



密级: 公开

# 本科生毕业设计(论文)

题 目: 基于车辆轨迹时空数据的  
城市热点预测模型研究

作 者: 胡成成

学 号: 41724260

学 院: 计算机与通信工程学院

专 业: 通信工程

成 绩: \_\_\_\_\_

2021 年 06 月



# 本科生毕业设计(论文)

题 目: 基于车辆轨迹时空数据的  
\_\_\_\_\_  
城市热点预测模型研究  
\_\_\_\_\_

英文题目: Urban hot spot prediction model based  
\_\_\_\_\_  
on spatiotemporal data of vehicle trajectory  
\_\_\_\_\_

学 院: 计算机与通信工程学院  
\_\_\_\_\_

班 级: 通信 1701  
\_\_\_\_\_

学 生: 胡成成  
\_\_\_\_\_

学 号: 41724260  
\_\_\_\_\_

指导教师: 隆克平 职称: 教授  
\_\_\_\_\_

指导教师: \_\_\_\_\_ 职称: \_\_\_\_\_



## 声 明

本人郑重声明：所呈交的论文是本人在指导教师的指导下进行的研究工作及取得研究结果。论文在引用他人已经发表或撰写的研究成果时，已经作了明确的标识；除此之外，论文中不包括其他人已经发表或撰写的研究成果，均为独立完成。其他同志对本文所做的任何贡献均已在论文中做了明确的说明并表达了谢意。

学生签名：\_\_\_\_\_ 年\_\_月\_\_日

导师签名：\_\_\_\_\_ 年\_\_月\_\_日



# 毕业设计(论文)任务书

一、学生姓名: 胡成成 学号: 41724260

二、题目: 基于车辆轨迹时空数据的城市热点预测模型研究

三、题目来源: 真实  、 自拟

四、结业方式: 设计  、 论文

五、主要内容:

基于城市出租车的时空轨迹数据, 利用核密度估计算法分析和呈现城市不同位置的车辆密度及交通热点。进一步, 分别考虑支持向量回归和不同结构的神经网络模型在时空轨迹数据的基础上预测交通热点随时间的变化, 并对算法性能进行评估, 为智能交通系统和城市车流管理提供数据支撑。课题的目的在于培养学生基本的科研文献查找阅读能力、英文文献翻译理解能力, 以及综合建模设计开发、解决复杂工程问题的能力。

六、主要(技术)要求:

- 1) 基于核密度估计算法评估城市区域的实时车流密度;
- 2) 基于支撑向量机回归, 依据历史车流密度预测未来车流密度;
- 3) 基于不同层级和神经元数量的神经网络依据历史车流密度预测未来车流密度;
- 4) 评估模型在未来不同时间尺度上的预测精度。

七、日程安排:

第1周: 确定毕业设计概要技术思路, 了解车联轨迹大数据格式及机器学习相关技术;

第2-4周: 进一步细化课题方案, 掌握时空数据挖掘、核密度估计的基本方法, 翻译一篇英文文献, 完成选题报告;

第5-8周: 利用核密度方法确定城市区域的车辆分布热点, 并根据热点转换的时空特征进行机器学习, 对上述算法进行Python实现, 准备中期答辩;

第9-12周: 进一步补充后续代码, 改进机器学习算法性能并评估效果;

第13-15周: 课题经验总结, 按照规范撰写毕业论文并准备进行答辩。

八、主要参考文献和书目:

- [1] X. Zhan, Y. Zheng, X. Yi and S. V. Ukkusuri. Citywide Traffic Volume Estimation Using Trajectory Data [J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 29(2):272–285.
- [2] T. Kyaw, N. N. Oo and W. Zaw. Estimating Travel Speed of Yangon Road Network Using GPS Data and Machine Learning Techniques[C]. 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). 2018, Chiang Rai, Thailand.
- [3] L. Huang and L. Xu. Research on Taxi Travel Time Prediction Based on GBDT Machine Learning Method[C]. 2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), Harbin, China, 2018.
- [4] W. Liu, Y. Watanabe and Y. Shoji. Vehicle-Assisted Data Delivery in Smart City: A Deep Learning Approach [J]. IEEE Transactions on Vehicular Technology, 2020, 69(11):13849–13860.
- [5] 羊琰琰. 基于出租车GPS数据的热点区域识别及寻客推荐模型研究[D]. 北京交通大学, 2020.

指导教师签字: 年 月 日

学 生 签 字: 年 月 日

系(所)负责人章: 年 月 日

## 摘要

智能交通在近年得到了学术界和产业界的广泛重视。尤其是随着道路网的不断完善，交通车流越来越庞大，交通流预测显得越来越重要，分析并预测交通状况和交通热点分布情况是交通管控的基础，对城市交通管控有着十分重要的意义。随着车辆轨迹大数据技术、人工智能和机器学习技术的发展，基于机器学习和大数据对车辆密度进行预测已成为重要的技术趋势。

本文基于车辆轨迹大数据，利用机器学习技术对城市交通热点进行预测，主要的研究内容和创新点罗列如下：

首先，建立车流密度提取模型，利用核密度估计算法从车辆轨迹时空数据中提取车辆密度特征，并实现热点预测的可视化。本文从交通属性中车辆密度的角度去分析，相比传统的车流量和车速属性，让交通预测具有更加全局的特征信息，为交通管控增添一个新的维度与视角。

其次，提出预测滑动窗口模型，构建预测所需要的训练数据集，并使用标准的归一化方法进行处理，利用支持向量回归算法进行出租车车辆密度预测和热点预测，最后借助公认的评价指标对模型性能进行评估。为后续神经网络预测工作提供基础性参考。

再次，利用经典的神经网络——多层感知器模型对比不同层数和不同神经元个数的网络结构的性能，并使用循环神经网络中的长短期记忆模型进行预测，完成北京市出租车热点预测并达到预期效果。本文为机器学习应用于交通领域的全局和局部预测提供了新的思路，为该方向的研究提供基础性指标参考。

最后，总结短时预测模式下本文所述模型在不同时间尺度下的预测性能，并提出长时预测的概念，为后续研究提供新的交通预测思路，将交通的短时预测方向扩充到长时预测的场景下。

**关键词： 机器学习，核密度估计，交通热点预测，支持向量回归，多层感知器**



# **Urban hot spot prediction model based on spatiotemporal data of vehicle trajectory**

## **Abstract**

In recent years, intelligent transportation has received extensive attention from academia and industry. Especially with the continuous improvement of the road network, the traffic flow is becoming larger and larger, and the traffic flow prediction is becoming more and more important. The analysis and prediction of traffic conditions and the distribution of traffic hotspots are the basis of traffic control, which is of great significance for urban traffic control. With the development of vehicle trajectory big data technology, artificial intelligence and machine learning technology, vehicle density prediction based on machine learning and big data has become an important technical trend.

In this thesis, machine learning technology is used to predict urban traffic hot spots based on vehicle track big data. The main research contents and innovation points are listed as follows.

Firstly, the vehicle flow density extraction model is established, and the kernel density estimation algorithm is used to extract the vehicle density features from the spatiotemporal data of vehicle tracks, and the visualization of hot spot prediction is realized. In this thesis, from the perspective of vehicle density in traffic attributes, compared with the traditional vehicle flow and speed attributes, the traffic prediction has more global characteristic information, adding a new dimension and perspective to traffic control.

Secondly, the prediction sliding window model was proposed to construct the training data set required for the prediction, and the standard normalization method was used for processing. The support vector machine regression algorithm was used for the taxi vehicle density prediction and hot spot prediction. Finally, the performance of the model was evaluated by the recognized evaluation indexes. It provides the basic flow for the subsequent neural network prediction.

Thirdly, the classical neural network called multi-layer perceptron model was used to compare the performance of network structures with different layers and neurons, and the long and short term memory model in the cyclic neural network was used to predict the hot spots of taxis in Beijing, and the expected results were achieved. This thesis provides a new idea for the application of machine learning to global and local prediction in the field of transportation, and provides a basic index

reference for the research in this direction.

Finally, the prediction performance of the model described in this thesis under the short-time prediction mode is summarized under different time scales, and the concept of long-time prediction is proposed to provide a new forecasting idea for the follow-up research, and the direction of short-time traffic prediction is extended to the scenario of long-time prediction.

**Key Words:** Machine Learning, Kernel Density Estimation, Traffic Hotspot Prediction, Support Vector Regression, Multi-layer Perceptron

# 目 录

摘要	I
Abstract	III
插图清单	IX
附表清单	XI
1 引 言	1
1.1 研究背景及意义	1
1.2 研究内容与方法	2
1.3 本文组织内容	2
2 文献综述	4
2.1 交通预测概述	4
2.2 核密度估计概述	5
2.3 交通预测中的机器学习方法	5
2.3.1 传统交通预测方法	5
2.3.2 基于机器学习的预测方法	6
3 基于核密度估计算法的城市车流密度提取模型	8
3.1 车辆轨迹时空数据概述	8
3.1.1 GPS 数据描述	8
3.1.2 Open Street Map	10
3.1.3 研究区域概述	10
3.2 车辆轨迹时空数据预处理	11
3.3 基于时空轨迹数据的核密度估计算法	12
3.3.1 核密度估计算法	12
3.3.2 核密度估计最适带宽计算	13
3.4 北京市出租车热点信息提取	14
3.4.1 实验环境	14
3.4.2 时空数据密度挖掘	15
3.4.3 模型参数设置	17
3.4.4 实验结果与分析	17
3.5 本章小结	22
4 基于支持向量回归的热点预测模型	23
4.1 基于滑动窗口的热点信息数据处理	23
4.1.1 滑动窗口模型	23

4.1.2 训练数据集构造	24
4.1.3 数据的归一化处理	26
4.2 支持向量回归预测	27
4.2.1 支持向量回归	27
4.2.2 支持向量回归参数	30
4.3 支持向量回归热点预测	31
4.3.1 实验环境	31
4.3.2 模型参数设置	31
4.3.3 预测评价标准	32
4.3.4 实验结果与分析	33
4.4 本章小结	39
5 基于神经网络的热点预测模型	40
5.1 神经网络概述	40
5.1.1 神经网络构成	40
5.1.2 损失函数与正则化	42
5.1.3 前向传播与反向传播	43
5.1.4 激活函数	44
5.2 优化方法	46
5.3 循环神经网络	47
5.4 MLP 热点预测	50
5.4.1 实验环境	50
5.4.2 模型参数设置及评价标准	50
5.4.3 实验结果与分析	51
5.5 LSTM 热点预测	56
5.5.1 实验环境	56
5.5.2 参数设置	56
5.5.3 实验结果与分析	57
5.6 本章小结	60
6 不同时间尺度下模型性能分析	61
6.1 不同时间尺度预测模型	61
6.1.1 短时预测模型	61
6.1.2 长时预测模型	61
6.2 不同时间尺度预测模型性能对比	62
6.2.1 实验参数设置	62

6.2.2 模型对比与分析 .....	62
6.3 本章小结 .....	63
7 结论与展望 .....	64
参考文献 .....	67
在学取得成果 .....	73
致 谢 .....	75



## 插图清单

图 1-1 本文研究内容概览 .....	2
图 3-1 城市车流密度提取流程 .....	8
图 3-2 研究区域提取结果 .....	11
图 3-3 数据预处理流程 .....	11
图 3-4 核密度估计示意图 .....	13
图 3-5 时间抽样过程示意图 .....	15
图 3-6 空间采样过程示意图 .....	16
图 3-7 车辆密度数据集成示意图 .....	16
图 3-8 不同带宽下核密度估计热力分布图 .....	18
图 3-9 周一早 8 点 (左) 和晚 8 点 (右) 出租车密度分布图 .....	19
图 3-10 周二早 8 点 (左) 和晚 8 点 (右) 出租车密度分布图 .....	19
图 3-11 周三早 8 点 (左) 和晚 8 点 (右) 出租车密度分布图 .....	19
图 3-12 周四早 8 点 (左) 和晚 8 点 (右) 出租车密度分布图 .....	20
图 3-13 周五早 8 点 (左) 和晚 8 点 (右) 出租车密度分布图 .....	20
图 3-14 周六早 8 点 (左) 和晚 8 点 (右) 出租车密度分布图 .....	20
图 3-15 周日早 8 点 (左) 和晚 8 点 (右) 出租车密度分布图 .....	21
图 4-1 基于支持向量回归的热点预测流程 .....	23
图 4-2 预测滑动窗口模型示意图 .....	25
图 4-3 训练数据集构造流程 .....	26
图 4-4 支持向量机基本思想示意图 .....	28
图 4-5 SVR 容忍度示意图 .....	28
图 4-6 不同预测窗口步长下晚 20 点预测对比图 .....	34
图 4-7 不同预测窗口步长下晚 22 点预测对比图 .....	35
图 4-8 不同预测窗口步长下晚 20 点预测误差分布图 .....	35
图 4-9 SVR 预测天安门附近拟合与预测对比 .....	36
图 4-10 SVR 预测北京科技大学附近拟合与预测对比 .....	36
图 4-11 SVR 预测首都机场附近拟合与预测对比 .....	37
图 4-12 SVR 晚 20 点全局预测结果与真实情况对比 .....	37
图 4-13 SVR 晚 21 点全局预测结果与真实情况对比 .....	38
图 4-14 SVR 晚 22 点全局预测结果与真实情况对比 .....	38

图 4-15 SVR 晚 20 点全局预测误差分布情况 .....	38
图 5-1 基于神经网络的热点预测流程 .....	40
图 5-2 三层基本神经网络结构示意图 .....	41
图 5-3 sigmoid 函数分布 .....	44
图 5-4 tanh 函数分布 .....	45
图 5-5 relu 函数分布 .....	45
图 5-6 循环神经网络序列化示意图 .....	48
图 5-7 RNN 网络内部结构 .....	48
图 5-8 RNN 反向传播过程 .....	49
图 5-9 LSTM 前向传播过程 .....	49
图 5-10 MLP 预测天安门附近拟合与预测对比 .....	53
图 5-11 MLP 预测北京科技大学附近拟合与预测对比 .....	54
图 5-12 MLP 预测首都机场附近拟合与预测对比 .....	54
图 5-13 MLP 晚 20 点全局预测结果与真实情况对比 .....	55
图 5-14 MLP 晚 21 点全局预测结果与真实情况对比 .....	55
图 5-15 MLP 晚 22 点全局预测结果与真实情况对比 .....	55
图 5-16 MLP 晚 20 点全局预测误差分布情况 .....	56
图 5-17 LSTM 预测网络结果概念图 .....	57
图 5-18 LSTM 预测天安门附近拟合与预测对比 .....	58
图 5-19 LSTM 预测北京科技大学附近拟合与预测对比 .....	58
图 5-20 LSTM 预测首都机场附近拟合与预测对比 .....	58
图 5-21 LSTM 晚 20 点全局预测结果与真实情况对比 .....	59
图 5-22 LSTM 晚 21 点全局预测结果与真实情况对比 .....	59
图 5-23 LSTM 晚 22 点全局预测结果与真实情况对比 .....	59
图 5-24 LSTM 晚 20 点全局预测误差分布情况 .....	60
图 6-1 长时预测和短时预测的对比 .....	62
图 6-2 短时预测 SVR, MLP 与 LSTM 不同尺度预测对比 .....	63

## 附表清单

表格 3-1 原始数据属性字段说明 .....	9
表格 3-2 原始数据样例 .....	9
表格 4-1 SVR 模型参数设置.....	31
表格 4-2 SVR 不同步长下预测下评价指标均值对比.....	33
表格 5-1 MLP 预测参数符号与赋值 .....	50
表格 5-2 MLP 不同窗口步长下评价指标均值对比 .....	51
表格 5-3 MLP 单隐藏层不同神经元个数性能对比（取前 8 项） .....	52
表格 5-4 MLP 双隐藏层不同神经元个数性能对比（取前 10 项） .....	53
表格 5-5 LSTM 模型指标参数.....	57
表格 6-1 不同时间尺度下不同模型参数设置 .....	62



# 1 引 言

## 1.1 研究背景及意义

随着道路网系统的不断建设与完善，人们生活水平的不断提升，交通车流辆也越来越大，对交通的管理与控制显得越来越重要。智能化的城市交通管理是现代化进程中不可缺少的，这就需要利用交通数据并进行处理，来预测未来的交通道路状况，分析城市交通热点的分布及变化，有利于人们出行的决策，判断城市交通拥塞与车辆密集地，避免进入交通密集地等待，影响正常的出行规划。相应地，交通机构也需要利用实时交通量信息来对交通进行干预，例如改变交通信号灯的时间或者关闭某些道路，以便在严重拥堵或紧急事件下做出反应。而影响交通状况的三个重要指标包括车辆的速度、车流量和车辆密度，三者的关系也称作 FD(Fundamental Diagram)<sup>[1]</sup>。其中车辆速度和车流量是目前广大研究者所关注的重点，在交通拥塞预测中，研究者们喜欢从行驶速度预测(TSE, Travel Speed Estimation)与交通流量预测(TVE, Traffic Volume Estimation)两个模型对交通状况进行预测<sup>[1][2]</sup>。这也源于目前的交通数据普遍来源于 GPS (Global Positioning System) 交通数据，而 GPS 交通数据包含了车辆的行驶速度、位置和时间等可以直接利用的数据信息<sup>[3]</sup>。

出租车交通系统作为城市交通系统的一部分，与公交车交通系统相比，没有很强的规律性，但相比于私家车交通系统，也没有很强的随机性。因此，研究出租车交通系统，一定程度上能反映出人们生活出行活动的规律，同时也能一定程度反映出城市出租车接客密集的热点区域。研究出租车出行密度分布与热点分布，相比于传统的利用车辆速度和街道车流量属性去研究，交通的全局性更加明显，有利于对出租车进行整体的调度与管控，同时也能研究局部地区出租车密度与分布在时间序列上的变化，获取出租车整体分布信息和局部动态信息。同时，前人利用城市出租车数据可以得到城市 OD(Origin-Destination) 热点<sup>[4]</sup>，便于出租车司机得知乘客上车地点分布，减少车辆空驶时间比例，从而减少不必要的资源浪费。同时，在交通预测的通信开销上也有研究机器学习方法来“绿色化”城市<sup>[5]</sup>。

城市热点预测是交通领域重要的研究方向之一。城市的热点区域间接地反映了城市居民活动较频繁的地区，挖掘这些热点区域一定程度上有利于为城市道路规划和城市规划管理提供依据。同时也有利于交通参与者提前了解，避开交通热点通行，减少不必要的时间开销，提高人们的出行效率。让城市

交通网络更加智能地判断拥塞地点，及时做出道路管理措施，为城市现代化建设提供“绿色”保障。本文利用北京市出租车数据进行处理，达到预测城市出租车热点分布的目的，为城市出租车调度与管控提供可靠的依据，提高人们的出行效率并减少资源的损耗。

## 1.2 研究内容与方法

本课题致力于利用城市车辆时空轨迹数据进行交通热点预测，研究的对象是北京市范围内整体出租车的分布。主要从两个方面进行展开：

(1) 针对北京市出租车原始数据，进行预处理，利用核密度估计算法估计城市出租车车流密度并进行可视化，并利用滑动窗口模型对训练数据集进行构造。

(2) 通过构造的训练数据集，利用基于支持向量回归算法进行预测，依据历史车流密度预测未来车流密度，并通过构建不同结构的神经网络进行预测，最后评估分析模型的性能。

具体研究内容如图 1-1 所示。

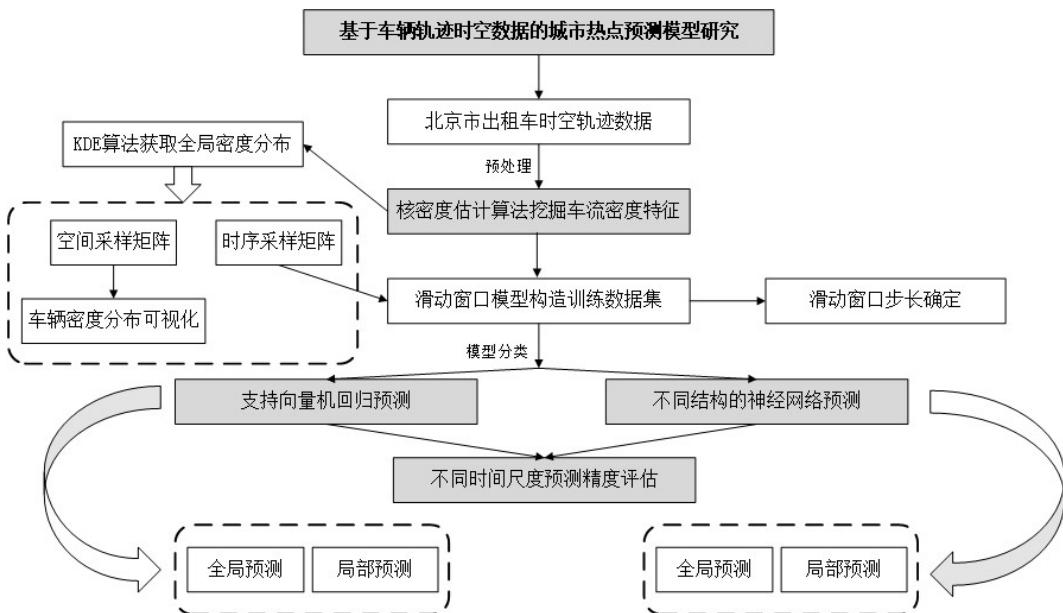


图 1-1 本文研究内容概览

## 1.3 本文组织内容

本文内容层次上分为以下几个模块。

第一章，引言。概括了本文研究北京市出租车数据的方法与内容，介绍本次研究的背景与意义。

第二章，文献综述。介绍了交通预测通常使用的参数与数据，目前常见的方法与模型，以及前人在该领域所作的成就及进展。

第三章，基于核密度估计算法的城市车流密度提取模型的研究。本章分别介绍了车辆轨迹时空数据和研究区域，车流数据的预处理方法，并对关键算法——核密度估计算法进行实现与应用，并将北京市出租车密度数据进行可视化。

第四章，基于支持向量回归的出租车城市热点预测模型研究。本章分别介绍了滑动窗口模型并借助该模型进行训练数据集的构造，对支持向量回归算法和数据归一化处理进行介绍，设置不同步长的滑动窗口进行预测，评估模型的性能。

