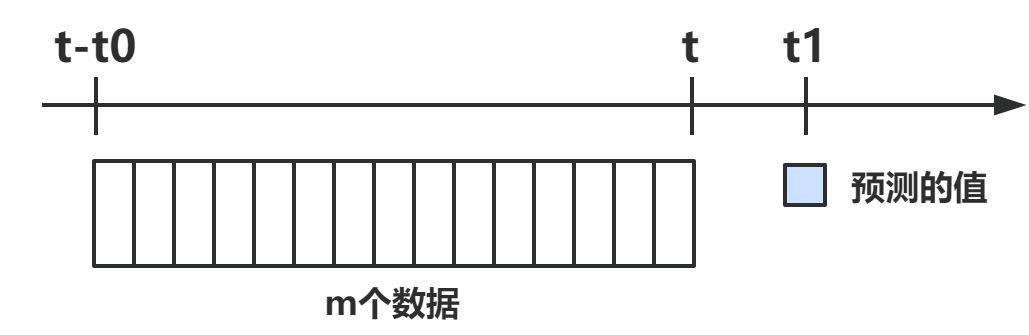


图 4.3 输入样本和预测值的关系

图中横向的为时间轴，其中t1>t，t1表示的是未来的某一时刻，模型需要预测的就是该时刻的某一烟气的排放数值。在这段时间内，有m个数据，这里的数据是形如的一组向量，向量的维度就是参考标签的个数，每个元素的下表都是一致的，意思是每一组向量都是所有标签在某个时刻的数值的组合。m的值是由时间窗口决定的，公式如下

这里要求时间窗口大小一定要是30的倍数，这样m才能取整数。输入样本的数据就是由多个连续的二维矩阵组成的三维矩阵。

在线性模型和神经网络模型的训练当中，由于模型本身原理的不同，对于输入样本的需求也存在些许差异。线性模型将每一组样本都独立开来训练，即每一组样本都是线性空间上的一个点，这时候样本的排列顺序是不重要的，即使将样本打乱，对于最终拟合的线性模型来讲，也没有太大的影响。但对于神经网络模型来讲，则不同。

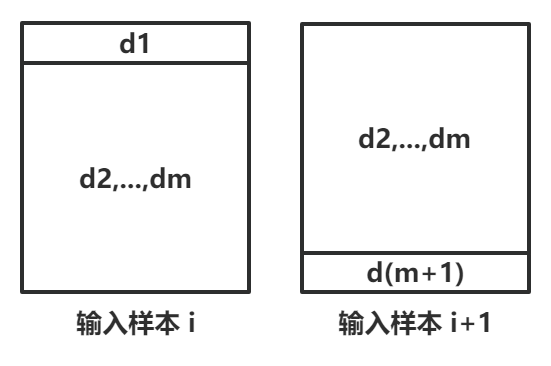


图 4.4神经网络相邻输入样本的关系

由于本次实验的神经网络模型是基于潜在时间关系的循环神经网络模型，所以需要输入样本严格按照时间顺序来分布，即在顺序上相邻的样本i和样本i+1的关系应如上图4.4所示。

本模型的输入是历史工况数据，并非所有在垃圾焚烧当中采集的工况数据都会作为模型训练的参考，相关的筛选和处理方法在第3章有详细的介绍，这里给出了所有作为输入样本的数据标签，数据标签的个数即样本当中单个数据向量的维度m，在本文当中的标签个数为28，具体标签如下表4.1所示。

表格 4.1 输入样本数据标签一览

|  |  |
| --- | --- |
| **数据标签1-14** | **数据标签15-28** |
| 锅炉上部烟温 | 省煤器后烟气压力 |
| 锅炉上部烟压 | 省煤器后烟温 |
| 锅炉第一通道中部烟温（左） | 省煤器后氧量 |
| 锅炉第一通道中部烟温（前） | 一次风量 |
| 锅炉第一通道中部烟温（右） | 二次风量 |
| 中下部烟压 | 炉后烟气压力（排放） |
| 锅炉第一通道中下部烟温（左） | 炉后烟气湿度（排放） |
| 锅炉第一通道中下部烟温（前） | 炉后烟气NOX（排放） |
| 锅炉第一通道中下部烟温（右） | 炉后烟气SO2（排放） |
| 锅炉第一通道下部烟温（左） | 炉后烟气HCL（排放） |
| 锅炉第一通道下部烟温（前） | 炉后烟气含尘（排放） |
| 锅炉第一通道下部烟温（右） | 炉后烟气CO（排放） |
| 顺推炉排左侧温度 | 烟气流速（排放） |
| 顺推炉排右侧温度 | 炉后烟气温度（排放） |

上表4.1的排列顺序，和真实训练模型时的排列顺序不一致，但是因为排列顺序和训练结果的关系不大，所以在这里仅列出需要参考的工业数据值。在实际的工业场景当中，可能存在更多的具有参考价值的数据标签，这个时候就需要扩充输入样本中每个数据向量的维度m，具体的操作和上文介绍的基本一致。

### 4.1.2 输出样本

输出样本是模型训练样本除输入样本之外的另一部分，由需要被预测的烟气数据组成。输出样本从数据存储方式来讲是就是一组向量，在不同的模型需求条件下，可以转化成二维或多维矩阵，如下所示

这组向量当中元素的总个数就是输出样本的总个数，这里的总个数和第4.1.1节种输入样本的总个数Sample是一致的，每一个输入样本对应着一个输出样本。同时，输入样本和输出样本存在严格的时间对应关系，输入样本和输出样本的时间间隔，就是预测的时间步长。

由于输出样本和输入样本存在着对应关系，所以本文当中的输出样本也不按照月份进行划分，是多个月份连续组成的样本集合。

输出样本也是从最初的数据集截取而来，同时也经过了上文介绍的数据重组阶段。原先的数据集形状如下，与输入数据来源的数据集一致，因为要保持时间上的对应关系，数据集已经经过了Min-Max规范化。

要转化为形如的输输出样本，同样需要拆分和重组。在拆分和重组的时候需要确定时间步长，这里的时间步长指的是在预测未来烟气数据的时候，需要确定是预测未来什么时刻的数值，该未来时刻和当前时刻的间隔就是时间步长，在本文的模型训练当中，称之为TimeAfter。这个时间步长TimeAfter和上文的窗口不同，是指预测时间的远近。下图4.5依旧保留了输入样本的示意图，在此基础上增加了输出样本。

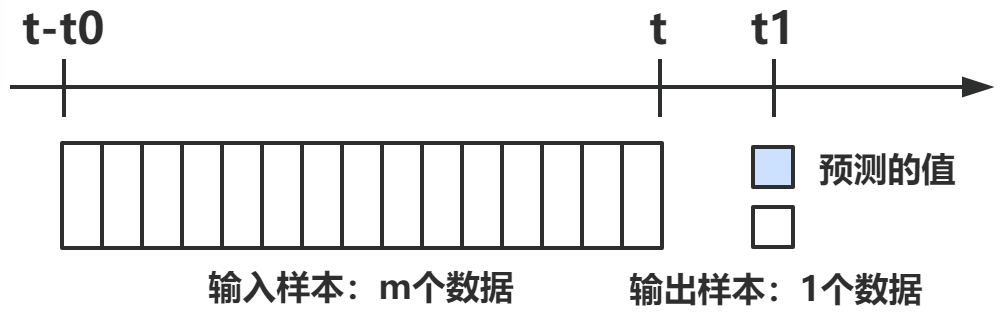


图 4.5 输出样本、输入样本和预测值的关系

图中横向的为时间轴，其中t1>t，t1表示的是未来的某一时刻，模型需要预测的就是该时刻的某一烟气的排放数值。可以看到，输出样本的数据，对应的就是数据集当中，烟气数值在t1时刻的排放数值，所以输出样本就是距离当前时刻固定时间步长的烟气数值，在这里t1由时间步长决定，表达式如下

和输入样本相同，输出样本集当中的每个样本都是在时间上相邻的数据，当然并不是所有输出样本都会在时间上相邻，因为在数据处理过程当中会出现一段时间的空缺，详见第3.3节的异常值处理部分，所以只能说数据在时间上由先后关系。

这里给出了所有作为输出样本的数据标签，每个数据标签都可以作为预测对象，由于本文追求模型预测的准确率，且不同数据标签的预测效果会有不同，所以输入样本的维度维持在1，下列标签都是环保达标当中需要考虑的废气排放数据，具体标签如下表4.2所示。

表格 4.2 输出样本数据标签一览

|  |
| --- |
| **数据标签** |
| 炉后烟气NOx（排放） |
| 炉后烟气SO2（排放） |
| 炉后烟气HCL（排放） |
| 炉后烟气含尘（排放） |
| 炉后烟气CO（排放） |

这里的各个标签都可以作为预测对象，上文讲到不同的数据标签的预测效果不同，这可能是由于不同数据标签的采样精度和传感器质量不同，为了更好的对烟气数据进行预测，输出样本的数据标签，不宜过多，本文当中的数据标签维度维持在1，即每次预测都仅仅针对1中烟气进行，且时间步长保持不变，即每个模型针对的都是特定烟气在特定时间步长下的预测问题。

### 4.1.3 模型内容

本文进行预测的模型主要由两大类，线性模型和神经网络模型。

在线性模型的训练当中，应该用了普通最小二乘法和岭回归两种算法。线性模型的训练相对来讲比较简单，具体步骤如下：

1. 确定线性回归方程，初始化参数矩阵；
2. 整理输入输出训练样本的格式；
3. 进行模型的训练；

具体的步骤主要就是以上的三步，其中线性回归方程由算法的不同可以分成两种，即普通最小二乘法和岭回归两种算法对应的方程，形式如下

其中式对应的是普通最小二乘法的参数矩阵求解，而式对应的式岭回归方法的参数矩阵求解。应用岭回归方法求解的时候，需要在模型训练之前设置参数，的大小未知，初始值可以设置为0.5，按照具体的实验和模型训练效果，可以在后续对进行适当的调整。

在神经网络模型的训练当中，主要应用了普通循环神经网络算法和LSTM的变体GRU算法。这几种算法在第2.2节都由较为详细的介绍，之所以选择GRU而不是LSTM，主要是考虑到训练时间的问题，由于GRU和LSTM两种模型的最终效果相近，而GRU的训练时间明显较短，所以在本文的研究对象具有庞大数据量的前提下，GRU明显更加的适合。

由于本文研究的预测问题是否为长期依赖问题未知，所以运用SimpleRNN和GRU同时进行训练，并对比训练结果，能够找到具有更好预测能力的模型。

SimpleRNN下图4.6所示。

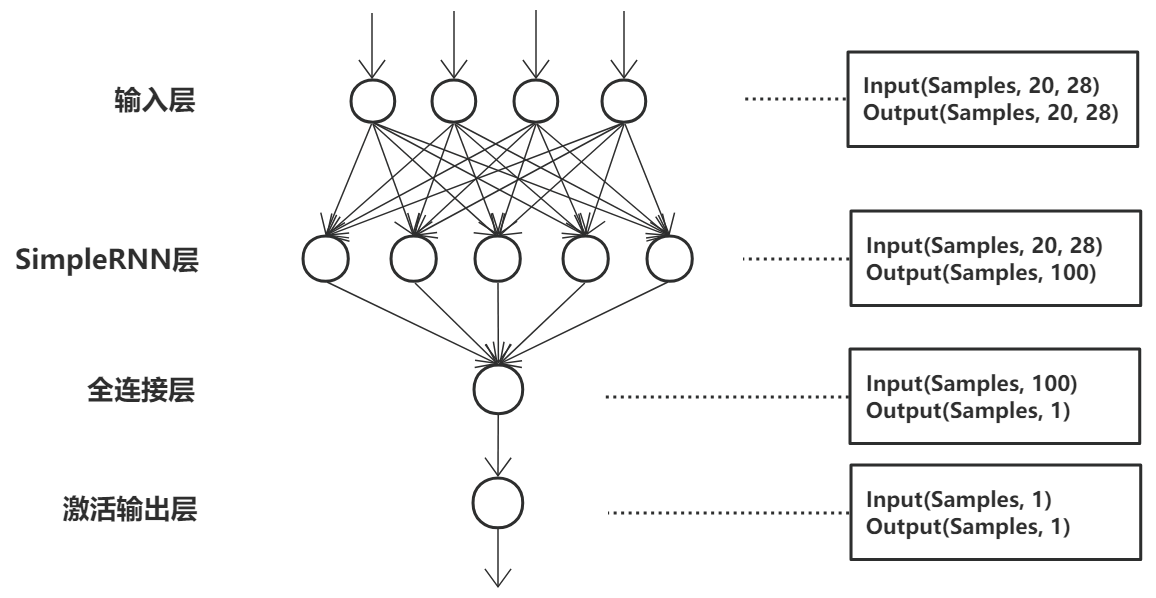


图 4.6 SimpleRNN网络结构图

在SimpleRNN模型当中，输入样本首先经过一层输入层，在这层当中数据的形状不变，图中右侧矩形当中的“Input”和“Output”代表该层的输入和输出数据的形状，第一位“Samples”指的是样本数量，由于训练的时候优化器的选择不同(见第2.4节优化器的相关介绍)，所以在单个迭代当中的数据数量不是固定的，可以通过后期的训练参数调整进行改变。“Samples”的大小决定了在一个迭代当中的数据量大小，“Samples”越大，训练时间越长，但相对的对于参数的改动也就越大，反之亦然。第二位，这里的数值是20，含义是每个样本当中含有的数据向量的个数，在4.1.1节输入样本的介绍当中参照图4.3，这里的20就对应图中的m，公式为

由此可见，图4.6当中的输入样本设计的时间窗口大小是600s，也就是10分钟。最后一位的数值代表每个数据向量的维度，代表的现实意义就是之前介绍的参考的工况数据标签的个数，和表4.1相对应，总共28个。

在SimpleRNN层进行相应的数值变换，这里每个神经节点都是一个RNN单元，总共有100个单元，每个单元有一个输出，所以在经过该层之后，输出数据的形状变为了一组维度为100的向量，注意到所有数值尽管在一维和二维之间转换，但是样本数量是不变的。

全连接层进行的操作是将100维的向量转化为1维向量，这里进行的是简单的线性变换，即基本的矩阵乘法，通过乘上一个系数矩阵改变数据的形式，不涉及到非线性的变化操作。

最后一层是激活输出层，该层需要用到之前在2.3.1节介绍的各种激活函数，由于不同激活函数的效果存在些许的差异，所以在本文进行模型训练的时候，比较了不同的激活函数对模型预测准确率的影响，最终选择了预测准确率和收敛速度相对最优的激活函数。在经过激活函数之后输出的一维向量，就是最后模型预测的值。