第五章，基于不同结构的神经网络的热点预测模型研究。本章分别介绍了神经网络的基本结构和原理，以及介绍了适用于时间序列预测的循环神经网络模型，通过设置不同结构的经典神经网络模型进行预测评估，并在此基础上利用较先进的长短期记忆模型进行预测，评估模型的性能。

第六章，对上述所用最优模型，从短时预测的角度在不同时间尺度下的预测进行研究，评估各个模型的优劣，并提出长时预测模型的基本概念。

第七章，结论与展望。总结了本文在北京市出租车数据集上进行处理和预测的工作，并对未来的工作进行展望。

## 2 文献综述

### 2.1 交通预测概述

城市交通预测是交通管理领域的一个重要研究方向，现代城市都在追求智慧化，城市化和人口的增长给城市的交通带来更大的压力与挑战。因此，智能交通系统的需求越来越大，准确的交通预测成为实际交通管控必不可少的部分。例如，交通量的预测可以缓解城市交通拥堵问题，出租车需求预测可以帮助出租车运营商及时将出租车分配到需求高的地方。

研究交通预测，首先需要明确交通预测的对象是什么，主要的预测任务是什么，从而明晰交通预测的属性。随着交通的发展必然也会涌入新的属性数据作为评价交通状况的指标，而目前的交通预测指标可为我们预测提供基础的属性值，交通预测的属性依据城市出行车辆的时空特性所产生，根据车辆本身具有的位置信息、速度信息等以及车辆集群所具有的流量、密度等信息可以延伸出交通预测的几个重要属性。从目前的主要研究工作来看，交通预测的主要任务包括以下几个方面<sup>[6]</sup>：

(1) 交通流量预测(Flow)，即预测某一段时间内通过道路上某一位置的车辆数量信息。

(2) 速度预测(Speed)，即预测某一段时间内道路上的车辆平均车速。

(3) 需求量预测(Demand)，即使用历史数据来预测某一个区域在未来某一段时间中的需求量，其中交通需求通常包括出租车和共享单车的需求。

(4) 占用率预测(Occupancy)，即预测某一段时间内车辆占用道路空间的程度。一般在测量时，需要考虑交通的组成和速度的变化，并提供更可靠的车辆占用道路的程度指标。

(5) 旅行时间预测(Travel time)，即在获取路网中任意两点的路线的情况下，预测从路线中的一个点到另一个点的旅行时间。

广大研究者针对交通预测的这几个属性指标展开自己的研究，同时，也有很多组织和个人贡献自己所搜集的交通数据集，常见的数据集包括：

(1) PeMS，即加州交通局性能测量系统，通过地图显示，由 39000 多个探测器实时采集。覆盖了美国加利福尼亚州所有主要都市区的高速公路系统。

(2) TaxiBJ，提供了北京市出租车 GPS 轨迹数据和对应时间段的天气数据信息，在 2013 年到 2016 年中四个时间段进行采集。

(3) SZ-taxi，提供了深圳市出租车 2015 年 1 月份的 GPS 轨迹数据。研

究区域包括 156 条主干道。每条道路采样频率每 15 分钟一次。

(4) NYC Taxi，提供的轨迹数据是纽约市 2009 年至 2018 年的出租车 GPS 数据。

(5) T-Drive，提供了北京出租车从 2015 年 2 月份到 2015 年 6 月份的大量轨迹。这些轨迹可用于计算每个区域的交通流。

(6) DiDi chuxing，滴滴数据开放计划提供真实和免费的脱敏数据。主要包括出行时间指数、多个城市的出行和轨道数据集。

这里列举出常见的交通数据集，还有很多开源数据集可供研究者去研究挖掘。交通预测的条件不断完善，使得研究者们可以从更多不同的角度和方法去实现自己的预测方案。

## 2.2 核密度估计概述

核密度估计算法作为一种非参数估计算法，常用于预估位置的概率密度分布，最初由 Rosenblatt 和 Emanuel Parzen 提出<sup>[6]</sup>，常用于地理空间分析领域，通过二维离散点生成三维连续的光滑曲面。相比于其他的空间分析方法，核密度估计算法的参数少，不易受人的主观因素影响，因此核密度估计算法的应用十分广泛。例如：地物空间及区域格局分析<sup>[7] [8]</sup>、疫情分析与地质灾害监测<sup>[10] [11]</sup>、路径分析优化<sup>[12]</sup>、遥感影像分析<sup>[13]</sup>、POI（Point of Interest）兴趣点分布分析<sup>[14]</sup>、点群制图分析<sup>[15]</sup> 等诸多领域。核密度估计算法在各大领域都展现了其优越的估计性能，在交通预测领域的应用甚少，有待研究人员的挖掘与使用。

## 2.3 交通预测中的机器学习方法

在智能交通系统（ITS, Intelligent Traffic System）中，借助历史交通状况特征准确预测未来短时的交通状况信息，对城市交通规划管控都很重要。短时交通预测<sup>[16]</sup> 是预测某一地区未来几分钟或几小时的交通流信息的变化（例如：速度、车流量等），常见的预测方法包括传统预测方法和基于机器学习和深度学习的方法。

### 2.3.1 传统交通预测方法

根据交通数据的特性和属性来看，用于交通预测的数据具有时序特征，符合时序预测的特点，传统的时序预测方法包括：

(1) ARIMA 模型(Autoregressive Integrated Moving Average model)<sup>[17]</sup>：该模型是时间序列预测分析经典的方法，主要思路是将预测随时间的变化序列当作一个随机过程，经过多次差分使不平稳的序列转化成平稳序列，然后建模成近似稳定的序列。短期预测效果不错，且模型的结构简单的特性，但具有只能应用于线性关系的局限性，不适应于交通流急剧变化不稳定的状况。针对该模型，文献[18] 将 ARIMA 与人工神经网络相结合，提出一种预测序列的新方法。后续还有很多研究人员对 ARIMA 模型进行不同方面的改进<sup>[19]</sup><sup>[20]</sup>。

(2) HA(History Average Model): 文献[21] 提出应用于城市交通控制系统的 HA 模型，算法具有简单、其参数可用最小二乘法估计的特点，可解决不同时间和时段中的交通状态信息变化问题，但模型具有静态特性的局限性，存在不纳入现输入状态的影响的情况，不能反映动态交通状态信息的不确定性与非线性特性，不能应对一些突发事件的发生。

(3) VAR 模型(Vector Autoregressive model): 文献[22] 提出 VAR 模型，使得预测的过程将单变量扩充到两个及以上，多用在多变量时间序列的预测，减少了预测中的不确定性，同时也能很好地反映交通状况的波动情况。该模型在预测精度上有着不错的效果，但是模型的参数比较多。

### 2.3.2 基于机器学习的预测方法

随着智能化的发展，机器学习和深度学习的方法越来越受关注，交通领域使用机器学习和深度学习的方法被发现有更加优越的性能，越来越多的模型被发现<sup>[23]</sup><sup>[24]</sup>。常用的方法包括：

(1) K 近邻 (KNN, K-Nearest Neighbors) 算法<sup>[25]</sup>：KNN 算法是非参数回归中最经典的算法之一。需要建立历史信息数据库，使得其具有足够大的容量，通过对历史数据库筛选识别并清洗，按照设定的相关要素从其中找到与当前预测数据最近最匹配的结果，从而预测下一时刻的交通量。文献[26] 提出了一种基于 KNN 的道路交通状态预测方法，验证了其可行性，且达到很高的预测精度。文献[27] 进一步优化了 KNN 的性能。

(2) 支持向量回归 (SVR, Support Vector Regression) 算法<sup>[28]</sup>：SVR 通过选取核函数，对支持向量回归机进行训练，随后向模型输入交通属性特征，预测下一时段的交通特征信息。SVR 在预测交通流量方面拥有优越的性能<sup>[29]</sup><sup>[30]</sup>，利用历史车流量预测未来车流具有很高的精度。

(3) 多层感知器 (MLP, Multilayer Perceptron) <sup>[31]</sup>：MLP 是最简单的

神经网络，从它的结构可以清晰地看到神经网络计算的过程，是最经典的神经网络。利用 MLP 预测城市车流量相比传统的非机器学习方法，有着更好的性能<sup>[32][33]</sup>。

(4) 卷积神经网络 (CNN, Convolutional Neural Networks)<sup>[34]</sup>：CNN 是多层感知机 (MLP) 的变种，通过建立一些局部的链接，共享一些参数，达到减少了网络权重参数的数目，优化网络性能，还降低了过拟合出现的目的。CNN 常用于计算机视觉训练计算，而在交通预测领域处理的序列数据一般都采用变种的 CNN。文献[35] 通过将序列中预测数据可用的信息特征构建成类似于图像的矩阵表示，再利用卷积神经网络进行预测，产生不错的效果。文献[36] 直接利用 PeMS 平台提供的全局道路拥塞等级图进行处理，并使预测输出结果也为交通拥塞等级图。

(5) 循环神经网络 (RNN, Recurrent Neural Network)<sup>[37]</sup>：RNN 结构的神经元间也建立了权重连接关系，一般处理时间序列数据。文献[38] 对比了 RNN 模型和传统预测模型，发现 RNN 在时序预测上非常不错的性能。文献[39] 将 RNN 预测城市交通乘客出行流量，实验得到不错的预测精度。

(6) 长短期记忆模型 (LSTM, Long Short-Term Memory)<sup>[40]</sup>：为了克服 RNN 模型的一些缺点，LSTM 模型能够学习更长的依赖关系。在交通预测中，LSTM 是现代很受欢迎的神经网络，在车流量预测和交通拥塞预测方面也表现出不错的效果<sup>[41][42][43]</sup>。文献[44] 将在 LSTM 模型中引入了注意力机制 (Attention Model)，使得预测效果进一步提升。

### 3 基于核密度估计算法的城市车流密度提取模型

本章主要研究基于核密度估计算法来提取城市车流密度数据，通过对北京市 28590 辆出租车一周内 GPS 轨迹定位数据进行挖掘，设置等间隔的时间序列，提取时间序列上采样时间点所对应的位置信息的核密度估计密度值，将车流密度数据集成便于后期进行预测分析。

基于核密度估计算法的城市车流密度提取模型的基本流程如图 3-1 所示。

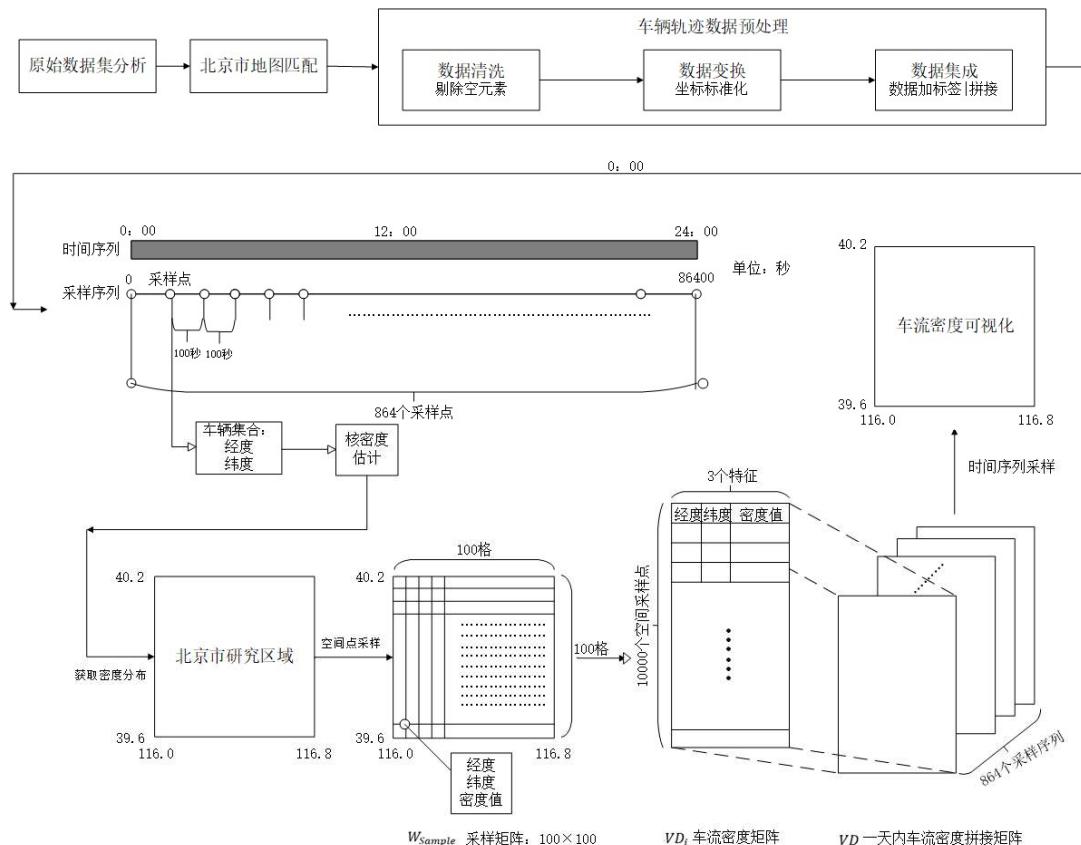


图 3-1 城市车流密度提取流程

#### 3.1 车辆轨迹时空数据概述

##### 3.1.1 GPS 数据描述

全球定位系统(GPS, Global Positioning System)是利用人造地球卫星技术与无线通讯技术为基础的定位系统，具有定位精度高等性质，在全球范围内的近地空间能够提供准确严密的地理位置信息、车辆出行速度以及定位时间信息。GPS 车辆轨迹数据则是通过车辆装载 GPS 定位设备来采集车辆的轨迹数据。而 GPS 数据往往包含多个字段，承载车辆多种不同角度的时空信息，其中从 GPS 推荐定位信息\$GPRMC 中提取车辆轨迹研究有价值的信息<sup>[2]</sup>，

可定义为：

$$veh_i = \{id, t_i, stat_i, lng_i, lat_i, v_i, dir_i\}$$

其中， $veh_i$ 是定位车辆第*i*条轨迹点数据； $id$ 是定位车辆的标识； $t_i$ 是定位点的世界标准(UTC, Coordinated Universal Time)时间； $stat_i$ 是定位状态，包括定位有效和定位无效两种状态； $lng_i, lat_i$ 分别表示定位时刻的经度和纬度； $v_i$ 是车辆定位时刻的行驶速度，单位公里每小时； $dir_i$ 是车辆定位时刻的车头朝向，初始角度以正北方向为0°基准，按顺时针方向递增。

本文研究以出租车GPS轨迹数据为依据，一般地，出租车轨迹时空数据相比于车辆轨迹时空数据会多出一个字段 $car_i$ ，即载客状态，有满载和空载两个状态，可表示为：

$$taxi_i = \{id, t_i, stat_i, lng_i, lat_i, v_i, dir_i, car_i\}$$

**表格 3-1 原始数据属性字段说明**

字段名	含义	格式
Longitude	定位点经度	度度秒秒.秒秒(ddmm.mm)
Latitude	定位点纬度	度度秒秒.秒秒(ddmm.mm)
Time	定位点定位时间	秒(s)

由于车联大数据通常具有数据量体量大、数据价值密度很低的特点，本文采用简化的出租车轨迹时空数据进行研究分析，数据字段只需要用于热点分析的经纬度字段和定位的时间字段。原始数据是北京市一周内的出租车定位数据和时间数据，存储在MAT文件中，每辆出租车的定位数据和时间数据分别在两个不同的MAT文件的元胞(Cell)数组中，但是数据都是一一对应的。具体字段说明如表3-1所示。

**表格 3-2 原始数据样例**

实例编号	经度原始值	纬度原始值	原始定位时间
1	11610997.	3985274.	0.000e+00
2	11610997.	3985274.	1.000e+01
3	11610997.	3985274.	2.000e+01
.....	.....	.....	.....
7551	11610943.	3985273.	7.550e+04
7552	11610943.	3985273.	7.551e+04
7553	11610943.	3985273.	7.552e+04

如表 3-2 是读取的部分原始数据，包括经纬度，时间数据。第一列为实例编号，第二列为经度数据，第三列为纬度数据，最后一列是时间数据，单位为秒，以每十秒为间隔采样一次。

### 3.1.2 Open Street Map

OpenStreetMap（简称 OSM）是一个开源的地图协作计划，旨在创造一个内容自由并且能让任何人编辑的世界开源地图。它是可以和网络上的任何人来一起编辑的开源地图服务。OpenStreetMap 借助公众的付出和没有报酬的贡献来完善地图相应的地理数据，包含了近乎完备的地理数据，为地理空间与规划相关领域感兴趣的人提供了更方便的基础工具。因此，OSM 是非营利性的，它将数据反馈给社区重新为其他的产品提供相应服务<sup>[45]</sup>。

OSM 的地图由广大使用者去制作与绘制，可以使用 GPS 装置、航空拍摄等方式记录。OSM 采用的是地心坐标系 WGS84，从 OSM 通过爬虫技术获取的地图数据能够作为热点预测的背景，为分析热点分布位置提供更好的视觉效果。

### 3.1.3 研究区域概述

北京市是我国的首都，同时也是我国各方面发展领航的首要城市之一，北京市地处东经  $115.7^{\circ}$  至  $117.4^{\circ}$ ，北纬  $39.4^{\circ}$  至  $41.6^{\circ}$ ，截至 2020 年，北京市分为东城区、西城区、朝阳区等 16 个市辖区。现居人口聚集，交通环境压力也相对较大，研究北京市的交通热点对于北京市交通规划和人们出行策略有着重要意义。由于北京市外围郊区存在大面积的道路低密度区域，研究这些区域的出租车轨迹数据意义并不大，因此我们将研究区域适当缩小为东经  $116.0^{\circ}$  至  $116.8^{\circ}$ ，北纬  $39.6^{\circ}$  至  $40.2^{\circ}$ ，如图 3-2 所示，我们将研究区域集中倾向于道路密集的城区。本文研究对象是北京市范围内出租车轨迹时空数据，研究的区域即北京市所框选的地理位置矩形区域。

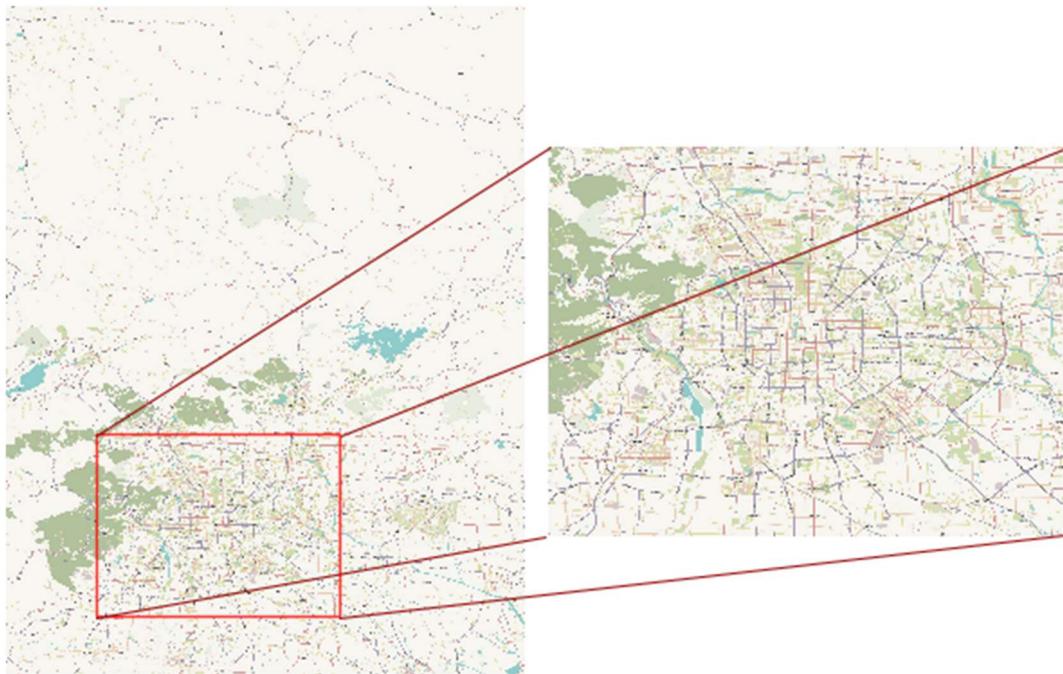


图 3-2 研究区域提取结果

北京市出租车一般通过添加 GPS 设备，连接信息处理平台，对 GPS 定位轨迹数据处理分析，挖掘利用有价值的信息，从而更好地运营管理出租车。本文研究北京市 28590 辆出租车的轨迹数据，分析预测出租车热点分布，从而为出租车运营管理提供一定的参考价值。

### 3.2 车辆轨迹时空数据预处理

北京市出租车原始数据由多个 MAT 文件分别存储，分为出租车的位置和时间数据信息，不利于处理分析。且数据由于采集过程中遇到各种不可控因素，造成 GPS 记录的经纬度数据缺失，部分车辆采集的数据完全为空，有些车辆采集的时间点数据存在缺失。因此需要把位置和时间数据进行整合，同时对每辆出租车进行标识，添加标识号。数据预处理工作是数据信息挖掘的基础，为后续车辆轨迹数据挖掘提供合理依据。数据预处理的流程如图 3-3 所示。

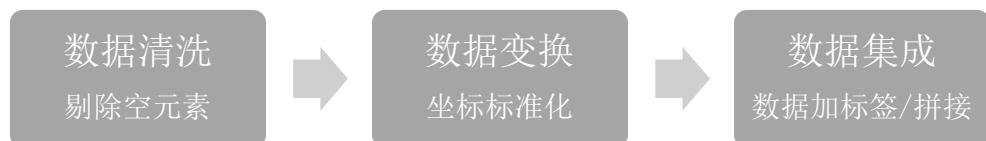


图 3-3 数据预处理流程

(1) 数据清洗：清洗过程需要遍历原始数据并逐条读取，对于采集的数

据完全为空的车辆数据进行剔除。

(2) 数据变换: 原始的坐标信息采用的标准为地心坐标系 WGS84 标准, 获取的 GPS 经纬度数据是实际的经纬度数据的十万倍, 需要将所有获取的经纬度数据进行变换处理, 将数据对 10 的五次方做商运算, 方便后续在开源的 Open Street Map 上进行数据投影工作。