可以看到最后输出的数据是形如(Samples, 1)的二维矩阵，这和之前4.1.2当中介绍的输出样本是一致的，上图中仅表示了模型当中的各个层，但是没有涵盖训练模型的较为重要的函数——损失函数(介绍见第2.3.2节损失函数)。通过损失函数计算模型输出与样本输出之间的误差关系，同时利用反向传播算法(第2.2.1节当中介绍)进行参数修正。不同的损失函数对模型训练的影响不大，本文采用的损失函数是”平均绝对误差(MAE)”，公式如下

GRU网络的结构图和SimpleRNN的基本一致，如下图4.7，只是在中间的隐藏层当中替换为GRU单元，数量同样是100个，这个单元数量不是固定的，具体数量应该由实验得出，本文后续在第4.3节神经网络模型当中会详细比较不同单元数对于训练结果和收敛速度的影响。

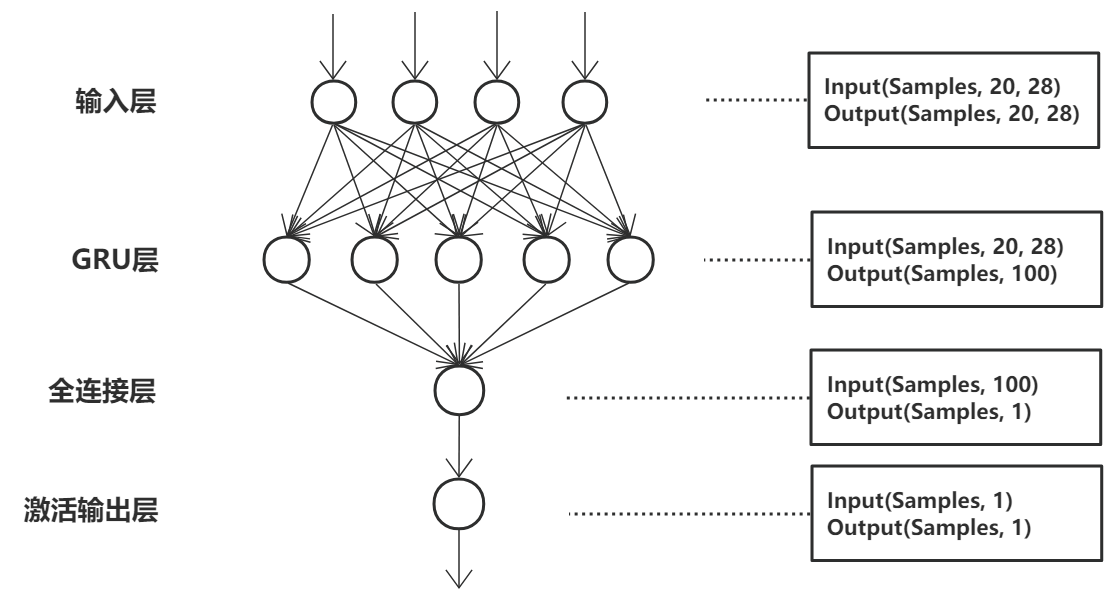


图 4.7 GRU网络结构图

全连接层的数目在上述几张结构图当中的单元数量都是1，这是因为激活函数不会改变数据的形式，所以为了使得模型输出和输出样本形状一致，所以增加了全连接层将数值向量的维度转化为1，可以在全连接层和GRU层之间增加隐藏层，本文在第4.3节研究讨论了增加隐藏层会带来的结果，在这里就不做赘述了。

和SimpleRNN网络相比，GRU网络由于GRU单元结构的复杂性(详见第2.2.4节)，训练的时间要更长一些，但相对的，在中长期预测上面会有更好的表现。

### 4.1.4 测试及可视化

需要指出的是，这里的测试并非是对模型总体表现做统一测试，仅仅是在训练模型的时候，依照对于测试集的表现程度，对于当前模型参数进行一个较为客观的优劣判断，根据测试的结果修正参数，在线性模型的训练当中，需要对于岭回归的参数 进行测试，找出最优的值；在神经网络模型的训练当中，需要进行更多的参数测试，主要如下表4.3所示。

表格 4.3 模型测试调整参数

|  |  |
| --- | --- |
| **参数名称** | **参数含义** |
| Units | GRU或SimpleRNN层的节点个数 |
| batch\_size | 训练中单次迭代的样本个数 |
| Epochs | 对整个数据集的遍历次数 |
| Activation | 激活函数 |

本章当中的预测曲线和测试集曲线的可视化采用的是Python中的matplotlib.pyplot库，以时刻为横坐标，烟气数值为纵坐标，绘制曲线；数值对比采用的是Excel绘制的图表。

本模型解决的是连续数值预测的回归问题，对模型的主要的评价参数是“解释方差分数”，公式如下：

其中代表测试集的真实值，代表模型计算得出的预测值，表示方差。最好的分数是1.0，分值越低，模型的预测效果或者说是拟合程度越差。

## 4.2 线性模型

### 4.2.1 最小二乘法可变参数设计

最小二乘法，也就是最普通的线性回归方法，应用在本文模型当中也较为简单。运用如下系数矩阵的计算公式，通过输入训练样本(输入样本和输出样本)，可以计算出相应的系数矩阵，进行下一步的测试和评价。

由上述公式可以看出，系数矩阵只和训练样本有关，因此该模型的训练速度也相当快，因为仅仅涉及了矩阵运算，不包含数据集遍历和迭代修正。

在模型训练的时候，对于每一种输出样本，即5种不同的废气指标进行了测试，训练集的个数为150,000，即有该数量的样本，测试集的数目为1,000，预测时间步长TimeAfter定为5，即预测距离当前时刻150s的时间间隔的烟气排放数值。具体的参数设置如下表4.4所示，这些参数在训练和测试当中维持不变。

表格 4.4 线性模型固定参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数数值** |
| INPUT\_DIM | 输入数据向量维度，即参考的工况数据标签的个数 | 28 |
| TIME\_AFTER | 预测时间步长，即预测时刻距离当前时刻的距离，单位30s | 5 |
| TRAIN\_SIZE | 训练集大小，即训练样本的总数 | 150,000 |
| TEST\_SIZE | 测试集大小，即测试样本的总数 | 5,000 |

考虑到不同的时间窗口可能对预测效果产生影响，所以本文比较了不同的时间窗口下的预测模型得分，同时，不同的烟气预测效果不同，因此对于每一种烟气都有相应的预测测试。下表4.5展示了在本次模型训练当中可变的参数。

表格 4.5 线性模型可变参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数数值** |
| GAS | 预测的气体种类 | CO, HCl, Smoke, NOx, SO2 |
| TimeStep | 时间窗口大小，即采用过去数值向量的个数 | {5, 10, 15, 20, 25, 30} |

**1）时间窗口大小的研究**

下图4.8是在预测对象为CO气体的时候，模型的解释方差分数和时间窗口大小的关系，图中的横坐标为时间窗口大小，单位为30s，纵坐标为解释方差分数，单位为1。该分数越高，说明模型的适应性越好。公式在4.1.4当中给出，即式。

可以看到，在时间窗口为4时，即采用的历史数据时过去150s的工况数据的时候，模型的预测效果最好。由于时间窗口不能是小数，所以最低的粒度就是1，所以选取4作为在预测CO气体时的时间窗口，本文的实验结果表明，当预测的气体不同时，最适合的时间窗口大小也不同，推断原因是反应在工厂垃圾焚烧工业环境当中，不同废气的产生源头不一样，流速和处理环节也不一样。

图 4.8 CO预测模型解释方差分数与时间窗口对应关系图

通过对各个烟气预测模型时间窗口的实验，得到的最适合的时间窗口大小如下表4.6所示，表中时间窗口的单位为1，每个单位时间长度为30s。

表格 4.6 线性模型最优时间窗口

|  |  |
| --- | --- |
| **烟气类型** | **最适合的时间窗口大小** |
| CO | 4 |
| HCl | 18 |
| Smoke(烟气含尘) | 16 |
| NOx | 22 |
| SO2 | 6 |

**2）不同烟气的预测效果**

预测的烟气一共有五种，不同烟气的预测效果不同，相应的解释方差分数如下图4.9所示，其中，烟尘预测的分数不好，CO的预测得分中等，其余三种烟气预测的得分较高。

从预测的得分上面来看，HCl、SO2和NOx三种气体的预测模型都较为理想，CO和Smoke的预测模型虽然得分相对偏低，但是未必就不能较好的反应烟气的变化规律，这个时候需要通过预测数据曲线和测试数据的曲线可视化对比，比较相应的曲线差异。

在这里需要指出的是，在实际工业环境当中，预测数值的精准度不是最重要的，重要的是能够及时预测出烟气的变化规律，这样能为工厂在垃圾焚烧处理当中提供预警——当烟气有上升的趋势时，就要及时的采取措施。

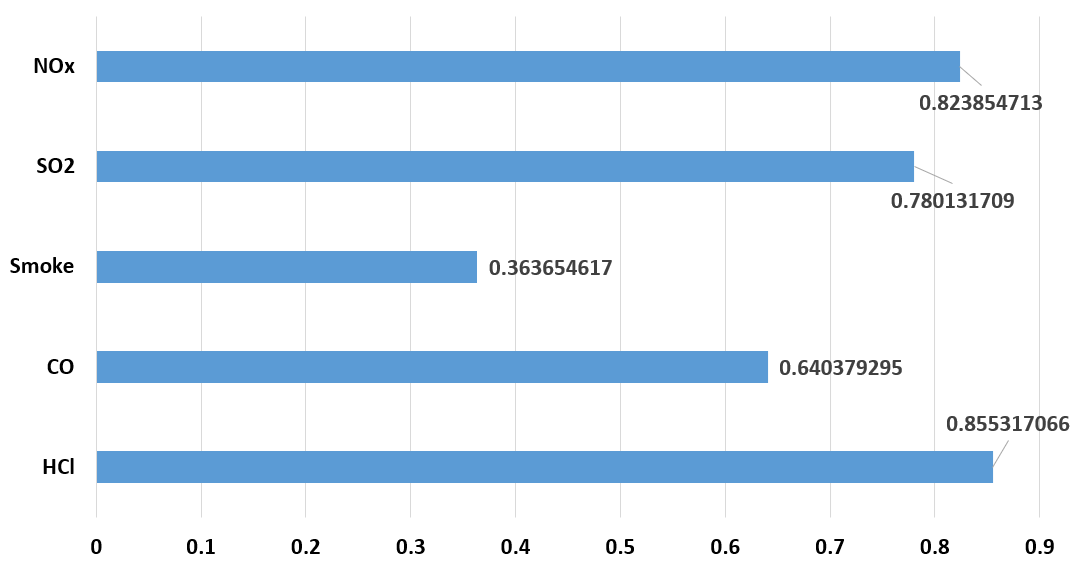


图 4.9 不同气体的最小二乘法模型解释方差分数

下文给出了五种气体的预测值和实际值的关系图，其中蓝色线为测试集当中的烟气变化曲线，即实际的烟气值，黄色线为模型预测的延期变化曲线，即预测值。通过进一步研究预测值和实际值的相对关系，能够对模型的精准程度进行更为直观的判断。

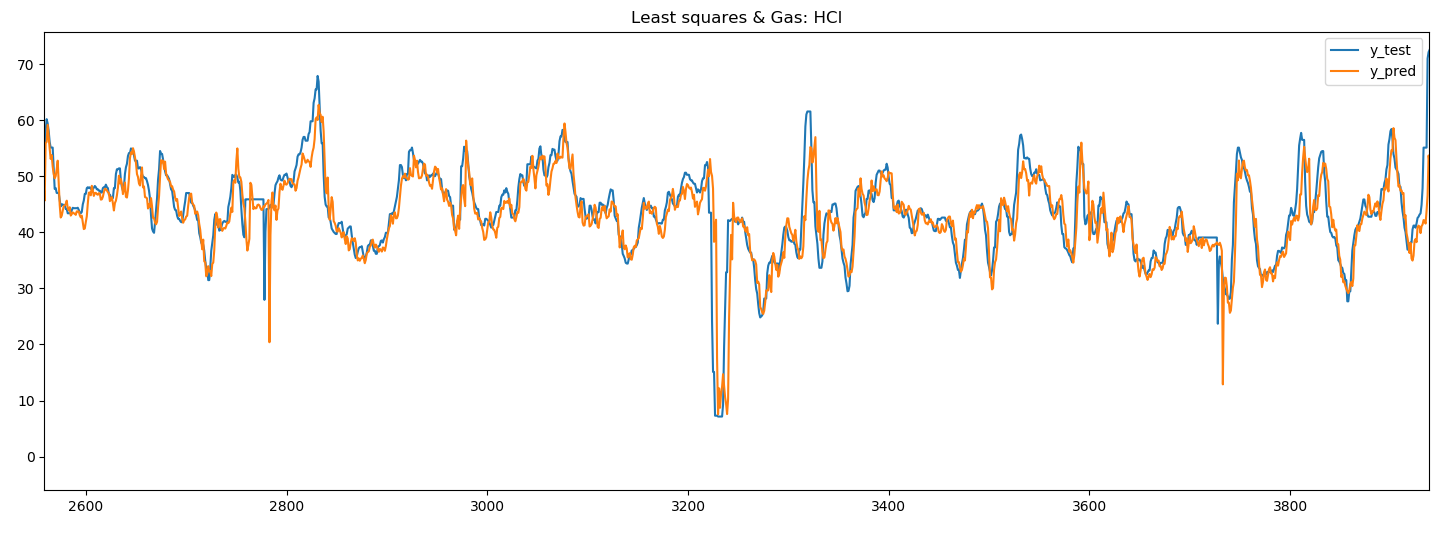


图 4.10 最小二乘法HCl预测曲线

上图4.10给出的时预测HCl烟气时的预测曲线和实际值曲线的关系，该模型的解释方差分数为0.855，可以看到预测曲线和实际值曲线吻合的还是不错的，同时也必须注意到，预测曲线在峰值附近存在着些许的滞后性，但不是很明显，这可能是由于在峰值附近会存在人为改动垃圾入炉速率的可能性，因为参考数据当中没有垃圾入炉量这个参数，所以会有一定的偏差，但从总体上来讲，预测效果还是相当不错的。

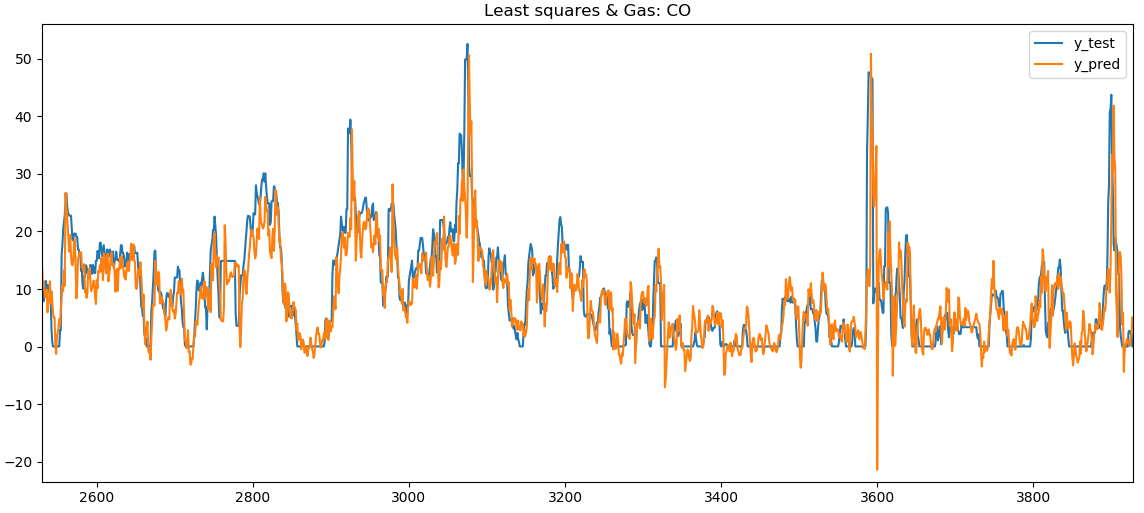


图 4.11最小二乘法CO预测曲线

上图4.11给出的时预测CO烟气时的预测曲线和实际值曲线的关系，该模型的解释方差分数为0.640，分数一般，但是可以看到，曲线的吻合程度还是令人满意的，由于数据集可能存在着些许未必是真实的值，反应在预测上面就是会出现较大的波动，如图在第3600个测试样本附近，出现了较大的曲线波动，这可能是有部分测试数据出现了偏差。在总体上来说，CO的烟气预测模型，能够反映出真实数据的变化规律，具有较大的参考价值。同时也存在相似的情况，仍然在部分峰值附近会存在滞后现象，但并不明显。

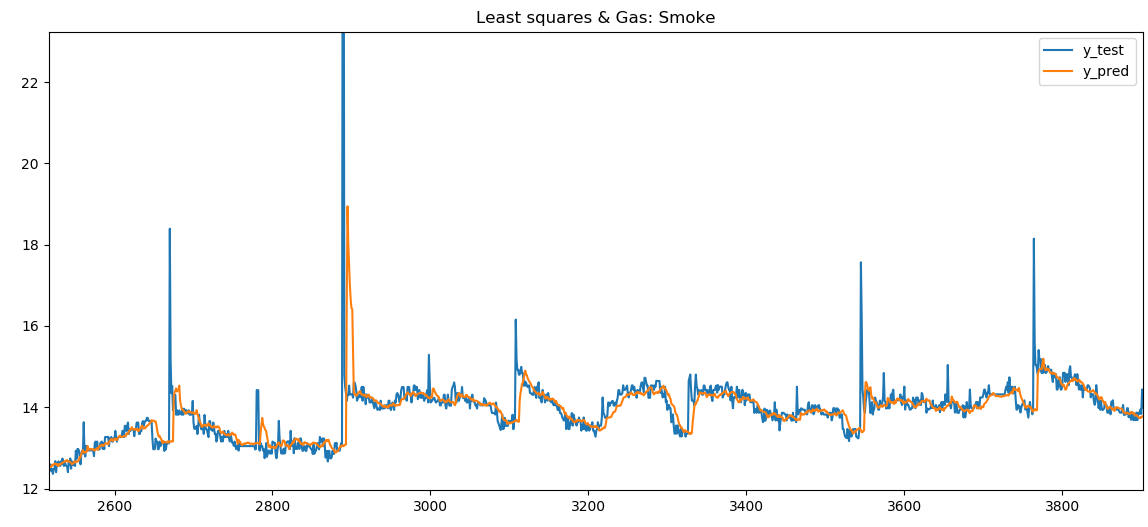


图 4.12最小二乘法烟尘预测曲线

上图4.12给出的时预测烟尘时的预测曲线和实际值曲线的关系，该模型的解释方差分数为0.364，属于一个比较低的值。从预测曲线图上面来看，其实在大多数预测点上，预测的数值还是比较准确的，但是在峰值上面的偏差很大。可以看到测试集的峰值出现的非常的突然，瞬时的斜率很大，考虑可能是其他的因素导致的。但在一般值附近，曲线还是有较好的表现的。

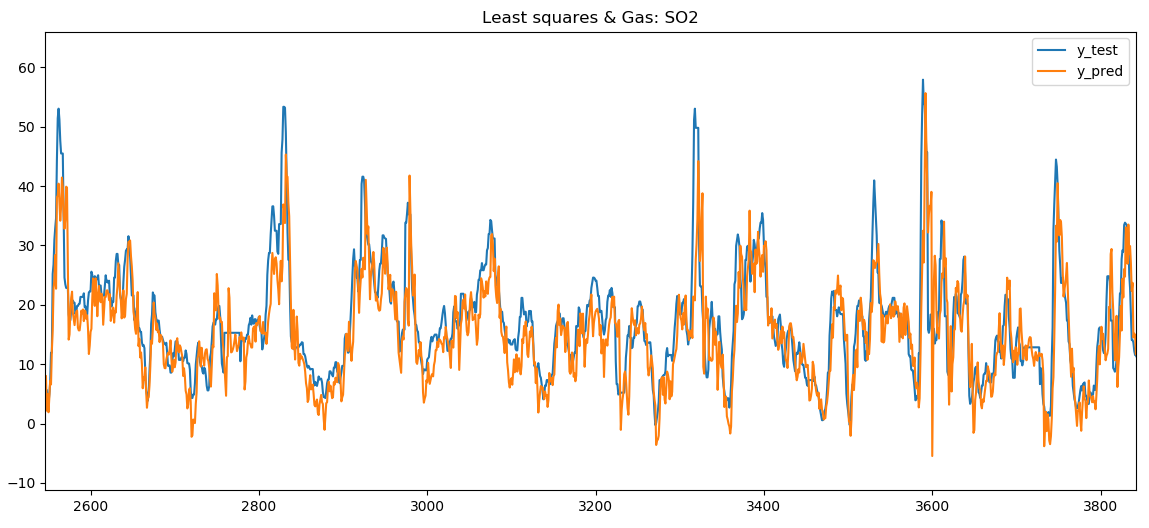


图 4.13最小二乘法SO2预测曲线

上图4.13给出的时预测SO2烟气时的预测曲线和实际值曲线的关系，该模型的解释方差分数为0.780，分数较高。同时可以看到在和实际曲线的比较当中，吻合的也还是不错的，但是在峰值附近会有一些偏差，往往是预测值小于峰值，同时滞后性也存在，考虑到这是线性模型，可能这种情况难以避免。推测之所以该模型的分数低于HCl，是因为在谷值的预测上会比真实值偏低一些。

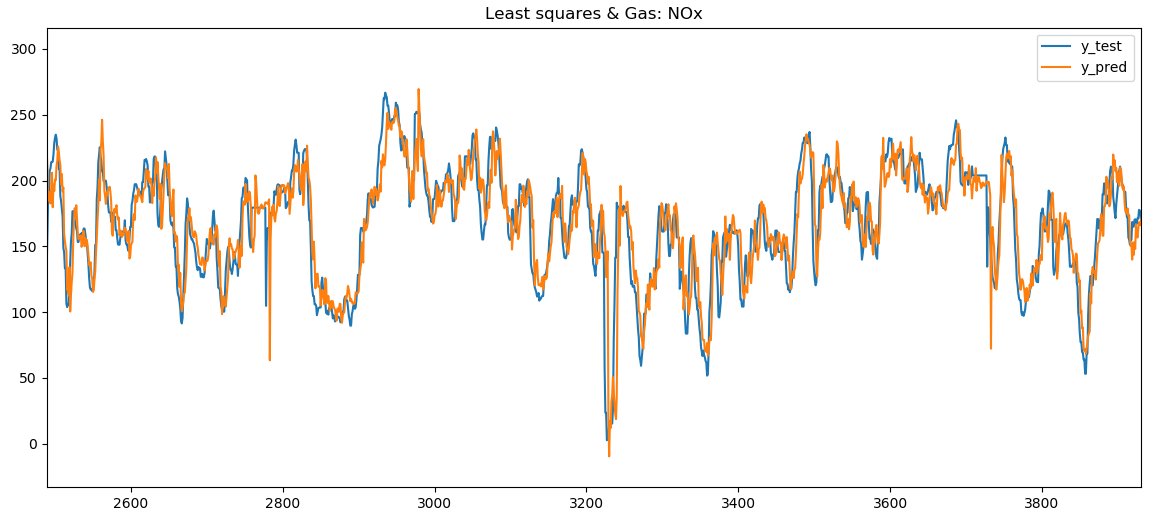


图 4.14最小二乘法NOx预测曲线

上图4.14给出的时预测NOx烟气时的预测曲线和实际值曲线的关系，该模型的解释方差分数为0.824，分数很高。该模型的预测曲线和实际值吻合的相当好，从图片上看到，除了有一些突然下降的谷值吻合的一般之外，在变动幅度相对正常的部分，预测值和实际值基本一致。从这里可以发现，其实之所以之前几个预测模型会在峰值部分出现些许的滞后性，可能是因为在突然变化的部分，没有相应的数据能够进行较好的拟合，也有可能是这样的突变点本来就比较少，在实际工业场景当中也只是偶尔出现。

### 4.2.2 岭回归可变参数设计

岭回归的不变参数和可变参数和最小二乘法一致，但和普通最小二乘法不同，岭回归需要在进行模型训练的时候设置的值，由于不同的数据条件下，的最适合的值不同，所以需要进行实验，得到一个相对最优的值。本文以CO气体为例，进行了的最优值实验。下图4.15展示了不同下，通过岭回归方法得到的模型分数的分布曲线，可以发现，在取0.05到0.20当中的某个值的时候，模型具有最好的解释方差分数，这个就应该是最适合本文面向的数据集的参数值。

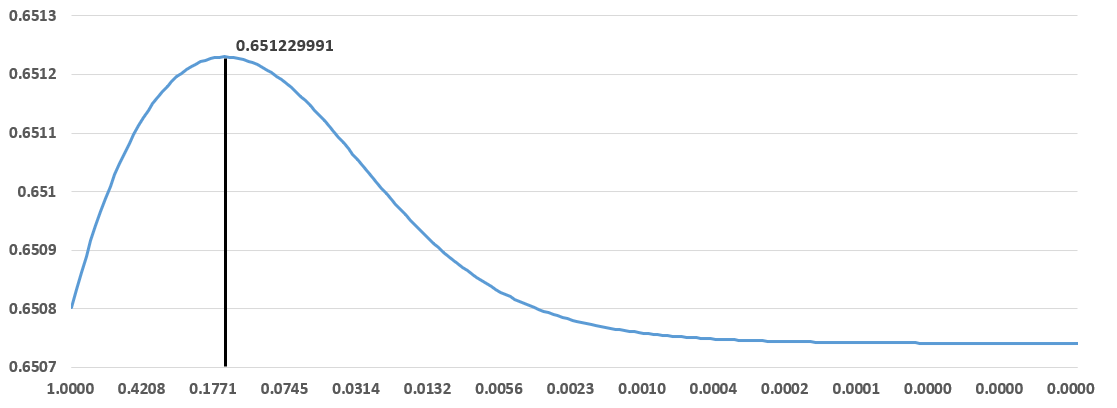


图 4.15 值(0, 1)和模型解释方差分数的关系图

为了找到的值，继续在区间(0, 0.5)之间，以0.001为间隔进行实验，得到的解释方差分数和的对应关系如下图4.16所示，最终选定0.156为本文岭回归模型的的值。需要注意的是，本次的时间间隔在0.001，因此找到的该值也只是近似最优解，但由于相邻的取值点所对应的分数相差已经不大，所以取该值作为最终的值也是合理的。

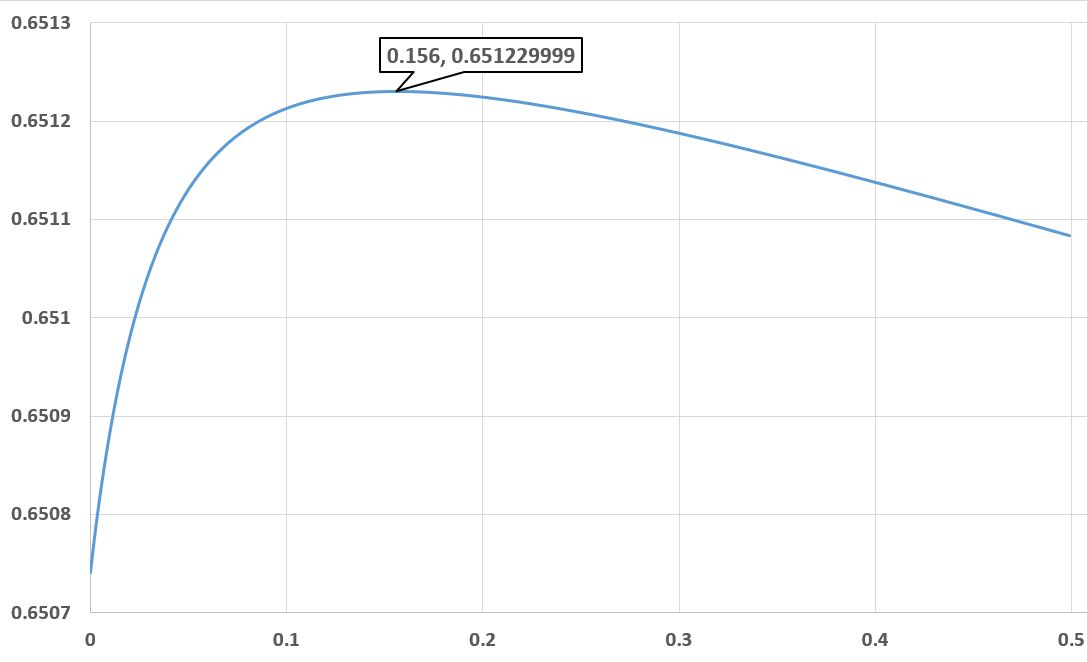


图 4.16 值(0, 0.5)和模型解释方差分数的关系图

在确定了的值之后，可以进行模型的训练和测试。

必须指出的是，不同的预测对象，其预测模型最合适的未必是相同的，在具体的操作过程当中，需要按照上文介绍的步骤一个个进行实验，得到最适合的值和对应的分数。最后得到的不同气体的值如下表4.7所示。