(3) 数据集成: 对一单车辆, 获取车辆位置信息和时间信息, 并在这基础上添加车辆标识信息, 实现单车数据的集成, 经集成后的出租车数据形成一个维度为  $T_i \times 4$  的信息矩阵, 其中  $T_i$  代表总行数, 表示信息采集的时间点, 4 表示数据的列数, 分别对应采样时间点上的车辆标识号、时间点、经度和纬度。并将所有单一处理的车辆数据信息进行拼接合并, 使得数据矩阵的行数变为  $\sum_N T_i$ , 其中 N 表示数据有效的车辆数目, 最终获得处理后的  $\sum_N T_i \times 4$  的车辆轨迹预处理数据。

### 3.3 基于时空轨迹数据的核密度估计算法

#### 3.3.1 核密度估计算法

对于给定的一个样本集, 获取该样本集的分布密度函数一般有以下两种方法:

(1) 参数估计方法: 假设所给的样本集合满足某种概率分布, 根据样本数据拟合出分布中所需要的参数, 例如: 似然估计法, 高斯混合法等, 但该方法通常需要一些先验知识, 很难拟合出真实的分布。

(2) 非参数估计法: 该方法不需要引入先验知识, 利用数据本身具有的特性, 拟合出数据的分布模型, 相比参数估计法有更不错的效果, 例如: 核密度估计法。

研究对象在空间上的出现, 带有很强的任意性, 但是某种空间作用会破坏这种任意性, 导致研究对象点在以不同的概率出现在任意位置。若某一区域出现的对象点很多, 表明存在一种作用关系使得该区域内的对象点出现更有可能, 反之则不太可能。基于此理论基础, 利用核密度估计方法就能很容易的找到出租车热点的分布情况。

获取出租车的空间分布, 一般需要利用热力图来进行可视化, 热力图可视化的原理包括为样方法和核密度估计法 (KDE, Kernel Density Estimation)。样方法是研究空间点模式中最基础且简单的方法, 通过将空间分割成若干个小栅格, 统计栅格内样本数目, 从而进行密度估计进行可视化。核密度估计

法则通过离散的数据点来构建平滑密度表面，从而实现将离散数据模型转化到连续数据模型。如图 3-4 是核密度估计的示意图。

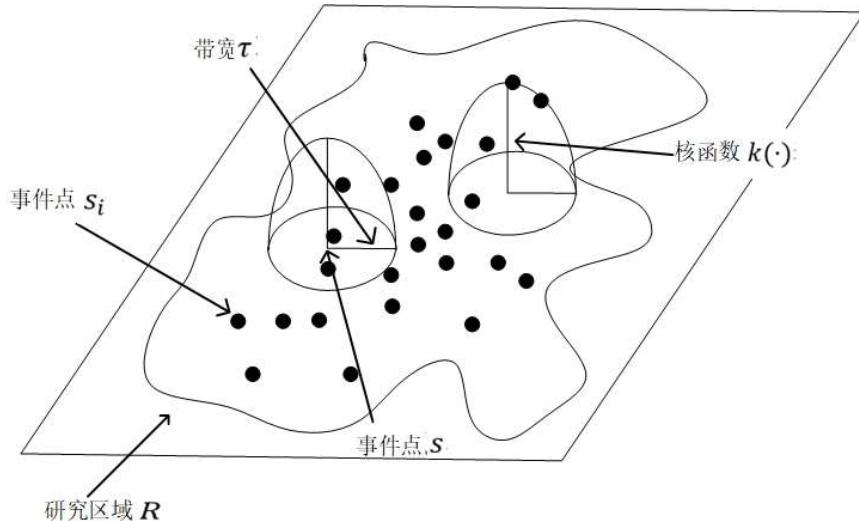


图 3-4 核密度估计示意图

核密度估计法可描述为<sup>[14] [15]</sup>：设在研究区域  $R$  内分布有  $n$  个事件  $s = |s_1, s_2, \dots, s_n|$ ,  $s$  处的点密度为  $\rho(s)$ , 密度估计值为  $\rho'(s)$  表示为：

$$\rho'(s) = \sum_{i=1}^n \frac{1}{\tau^2} k\left(\frac{s-s_i}{\tau}\right) \quad (3-1)$$

其中,  $k(\cdot)$  代表核密度估计的核函数,  $\tau > 0$ , 代表核密度估计带宽, 根据示意图所示, 带宽可表示为以  $s$  为中心的参与计算的作用范围, 不同的带宽会影响到结果的视觉光滑程度; 公式中  $s - s_i$  代表点  $s$  和点  $s_i$  之间的距离。

一般采用不同的核函数  $k(\cdot)$ , 得到的结果差异也比较大。常见的核函数包括高斯, 线性, 余弦等函数。本文实验拟用高斯核进行建模分析, 具体表示为:

$$\rho'(s) = \rho'(x, y) = \frac{1}{n\tau^2} \sum_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma^2\tau^2}} \quad (3-2)$$

其中,  $\sigma$  表示高斯核的标准差。

### 3.3.2 核密度估计最适带宽计算

带宽  $\tau$  的确定影响核密度估计算法的效果。前人的经验表明, 带宽  $\tau$  的值调整的大一些, 可以得到更加光滑的估计结果, 但容易忽略掉一些局部密度信息; 而将  $\tau$  的值设的太小, 会得到表面不够光滑且不平坦的核密度曲面, 即局部的特征得到凸显放大, 一定程度弱化了稍大尺度上的全局特征。在机器学习理论中, 我们可以使用交叉验证方法择优带宽  $\tau$ 。对  $\tau$  参数的选择使用平均积分平方误差 (MISE, Mean Integrated Squared Error) 来计算:

$$MISE(\tau) = E[\int (\rho'(s) - \rho(s))^2 ds] \quad (3-3)$$

在弱假设情况下,  $MISE(\tau) = AMISE(\tau) + o(\frac{1}{n\tau} + \tau^4)$ , 其中 $AMISE$ 为渐进的 $MISE$ , 其中 $AMISE$ 有:

$$AMISE(\tau) = \frac{R(k)}{n\tau} + \frac{1}{4}m_2(k)^2\tau^4R(\rho'') \quad (3-4)$$

其中:

$$R(k) = \int k(s)^2 ds, m_2(k) = \int s^2 k(s) ds \quad (3-5)$$

那么, 使 $MISE(\tau)$ 最小, 则转化为求极点问题:

$$\frac{\partial AMISE(\tau)}{\partial \tau} = -\frac{R(k)}{n\tau^2} + m_2(k)^2\tau^3R(\rho'') = 0 \quad (3-6)$$

$$\tau_{AMISE} = \frac{R(k)^{0.2}}{m_2(k)^{0.4}R(\rho'')^{0.2}n^{0.2}} \quad (3-7)$$

除此之外, 常用的三种最适带宽的计算方法如下:

- (1) 先计算整个样本点的平均中心, 根据每个样本点到该中心的距离, 然后取距离的中位数 $D_m$ , 并计算它们距离的标准差 $SD$ ,  $n$ 为事件点个数, 则 $\tau$ 值满足公式<sup>[46]</sup>:

$$\tau = 0.9 \times \min (SD, \sqrt{\frac{1}{\ln(2)} \times D_m}) \times n^{-0.2} \quad (3-8)$$

- (2) 设 $A$ 为研究区域的面积,  $n$ 为事件点个数, 则 $\tau$ 值满足<sup>[47]</sup>:

$$\tau = (0.68 \times n)^{-0.2} \times \sqrt{A} \quad (3-9)$$

- (3)  $k$ 阶最邻近距离方法来确定 $\tau$ 值,  $d_{ij}$ 代表 $k$ 阶最邻近距离, 表示从第一个对象点依次去计数到第 $k$ 个最邻近对象点的距离平均值。 $k$ 值的作用与带宽 $\tau$ 的作用十分相似,  $k$ 值与带宽参数 $\tau$ 的关系呈现正相关,  $k$ 值愈大, 生成的密度曲面也就越光滑<sup>[48]</sup>。

$$\tau = \sum_{i=1}^n \sum_{j=1}^k \frac{d_{ij}}{kn} \quad (3-10)$$

通过不同的核密度估计带宽使用, 可以观察到数据分布的微妙变化, 从中筛选出最佳的带宽进行估计。

### 3.4 北京市出租车热点信息提取

本节对出租车热点信息提取进行实验, 配置实验环境, 实现时空数据密度挖掘流程, 对核密度估计算法参数设置, 最后进行实验得出结论。

#### 3.4.1 实验环境

本文的实验运行环境配置如下:

- (1) 处理器: Intel® Core(TM) i7-9700 CPU @ 3.00 GHz
- (2) 内存(RAM): 16GB
- (3) 操作系统: Windows 64 位
- (4) 编程语言: Python3.7
- (5) 机器学习框架: scikit-learn 0.24.1
- (6) 工具包与开放库: Pandas、Numpy、Matplotlib、Scipy 等

### 3.4.2 时空数据密度挖掘

时空数据密度挖掘主要包括以下几个步骤:

- (1) 实验数据分析: 通过 3.2 节预处理的方法, 可以获取到北京市一周内的所有车辆的轨迹定位数据, 以车辆在星期一的数据为例, 获取到的预处理数据是  $\sum_N T_i \times 4$  尺寸的车辆轨迹数据。
- (2) 实验数据时间抽样: 由于原始数据时间以秒计算, 即每十秒采集一次位置数据, 因此时间序列一天内有  $60 \times 60 \times 24 \div 10 = 8640$  个时间点, 通过对预处理的数据可按照时间进行提取, 例如: 可获取某一时间点所有车辆的位置信息(经纬度), 利用这一时刻的位置信息进行核密度估计。由于车辆数据十分庞大, 这里以 100 秒的时间间隔进行采样, 一天则可获取共 864 个时间序列点, 如图 3-5 所示, 将采样的数据为下一步核密度估计提供输入。

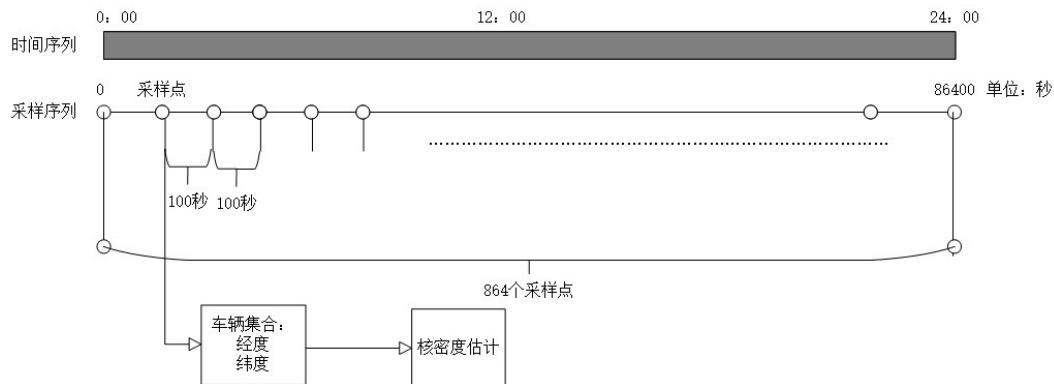


图 3-5 时间抽样过程示意图

- (3) 核密度估计获取车流密度数据: 对获取到的一天内的 864 个时间采样点的位置信息依次进行核密度估计, 获取到这些采样时间点的核密度估计模型。其次, 对研究区域  $Bounds = [116.0, 39.6, 116.8, 40.2]$ , 该数组解释为 [最小经度, 最小纬度, 最大经度, 最大纬度], 通过一个车流密度采样矩阵  $W_{Sample}$  进行密度提取, 其中, 采样矩阵的尺寸选取尤为重要。采样矩阵的尺寸过小, 会导致后续预测出租车热点精度不足; 采样矩阵尺寸过大, 会使预

测的计算时间复杂度以2次幂增长。因此，本次实验采用的 $W_{Sample}$ 矩阵尺寸为 $100 \times 100$ ，即将研究区域划分为 $99 \times 99$ 的小方格，对每个方格的顶点处进行密度采样，如图3-6是数据采样过程。

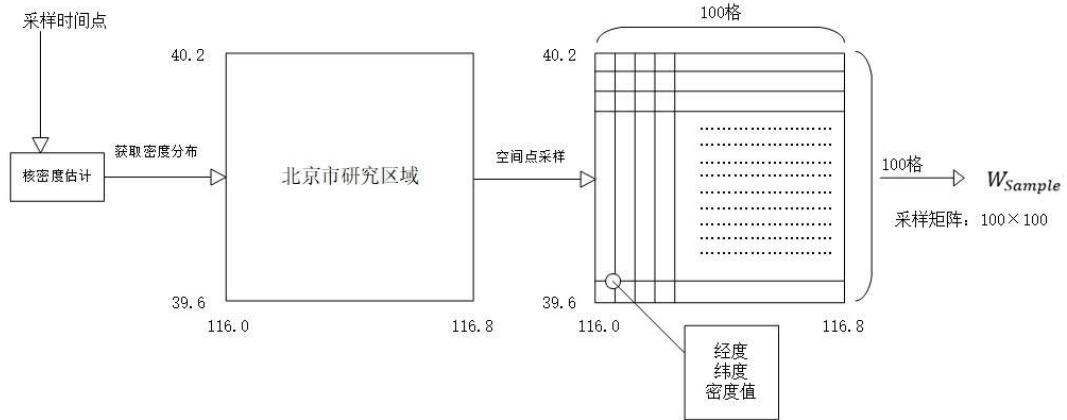


图3-6 空间采样过程示意图

(4) 车流密度数据集成：核密度估计数据将某一时刻 $T_i$  ( $i = 1, 2, \dots, 864$ ) 获取的采样点经纬度信息与该点的车流密度信息进行拼接，可以得到一个大小为 $10000 \times 3$ 的车流密度矩阵 $VD_i$ ，其中矩阵的行表示采样位置点个数，即车流密度采样矩阵大小 $100 \times 100 = 10000$ ，矩阵的第一列表示对应采样点的经度，第二列表示对应采样点的纬度，第三列表示对应采样点的车流密度。那么对于一天内的采样时间集合 $T = \{T_1, T_2, \dots, T_{864}\}$ ，可以得到一天内所有采样时刻的车流密度信息矩阵 $VD$ ，该矩阵大小为 $864 \times 10000 \times 3$ ，其中矩阵的三个维度分别表示为[时间序列数，研究区域采样点个数，车流密度信息]。如图3-7是数据的集成过程。

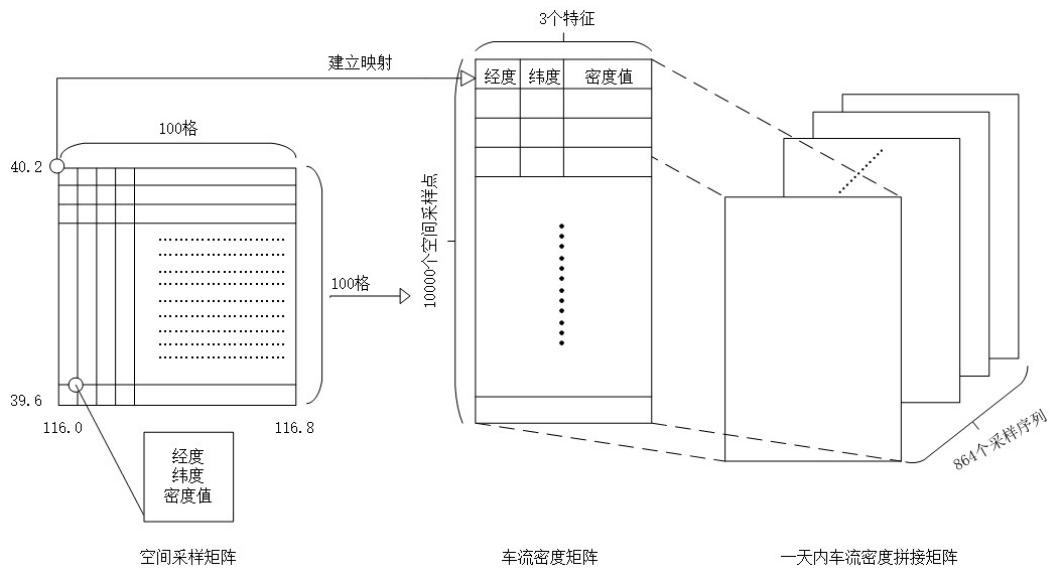


图3-7 车辆密度数据集成示意图

### 3.4.3 模型参数设置

核密度估计算法需要最重要的两个参数确定：核函数的选取和核密度估计带宽选取。本文采用核密度估计的核函数为高斯（Guassian）核函数。通过设置不同的带宽来观察核密度估计的效果，有助于对核密度估计带宽的理解，具体带宽设置为：0.1km, 0.25km, 0.5km, 0.75km, 1km, 2km, 3km, 6km, 10km 共 9 组。选出最合适的带宽后，以周一到周日早上八点（折合一天第 28800 秒，对应第 288 个时间采样点）和晚上八点（折合一天第 72000 秒，对应第 720 个时间采样点）的密度数据，以研究区域 $Bounds = [116.0, 39.6, 116.8, 40.2]$ 来进行车流密度分布的描绘。

其中，热力图描绘以地图实际的经纬度信息刻画，核密度估计的带宽以单位为千米（km）刻画，经纬度数据与公里的转化公式为：

$$\Delta Lon = \frac{\Delta km \cdot 360}{2\pi \cdot R_e \cdot \cos(\frac{(Lat_1 - Lat_2)\pi}{360})} \quad (3-11)$$

$$\Delta Lat = \frac{\Delta km \cdot 360}{2\pi \cdot R_e} \quad (3-12)$$

其中， $\Delta Lon$ ,  $\Delta Lat$ 表示经纬度的变化量， $\Delta km$ 表示公里变化量， $R_e$ 是地球的半径（单位为千米）， $Lat_1$ ,  $Lat_2$ 表示研究区域的最小纬度和最大纬度。

可以看出，纬度的变化和公里数的变化呈线性相关，主要是所有的经线长度都一样，而经度的变化与公里数的变化涉及到不同的纬度上对应数值不一样，因为在赤道纬度线长最大，南北极的纬度线长为 0。

### 3.4.4 实验结果与分析

基于核密度估计的城市车联密度提取模型的实验思路是：首先，通过设置不同长度的带宽，检验核密度估计的效果，选出效果最好的核密度估计带宽来对每天不同时刻和不同天的同一时刻进行描绘。

#### （1）不同带宽下核密度估计车流密度提取效果

根据设置的带宽组：0.1km, 0.25km, 0.5km, 0.75km, 1km, 2km, 3km, 6km, 10km。以周一早上八点（折合一天第 28800 秒，对应第 288 个时间采样点）的数据进行核密度估计，得到如图 3-8 效果。

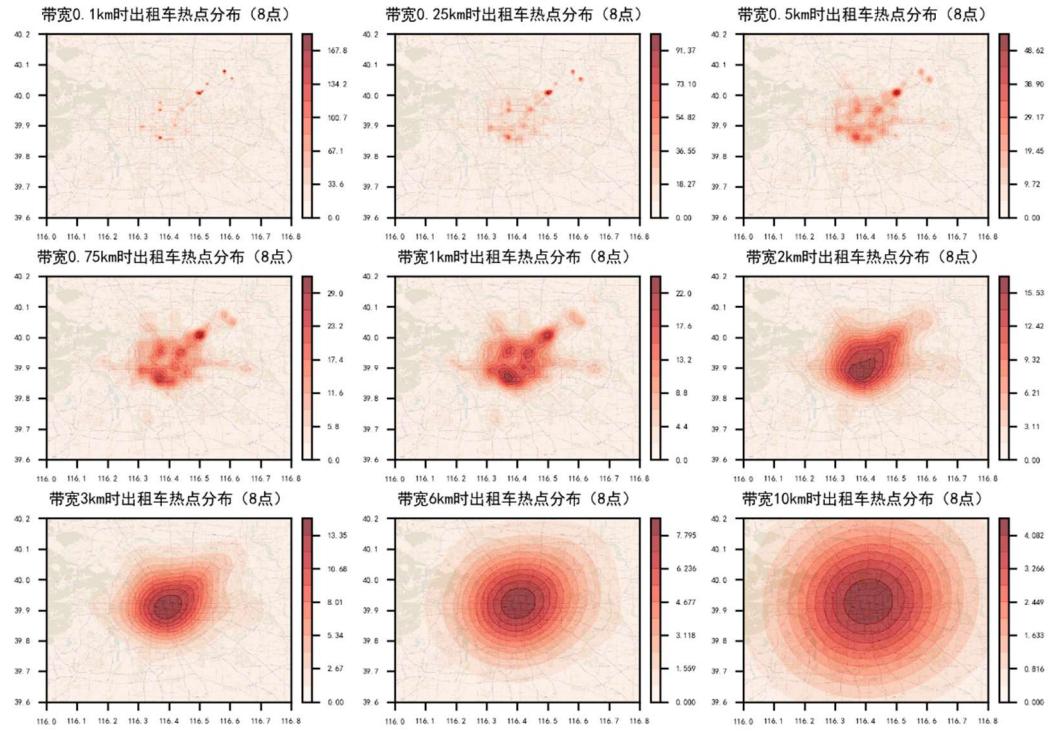


图 3-8 不同带宽下核密度估计热力分布图

从图中可以看出，不同的带宽进行核密度估计的效果差异十分明显。当选择较小的核密度估计带宽时，估计出的密度分布出现较多的热点与冷点区域，能够将核密度估计的局部密度信息保留；当增大带宽时，能使得热点在全局范围内的表现更加明了，但是会失去局部分布的特征。

综合考虑局部热点分布特征和全局热力分布层次特征，选择带宽为 0.5km 时效果最佳，作为后续分析预测的带宽选择。

## (2) 最佳带宽下不同时间点车流密度提取效果

以最佳带宽 0.5km 对北京市出租车周一到周日早上 8:00（折合一天第 28800 秒，对应第 288 个时间采样点）和晚上 20:00（折合一天第 72000 秒，对应第 720 个时间采样点）的位置数据进行核密度估计，如图 3-9 至 3-15 所示。