表格 4.7 不同烟气对应的最佳值

|  |  |
| --- | --- |
| **烟气类型** | **最适合的** |
| CO | 0.119 |
| HCl | 0.049 |
| Smoke(烟气含尘) | 1.516 |
| NOx | 0.001 |
| SO2 | 3.861 |

由于岭回归是普通最小二乘法的改进版，所以时间窗口的大小维持不变，依旧保持在4。进行训练之后，不同气体的预测模型的解释方差分数如下图4.17所示，图中包括进了普通最小二乘法的分数，以作对比。

从下图可以看到，尽管岭回归的确在一定程度上提高了模型的预测精准度，但是从总体的分数分布上面来看并没有提升太多，在模型精准度较高的NOx和HCl烟气上面，甚至基本一致，难以提升。

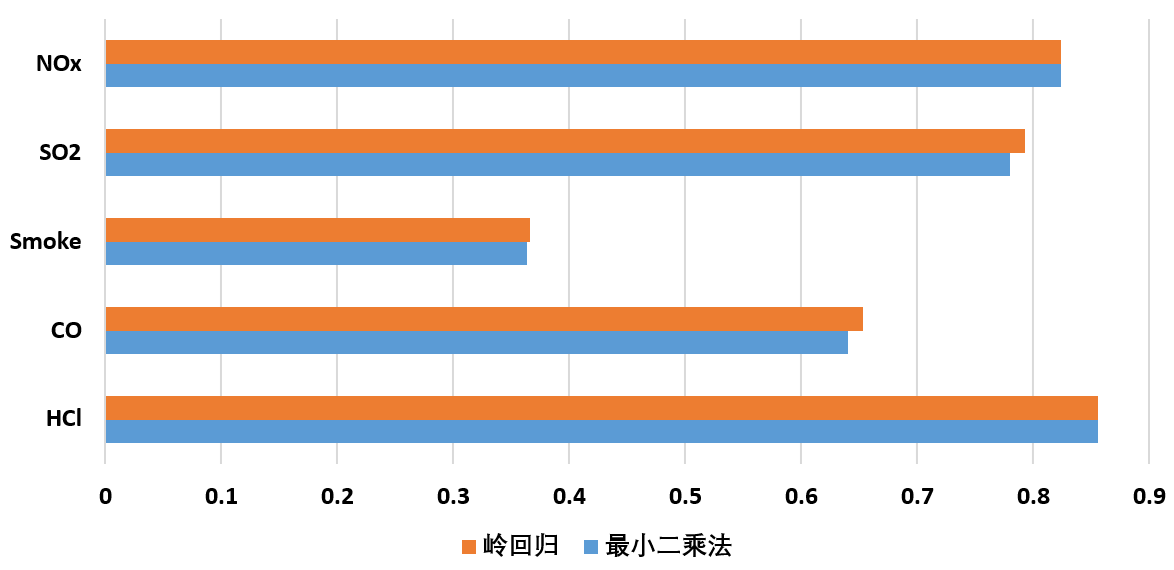


图 4.17 不同气体的岭回归模型解释方差分数

从里也看出线性模型对于数据集的拟合程度已经到达了极限。下一节4.3节将会探讨基于循环神经网络的模型在本文探讨的工业环境下的预测能力。

## 4.3 神经网络模型

本文探讨的神经网络模型主要分成两种——SimpleRNN模型和GRU模型。这两种模型的网络结构图详见第4.1.3节模型内容，本节将详细介绍这两种模型的训练过程以及最终形成的模型参数。

和线性模型在训练之前需要确定不变参数和可变参数一样，本文在训练神经网络模型的时候也确定了不变参数和可变参数，不变参数具体如下表4.8所示，需要指出的是，迭代训练的样本数目BATCH\_SIZE和训练集遍历次数EPOCHES仅和最终收敛的模型参数有关，如果这两个数值在合适的程度，那么就可以确保模型最终是收敛到合适的范围，本文将这两个值固定在相应的数值，是基于大量实验得出的较为合理的行为，如果数据集发生变化，那么就需要调整这两个参数，以确保模型训练能够顺利的收敛。损失函数和优化器都关系到模型的收敛速度，这里给出的参考函数也是基于本文采用数据集的特点设定的。

表格 4.8 神经网络模型固定参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数数值** |
| INPUT\_DIM | 输入数据向量维度，即参考的工况数据标签的个数 | 28 |
| TIME\_AFTER | 预测时间步长，即预测时刻距离当前时刻的距离，单位30s | 5 |

表格 4.8（续） 神经网络模型固定参数

|  |  |  |  |
| --- | --- | --- | --- |
| TRAIN\_SIZE | 训练集大小，即训练样本的总数 | 150,000 | |
| TEST\_SIZE | 测试集大小，即测试样本的总数 | 5,000 | |
| BATCH\_SIZE | 每次进入模型训练的样本数目 | 20 | |
| EPOCHES | 训练集遍历的次数 | 10 | |
| LOSS | 损失函数 | MAE | |
| OPTIMIZER | 优化器 | Adam(0.001) |

神经网络模型当中的可变参数如下表4.9所示，和线性模型相同的是，需要对不同的烟气分别建立模型，进行预测和可视化，类似的，还要探究不同的时间窗口对于模型预测能力的影响。在神经网络模型当中，还存在着两个比较重要的可变参数，一个是Activation激活函数，在第2.3.1节当中简要介绍了主要的几种激活函数，神经网络的训练环节，会用到多种激活函数，通过比较不同激活函数的效果，选取相对最优的激活函数；另一个参数是网络层的节点Units，节点数并不是越大越好的，节点数少的时候，训练速度快，但是表达能力较弱，节点多的时候，训练速度慢下来，但是表达能力会有所上升，本文将比较不同数值的Units对于模型训练速度和精准程度的影响。

**表格 4.9 神经网络模型可变参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数数值** |
| GAS | 预测的气体种类 | CO, HCl, Smoke, NOx, SO2 |
| TimeStep | 时间窗口大小，即采用过去数值向量的个数 | (2, 30) |
| Activation | 激活函数 | 激活函数集 |
| Units | SimpleRNN或GRU层的节点个数 | (10, 200) |

由于GRU和SimpleRNN的网络结构类似，所以在预测时间窗口大小的时候，仅以训练速度较快的SimpleRNN模型作为实验对象。

### 4.3.1 时间窗口设计

下图4.18是SimpleRNN模型在对CO烟气进行预测时的解释方差分数和时间窗口的对应关系图，可以看到，时间窗口和模型经过测试集测试之后的得分并不是线性相关的，总体在一个范围波动，考虑这是因为不同的时间窗口会导致模型出现局部最优节，最后没有收敛到一个较为理想的参数矩阵附近。但大多数较为合理的点时出现在时间窗口为12-21之间，因此选取21作为后续模型的时间窗口大小。

可以发现，模型的分数具有一定的波动性，起初考虑是因为模型最终没有收敛，但是通过多次训练发现，在时间窗口确定的情况下，加长训练样本遍历次数并不能该改善模型分数。所以对于不同预测时间步长，需要分别寻找合适的时间窗口，因为时间窗口的不同可能导致相差0.1分这种较为大的分分差。

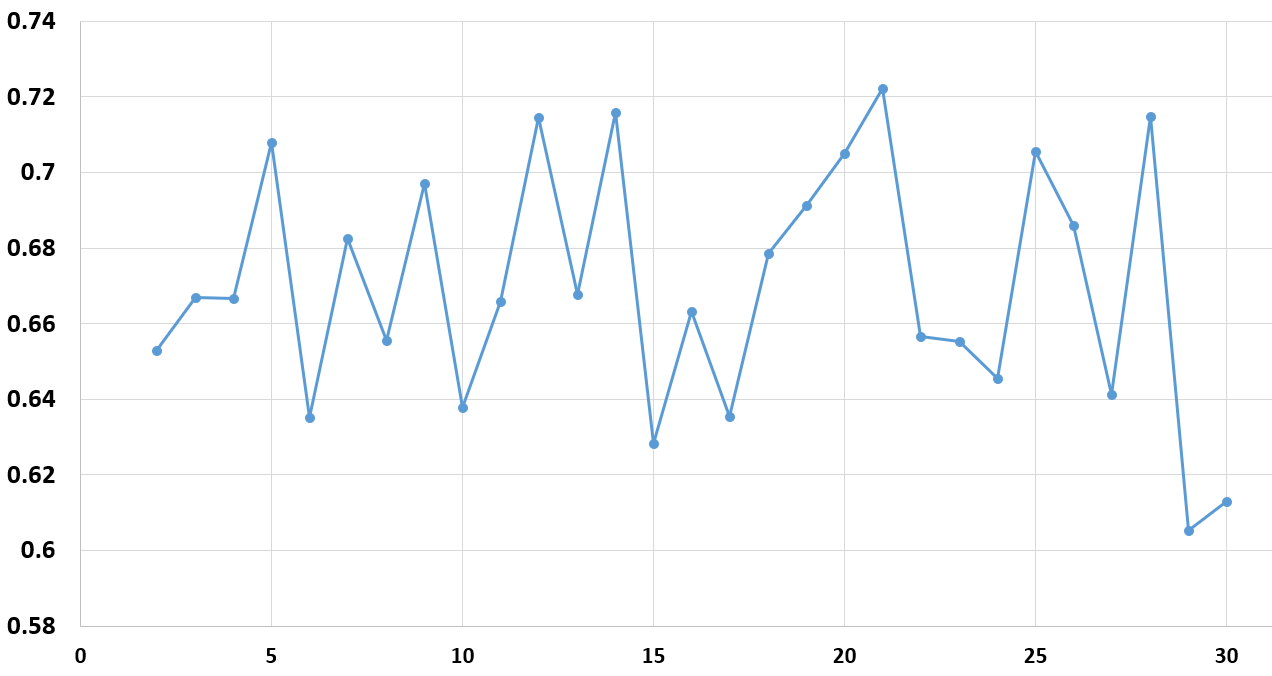


图 4.18 CO-SimpleRNN-5模型分数与时间窗口对应关系图

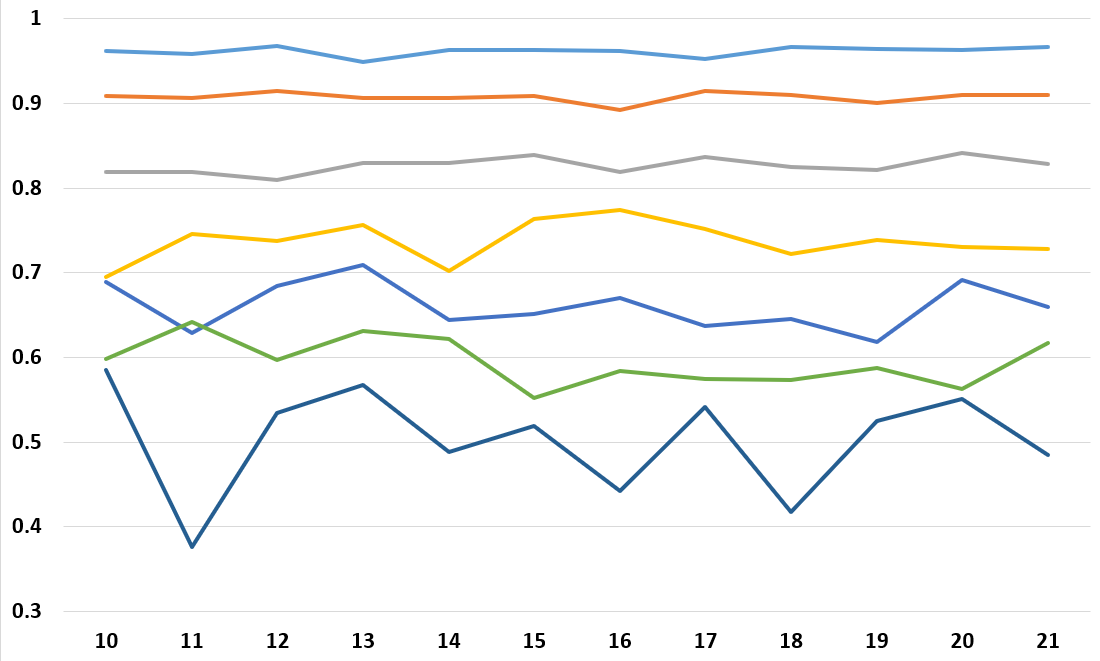


图 4.19 不同预测时间步长下时间窗口和模型分数的关系

因此对于SimpleRNN和GRU模型来讲，不同的预测时间步长，有不同的最合适的时间窗口，为了提高模型的预测准确率，应该通过实验得到每个时间步长下最适合的窗口大小。本文进一步探究了在不同预测时间步长下，SimpleRNN模型在预测CO气体时的分数，如上图4.19所示。

图中横坐标为时间窗口大小，纵坐标为分数，从上往下的曲线的时间步长为从1到7，可以发现，在模型预测效果比较好的情况下，曲线是比较稳定的，在模型预测效果较差的情况下，曲线的波动相当不稳定。出于模型训练和预测效果的综合考虑，将时间窗口定为20。

### 4.3.2 激活函数选择

上文讲到，不同的激活函数会影响训练所得模型的最终分数，也会在模型训练的时候影响模型的收敛速度。如hard\_sigmoid函数和sigmoid函数，由于其公式的不同， sigmoid训练时的计算速度明显更快。

本节对不同激活函数对模型的影响做了比较，建立模型，探究不同的激活函数所花费的训练时间，以及，最终训练所得的分数。这里为了节省时间，将训练样本的遍历次数设置为了5，每次进入模型的样本数设置为50。具体的参数如下表4.10所示。由于数据集足够大，能够充分保证模型已经收敛。

表格 4.10 激活函数比较时不变的模型参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数数值** |
| INPUT\_DIM | 输入数据向量维度，即参考的工况数据标签的个数 | 28 |
| TIME\_AFTER | 预测时间步长，即预测时刻距离当前时刻的距离，单位30s | 5 |
| TRAIN\_SIZE | 训练集大小，即训练样本的总数 | 150,000 |
| TEST\_SIZE | 测试集大小，即测试样本的总数 | 5,000 |
| BATCH\_SIZE | 每次进入模型训练的样本数目 | 50 |
| EPOCHES | 训练集遍历的次数 | 5 |
| LOSS | 损失函数 | MAE |
| OPTIMIZER | 优化器 | Adam(0.001) |
| Model Type | 模型种类 | SimpleRNN |
| TIME\_STEP | 时间窗口 | 21 |

下图4.20是模型在不同激活函数的影响下，最终得分的变化图，图4.21是模型在不同激活函数下，训练时间的变化图。

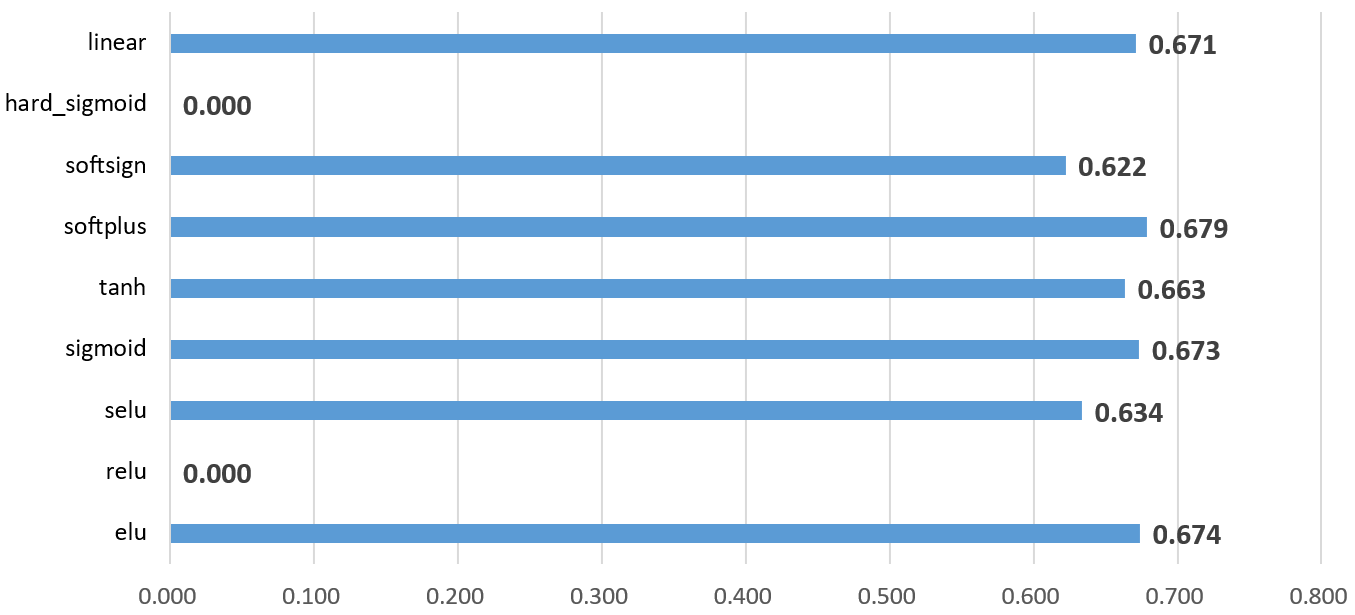


图 4.20 不同激活函数对模型分数的影响

在图4.20当中，激活函数relu和hard\_sigmoid所对应的模型最终得分都是0分，说明这两个激活函数并不适合CO的预测模型训练。这可能是由于数据集的特点，在利用这两个激活函数进行训练的时候，出现了梯度消失，无法更新参数的问题。

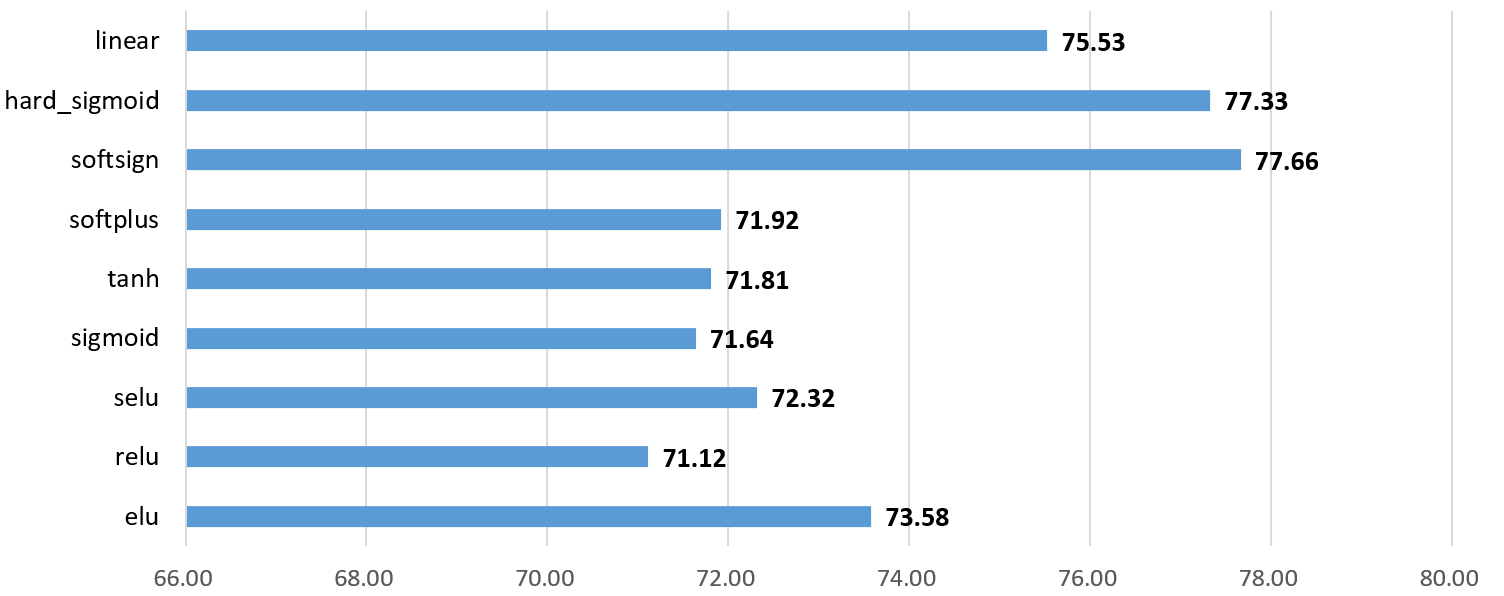


图 4.21 不同激活函数对训练时间的影响

可以看到，不同激活函数的训练时间有些许的差异，其中上图的单位是s，在遍历5次数据集的时候得到的时间差在6s左右，距离也不是很大。

综上所述，本文需要选取的激活函数，应该有较好的普适性、较好的最终训练模型分数以及较为合理的训练时间，因此本文选取了sigmoid函数作为激活函数。

### 4.3.3 节点个数设计

在第2.2节介绍神经网络的时候讲到，神经网络的节点个数影响到了网络的对于复杂结构的表达能力，不同的节点个数会带来不同的模型预测结果。当然，节点个数并不是越多越好的，应该找到最适合的个数，相应的，也要权衡训练时间和最终模型的得分关系。

本节依旧按照控制变量的原则，在训练模型之前确定了一系列的参数，这些参数在训练过程当中都是不会改变的，具体如下表4.11所示。

表格 4.11 节点个数比较时不变的模型参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数数值** |
| INPUT\_DIM | 输入数据向量维度，即参考的工况数据标签的个数 | 28 |
| TIME\_AFTER | 预测时间步长，即预测时刻距离当前时刻的距离，单位30s | 5 |
| TRAIN\_SIZE | 训练集大小，即训练样本的总数 | 150,000 |
| TEST\_SIZE | 测试集大小，即测试样本的总数 | 5,000 |
| BATCH\_SIZE | 每次进入模型训练的样本数目 | 50 |
| EPOCHES | 训练集遍历的次数 | 10 |
| LOSS | 损失函数 | MAE |
| OPTIMIZER | 优化器 | Adam(0.001) |
| Model Type | 模型种类 | SimpleRNN |
| TIME\_STEP | 时间窗口 | 21 |

注意到和第4.3.2节稍有不同的是，对样本集的遍历次数从5次改为了10次，这是因为考虑在节点个数过多的时候，5次的遍历次数并不能很好的使模型达到收敛，因此考虑增加遍历次数确保最后得到的分数都是模型稳定收敛之后的分数。

同样，本次通过改变循环神经网络层的节点个数，并探究模型最终得分和模型训练花费时间的关系。

下图4.22是模型在不同节点个数下，最终得分的变化图，图4.23是模型在不同节点个数下，训练时间的变化图。

可以看到节点个数和模型分数的关系并不是严格的线性关系，但在一定范围内，在这里是40-130之间，模型最终的得分都还是比较平稳的。

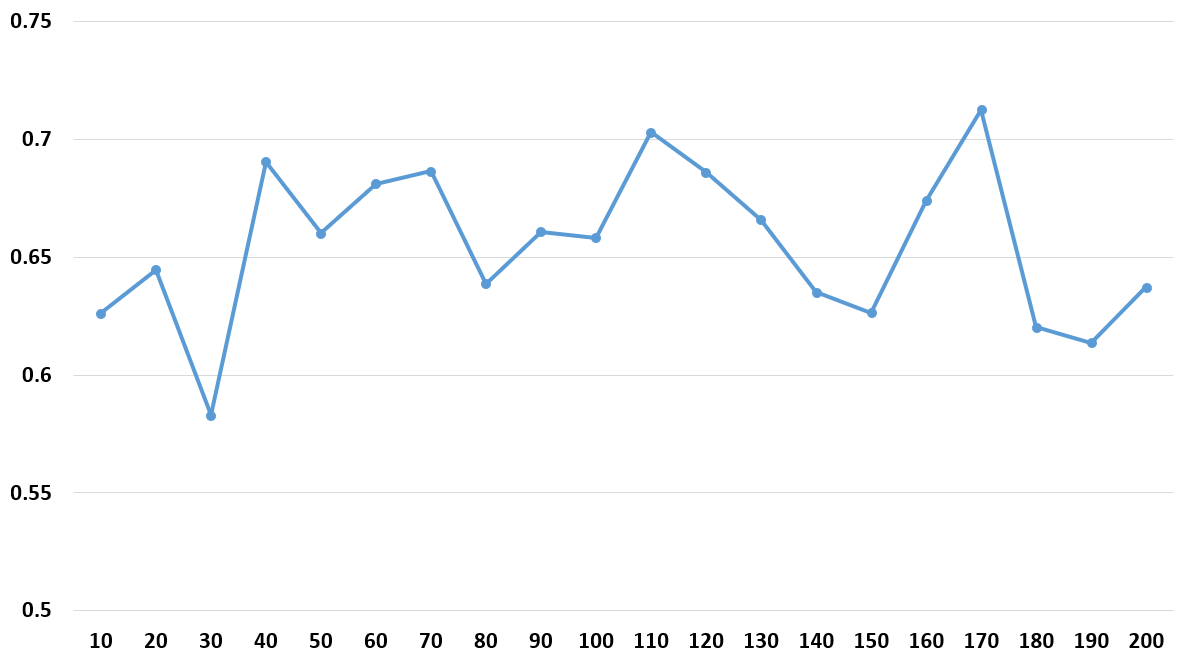


图 4.22 不同节点个数对模型分数的影响

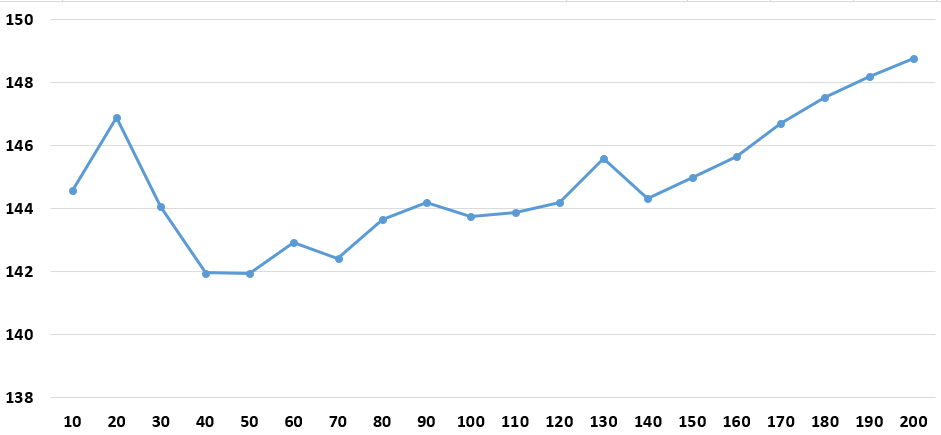


图 4.23 不同节点个数对训练时间的影响

由于遍历数据集的次数为10，所以尽管在图中最大最小值的差值在5s左右，其实像擦的时间并不是很多。可以发现，节点个数的增加的确会使得模型训练时间上升，但是由于幅度不大，因此在考虑节点个数的时候应该更加关注最终获得模型的分数。

考虑到不同模型的最优节点个数可能不同，因此对GRU模型也做了相似的实验，最后确定了网络节点个数如下表4.12所示。

表格 4.12 节点个数比较时不变的模型参数

|  |  |
| --- | --- |
| **模型类别** | **节点个数** |
| SimpleRNN | 110 |
| GRU | 100 |

# 5 测试与分析

## 5.1 测试目标

本文的测试目标是要验证模型对于不同烟气的预测效果，同时比较不同模型在相同预测条件下的表现能力。

其中预测的烟气一共有五种，分别是一氧化碳(CO)，二氧化硫(SO2)，氯化氢(HCl)，氮氧化物(NOx)和烟尘。不同的气体在不同的预测时间步长上有不同的预测模型，这里的预测时间步长，指的是当前时刻点和预测时刻点之间的时间间隔，单位为30s，即一个步长为30s，该单位并不是随机指定的，而是根据实际工业场景当中传感器采样特征制定的，详见第3.1.3节关于数据特性的研究。

本文的预测集中在短期预测上面，由于垃圾焚烧原料成分未知、垃圾入炉燃烧的速率未知，所以在长期预测上面存在着条件不足的问题，可以从后续的测试图像当中看到，当时间步长逐渐增大的时候，预测模型的准确率也在逐渐下降。

由于是预测烟气排放值是一个连续的数值，因此本文的预测问题也应是回归问题，结合回归模型评估的相关资料，本节为了衡量模型的预测准确率，提供了4种测评指标，下文将详细介绍这五种指标的内容。

1. **解释方差分数**

解释方差分数的公式如下：

其中代表测试集的真实值，代表模型计算得出的预测值，表示方差。最好的分数是1.0，分值越低，模型的预测效果或者说是拟合程度越差。可以从该分数上面看出模型的预测效果，然而，该分数低的时候，模型未必就没有价值，如在预测烟尘时的时候，模型的分数很低，但是对测试值和真实值的拟合来讲，出了高点的拟合一般，在正常值附近时拟合的都比较好，所以需要结合曲线可视化结果来一起分析。

1. **平均绝对误差**

平均绝对误差的公式如下：

其中代表测试集的真实值，代表模型计算得出的预测值。平均绝对误差很好理解，就是简单的误差绝对值加和取平均，需要看到的时，不同的数据基数下，该值具有不同的意义。如果数值的基数较大，如10,000~20,000，那么MAE就是为10-100也不算大。