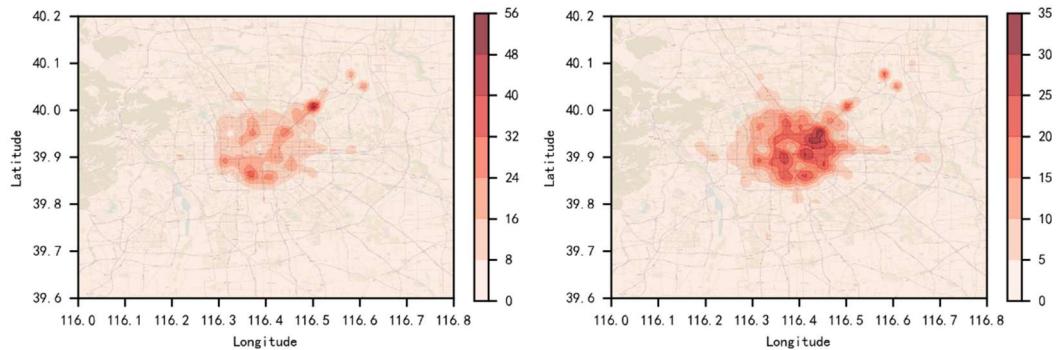


图 3-9 周一早 8 点（左）和晚 8 点（右）出租车密度分布图

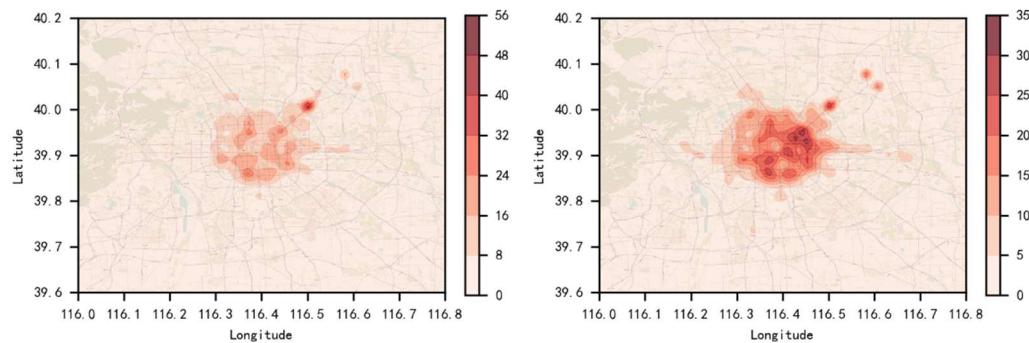


图 3-10 周二早 8 点（左）和晚 8 点（右）出租车密度分布图

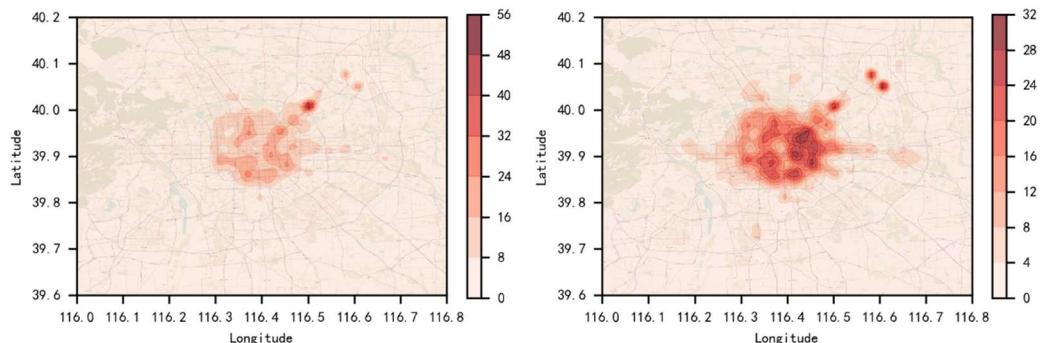


图 3-11 周三早 8 点（左）和晚 8 点（右）出租车密度分布图

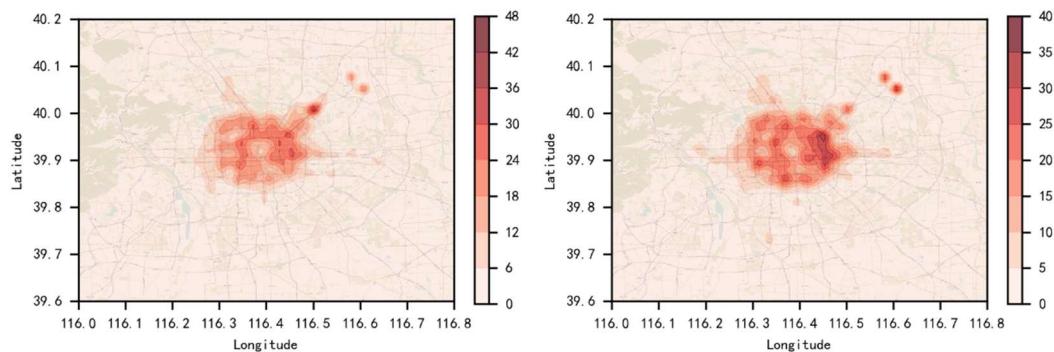


图 3-12 周四早 8 点（左）和晚 8 点（右）出租车密度分布图

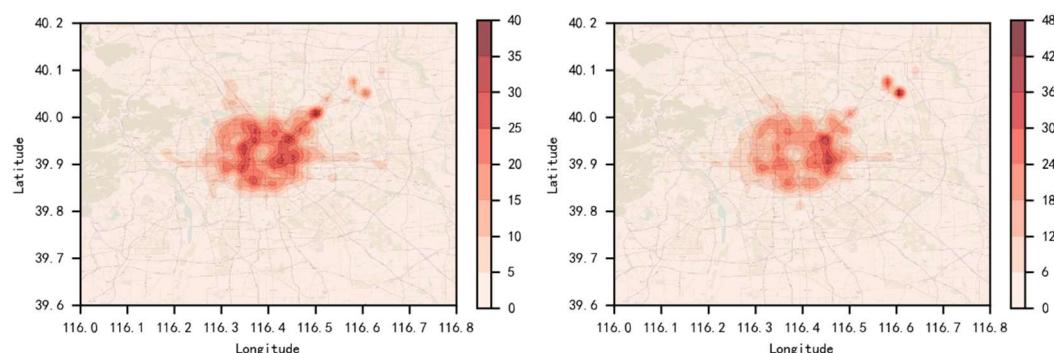


图 3-13 周五早 8 点（左）和晚 8 点（右）出租车密度分布图

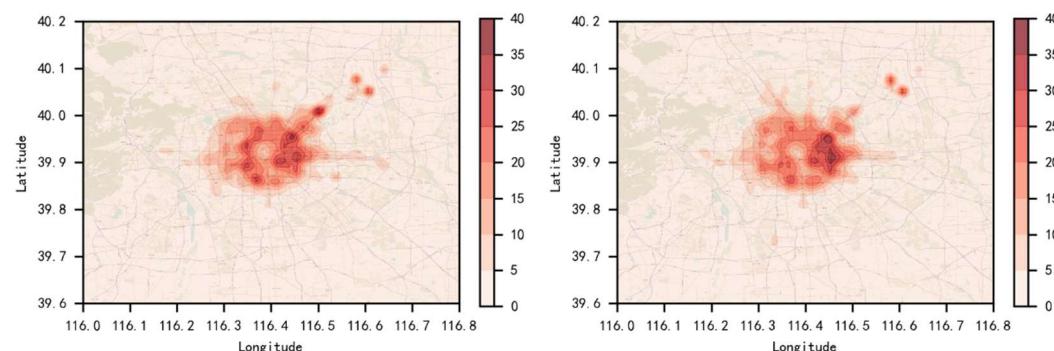


图 3-14 周六早 8 点（左）和晚 8 点（右）出租车密度分布图

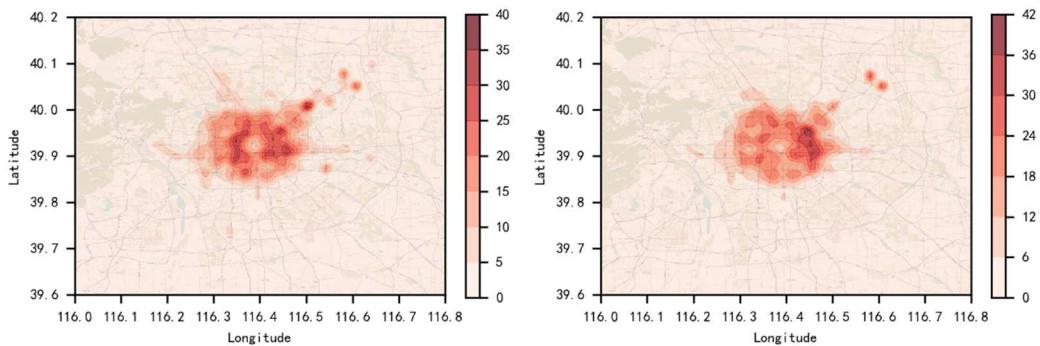


图 3-15 周日早 8 点（左）和晚 8 点（右）出租车密度分布图

观察核密度估计密度分布图，从全局来看，不同时刻的出租车热点分布存在明显差异，密度峰值也存在差异；从局部来看，不同时刻存在着相近的出租车热点，且局部分布存在很高的相似性。

从全局角度分析来看：由于出租车核密度估计以北京市整个区域所有 28590 辆出租车进行估计，不同时间点研究区域内车辆的集中度不一致。从不同时刻纵向的密度分布集中程度来看，观察周一到周日早晨 8:00 的密度热力图，周一至周三早晨 8:00 的出租车密度峰值接近 56，周四早晨 8:00 的出租车密度峰值接近 48，周五至周日早晨 8:00 的出租车密度峰值仅接近 40，说明一周内从周一到周日早晨 8:00 时刻出租车热点密度峰值逐渐减小，也意味着出租车在从全局的角度看越来越相对分散。同时，从热点数目来看，一周内前三天的热点数目相对较少，后四天的热点数目相对更多，且热点更集中分布在城区四环内。观察周一到周日晚上 20:00 的密度热力图，周一至周二晚上 20:00 的出租车密度峰值接近 35，周三接近 32，周四和周六接近 40，周五接近 48，周日接近 42，可以看出周一到周日晚上 20:00 的出租车密度分布集中程度呈波动状态。从不同时刻横向的密度分布集中程度来看，可以看出，周一至周四早晨到晚上的出租车明显从相对集中在城区某些高热点区到相对分散在城区，周五则出租车明显从相对分散在城区到相对集中在城区某些高热点区，周末则在早晚处于相对均衡的状态。

从局部角度分析来看，北京市出租车热点分布存在一些热点的持续时间长和热点的动态变化两大特征。从纵向角度来看，周一到周日早上 8:00 的出租车相对更集中在研究区域，局部密度峰值有更热的存在，出租车经常性热点区域包括一周内早上 8:00 始终存在东经  $116.5^{\circ}$ ，北纬  $40.0^{\circ}$  附近的出租车热点，即北京市朝阳区东园文化创意广场附近，从附近的兴趣点 (POI, Point of Interest) 来看，附近有接连的汽车服务中心和汽车维修中心，出租车密度高，很大概率是因为多数出租车在此处维护修理，同时附近地处

居民活动热区，使得出租车车流密度相对集中。出租车动态性比较大的热点则在北京市区内四环存在，众多出租车聚集热点在每天同一时刻变化性较大，一定程度呈现了出租车在城区接客的动态变化，反映出居民出行的频繁地点。从横向的角度来看，晚上 20:00 时位于东经  $116.5^{\circ}$ ，北纬  $40.0^{\circ}$  附近的出租车热点明显没有早晨密度高，而城区内热点会出现热点密度最高点，可见，晚上出租车的活动相对分布在更密集的东四环附近的城区，是出租车聚集的强热点地区。

### 3.5 本章小结

本章是本文探讨城市车流密度预测的基础。针对北京市 28590 辆出租车一周内的原始轨迹数据，分析观察数据特征和结构，并对其进行适当预处理工作；利用爬虫技术获取开源地图北京市背景，实现数据点与真实地图位置的匹配；并利用核密度估计算法实现时空数据密度的挖掘，通过实验得到最佳核密度估计带宽  $0.5\text{km}$ ，并依据此获取一天内的时序车流密度信息数据集。该模型的工作完成了原始数据到车流密度挖掘并可视化的过程，为后续利用历史车流密度信息预测未来车流密度信息做好准备。同时，利用核密度估计的思想去挖掘车辆信息，从全局和局部两个角度去分析，有着更宏观的研究意义。

## 4 基于支持向量回归的热点预测模型

本章主要研究思路：首先利用滑动窗口模型构造训练数据集，并归一化处理数据，让模型预测在训练时收敛更快；其次，介绍支持向量回归及其参数，并对模型参数进行设置；最后，利用支持向量回归模型对核密度数据进行预测，并对模型进行评估。

基于支持向量回归的热点预测模型的基本流程如图 4-1 所示。

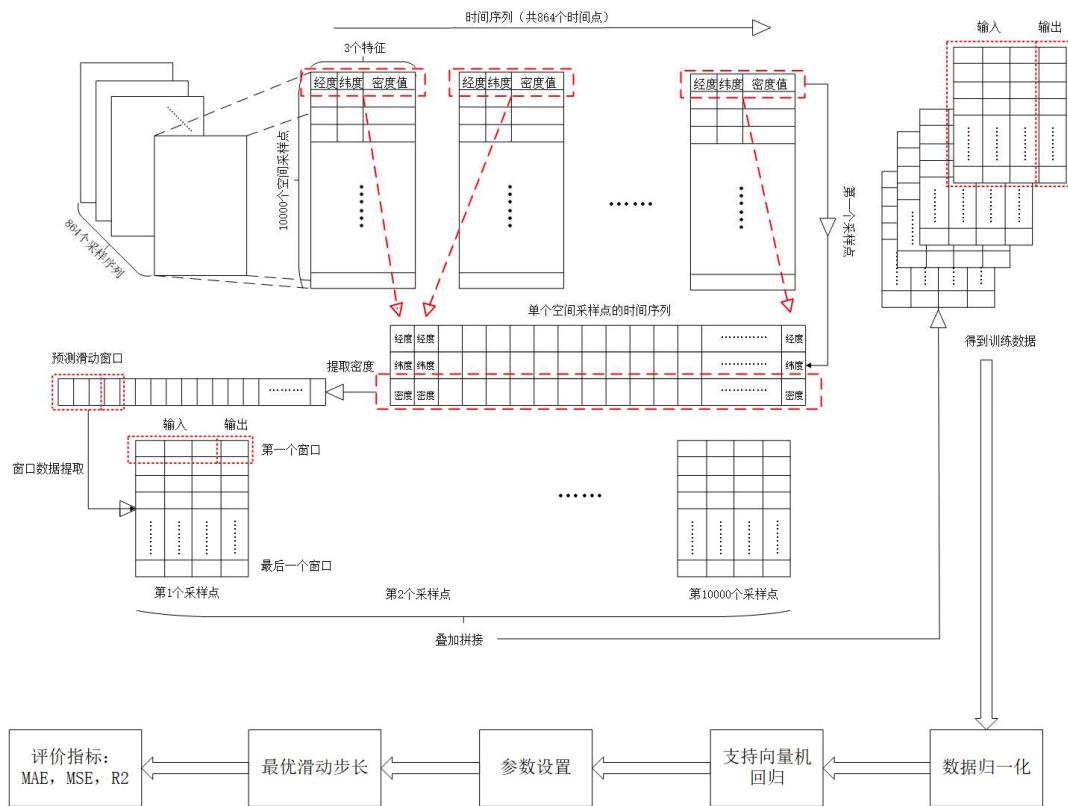


图 4-1 基于支持向量回归的热点预测流程

### 4.1 基于滑动窗口的热点信息数据处理

#### 4.1.1 滑动窗口模型

通过基于核密度估计算法的城市车流密度提取模型已经可以得到北京市一周内任意一天的车流密度信息矩阵  $VD$ ，在 3.4.2 节中，可以得到该矩阵大小为  $864 \times 10000 \times 3$ ，其中矩阵的三个维度分别表示为[时间序列数，研究区域采样点个数，车流密度信息]。以周一的数据为例，经过提取的数据是一个三维矩阵，在时间序列轴上，可以挖掘出每个采样点的车流密度信息的时间序列，作为后续预测的依据。

滑动窗口模型通过构造滑动窗口 $W_L$ , 实现对密度信息数据进行挖掘和构造, 滑动窗口可以通过三个属性来确定, 包括: 置于窗口的数据集对象 $Data_s$ , 窗口开始位置 $P_{start}$ , 滑动窗口的步长量 $L$ 。针对滑动窗口步长, 可以将其分为单步长滑动窗口模型和多步长滑动窗口模型<sup>[49]</sup>。

### (1) 单步长滑动窗口模型

对于给定的一个时间序列数据 $X = \{x_1, \dots, x_i, \dots\}$ , 单步长滑动窗口模型定义为: 构造一个长度为 1 个单位的滑动窗口 $W$ , 从序列 $X$ 开头将滑动窗口依次逐步向后滑动, 提取出滑动窗口 $W$ 内数据 $\{x_i\}$ , 将获取的滑动窗口数据按列拼接形成单步长窗口数据集。

### (2) 多步长滑动窗口模型

同单步长滑动窗口类似, 给定时间序列数据 $X = \{x_1, \dots, x_i, \dots\}$ , 多步长滑动窗口模型定义为: 构造步长为 $L$ 的窗口 $W_L$ , 从序列 $X$ 开头将滑动窗口依次逐步向后滑动, 提取出滑动窗口 $W$ 内数据 $\{x_i, x_{i+1}, \dots, x_{i+L-1}\}$ , 将获取的滑动窗口数据按列拼接形成多步长窗口数据集。

## 4.1.2 训练数据集构造

机器学习中, 我们使用监督学习的方法, 把需要的数据集分为输入数据和输出数据, 即表示为 $Data = \{X, Y\}$ , 非监督学习数据集只有输入 $Data = \{X\}$ 。而监督学习的目的是建立数据集里 $X \rightarrow Y$ 的映射, 即模型的输入可以一一映射到模型的输出。那么对应的数据集可表示为 $Data = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , 其中,  $x_i \in X$ 为输入值,  $y_i \in Y$ 为输出值, 通过学习训练得到模型, 回归决策模型可表示为 $y = f(x)$ 。

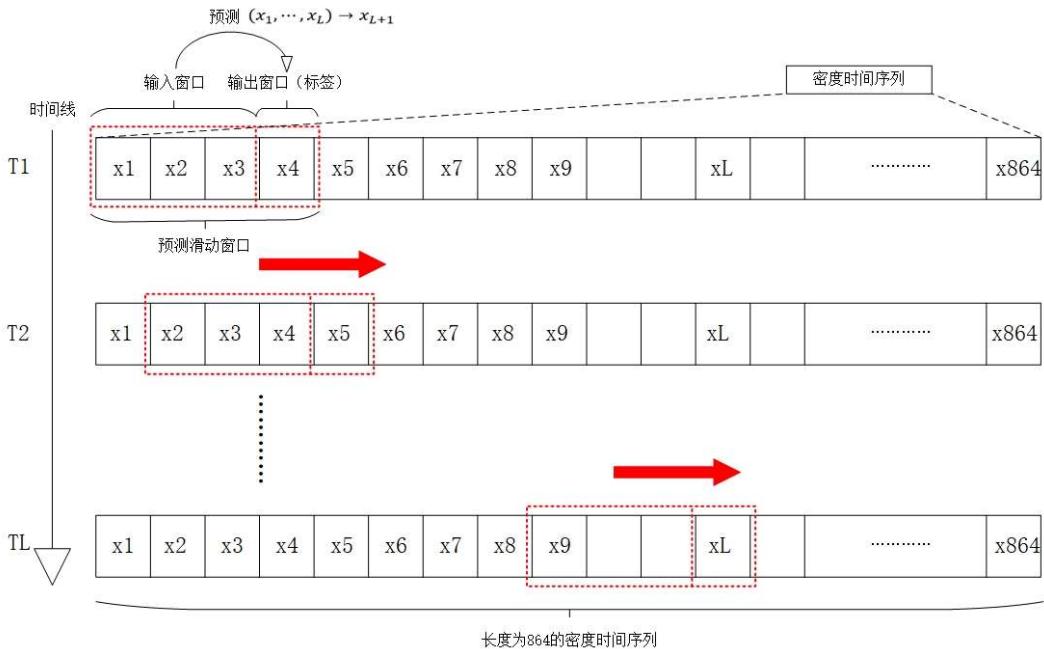


图 4-2 预测滑动窗口模型示意图

那么对于训练所需要的数据集，就需要对输入输出数据进行构造，由于需要对某一时刻密度值进行预测，可以分析出 $Y$ 是一个一维向量，而对于输入数据取决于滑动窗口的步长。即对于一个时间序列，在滑动窗口后加上一项输出项标签，从而形成预测滑动窗口。具体实现如图 4-2 所示。

对时间序列数据 $X = \{x_1, \dots, x_i, \dots\}$ ，对于单步长的预测窗口 $W$ 可建立映射关系 $\{x_1 \rightarrow x_2, x_2 \rightarrow x_3, \dots, x_i \rightarrow x_{i+1}, \dots\}$ ，将预测数据用 $Y$ 来表示，即 $\{x_1 \rightarrow y_1, x_2 \rightarrow y_2, \dots, x_i \rightarrow y_i, \dots\}$ 。对于多步长（步长 $L$ ）的预测窗口 $W_L$ 可建立映射关系 $\{(x_1, \dots, x_L) \rightarrow x_{L+1}, (x_2, \dots, x_{L+1}) \rightarrow x_{L+2}, \dots, (x_i, \dots, x_{i+L-1}) \rightarrow x_{i+L}, \dots\}$ ，将预测数据用 $Y$ 来表示，即 $\{(x_1, \dots, x_L) \rightarrow y_1, (x_2, \dots, x_{L+1}) \rightarrow y_2, \dots, (x_i, \dots, x_{i+L-1}) \rightarrow y_i, \dots\}$ 。而对于本文研究对象，时间序列数据 $X$ 只有一个特征，即某一采样点的密度值，可表示为 $x_i = \{\rho_i\}$ ，其中 $\rho_i$ 是第*i*时刻的密度值。