1. **中间绝对误差**

中间绝对误差的公式如下：

其中代表测试集的真实值，代表模型计算得出的预测值，表示取中位数。可以看到在这个计算公式当中，排除了离群值的影响，具有一定的参考价值，尤其时在离群值较大、较多的情况下。

1. **R2分数**

R2分数的公式如下：

其中代表测试集的真实值，代表模型计算得出的预测值，代表测试集的平均值。该公式提供了未来样本能否被模型有效预测的量度。最佳分数为1.0，可以为负数，当模型为负数时就是更糟糕的情况。

## 5.2 测试策略

本文的测试主要分成以下几个步骤进行：

1. 测试普通最小二乘法模型，在不同气体不同时间步长下的测试指标
2. 测试岭回归模型，在不同气体不同时间步长下的测试指标
3. 测试SimpleRNN模型，在不同气体不同时间步长下的测试指标
4. 测试GRU模型，在不同气体不同时间步长下的测试指标

主要的不可变数据如下表5.1所示。

表格 4.8 模型测试不可变量

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数数值** |
| INPUT\_DIM | 输入数据向量维度，即参考的工况数据标签的个数 | 28 |
| TIME\_AFTER | 预测时间步长，即预测时刻距离当前时刻的距离，单位30s | 相对应的最佳数值，具体见第4章 |

表格 4.8（续） 模型测试不可变量

|  |  |  |
| --- | --- | --- |
| TRAIN\_SIZE | 训练集大小，即训练样本的总数 | 150,000 |
| TEST\_SIZE | 测试集大小，即测试样本的总数 | 5,000 |
| TIME\_STEP | 训练样本时间窗口大小 | 相对应的最优值 |
| ALPHA | 岭回归模型参数 | 与烟气对应的最优参数值 |
| BATCH\_SIZE | 每次进入模型训练的样本数目，神经网络模型使用 | 30 |
| EPOCHES | 训练集遍历的次数，神经网络模型使用 | 10 |
| LOSS | 损失函数，神经网络模型使用 | MAE |
| OPTIMIZER | 优化器，神经网络模型使用 | Adam(0.001) |

主要的可变数据如下表5.2所示。

表格 5.2 模型测试可变量

|  |  |
| --- | --- |
| **模型种类** | 最小二乘法，岭回归，SimpleRNN，GRU |
| **预测时间步长** | [1, 10] |
| **预测对象** | 一氧化碳(CO)，二氧化硫(SO2)，氯化氢(HCl)，氮氧化物(NOx)，烟尘 |

在获取指标数据之后，对比不同烟气下的各种模型预测表现能力。

不同的烟气，在不同的时间步长下有相应的预测效果曲线，每种预测指标应该分别绘图，同时，不同模型的在同一种预测条件下的预测指标应该绘制于同一张图，方便对比和分析。

最后，应该对于每种气体，选取相对较好的预测模型，进行预测曲线和测试数据曲线的可视化，对比两种曲线的吻合情况。

## 5.3 测试结果与分析

### 5.3.1 模型测试结果与分析

**1. 最小二乘法模型测试结果与分析**

在不同的预测时间步长、不同的烟气预测对象下，对该模型的预测指标进行了测试。测试时采用的训练数据集为和第4章当中的数据集相同，测试数据和训练数据来源于不同的时间段，都是连续的。解释方差分数如下图5.1所示，平均绝对误差如下图5.2所示。

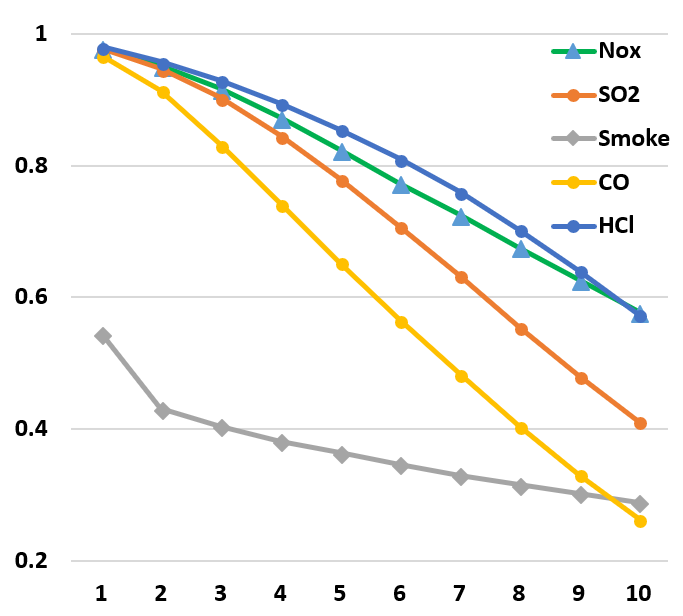
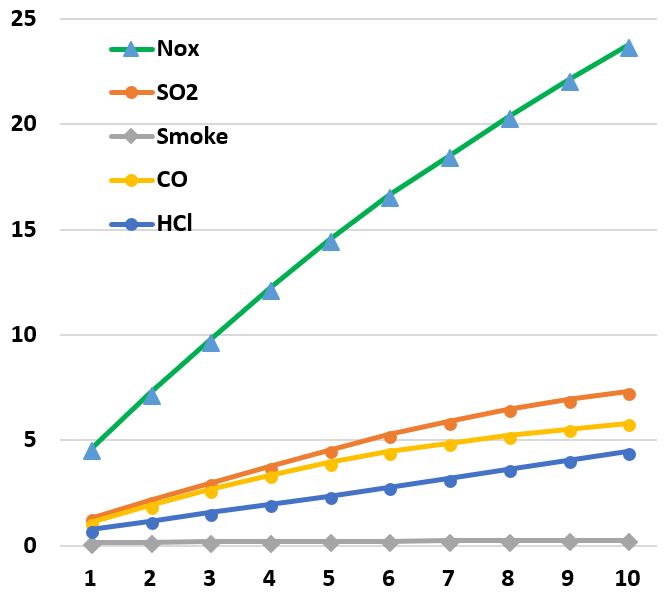
 

图 5.1 最小二乘法模型解释方差分数 图 5.2 最小二乘法模型平均绝对误差

R2分数和解释方差分数、中间绝对误差和平均绝对误差的图像类似，因此本文仅展示上述两张图。可以发现随着预测时间步长的增加，预测模型分分数和误差大小也在不断的增加，从分数的角度上看，HCl、NOx、SO2的模型较好，CO的模型一般，烟尘的模型较差，但是从误差上来讲，反而烟尘的误差最低，考虑这可能是由于烟尘数据曲线当中会存在较多突变点，影响了分数的计算。

如果单以解释方差分数作为评判标准的话，线性模型在预测时间步长为5以内，即预测时间不超过150s的时候，性能是比较出色的。

结合具体的曲线来看，如下图5.3，展示了在时间步长为5时的预测曲线和真实值曲线的关系，发现拟合的还是相当不错的，尽管在小范围的波动较为频繁，但是总体上还是表现出了真实值的波动规律。

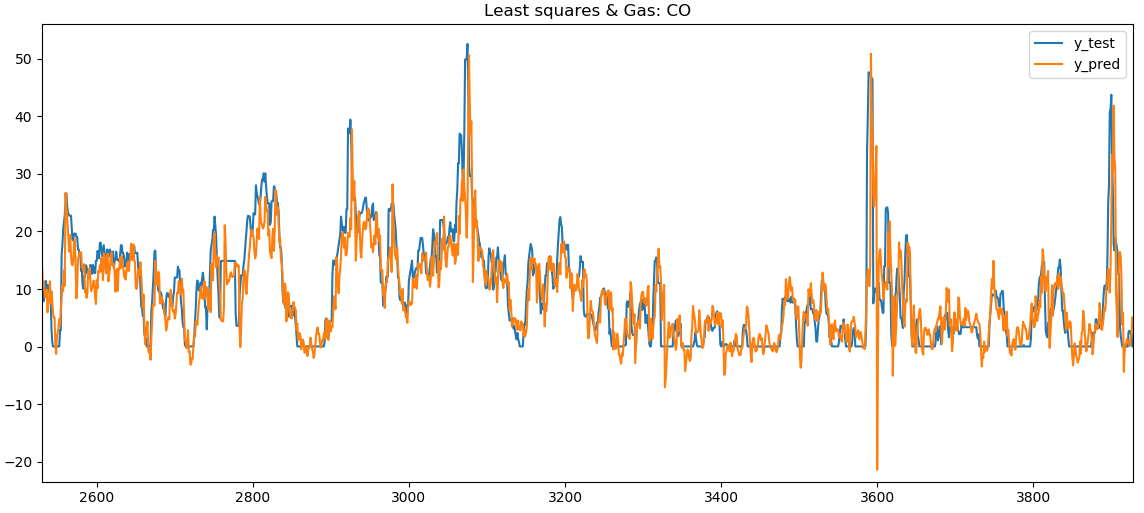


图 5.3 预测时间步长为5的CO预测曲线

岭回归模型仅仅是对最小二乘法模型做了些许优化，因此绘制的图像曲线基本不变。岭回归稍稍提升了线性模型预测的分数，但是提升程度较低，基本只有0.05左右。

**2. SimpleRNN模型测试结果与分析**

神经网络模型测试当中的固定参数在第5.2节测试策略当中已经给出，其中对于数据集的遍历次数已经足够大以保证最终模型能够收敛。神经网络模型训练所使用的训练集、测试所使用的测试集均和线性模型一致，模型的解释方差分数如下图5.4所示，平均绝对误差如下图5.5所示。

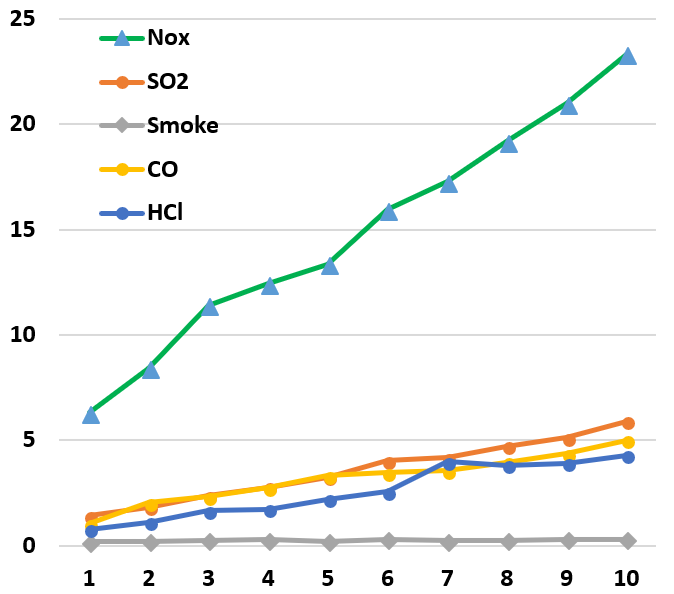
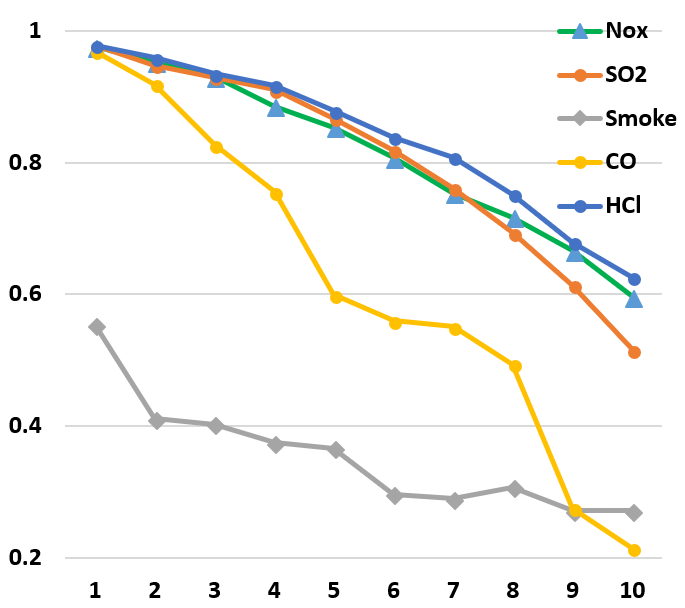


图 5.4 SimpleRNN模型解释方差分数 图 5.5 SimpleRNN模型平均绝对差

R2分数和解释方差分数、中间绝对误差和平均绝对误差的图像类似，所以此处仅展示上述两张图，已经足够做一个较为客观的评价。

和线性模型一致，在预测时间步长增大的时候，模型的预测得分呈现下降趋势，同时模型预测的平均绝对误差也在增加，但误差除了NOx显著增加之外，其他的烟气误差并不是很大，考虑到这或许时由于NOx烟气的波动幅度较大引起的。

和线性模型不同的时，模型的预测分数并不是平稳下降的，在小的时间预测步长范围之内的变化率可能会出现波动，如图5.4当中CO气体的4-5、8-9阶段，都出现一个急速的下降。考虑到模型拟合的可能时非线性关系，所以这也是可以理解的。

在SimpleRNN的预测当中，SO2气体的分数变化曲线要明显好于线性模型，其余气体的模型和线性模型的效果接近。这和烟气真实曲线的分布规律有一定的关系。

为了进一步观察SimpleRNN的预测效果，下图5.5可视化了预测曲线和真实值曲线的关系，其中预测的气体时NOx，预测时间步长为5。可以看到预测曲线和真实值曲线基本一致，但是在波动激烈的地方会出现一定的偏差，这个偏差往往时稍显滞后的，在第4.2节分析过该现象的原因，就是影响烟气波动的某个数值没有参考进模型的训练当中，推测是垃圾的入炉焚烧的速率。