因此，通过上述对预测窗口描述总结，对出租车车辆密度预测模型的数据集构造流程如下：

- 1) 将车流密度信息矩阵 $VD$ 按第二维度进行划分成若干个采样点时间序列信息，记为 $VD_s$ ，表示第*s*个采样点序列信息，其中 $s = 1, 2, \dots, 10000$ 。
- 2) 对于任意一个采样点时间序列信息 $VD_s$ ，是一个 $864 \times 3$ 的二维数组，其中第一维度记一天内的采样时间点的数目，第二维度的前 2 个特征是经纬度信息，第 3 个特征是我们需要的对应位置的密度信息，因此，将第二维度的第三个特征进行提取，成为时间序列密度，记为 $\rho_s$ 。对于时间序列密度 $\rho_s$ ，

它是一个一维的向量，其中的每一个元素代表第 $s$ 个采样点一天内采样时间上的密度值。

3) 在时间序列密度 $\rho_s$ 上加预测滑动窗口，取步长为 $L$ ，即每连续的 $L$ 个密度值来预测下一时刻的密度值，形成数据集的一项 $(x_i, \dots, x_{i+L-1}) \rightarrow x_{i+L}$ ，将预测滑动窗口从时间序列密度上从队头滑到队尾，得到第 $s$ 个采样点的训练数据 $Data_s$ ，它是一个 $(864 - L - 1) \times (L + 1)$ 维度的二维数组数据。

4) 对研究区域内 10000 个采样点数据进行上述步骤 2) 和 3) 重复操作，将获取到的所有 $Data_s$ 进行叠加，形成 $10000 \times (864 - L - 1) \times (L + 1)$ 的三维训练数据 $Data$ 。

如图 4-3 所示，是训练数据构造的全过程。

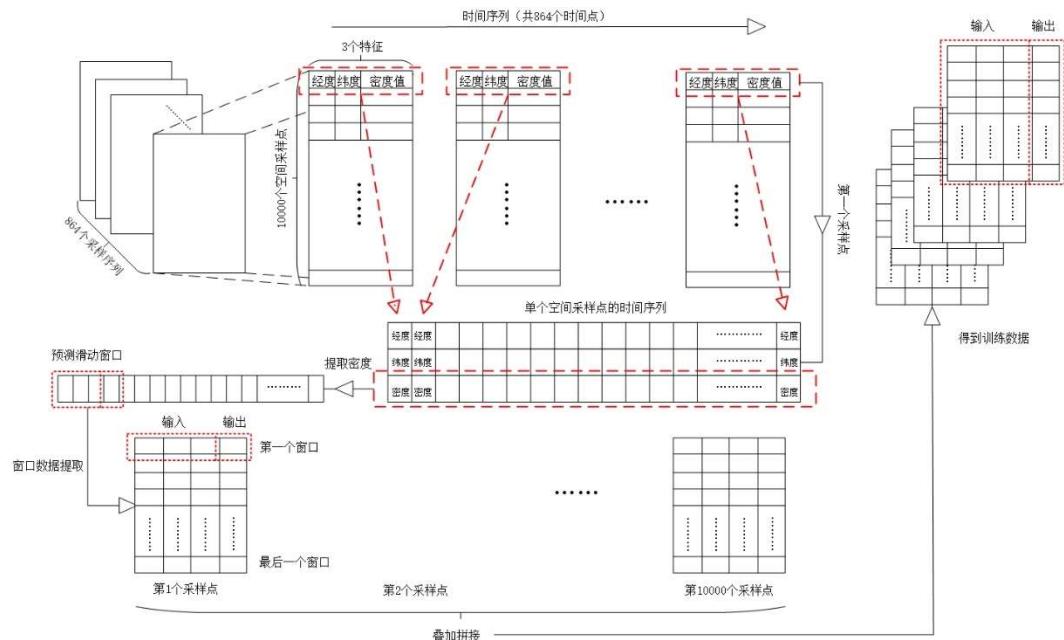


图 4-3 训练数据集构造流程

### 4.1.3 数据的归一化处理

一般地，不同的数据属性因其物理意义不同，量纲因此有差异，使得不同属性的数值没有比较性，为了消除这些影响，减小参数的大小，将数据进行归一化处理在进行比较更能挖掘出数据特征，同时，归一化的处理的目的是一定程度上加速求解最优解的速度。其方法包括：Min-Max 方法和 Z-Score 方法<sup>[50]</sup>。

#### (1) Min-Max 归一化方法

Min-Max 归一化是要借助线性变换，来处理原始的数据，建立一个映射关系，让输出结果映射在 0 到 1 的区间。具体的映射公式描述为：

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (4-1)$$

其中， $X$ 为待归一化数据， $X_{norm}$ 为归一化后数据， $X_{max}$ 为数据 $X$ 中的最大值， $X_{min}$ 为数据 $X$ 中的最小值。

## (2) Z-Score 归一化方法

Z-Score 归一化是借助数据的平均值和方差属性来进行归一计算，使得处理后的结果符合标准的正态分布 $N(0,1)$ 。具体的映射公式描述为：

$$X_{scale} = \frac{X - X_{mean}}{S} \quad (4-2)$$

其中， $X$ 为待归一化数据， $X_{scale}$ 为归一化后数据， $X_{mean}$ 为数据 $X$ 中的均值， $S$ 为数据 $X$ 中的方差。

本文采用 Min-Max 方法对构建的训练数据进行统一处理。

归一化后的数据想要还原为原始数据可以通过归一化还原实现，对相应的方法取逆运算即可。

## 4.2 支持向量回归预测

### 4.2.1 支持向量回归

机器学习方法现在被应用在越来越多的领域，提供分类和预测的建模方法。支持向量机（SVM，Support Vector Machine）是其中一种高效常用的方法，在函数回归，序列预测等领域影响深远，目前是机器学习领域内的关注热点。其实质是基于统计学习的方法和理论去实现。

支持向量机的基本思想是寻找正反数据区分的最大化间隔的界限，利用线性分割来分类，但往往低维度无法做到线性分割，需要在某一更高维度空间内建立一个界限使得正反数据的间隔最大化<sup>[51]</sup>。同时，该问题也转化为了求解最优化的问题，在这个过程中需要经过数据维度提升和下降。维度提升简单来看就是把低维数据映射到高维空间，这一过程通过非线性手段实现，然后在高维空间中解决低维数据的非线性问题。如图 4-4 反映了这一过程直观的图示。

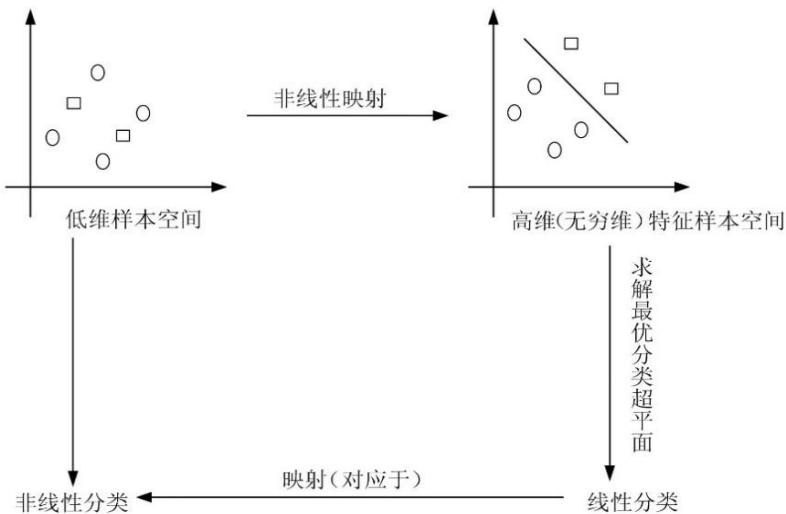


图 4-4 支持向量机基本思想示意图

而 SVR (Support Vector Regression) 与 SVM 的思想方法相同，只是支持向量回归用于做回归预测问题，支持向量机用于做分类预测问题。对于支持向量回归 (SVR) 问题，可以描述为：

设训练样本 $\{x_i, y_i\}_{i=1}^m$ ，其中 $x_i$ 是输入模式的第*i*个样本， $y_i$ 是对应的输出结果。希望学习得到一个回归模型 $f(x) = w \cdot x + b$ ，使得 $f(x)$ 与  $y$  尽可能接近，其中：模型的参数包括 $w$ 和 $b$ 。

给定样本 $(x, y)$ ，依据传统回归模型的计算损失的方法，需要通过 $f(x)$ 与  $y$  之间的差值来衡量， $f(x)$ 与  $y$  相同时损失为 0。支持向量回归与此有些差异，在预测前就假设 $f(x)$ 与  $y$  之间的忍耐误差最多有 $\epsilon$ 限制，仅当 $f(x)$ 与  $y$  之间的正负偏差大于 $\epsilon$ 时才计算损失。如图 4-5 是直观的 SVR 误差容忍度。

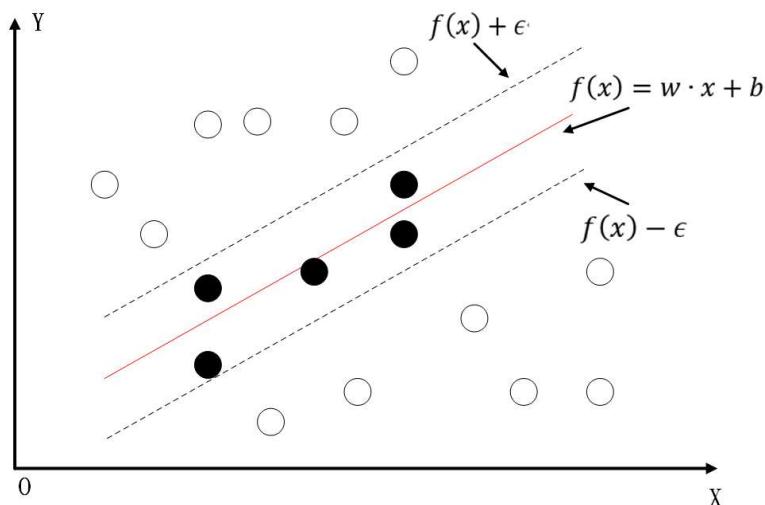


图 4-5 SVR 容忍度示意图

于是，SVR 问题转化为最优化问题：

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m l_\epsilon(f(x_i) - y_i) \quad (4-3)$$

其中,  $C$  为正则化参数,  $l_\epsilon$  表示为:

$$l_\epsilon(z) = f(x) = \begin{cases} 0, & \text{if } |z| < \epsilon \\ |z| - \epsilon, & \text{otherwise} \end{cases} \quad (4-4)$$

引入松弛变量  $\varepsilon_i$  和  $\tilde{\varepsilon}_i$ , 优化问题转化为:

$$\begin{aligned} & \min_{w,b,\varepsilon_i,\tilde{\varepsilon}_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\varepsilon_i + \tilde{\varepsilon}_i) \\ & \text{s.t. } f(x_i) - y_i \leq \epsilon + \varepsilon_i \\ & \quad y_i - f(x_i) \leq \epsilon + \tilde{\varepsilon}_i \\ & \quad \varepsilon_i \geq 0, \tilde{\varepsilon}_i \geq 0, i = 1, 2, \dots, m \end{aligned} \quad (4-5)$$

引入拉格朗日乘子  $\mu_i \geq 0, \hat{\mu}_i \geq 0, \alpha_i \geq 0, \hat{\alpha}_i \geq 0$ , 得到函数:

$$L(w, b, \alpha, \hat{\alpha}, \varepsilon_i, \tilde{\varepsilon}_i, \mu_i, \hat{\mu}_i) = \min_{w,b,\varepsilon_i,\tilde{\varepsilon}_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\varepsilon_i + \tilde{\varepsilon}_i) - \sum_{i=1}^m \mu_i \varepsilon_i - \sum_{i=1}^m \mu_i \varepsilon_i + \sum_{i=1}^m \alpha_i (f(x_i) - y_i - \epsilon - \varepsilon_i) + \sum_{i=1}^m \hat{\alpha}_i (f(x_i) - y_i - \epsilon - \tilde{\varepsilon}_i) \quad (4-6)$$

求偏导, 令偏导等于 0, 可得:

$$\begin{aligned} w &= \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) x_i \\ 0 &= \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) \\ C &= \alpha_i + \mu_i \\ C &= \hat{\alpha}_i + \hat{\mu}_i \end{aligned} \quad (4-7)$$

代入到拉格朗日函数中, 转化为 SVR 对偶问题:

$$\begin{aligned} & \max_{\alpha_i, \hat{\alpha}_i} \sum_{i=1}^m y_i (\hat{\alpha}_i - \alpha_i) - \epsilon (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) x_i^T x_j \\ & \text{s.t. } \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) = 0 \\ & \quad 0 \leq \hat{\alpha}_i, \alpha_i \leq C \end{aligned} \quad (4-8)$$

上述过程满足 KKT (Karush-Kuhn-Tucke) 条件:

$$\begin{aligned} \alpha_i (f(x_i) - y_i - \epsilon - \varepsilon_i) &= 0 \\ \hat{\alpha}_i (f(x_i) - y_i - \epsilon - \tilde{\varepsilon}_i) &= 0 \\ \hat{\alpha}_i \alpha_i &= 0, \varepsilon_i \tilde{\varepsilon}_i = 0 \\ (C - \alpha_i) \varepsilon_i &= 0, (C - \hat{\alpha}_i) \tilde{\varepsilon}_i = 0 \end{aligned} \quad (4-9)$$

从上式可以看出, 当且仅当  $f(x_i) - y_i - \epsilon - \varepsilon_i = 0$  时  $\alpha_i$  可以取到非 0 值, 当且仅当  $f(x_i) - y_i - \epsilon - \tilde{\varepsilon}_i = 0$  时  $\hat{\alpha}_i$  可以取到非 0 值。也就是仅当样本不落在  $\epsilon$  间的间隔带中,  $\hat{\alpha}_i, \alpha_i$  才能取到非 0 值, 除此之外,  $f(x_i) - y_i - \epsilon - \varepsilon_i = 0$  和  $f(x_i) - y_i - \epsilon - \tilde{\varepsilon}_i = 0$  不能够同时成立, 因此,  $\hat{\alpha}_i, \alpha_i$  至少有一个为 0。因此

SVR 的解如下：

$$f(x) = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) x_i^T \cdot x + b \quad (4-10)$$

能使上式  $\hat{\alpha}_i - \alpha_i \neq 0$  的数据点我们称为 SVR 的支持向量，它们落在了设定的容忍度  $\epsilon$  间隔区域之外的地方，这时候，模型实际训练就是利用找到的 SVR 的支持向量，即训练数据的部分，具有稀疏性。

得到 SVR 解后，依据 KKT 条件，每个样本应该都有  $(C - \alpha_i)\varepsilon_i = 0$  且  $\alpha_i(f(x_i) - y_i - \epsilon - \varepsilon_i) = 0$ 。于是，在得到  $\alpha_i$  后，若  $0 < \alpha_i < C$ ，则必有  $\varepsilon_i = 0$ ，进而得到：

$$b = y_i + \epsilon - \sum_{j=1}^m (\hat{\alpha}_j - \alpha_j) x_j^T \cdot x_i \quad (4-11)$$

实际运用过程使用更稳定的方案：选取尽可能多的符合条件  $0 < \alpha_i < C$  的数据来计算  $b$  的平均值，并引入核函数，用  $x$  替代  $K(x)$ ，SVR 最终的结果为：

$$f(x) = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) K(x_i)^T \cdot K(x_i) + b \quad (4-12)$$

## 4.2.2 支持向量回归参数

在进行 SVR 预测时，模型有几个重要的参数需要根据实际情况进行选择调试。主要包括核（Kernel）函数类型的选择和惩罚因子 C 的确定。

(1) 核 (Kernel) 函数类型：包括线性 (linear) 核，多项式 (poly) 核，高斯核 (RBF)，sigmoid 核。

- 1) 线性核：优点是一般方案首选，简单，参数少，速度快，对于一般数据预测效果就很不错；缺点是只能解决线性可分问题。
- 2) 多项式核：优点是可解决非线性问题，可主观设置幂数；缺点是大数量级的幂数会产生很大的偏差效果，参数相对较多进行选择。
- 3) 高斯核：优点是决策边界更多样，只有一个参数，可以将数据映射到无限维度，比多项式核参数更容易选择；缺点是解释性差，容易过拟合。
- 4) Sigmoid 核：此时支持向量机可以理解为一种多层次感知器神经网络。

(2) 惩罚因子 C 确定：惩罚因子 C 表征的是数据相对整体偏离的重视程度，C 越大表明越受关注，越不想去丢掉它们。C 值大时对误差预测的惩罚增大，C 值小时对误差预测的惩罚减小。当 C 足够大，趋近无穷大极限表示不允许有误差，从而出现过拟合；当 C 趋于 0 极限时，表示预测不再关注正确性，出现欠拟合现象。一般通过不同数量级的 C 来寻找最优的 C 来进行预测分析。

## 4.3 支持向量回归热点预测

本节利用支持向量回归算法来实现对车流密度和热点分布的估计，首先配置实验环境，调整模型参数，确定评价标准，最后进行本节的预测实验。

### 4.3.1 实验环境

本文的实验运行环境配置如下：

- (1) 处理器：Intel® Core(TM) i7-9700 CPU @ 3.00 GHz
- (2) 内存(RAM): 16GB
- (3) 操作系统：Windows 64 位
- (4) 编程语言：Python3.7
- (5) 机器学习框架：scikit-learn 0.24.1
- (6) 工具包与开放库：Pandas、Numpy、Matplotlib 等

### 4.3.2 模型参数设置

- (1) 支持向量回归参数设置

通过支持向量机进行回归预测，需要设置两个基本参数，即核函数的选择和正则化参数 C 的选择。具体如表 4-1 所示。

**表格 4-1 SVR 模型参数设置**

参数	符号	取值
核函数	$K(\cdot)$	Linear（线性）
正则化系数	C	10

- (2) 训练集和测试集的划分

前文通过预测滑动窗口构造了可以用作训练的数据集  $Data$ ，对于单个采样点数据集是一个尺寸为  $(864 - L - 1) \times (L + 1)$  大小的矩阵，本实验采用一天内前 700 个采样时间序列作为训练集  $Data_{train}$ ，剩余时间为测试集  $Data_{test}$ 。每个采样点的时间序列都采用这种划分方式，得到训练集是尺寸为  $10000 \times 700 \times (L + 1)$  的矩阵数据，测试集是尺寸为  $10000 \times (164 - L - 1) \times (L + 1)$  的矩阵数据。

### 4.3.3 预测评价标准

#### (1) 平均绝对误差 (Mean Absolute Error, MAE)

MAE 指参数的预测值与真实值差值的绝对值的期望，可以评价模型的拟合效果，MAE 越小，模型拟合程度越好，MAE 越大，模型拟合效果越差。MAE 计算公式为：

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (4-13)$$

#### (2) 均方误差 (Mean Squared Error, MSE)

MSE 指参数的预测值与真实值差值的平方的均值，评价模型的拟合程度，MSE 越小，模型拟合程度越好，MSE 越大，模型拟合程度越差。MSE 计算公式为：

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4-14)$$

#### (3) 判定系数 $R^2$

研究  $R^2$  score，涉及到三个统计学概念：

1) 回归平方和 (SSR): 是预测值与平均值的误差，反映自变量与因变量之间的关联程度的偏差平方和。

$$SSR = \sum_{i=1}^N (y_i - \bar{y})^2 \quad (4-15)$$

2) 残差平方和 (SSE): 即预测值与真实值的误差，反映模型拟合效果。

$$SSE = \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4-16)$$

3) 总离平方和 (SST): 即平均值与真实数据值的误差，反映真实数据与数学期望的偏离程度。

$$SST = \sum_{i=1}^N (\bar{y} - \hat{y}_i)^2 \quad (4-17)$$

决定系数  $R^2$ ，表示为：

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (\bar{y} - \hat{y}_i)^2} = 1 - \frac{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}}{\frac{\sum_{i=1}^N (\bar{y} - \hat{y}_i)^2}{N}} = 1 - \frac{MSE(y, \hat{y})}{Var(y)} \quad (4-18)$$

其中  $R^2 \in (-\infty, 1]$ ，在几种情况下效果分析如下：

1)  $R^2 = 1$ ，分子 MSE 为 0，此时是  $R^2$  所有取之中最大的情况，即预测模型没有误差，模型结果完美拟合。一般情况下模型不可能完美拟合，只会趋近于 1，这时候的模型就可以认为效果很不错了，随着  $R^2$  的减小，模型的预测误差会越来越大。

2)  $R^2 = 0$ ，此时模型的预测结果和直接取平均结果效果相同。

3)  $R^2 < 0$ ，此时训练模型的误差甚至不如直接赋予数据均值，出现这种

情况，通常是模型可能性质上不满足线性关系，而误使用了线性预测模型，产生误差很大。

#### 4.3.4 实验结果与分析

基于支持向量回归的热点预测模型的实验思路如下：通过处理好的北京市一周内的数据集和支持向量回归的参数确定，首先验证不同预测滑动窗口步长下的预测效果，并通过评价指标量化得到最优的预测滑动窗口步长；选取最优的滑动窗口步长进行预测，选取北京市三个地点进行密度预测曲线绘制分析；最后，选取三个时刻预测结果与真实热点分布对比分析。

##### (1) 不同预测滑动窗口步长下的预测效果

预测滑动窗口步长决定着被预测时间点受过去时间线的影响的程度，实验采用预测滑动窗口步长组：2，3，4，5，6，7，8，9。实验原始数据采用周一全天车辆估计数据，通过基于核密度估计算法的城市车流密度提取模型，再通过不同步长的设置，根据4.1.2节训练数据进行构造，生成每组需要的训练数据。并根据4.3.2节中的训练集和测试集进行划分，通过支持向量回归模型学习得到不同步长下所有采样点时间序列的测试集评价指标的均值对比如表4-2所示。

**表格 4-2 SVR 不同步长下预测下评价指标均值对比**

窗口步长	MSE	MAE	R <sup>2</sup>
2	0.0994	0.1305	<b>0.4785</b>
3	<b>0.0992</b>	<b>0.1302</b>	0.4737
4	0.1001	0.1305	0.4695
5	0.1006	0.1308	0.4659
6	0.1007	0.1310	0.4617
7	0.1002	0.1307	0.4573
8	0.0999	0.1306	0.4579
9	0.0999	0.1305	0.4569

从表4-2中可以看出，不同预测滑动窗口的步长对SVR预测的效果影响并不是很大，综合三项指标来看，步长为3的预测窗口的效果最佳，MSE和MAE误差均最小，且R<sup>2</sup>值也处于较高水平。其实，不难从步长的物理意义看出，步长越长，说明预测结果的值受过去更多的时间采样点影响，容易产生预测结果的太过依赖于历史值，在发生个别数据突变的情况下预测偏差过大；

步长越短,说明预测结果受历史数据的影响越小,受个别异常值的影响更大。

同时,取周一晚上 20:00 和 22:00 的不同预测窗口步长下密度分布热力图的预测结果与真实分布对比,得到如图 4-6 和图 4-7 的结果。可以看出,不同步长的滑动窗口构造的数据集对 SVR 预测结果影响并不是很大,通过对所有预测结果与真实情况做差值,判断预测误差在每个位置的分布情况,具体如图 4-8 所示,颜色越深蓝的地方表示预测的误差越大,越偏白的地方误差越小。相比较而言,步长为 3 的窗口数据集更接近真实情况。

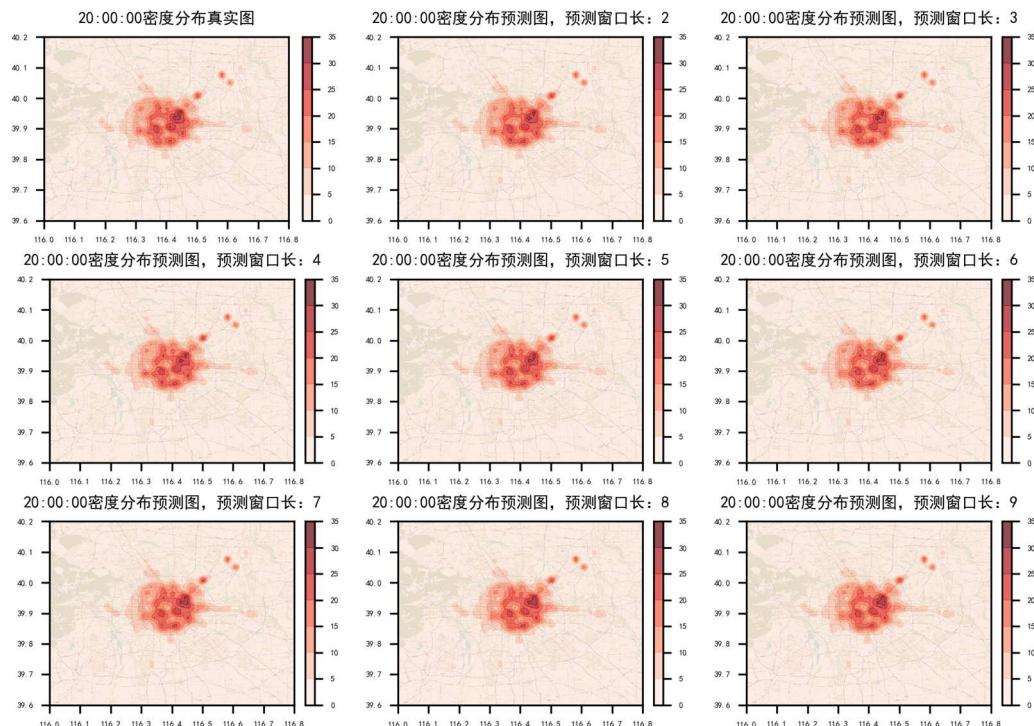


图 4-6 不同预测窗口步长下晚 20 点预测对比图

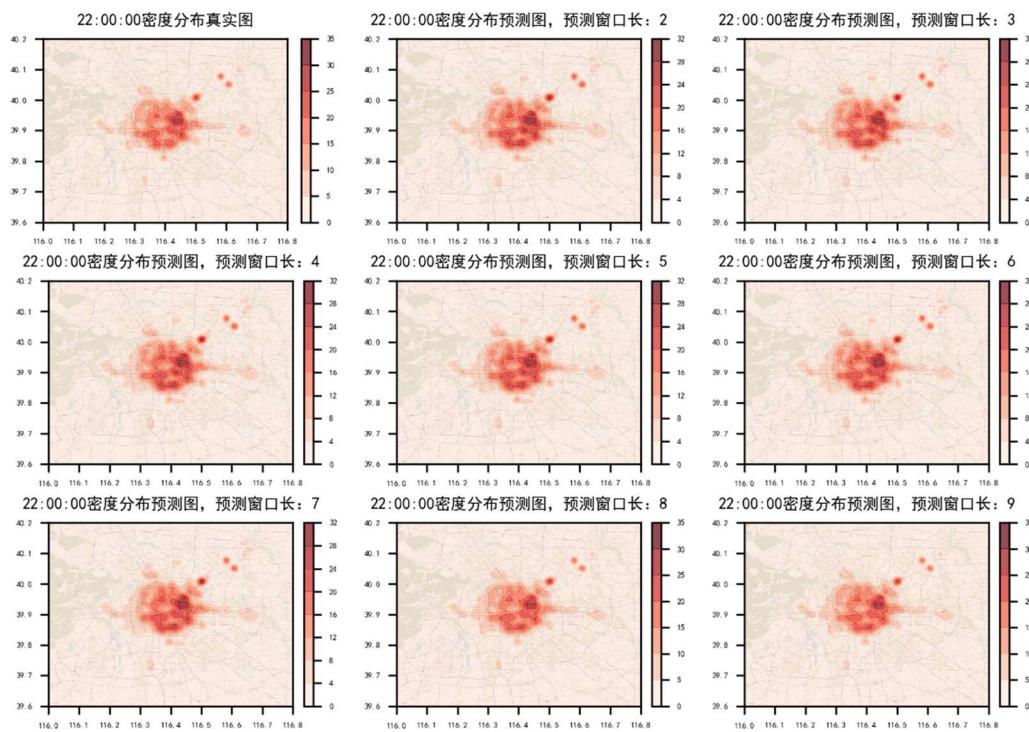


图 4-7 不同预测窗口步长下晚 22 点预测对比图

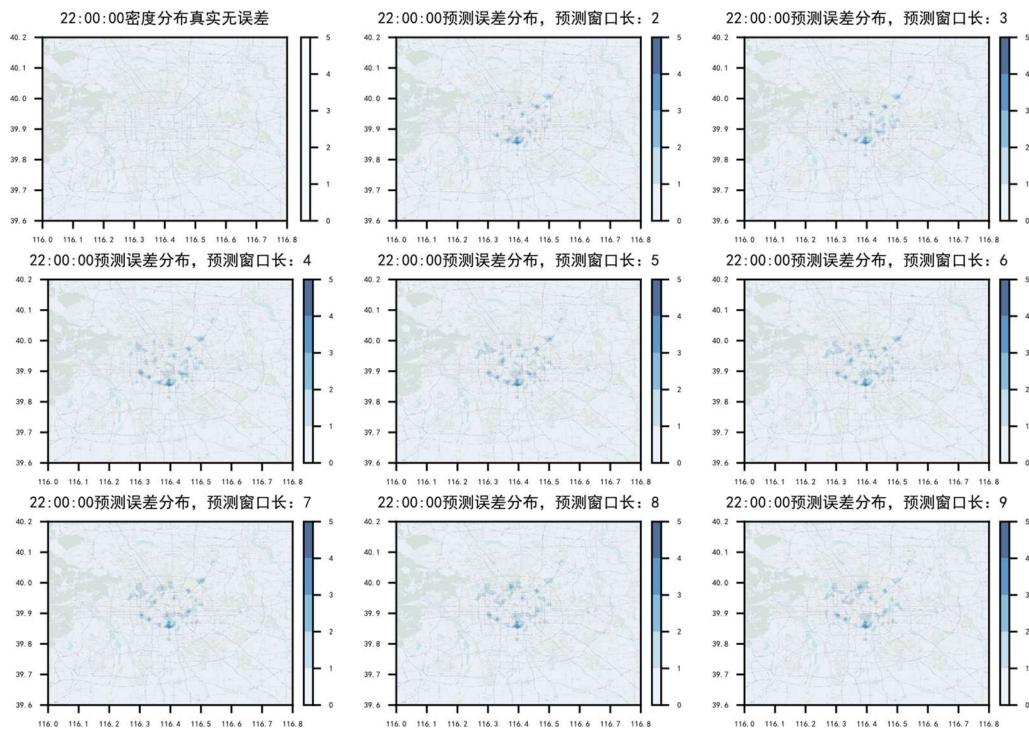


图 4-8 不同预测窗口步长下晚 20 点预测误差分布图

## (2) 最优预测滑动窗口步长下单点密度预测

通过对不同预测滑动窗口步长进行比对,得到最优步长为 3 的预测模型。选取天安门故宫附近(东经  $116.408^{\circ}$ , 北纬  $39.916^{\circ}$ )、北京科技大学西门附近(东经  $116.36^{\circ}$ , 北纬  $39.996^{\circ}$ )和首都机场附近(东经  $116.6^{\circ}$ , 北纬  $40.054^{\circ}$ )三个地点观察局部的密度预测效果,如图 4-9 至 4-11 所示。可以看出,针对局部单点 SVR 的预测效果十分优越,不仅整体上符合预测预期,并且在一些车流密度突变点上也能做到很好的预测反应。

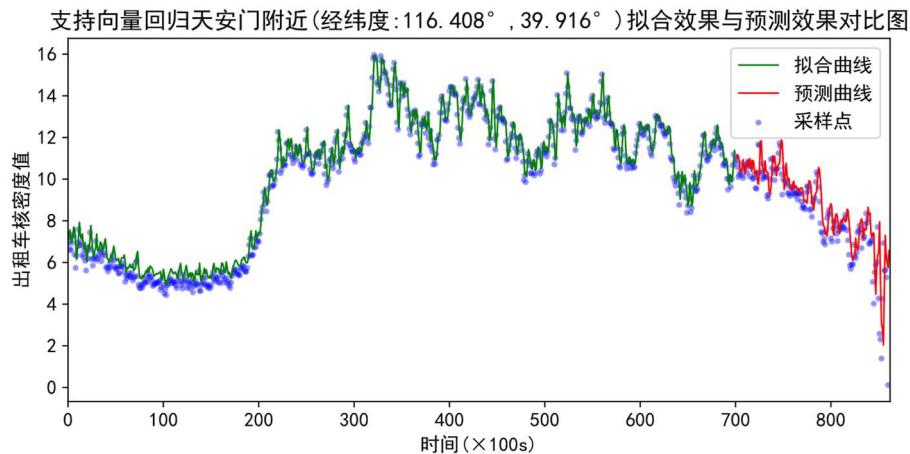


图 4-9 SVR 预测天安门附近拟合与预测对比

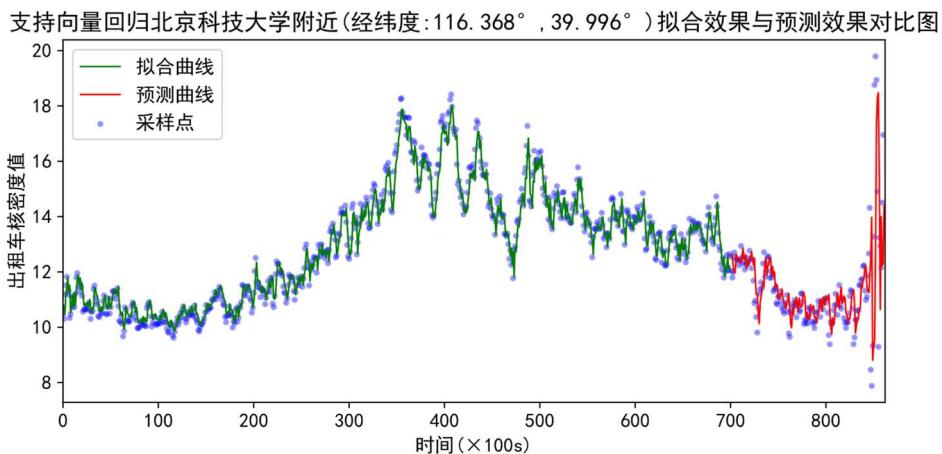


图 4-10 SVR 预测北京科技大学附近拟合与预测对比

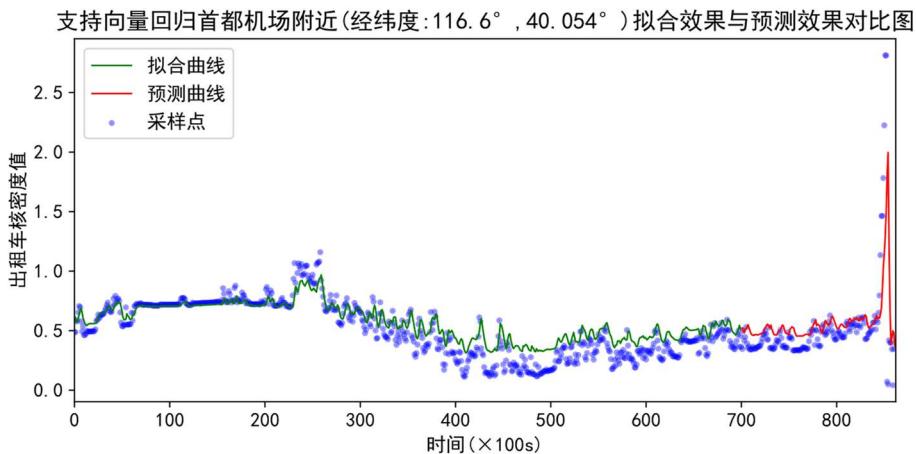


图 4-11 SVR 预测首都机场附近拟合与预测对比

### (3) 最优预测滑动窗口步长下采样时间点热点分布预测对比

最优预测模型即窗口步长为 3 下的情况下针对若干时间点采样，取晚上 20:00, 21:00, 22:00 的时间点进行密度分布热力图预测结果与真实情况对比，如图 4-12 至 4-14 所示。为了更直观的对比预测的误差分布，通过将预测密度分布与真实密度分布做差值处理，以 20:00 为例，如图 4-15 所示。从全局的预测结果来看，SVR 预测效果和实际真实情况只存在细微的差距，很少出现局部热点分裂和局部热点等级下降或上升的情况，预测效果优越。

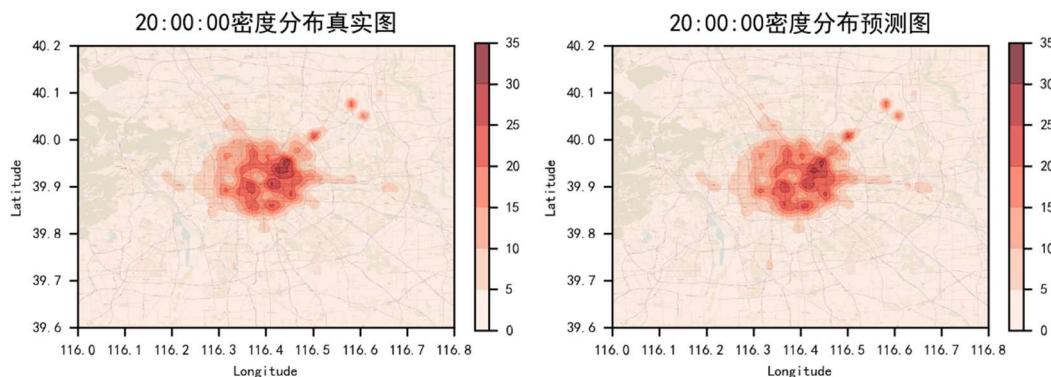


图 4-12 SVR 晚 20 点全局预测结果与真实情况对比

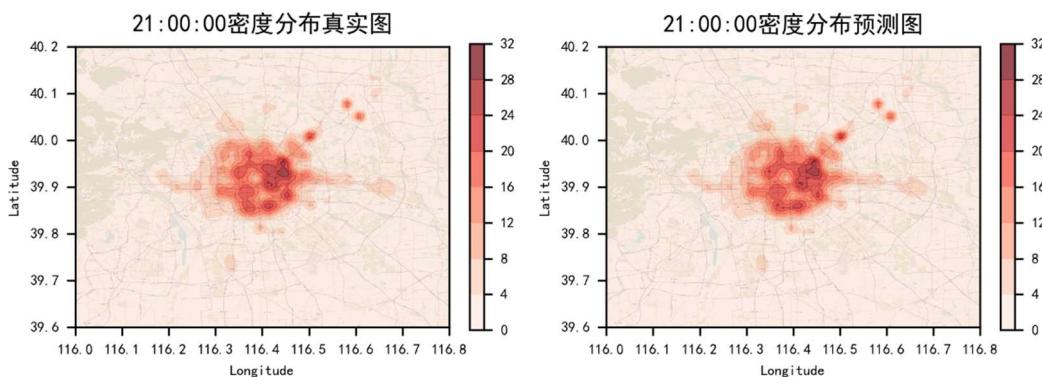


图 4-13 SVR 晚 21 点全局预测结果与真实情况对比

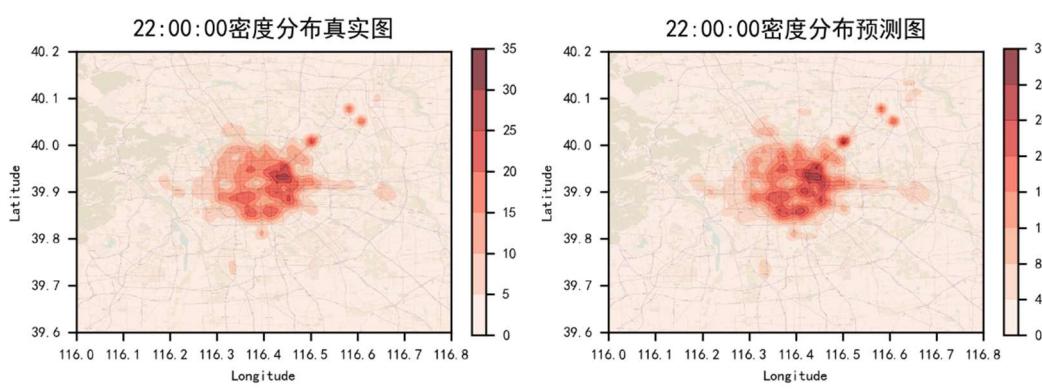


图 4-14 SVR 晚 22 点全局预测结果与真实情况对比

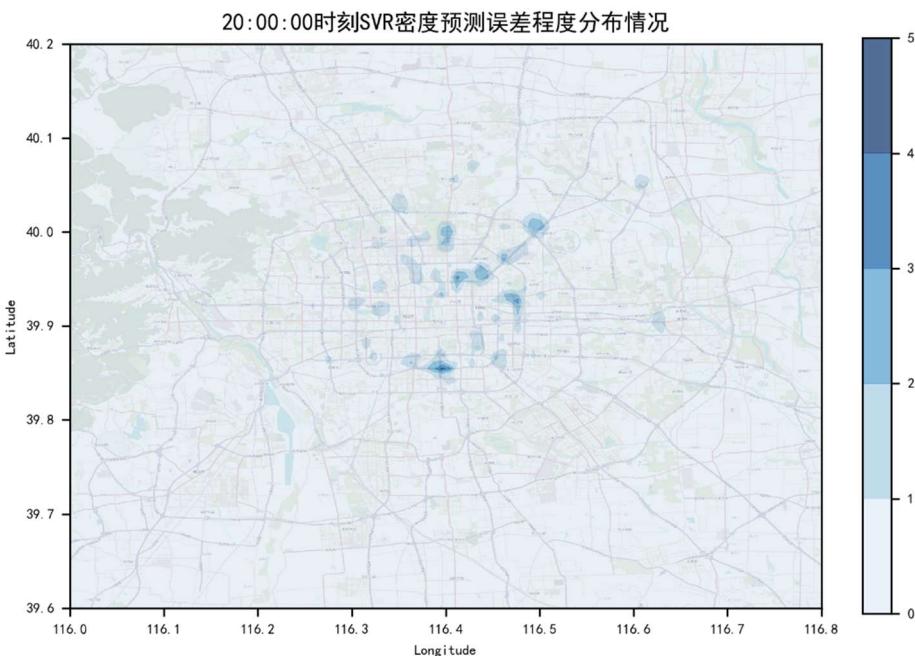


图 4-15 SVR 晚 20 点全局预测误差分布情况

#### 4.4 本章小结

本章是本文的核心探讨内容之一，针对预处理完毕的北京市车辆轨迹数据，提出滑动窗口模型，完成对训练数据集的构造，并对训练数据进行归一化处理让后续模型训练更容易收敛；获取训练数据，利用支持向量回归算法进行模型的预测，得出最佳的预测滑动窗口步长为 3 的结论，并利用最优模型进行局部单点和全局的预测，从实验结果可以看出预测效果符合预期，为后续神经网络预测工作提供流程借鉴和思路。

## 5 基于神经网络的热点预测模型

本章主要研究思路：首先对神经网络的结构、损失函数、前向传播与反向传播等进行介绍说明；其次，使用不同结构的经典多层神经网络模型测试，并对模型进行评估；最后，使用较为先进的长短期记忆模型测试，并对模型进行评估。