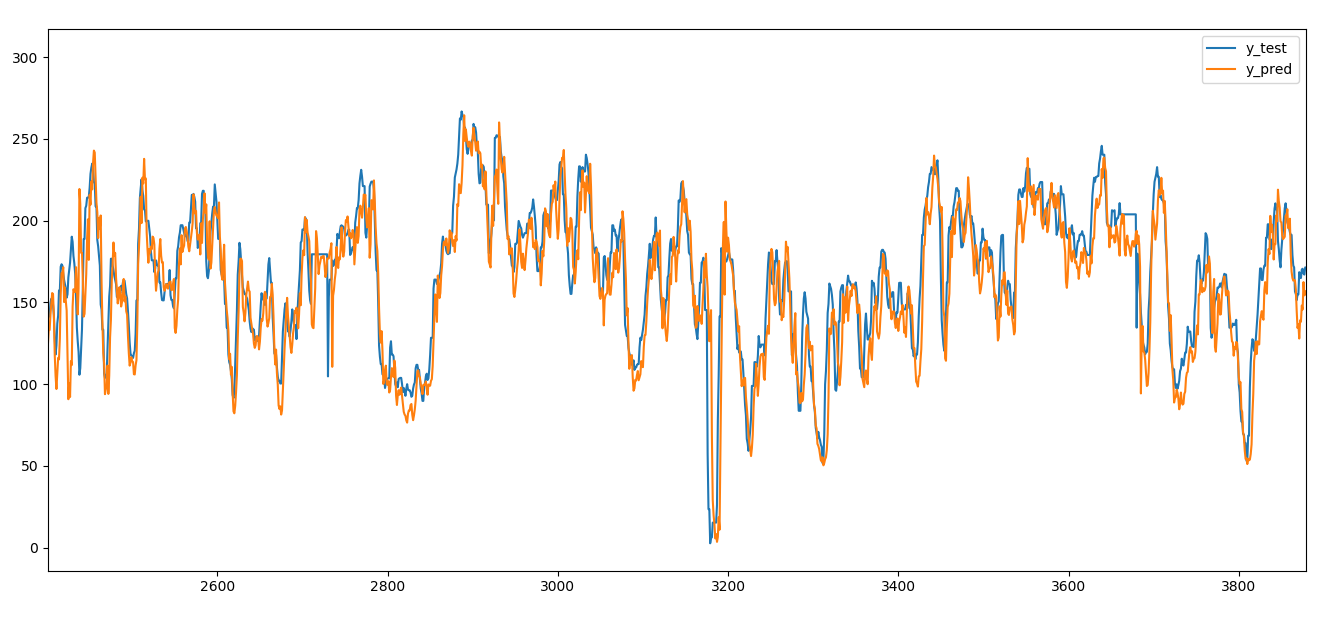


图 5.5 预测时间步长为5的NOx预测曲线

GRU模型的相关测试指标图像和SimpleRNN模型类似，在此就不作赘述。

### 5.3.2 模型对比与分析

本小节按照预测烟气的类别不同，分别做了模型对比的图像，不同的图像当中，预测烟气确定，纵轴为预测分数，横轴为预测时间步长。从图5.6到图5.10分别展示了NOx，SO2，烟尘，CO，HCl一共5种气体的模型对比。

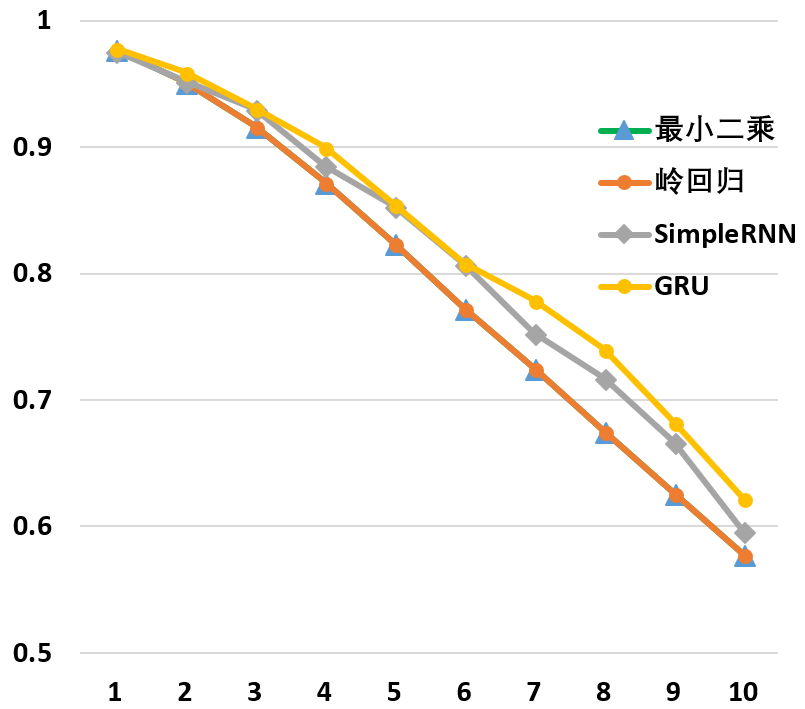
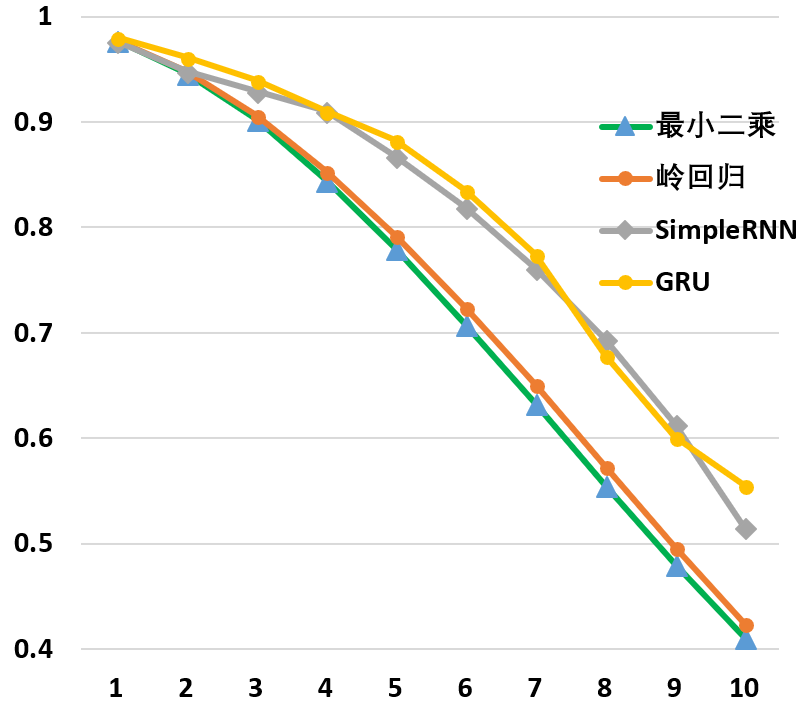
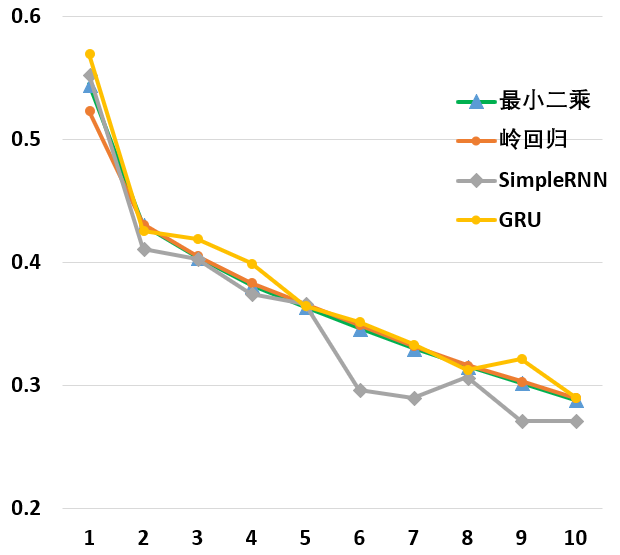
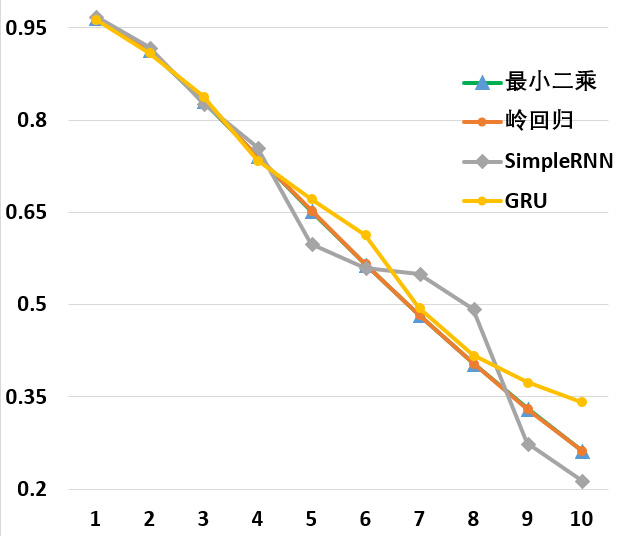
 

图 5.6 预测NOx时各模型解释方差分数 图 5.7 预测SO2时各模型解释方差分数

**图 5.8 预测烟尘时各模型解释方差分数 图 5.9 预测CO时各模型解释方差分数**

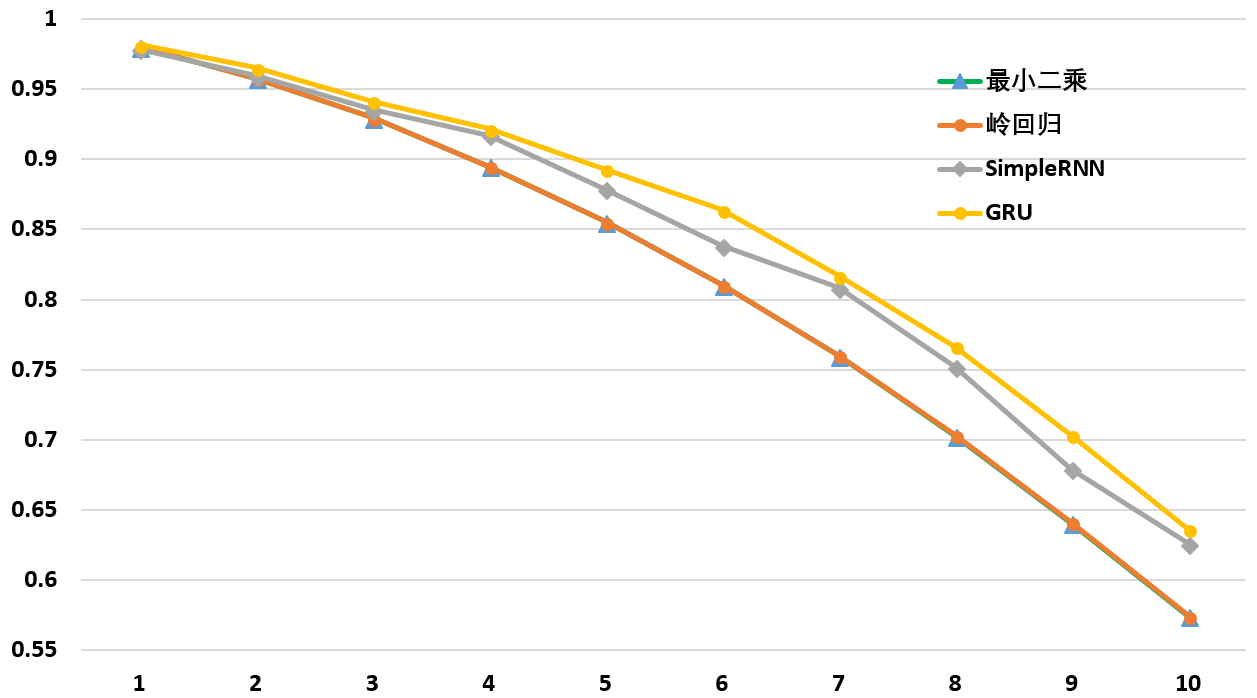


图 5.10 预测HCl时各模型解释方差分数

从5种烟气的预测模型的解释方差分数曲线可以看出，在预测CO、烟尘的时候，神经网络模型和线性模型的分数差别不大，但是在预测HCl、NOx、SO2这三种烟气的时候，神经网络模型最终得到的解释方差分数要明显优于线性模型，其中，在对比SimpleRNN模型和GRU模型的时候，发现GRU模型的分数基本上略高于前者，说明在处理本文针对的垃圾焚烧烟气预测问题上，面向长期依赖的问题的GRU模型具有更好的性能，同时也表明烟气和历史工况数据可能存在长期依赖关系，但不明显(GRU和SimpleRNN的分数差别不太大)。

从曲线的波动性来看，线性模型的两种方法——岭回归和最小二乘法的波动性不强，曲线非常的平滑，而GRU会有小幅度的波动，SimpleRNN则会出现较大的波动。因此在选择模型的时候，考虑到模型的稳定性和综合的预测分数，选择相对最优的模型，应该时GRU模型。

可以看到，尽管GRU相比其他模型提高了相同预测时间步长下的解释方差分数，但是依旧不能有效延缓随步长增大，解释方差分数曲线的下降趋势。然而尽管解释方差分数在下降，模型对于烟气的预测依旧是具有参考价值的，下图5.11给出了在预测时间步长为8，预测对象为HCl烟气时GRU模型的预测曲线，和真实值的曲线基本上时吻合的，在波动幅度较大的地方吻合一般，但是总体上能够反映出真实值的变化规律。