基于神经网络的热点预测模型的流程如图 5-1 所示。

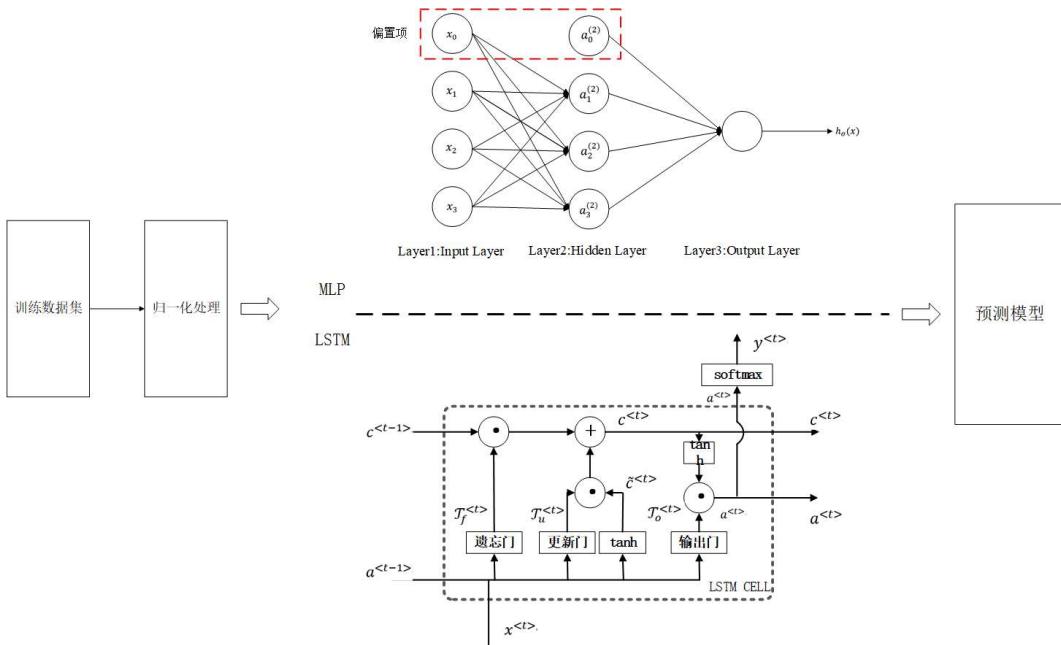


图 5-1 基于神经网络的热点预测流程

### 5.1 神经网络概述

#### 5.1.1 神经网络构成

人工神经网络（ANN, Artificial Neural Network），简称神经网络（NN, Neural Network），借鉴了生物学神经网络的结构，进行相似性建模而成，是一种数学模型或计算模型。现代神经网络一般利用非线性的方法，通常基于统计学方法来使用输入与输出的关系进行数据建模，或用来拟合数据的模式。我们可以认为神经网络就是一个运算模型，它由最基本的神经元单位组成。每个节点代表一种输出规则，称为激活函数。其中，每两个神经元节点间的关系代表一个权重值，这个权重值表示通过该联结关系的信号的加权，与人工神经网络的记忆相似。最终，整个网络的输出结果取决于神经网络中权重的分配，激活函数选择，以及网络连接的结构方式。而神经网络本质性原理

就是对已有的某种算法或映射关系进行逼近计算。

如图 5-2 是结构简单的神经网络，由输入层、输出层、中间隐藏层构成。

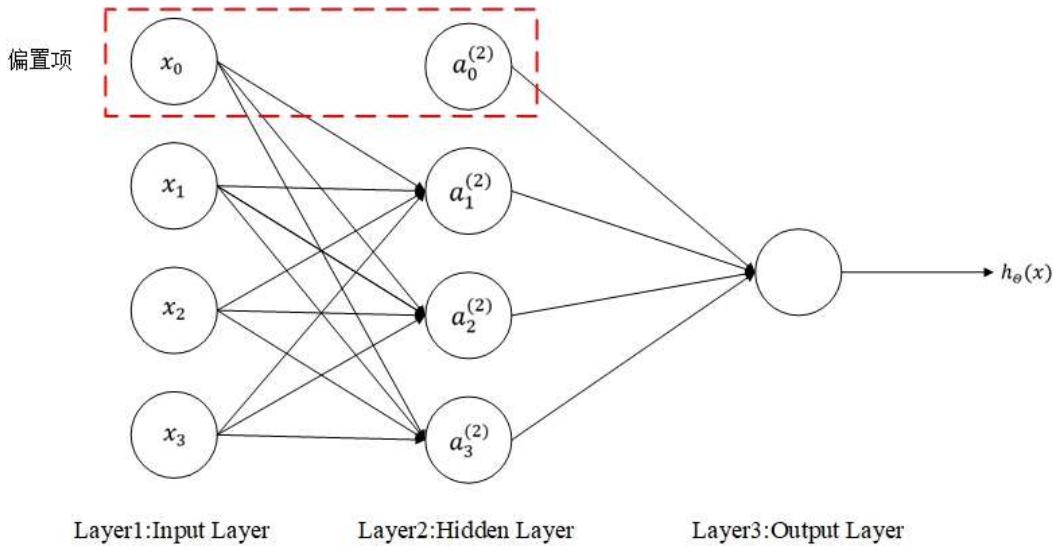


图 5-2 三层基本神经网络结构示意图

其中， $\theta^{(j)}$ 表示从第 $j$ 层映射到第 $j + 1$ 层时的权重关系矩阵， $\theta^{(j)}$ 的尺寸为：大小为 $j + 1$ 层的神经元数乘以第 $j$ 层的神经元数加一的矩阵。 $a_i^{(j)}$ 表示第 $j$ 层的第 $i$ 个激活单元。

依据图 5-2 的模型，中间层激活单元计算结果和最终输出结果分别如下所示：

$$\begin{aligned} a_1^{(2)} &= g(\theta_{10}^{(1)}x_0 + \theta_{11}^{(1)}x_1 + \theta_{12}^{(1)}x_2 + \theta_{13}^{(1)}x_3) \\ a_2^{(2)} &= g(\theta_{20}^{(1)}x_0 + \theta_{21}^{(1)}x_1 + \theta_{22}^{(1)}x_2 + \theta_{23}^{(1)}x_3) \\ a_3^{(2)} &= g(\theta_{30}^{(1)}x_0 + \theta_{31}^{(1)}x_1 + \theta_{32}^{(1)}x_2 + \theta_{33}^{(1)}x_3) \\ h_\theta(x) &= g(\theta_{10}^{(2)}a_0^{(2)} + \theta_{11}^{(2)}a_1^{(2)} + \theta_{12}^{(2)}a_2^{(2)} + \theta_{13}^{(2)}a_3^{(2)}) \end{aligned} \quad (5-1)$$

神经网络的输入一般与数据集的特征数有关，网络的隐藏层数和隐藏层的神经元个数是可以调整的参数，网络的输出则取决于数据集输出的特征数，本文的实验就需要对神经网络隐藏层的层数和神经元个数进行调整，找出适合出租车密度数据的神经网络结构。

确定了神经网络的结构后，需要通过神经网络的算法来不断迭代求得更优的参数 $\theta$ ，使得建立输入和输出的模型更加接近实际问题，同时，让预测的输出值与实际的输出值更加接近，这就需要模型的损失函数来衡量，同时神经网络的预测输出值通过前向传播来实现，前向传播的过程来计算损失函数，通过反向传播来调整网络参数，直到达到一个不错的预期效果。

多层感知器(MLP, Multi-Layer Perceptron)模型是一种经典的神经网络，

通过前向结构建立输入到输出的映射关系。除了输入输出层，中间可以有很多隐藏层，即神经网络线性回归预测常见的模型。

### 5.1.2 损失函数与正则化

#### (1) 损失函数 (Loss Function)

神经网络的结构确定后，需要通过前向传播，一层一层的计算得到最终的输出预测值，计算损失函数，衡量模型的性能。对于线性回归问题，损失函数计算公式一般利用最小二乘法，具体表达式为：

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 \quad (5-2)$$

其中， $J(\theta)$ 是损失函数， $m$ 是数据集输入数据量， $h_\theta(x^{(i)})$ 是输入数据 $x^{(i)}$ 的预测值， $y^{(i)}$ 是对应的真实值。

而对于逻辑回归，其损失函数表示为：

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))] \quad (5-3)$$

损失函数的值越小代表越好，它意味着预测值与真实值更靠近，预测效果更佳。最小化损失函数则需要借助神经网络的反向传播算法进行实现。

#### (2) 正则化 (Regularization)

线性回归和逻辑回归的应用能处理很多难题，但将它们应用到某些特定的机器学习场合时，可能会出现过拟合(over-fitting)的现象，从而会导致模型效果很差。对于出现过拟合问题，一般有两种解决办法：

- 尝试舍弃一些不能帮助我们去正确预测的属性。可以是手工选择保留，或借助一些专门的算法来择优，例如：PCA (Principal Component Analysis)。
- 正则化。保留所有的特征，但是减少参数的大小。

正则化就是对一些特定的特征给予一定的惩罚，调节正则化参数可以使模型向更优的方向上靠近。例如线性回归，正则化后的损失函数为：

$$J(\theta) = \frac{1}{2m} [\sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2] \quad (5-4)$$

其中， $\lambda$ 为正则化参数 (Regularization Parameter)。

同理，正则化的逻辑回归损失函数表示为：

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m [-y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \quad (5-5)$$

加入正则化方法的神经网络能够一定程度克服过拟合问题，但是引入了

新的超参数 $\lambda$ ,  $\lambda$ 的取值很大程度上影响到模型的性能, 该参数设置不好可能让模型出现过拟合或欠拟合现象, 如果参数 $\lambda$ 过大, 会导致参数都最小化了, 出现欠拟合现象。因此,  $\lambda$ 的选择需要实际的参数调试。

### 5.1.3 前向传播与反向传播

神经网络训练参数需要做梯度下降来优化参数, 具体的过程由前向传播和反向传播来实现。

#### (1) 前向传播 (Forward Propagation)

前向传播计算每一个神经元节点的值, 一层一层最终计算出损失函数的值。以图 5-2 为例, 正向传播过程可以描述为:

$$\begin{aligned} z^{[1]} &= W^{[1]}x + b^{[1]} \\ a^{[1]} &= g^{[1]}(z^{[1]}) \\ z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} \\ a^{[2]} &= g^{[2]}(z^{[2]}) \\ h_{\theta}(x) &= a^{[2]} \end{aligned} \quad (5-6)$$

其中,  $W^{[i]}$ 表示第*i*层神经网络权重参数,  $b^{[i]}$ 表示第*i*层神经网络偏置参数, 它们组成神经网络模型参数 $\theta^{[i]}$ 。 $x$ 表示模型的输入特征向量,  $z^{[i]}$ 是第*i*层神经网络单元处输出,  $g^{[i]}$ 是第*i*层神经网络使用的激活函数,  $a^{[i]}$ 是第*i*层神经网络节点通过激活函数的输出,  $h_{\theta}(x)$ 是神经网络的预测输出。

#### (2) 反向传播 (Back Propagation)

反向传播是根据损失函数对每层神经网络逆向求导, 利用梯度下降的思维进行最优化神经网络参数的寻找, 具体过程可以描述为:

$$\begin{aligned} dz^{[2]} &= A^{[2]} - Y \\ dW^{[2]} &= \frac{1}{m} dz^{[2]} A^{[1]T} \\ db^{[2]} &= \frac{1}{m} sum(z^{[2]}) \\ dz^{[1]} &= dW^{[2]T} dz^{[2]} * g^{[1]'}(z^{[1]}) \\ dW^{[1]} &= \frac{1}{m} dz^{[1]} x^T \\ db^{[1]} &= \frac{1}{m} sum(z^{[1]}) \end{aligned} \quad (5-7)$$

其中,  $Y$ 输出的真实值,  $A^{[i]}$ 是数据通过第*i*层神经网络节点的输出向量,  $dz^{[i]}$ 是第*i*层神经网络节点输出值的导数,  $dW^{[i]}$ 是第*i*层神经网络权重参数的导数,  $db^{[i]}$ 是第*i*层神经网络偏置参数的导数,  $sum(\cdot)$ 是求和函数,  $g^{[1]}'$ 是第

一层神经网络激活函数的求导。

### 5.1.4 激活函数

在神经网络中，我们通常引入激活函数（Activation Function）将非线性属性引入网络，来满足很多操作的非线性要求。激活函数应当具有可微，非线性，单调等性质。这样便于神经网络中反向传播进行求导，以下是几种常见的激活函数<sup>[49]</sup>。

#### (1) sigmoid 函数

该函数一般用于网络中的隐藏层和输出层，通过非线性压缩，将输出数据映射到 (0, 1) 区间，sigmoid 函数公式为：

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (5-8)$$

sigmoid 函数分布如图 5-3 所示。

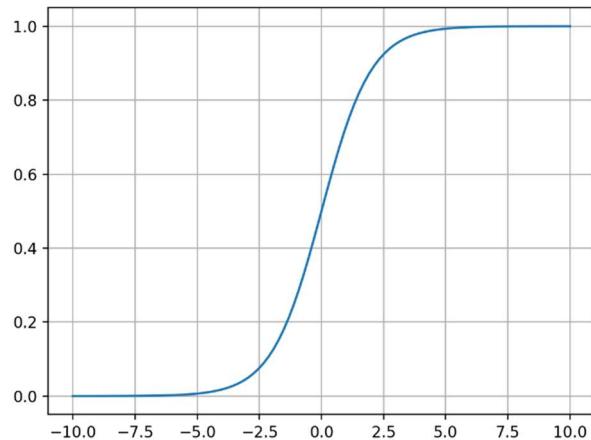


图 5-3 sigmoid 函数分布

sigmoid 函数将输出映射到 (0, 1) 区间，且其求导结果可用自己表示，但随着网络层数的不断变深，该函数求导输出会不断趋于 0，也就是所谓的梯度消失。除此之外，sigmoid 函数均值不是 0，若输入的符号是同向的，那么，函数输出值在求导后也是同向的，这使得网络收敛速度比较慢。

#### (2) tanh 函数

tanh 函数是有些类似于 sigmoid 函数，其公式表示为：

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5-9)$$

tanh 函数分布如图 5-4 所示。

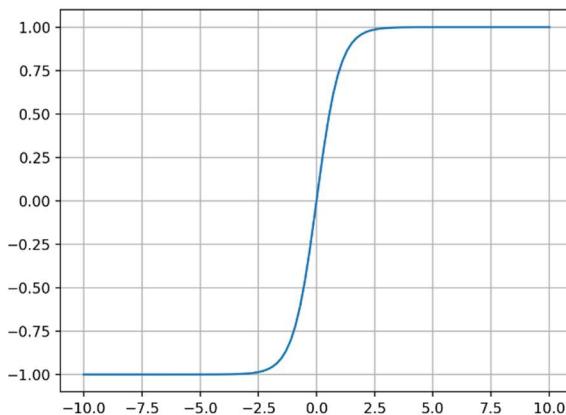


图 5-4 tanh 函数分布

该函数克服了均值不为 0 的缺点，将输出的数据映射的 (-1, 1) 区间，符合零均值分布，收敛速度得到了一定的改善。其中，tanh 函数可用 sigmoid 函数表示，因此同 sigmoid 函数相似，网络的深入会产生梯度消失现象。

### (3) relu 函数

relu 函数针对梯度消失问题对前两种方法做了一些改进，其表达式为：

$$\text{Relu}(x) = \max(0, x) \quad (5-10)$$

relu 函数分布如图 5-5 所示。

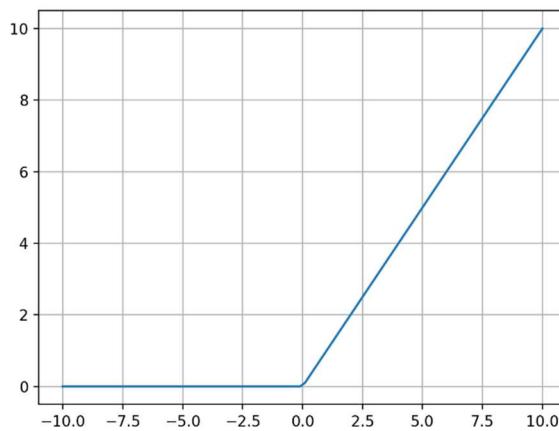


图 5-5 relu 函数分布

可以看出，该函数是一个分段函数，输入为小于等于 0 时，是恒定的 0 常数输出，否则，relu 的输入与输出值呈线性关系。该函数分布不是零均值，但不会出现收敛速度较慢的情况，主要由于 relu 函数不存在指数运算，它的导数值为常数，计算量小，求导迅速，因此在反向传播求导的过程中避免了

梯度消失的问题。

## 5.2 优化方法

机器学习的方法是一个非常依赖经验的过程，随着迭代数量的增加，需要训练更多对比模型，才能找到合适训练模型，而优化算法能够更快速地训练模型。下面是几种常见的优化方法。

### (1) 梯度下降法

梯度下降法 (GD, Gradient Descent) 是最常见的算法，指的是让参数沿着梯度的反方向更新，同时梯度的方向就是函数变化最快的方向，从而以比较快的速度减小损失函数的值。重复求取梯度，最后就能得到局部最优值。

梯度下降公式如下：

$$\theta = \theta - \alpha \nabla_{\theta} J(\theta) \quad (5-11)$$

其中， $\theta$ 为网络模型参数， $\alpha$ 为神经网络的学习率， $J(\theta)$ 是损失函数。

### (2) 动量梯度下降法

动量梯度下降法 (Momentum) 设定一个较小梯度初始值，通过赋予一定的加速度，来让改值不断增加趋于一个稳定状态，具体参数更新公式如下：

$$\begin{aligned} v_{d\theta} &= \beta v_{d\theta} + (1 - \beta)d\theta \\ \theta &= \theta - \alpha v_{d\theta} \end{aligned} \quad (5-12)$$

其中， $\theta$ 为网络模型参数， $\alpha$ 为神经网络的学习率， $v_{d\theta}$ 是指数加权平均数， $\beta$ 是加速度超参数。

动量梯度下降法的梯度更新通过加速的累计过程，当梯度下降的方向与加速度的方向一致时，动量项的值增加，反之相应的动量项会减少，从而达到加快网络收敛的目的。但是，动量梯度下降的方向在最坏情况下可能错过最优解。

### (3) RMSprop

RMSprop (Root Mean Square prop) 算法，不会出现因梯度累积后而导致随着网络层数深入而学习率变得越来越小的现象，具体参数更新如下：

$$\begin{aligned} S_{d\theta} &= \beta S_{d\theta} + (1 - \beta)d\theta^2 \\ \theta &= \theta - \alpha \frac{d\theta}{\sqrt{S_{d\theta}}} \end{aligned} \quad (5-13)$$

其中， $\theta$ 为网络参数， $\alpha$ 为神经网络的学习率， $S_{d\theta}$ 是指数加权平均数， $\beta$ 是超参数。

### (4) Adam

Adam (Adaptive Moment Estimation) 算法<sup>[52]</sup> 借鉴了 Momentum 算法以

及 RMSprop 算法，具体参数更新如下：

首先初始化中间参数：

$$v_{d\theta} = 0, S_{d\theta} = 0$$

在第 t 次迭代中执行：

$$\begin{aligned} v_{d\theta} &= \beta_1 v_{d\theta} + (1 - \beta_1) d\theta \\ S_{d\theta} &= \beta_2 S_{d\theta} + (1 - \beta_2) d\theta^2 \\ v_{d\theta}^{correct} &= \frac{v_{d\theta}}{1 - \beta_1^t} \\ S_{d\theta}^{correct} &= \frac{S_{d\theta}}{1 - \beta_2^t} \\ \theta &= \theta - \alpha \frac{v_{d\theta}^{correct}}{\sqrt{S_{d\theta}^{correct} + \epsilon}} \end{aligned} \quad (5-14)$$

其中， $\theta$  为网络参数， $\alpha$  为神经网络的学习率， $v_{d\theta}$  和  $S_{d\theta}$  是指数加权平均数， $\beta_1$  和  $\beta_2$  是超参数。 $v_{d\theta}^{correct}$  和  $S_{d\theta}^{correct}$  是对  $v_{d\theta}$  和  $S_{d\theta}$  的修正， $\epsilon$  为超参数。一般地， $\beta_1$  设置为 0.9， $\beta_2$  设置为 0.999， $\epsilon$  设置为  $10^{-8}$ 。

Adam 算法结合了 RMSprop 算法与 Momentum 算法的优点，能够处理梯度稀疏和非平稳问题。

### 5.3 循环神经网络

#### (1) 循环神经网络

循环神经网络 (RNN, Recurrent Neural Network) 是一种特殊的神经网络，序列化是它最明显的特征，通过在网络中加入序列的概念，输入的数据在序列上依次通过所有循环单元，达到训练模型的效果<sup>[39]</sup>。RNN 按链式结构的连接。其处理序列数据的性能很高，常用于自然语言处理 (NLP, Natural Language Processing) 领域，其结构如图 5-6 所示。

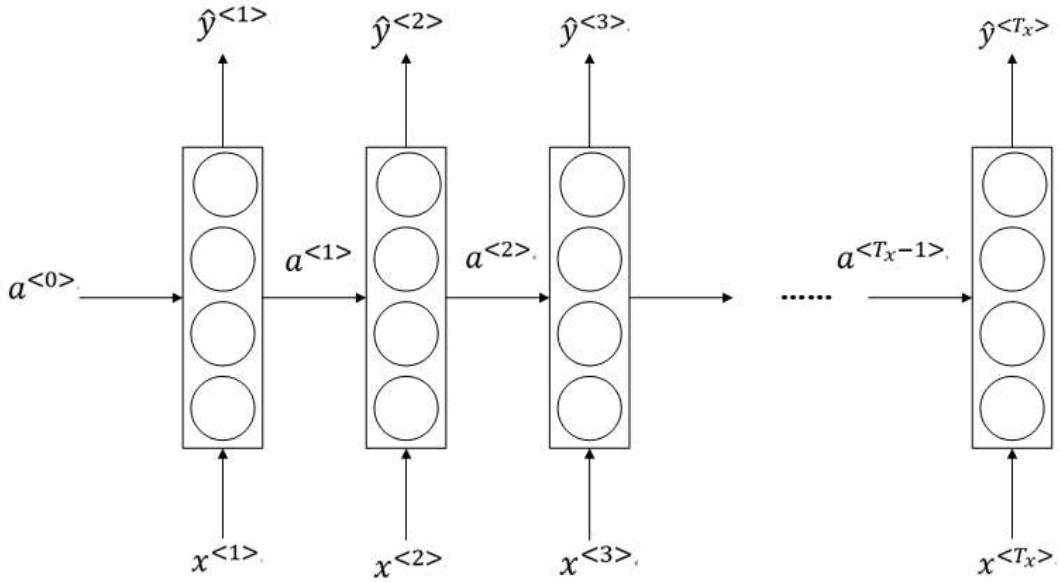


图 5-6 循环神经网络序列化示意图

一般情况下，在 $t$ 时刻 RNN 的前向传播可以表示为：

$$\begin{aligned} a^{<t>} &= g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \\ \hat{y}^{<t>} &= g_2(W_{ya}a^{<t>} + b_y) \end{aligned} \quad (5-15)$$