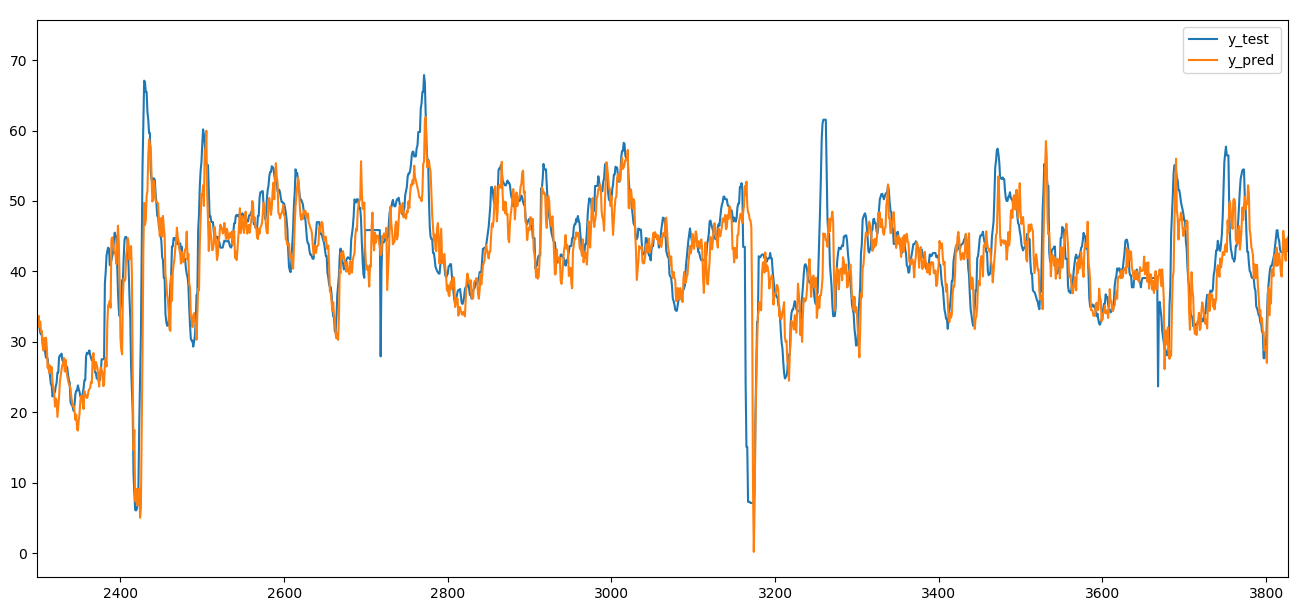
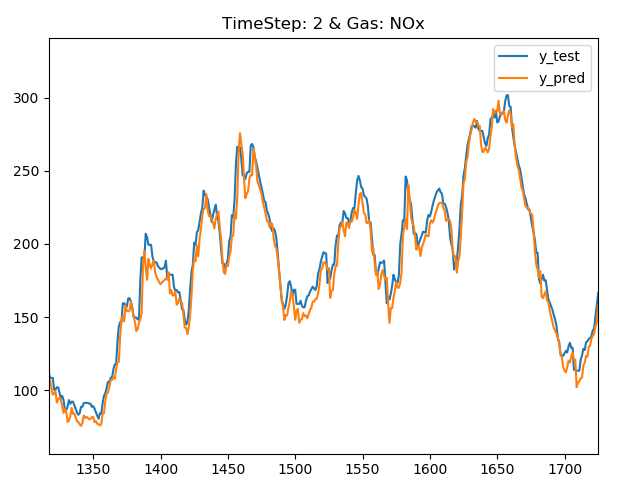
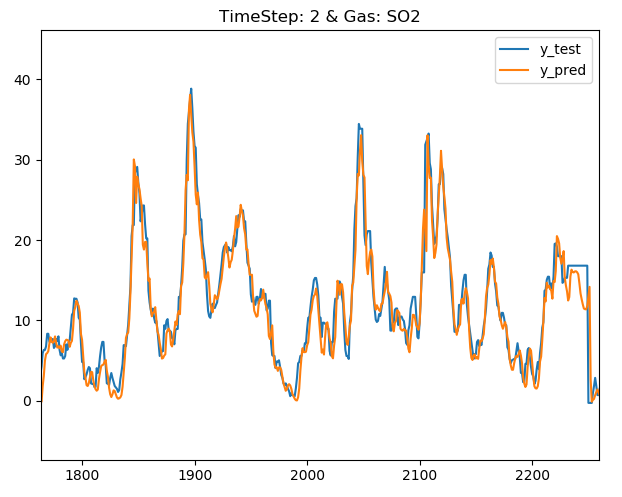


图 5.11 预测HCl的GRU模型预测值曲线

下图5.11给出了其他四种气体，在预测步长为2时，即时间间隔为1分钟时的预测曲线，其中预测模型采用了GRU模型，由于线性模型在烟气的预测上拟合更好，所以烟尘预测的模型采用的是岭回归方法。

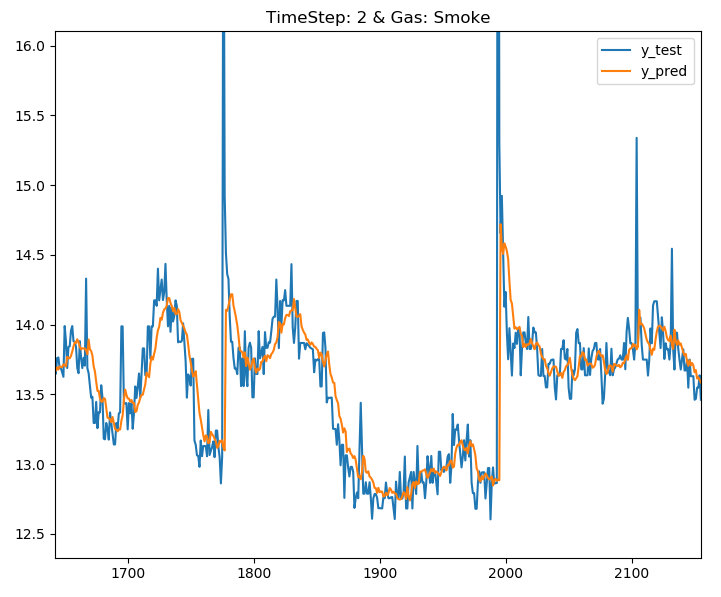
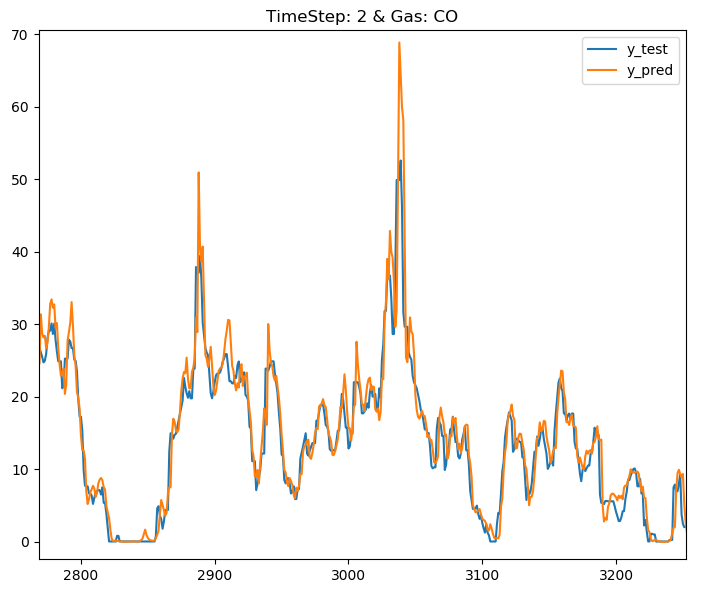
 

图 5.11 预测HCl的GRU模型预测值曲线

可以看到除了烟尘预测曲线在突然增高点的拟合一般，其他的曲线拟合的都是较为理想的，在短期预测上的准确率很高。

# 6 总结与展望

## 6.1 全文成果总结

本文针对以炉排炉为焚烧技术的垃圾焚烧厂排放烟气预测问题，提出了基于机器学习理论的预测模型。

在近来垃圾焚烧烟气研究问题上，中外普遍采用的是基于垃圾焚烧成分分析的技术，浙江大学的张东平博士提出的BP神经网络预测模型[11]也是以垃圾成分作为模型的输入。但是在实际的工业场景当中，没有分析垃圾成分的条件，甚至连垃圾入炉的速率都很难获取。

基于这种实际情况，本文提出了不依赖垃圾成分分析和入炉速率的烟气预测模型，首次将循环神经网络应用到了垃圾焚烧烟气预测问题当中。

本文通过对垃圾焚烧工况数据的特性研究，通过合理筛选可用数据、确定时间间隔、设计样本形式，形成了可供模型训练的，较为准确且理想的数据集，同时形成了一套完整的工况数据筛选流程，如果后续增加了新的数据标签，可以按照上述流程直接添加进数据集。

在模型的设计方面，本文结合了机器学习相关的理论，将普通最小二乘法、岭回归法、普通循环神经网络、GRU神经网络分别具体实现，按照数据集的特性，分不同的预测烟气探究了最适合的时间窗口大小、参数、网络节点个数等各种模型参数，同时在神经网络模型当中结合循环神经网络和多层神经网络设计了合理的网络模型，并进行了激活函数的测试，选取了较优的激活函数。

本文提出的模型在实际工业场景当中具有较大的实用价值，其中在短期预测，时间间隔为30s的预测方面，线性模型的预测效果好，且训练速度快，能够快速的进行短时间的烟气数值预测，而在较长的时间间隔上，神经网络的预测较为理想，虽然训练的时间较长，但是权衡提升的准确率来讲还是值得的。

相比于过去传统的烟气预测模型，本文提出的模型主要有以下几个优点：

1. 短期预测精准度高，相比于普遍的偏差10%(绝对平均误差10左右)的传统模型[11]，本文模型在短期预测上，除了NOx气体的误差较大外，其余气体的绝对平均误差均在5以内；
2. 依赖人工因素少，由于所有数据均来源与传感器采样，且本文已经找到了相对最优的时间窗口，所以后续的模型推广可以直接采用该时间窗口进行训练；
3. 优化潜力大，由于具体工业环境当中缺少相应的传感器和数据来源，所以本文提出的模型训练数据标签不包含垃圾成分和垃圾入炉速度，考虑在增设监控垃圾入炉速率的设备之后，模型的预测能力会有进一步的提升
4. 推广价值大，虽然本文采用的数据来源是基于炉排炉焚烧技术的垃圾焚烧工厂，但是由于不需要具体分析数据标签的含义，所以在应用其他技术的垃圾焚烧工厂当中也具有尝试的价值。

本文最终得出的模型，在垃圾焚烧的短期预测上具有较大的实践价值，同时为不依赖垃圾成分分析的预测技术提供了一种可行的技术路线和理论思路。

## 6.2 研究展望

由于受到人工费用和物理传感器设备的限制，所以实际工业环境采集数据标签的种类还是比较少的，在添加新的相关传感器以及推广垃圾分类之后，能够为模型带来更多有效的数据标签，这在一定程度上能够提升模型的预测能力。

模型在长期预测上面的能力有所欠缺，这是因为生活垃圾焚烧的决定性因素还是在于垃圾焚烧的种类和投放速率，现阶段由于实际工厂的经济效益问题，往往无法解决这个问题。

但换一种思路来看，如果有条件做垃圾焚烧种类和投放速率的实验，是有可能通过设计模型，利用工况数据反推导出上述的两种数据标签的。

如果能够提供较为理想的工业模拟环境，可以通过人为的进行焚烧实验，通过设计模型来预测垃圾焚烧种类和投放速率，同时将这两个预测所得的数据标签送入本文提出的模型当中，也许可以提高模型的准确率。

本文认为在垃圾历史工况数据当中存在着隐含的时间关系，事实证明的确循环神经网络的预测性能要优于线性模型，但是是否除了隐含的时间关系之外，还有其他的隐含条件在数据集当中，会在较大程度上影响未来的烟气排放数值，这个是未知的，也是下一步进行烟气预测研究的一个可以考虑的方向。

# 参考文献

1. 徐艳萍,黄力华,崔方娜.国内城市生活垃圾焚烧技术应用现状与进展[J].广东化工,2015,42(12):140-141+134.
2. 宋志伟,吕一波,梁洋,杨伟东.国内外城市生活垃圾焚烧技术的发展现状[J].环境卫生工程,2007(01):21-24.
3. Estelle Desroches-Ducarne, J.Christophe Dolignier, Eric Marty, Gérard Martin, Lucien Delfosse,Modelling of gaseous pollutants emissions in circulating fluidized bed combustion of municipal refuse,Fuel,Volume 77, Issue 13,1998,Pages 1399-1410
4. J. Soria, D. Gauthier, Q. Falcoz, G. Flamant, G. Mazza, Local CFD kinetic model of cadmium vaporization during fluid bed incineration of municipal solid waste, Journal of Hazardous Materials, Volumes 248–249, 2013, Pages 276-284
5. Quentin Falcoz, Daniel Gauthier, Stéphane Abanades, Fabrice Patisson, Gilles Flamant, A general kinetic law for heavy metal vaporization during municipal solid waste incineration, Process Safety and Environmental Protection, Volume 88, Issue 2, 2010, Pages 125-130
6. Kum-Lok Hwang, Sang-Min Choi, Moon-Kyung Kim, Jong-Bae Heo, Kyung-Duk Zoh, Emission of greenhouse gases from waste incineration in Korea, Journal of Environmental Management, Volume 196, 2017, Pages 710-718
7. Sangwon Park, Jun-Ho Choi, Jinwon Park, The estimation of N2O emissions from municipal solid waste incineration facilities: The Korea case, Waste Management, Volume 31, Issue 8, 2011, Pages 1765-1771
8. Eliza Harris, Kerstin Zeyer, Rainer Kegel, Beat Müller, Lukas Emmenegger, Joachim Mohn, Nitrous oxide and methane emissions and nitrous oxide isotopic composition from waste incineration in Switzerland, Waste Management, Volume 35, 2015, Pages 135-140
9. 何品晶,陈淼,杨娜,邵立明.我国生活垃圾焚烧发电过程中温室气体排放及影响因素——以上海某城市生活垃圾焚烧发电厂为例[J].中国环境科学,2011,31(03):402-407.
10. 杨仕桥. 生活垃圾焚烧发电厂酸性气体排放控制研究[D].清华大学,2013.
11. 张东平. 城市生活垃圾流化床焚烧过程酸性气体排放及其人工神经网络预测[D].浙江大学,2003.
12. 机器学习 周志华 P.52
13. Bretscher, Otto (1995). Linear Algebra With Applications (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
14. Stigler, Stephen M. (1981). "Gauss and the Invention of Least Squares". Ann. Stat. 9 (3): 465–474. doi:10.1214/aos/1176345451
15. Schölkopf B, Smola A J. Learning with kernels: support vector machines, regularization, optimization, and beyond[M]. MIT press, 2002.
16. Zou H, Hastie T. Regularization and variable selection via the elastic net[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2005, 67(2): 301-320.
17. Gal Y, Ghahramani Z. A theoretically grounded application of dropout in recurrent neural networks[C]//Advances in neural information processing systems. 2016: 1019-1027.
18. Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

作者简历

姓名：周天昱 性别：男 民族：汉

出生年月：1995-09-28 籍贯：浙江省诸暨市

2011.09-2014.06 杭州二中

2014.09-2018.06 浙江大学攻读学士学位

**本科生毕业论文任务书**

**一、题目：垃圾焚烧中的烟气预测研究**

**二、指导教师对毕业论文（设计）的进度安排及任务要求：**

**起讫日期 20 年 月 日 至 20 年 月 日**

**指导教师**（**签名） 职称**

**三、系或研究所审核意见:**

**负责人**（**签名）**

**年 月 日**

**毕 业 论 文（设计） 考 核**

**一、指导教师对毕业论文（设计）的评语：**

**指导教师(签名）**

**年 月 日**

**二、答辩小组对毕业论文（设计）的答辩评语及总评成绩：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **成绩**  **比例** | **文献综述**  **占（10%）** | **开题报告**  **占（15%）** | **外文翻译**  **占（5%）** | **毕业论文（设计）质量及答辩**  **占（70%）** | **总评成绩** |
| **分值** |  |  |  |  |  |

**答辩小组负责人（签名）**

**年 月 日**

第二部分

**文献综述和开题报告**

使用《浙江大学本科生毕业论文（设计）文献综述和开题报告（模板）》