其中， $a^{<t>}$ 表示第 $t$ 个时间序列的输出， $g_1$ 和 $g_2$ 表示激活函数， $W_{aa}$ 、 $W_{ax}$ 和 $W_{ya}$ 表示对应的权重系数， $b_a$ 和 $b_y$ 表示偏置参数。RNN 的神经网络的结构如图 5-7 所示。

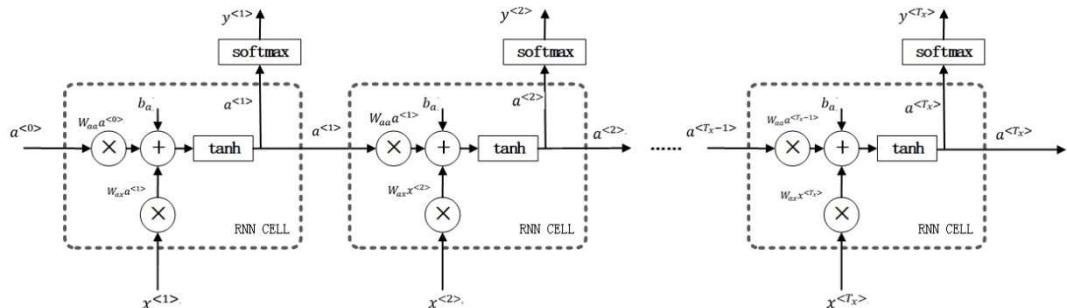


图 5-7 RNN 网络内部结构

其反向传播过程如图 5-8 所示。

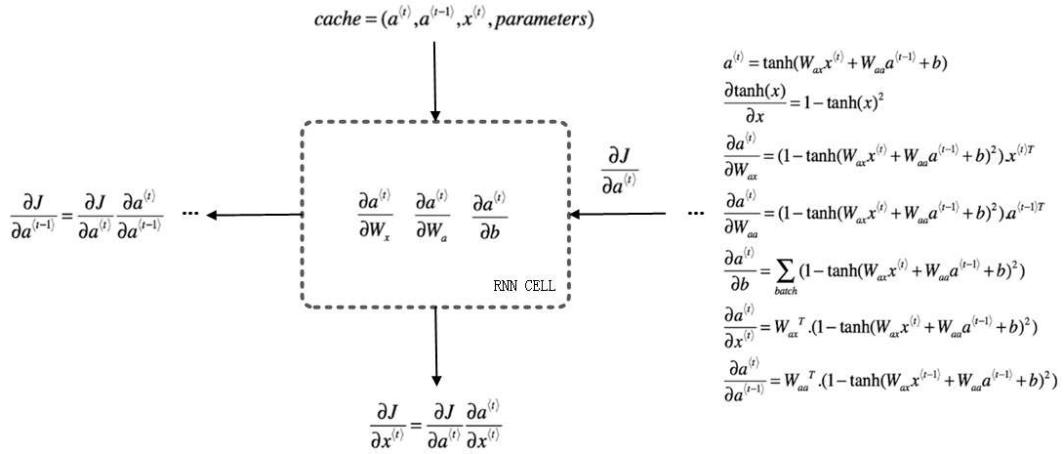


图 5-8 RNN 反向传播过程

## (2) 长短期记忆模型

长短期记忆 (LSTM, Long Short Term Memory) 模型是变种的 RNN 网络，拥有比一般 RNN 模型更好的性能，是为了解决一般的 RNN 存在的一些局限性设计出来的<sup>[41]</sup>，如图 5-9 是 LSTM 的前向传播过程。

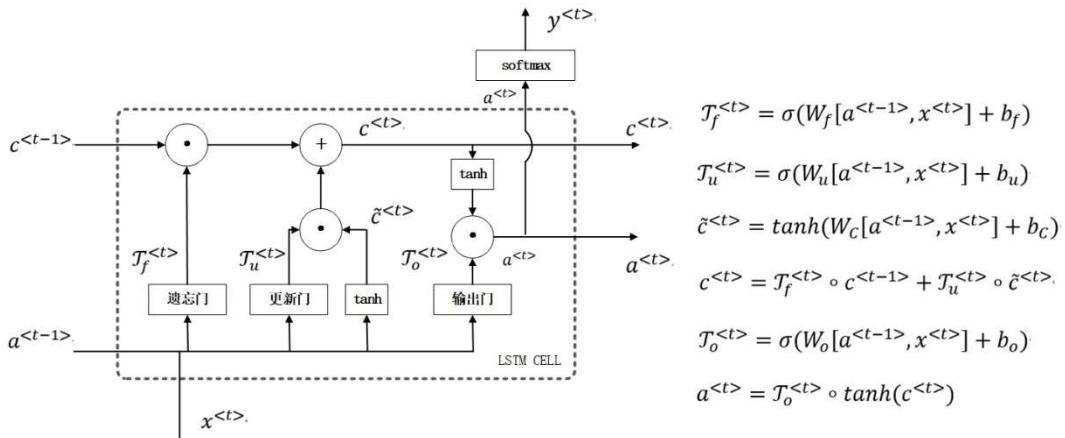


图 5-9 LSTM 前向传播过程

前向传播的具体描述如下：

$$\begin{aligned} T_f^{<t>} &= \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f) \\ T_u^{<t>} &= \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u) \\ \tilde{c}^{<t>} &= \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \\ c^{<t>} &= T_f^{<t>} \circ c^{<t-1>} + T_u^{<t>} \circ \tilde{c}^{<t>} \\ T_o^{<t>} &= \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o) \\ a^{<t>} &= T_o^{<t>} \circ \tanh(c^{<t>}) \end{aligned} \tag{5-16}$$

其中， $T_f^{<t>}$  表示遗忘门， $T_u^{<t>}$  表示更新门， $T_o^{<t>}$  表示输出门， $\sigma$  表示激活函数， $W_f$ 、 $W_u$ 、 $W_c$  和  $W_o$  表示对应的权重系数， $b_f$ 、 $b_u$ 、 $b_c$  和  $b_o$  表示偏

置参数。 $c^{<t>}$ 为记忆细胞， $\tilde{c}^{<t>}$ 是代替记忆细胞的候选值， $a^{<t>}$ 是一个神经元的输出值。

## 5.4 MLP 热点预测

本节利用 MLP 多层感知器模型来实现对车流密度和热点分布的估计，首先配置实验环境，调整模型参数，确定评价标准，最后进行本节的预测实验。

### 5.4.1 实验环境

本文的实验运行环境配置如下：

- (1) 处理器：Intel® Core(TM) i7-9700 CPU @ 3.00 GHz
- (2) 内存(RAM)：16GB
- (3) 操作系统：Windows 64 位
- (4) 编程语言：Python3.7
- (5) 机器学习框架：scikit-learn 0.24.1
- (6) 工具包与开放库：Pandas、Numpy、Matplotlib 等

### 5.4.2 模型参数设置及评价标准

- (1) 神经网络参数设置

**表格 5-1 MLP 预测参数符号与赋值**

参数	符号	取值
优化方法	-	SGD
正则化系数	$\lambda$	0.0001
学习率	$\alpha$	0.01
激活函数	$Activation(\cdot)$	Relu
最大迭代次数	$n$	1000
神经网络层数	Layers	2-3
隐藏层神经元个数	Num	1-30

通过神经网络进行回归预测，需要设置以下几个参数，即神经网络层数，每层神经网络的个数，正则化参数 $\lambda$ ，优化方法等参数的设置。其中神经网络

隐藏层层数和神经元个数是需要实验对比的，设置一组可选范围，具体参数设置如表 5-1 所示。

### (2) 训练集和测试集的划分

训练集和测试集的划分与 4.3.2 节一致，划分后可以得到训练集是尺寸为  $10000 \times 700 \times (L + 1)$  的矩阵数据，测试集是尺寸为  $10000 \times (164 - L - 1) \times (L + 1)$  的矩阵数据。

### (3) 评价指标

评价指标与支持向量回归预测相同，参考 4.3.3 节，使用 MAE，MSE 和  $R^2$  进行模型评价。

## 5.4.3 实验结果与分析

基于神经网络回归的热点预测模型的实验思路如下：将处理好的北京市一周内的数据集和神经网络参数确定后。首先，对比不同预测滑动窗口步长下的预测效果，利用预测滑动窗口步长最优的数据集，设置两层神经网络（即一层隐藏层），对隐藏层不同神经元个数设置训练模型，对比它们的预测效果并进行指标分析；其次，利用预测滑动窗口步长最优的数据集，设置三层神经网络（即两层隐藏层），对隐藏层不同神经元个数设置训练模型，对比它们的预测效果并进行指标分析；最后，选取最优模型预测北京市三个地点的密度变化曲线绘制分析，以及选取三个时刻预测结果与真实热点分布对比分析。

### (1) 不同预测滑动窗口步长下的预测效果

同 SVR 预测相同，实验采用预测滑动窗口步长组：2，3，4，5，6，7，8，9。测试集评价指标的均值对比如表 5-2 所示。

**表格 5-2 MLP 不同窗口步长下评价指标均值对比**

窗口步长	MSE	MAE	$R^2$
2	1.2593	0.4616	0.1554
3	1.1724	0.3679	0.2559
4	1.2473	<b>0.3596</b>	<b>0.3097</b>
5	1.4757	0.5086	0.0661
6	<b>0.8487</b>	0.3670	0.1854
7	1.1860	0.4130	0.0667
8	2.0475	0.5388	0.0885
9	1.7051	0.5391	0.0030

从表 5-2 中可以看出, 不同预测滑动窗口的步长对 MLP 预测的效果影响比较大, 综合三项指标来看, 步长为 4 的预测窗口的效果最佳, MAE 误差最小, 且  $R^2$  值得分最高, MSE 误差相对较小。

### (2) 最优预测滑动窗口步长情况下单隐藏层神经网络模型分析

通过不同预测滑动窗口步长对比分析得到最优预测窗口步长为 4, 对于一层隐藏层的 MLP 网络, 实验采用神经网络隐藏层神经元个数包括: 1, 2, 3, ……, 30。实验原始数据采用周一全天车辆估计数据, 通过基于核密度估计算法的城市车流密度提取模型, 再通过不同步长的设置, 根据 4.1.2 节训练数据进行构造, 生成每组需要的训练数据。并对训练集和测试集进行划分, 通过单隐藏层 MLP 模型学习得到不同隐藏层神经元个数下所有采样点时间序列的评价指标的均值对比, 取性能最高的前 8 项如表 5-3 所示。

**表格 5-3 MLP 单隐藏层不同神经元个数性能对比 (取前 8 项)**

隐藏层神经元个数	MSE	MAE	$R^2$
<b>18</b>	<b>0.198314</b>	<b>0.186148</b>	<b>0.389775</b>
23	1.107259	0.357752	0.313324
19	0.658222	0.300784	0.196428
22	1.243477	0.379973	0.306154
24	1.285818	0.406511	0.274814
27	1.247568	0.395029	0.272793
28	1.282963	0.410393	0.252896
25	1.219264	0.396595	0.244509

从表中可以看出, 单隐藏层下的神经元个数为 18 时, MSE, MAE 和  $R^2$  都取得了最好的效果。同时, 单隐藏层神经元个数为 18 的网络结构性能较大幅度超过了其他测试的网络结构。

### (3) 最优预测滑动窗口步长情况下双隐藏层神经网络模型分析

对于双隐藏层神经网络模型, 需要对两个隐藏层神经元个数分别设置, 对于隐藏层 1 给出神经元个数范围为: 1 到 30, 对于隐藏层 2 给出神经元个数范围为: 1 到 30。对不同模型对比测试, 取性能前 10 的网络结构指标如表 5-4 所示。

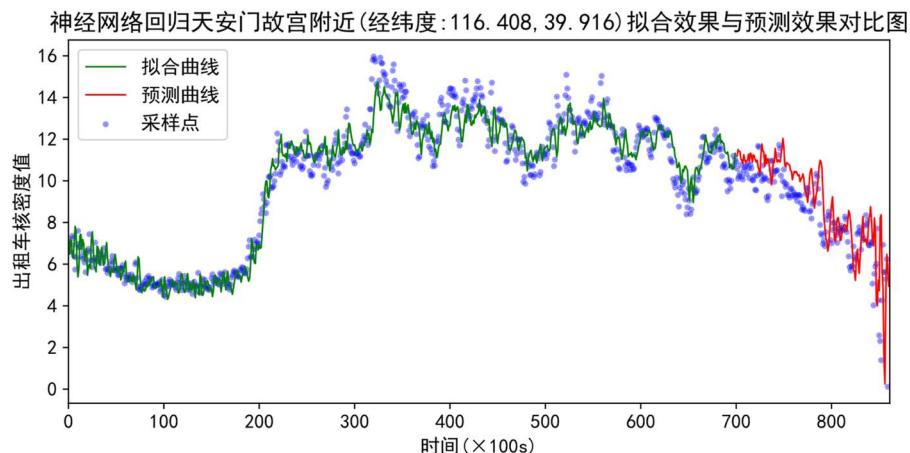
从表中可以看出, 双隐藏层下的隐藏层 1 神经元个数为 5, 隐藏层 2 神经元个数为 11 时, 三项指标综合取得最优效果, 但相对于单隐藏层神经网络结构效果欠佳。

**表格 5-4 MLP 双隐藏层不同神经元个数性能对比（取前 10 项）**

隐藏层 1 神经元数	隐藏层 2 神经元数	MSE	MAE	$R^2$
5	9	0.877985	0.334246	<b>0.220625</b>
5	11	<b>0.799377</b>	<b>0.322412</b>	0.202452
5	10	0.856919	0.333028	0.185805
12	17	1.185922	0.401819	0.184377
12	18	1.213516	0.408257	0.180598
5	8	1.055577	0.378129	0.178681
12	16	1.206223	0.406419	0.177992
12	15	1.225503	0.410293	0.177404
5	12	1.096969	0.385425	0.177065
12	19	1.205546	0.406863	0.176943

**(4) 最优模型下单点密度预测曲线**

通过不同神经网络的比对，得到最优网络结构为：单隐藏层神经元数目为 18 的结构。选取天安门故宫附近（东经  $116.408^\circ$ ，北纬  $39.916^\circ$ ）、北京科技大学西门附近（东经  $116.36^\circ$ ，北纬  $39.996^\circ$ ）和首都机场附近（东经  $116.6^\circ$ ，北纬  $40.054^\circ$ ）三个地点观察局部的密度预测效果，如图 5-10 至 5-12 所示。可以看出，对于局部单点的 MLP 预测效果还是不错的，但在某些区间预测的效果偏差有些大，预测效果相比于 SVR 预测有肉眼可见的不足。

**图 5-10 MLP 预测天安门附近拟合与预测对比**

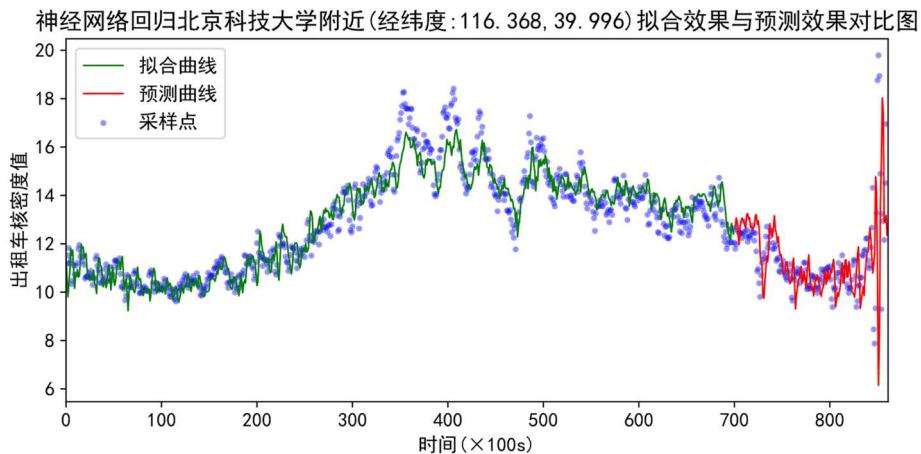


图 5-11 MLP 预测北京科技大学附近拟合与预测对比

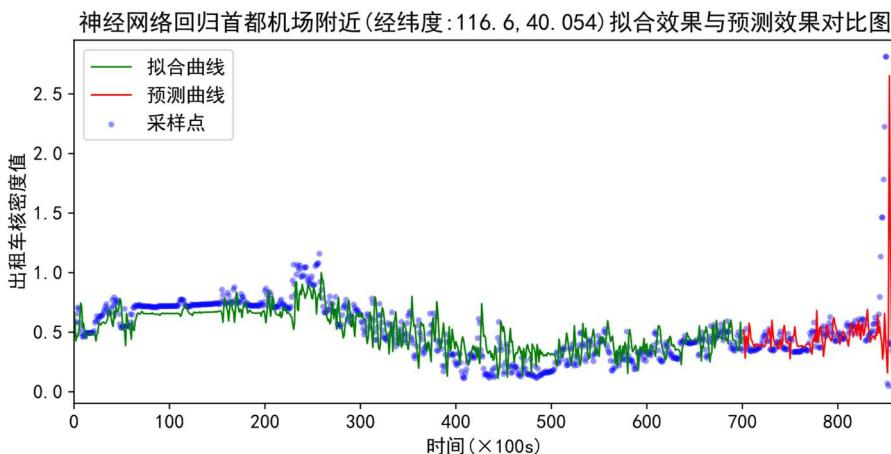


图 5-12 MLP 预测首都机场附近拟合与预测对比

### (5) 最优模型下预测效果与真实效果对比分析

最优的神经网络预测模型下进行若干时间点采样，取晚上 20:00, 21:00, 22:00 的时间点进行密度分布热力图预测结果与真实情况对比，如图 5-13 至 5-15 所示。为了更直观的对比预测的误差分布，通过将预测密度分布与真实密度分布做差值处理，以 20:00 为例，如图 5-16 所示。可以看出，从全局的预测来看，多层神经网络预测在局部的热点上出现层级的下降和上升，预测效果有一定的偏差。

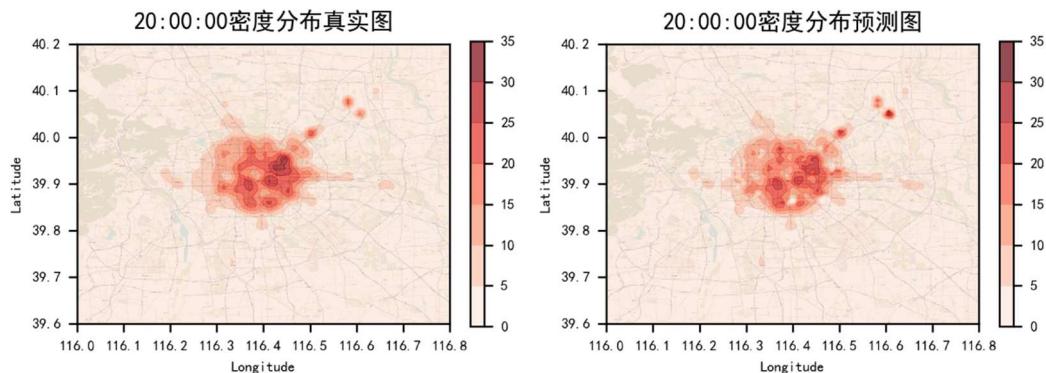


图 5-13 MLP 晚 20 点全局预测结果与真实情况对比

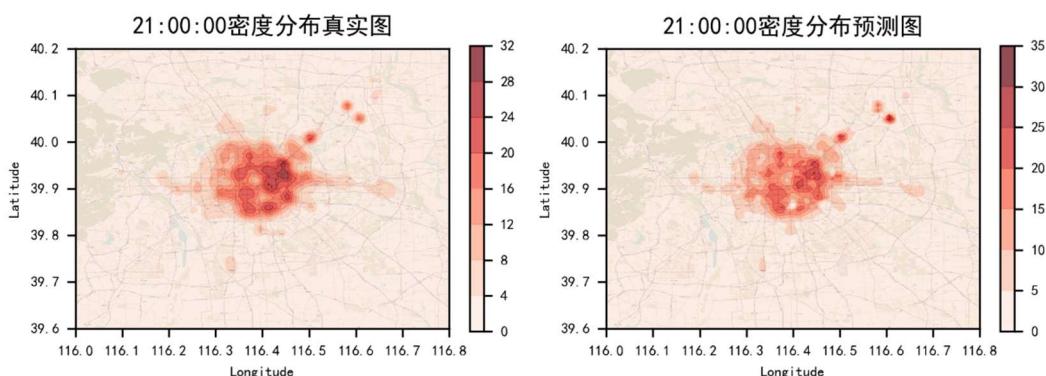


图 5-14 MLP 晚 21 点全局预测结果与真实情况对比

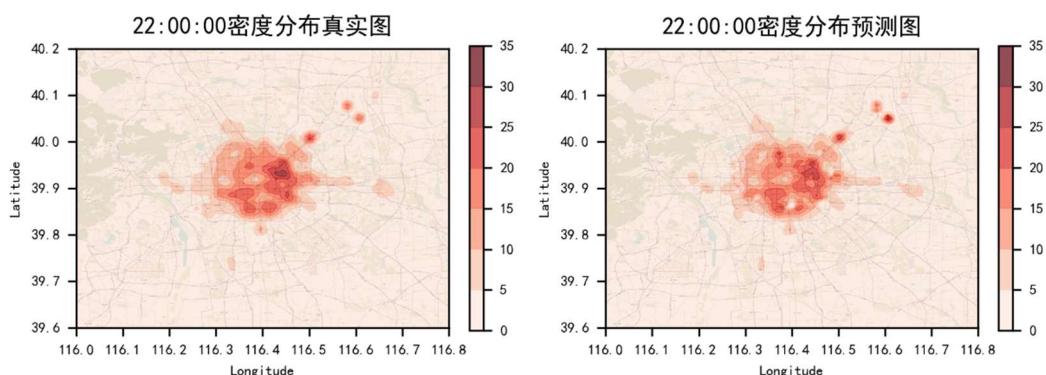


图 5-15 MLP 晚 22 点全局预测结果与真实情况对比

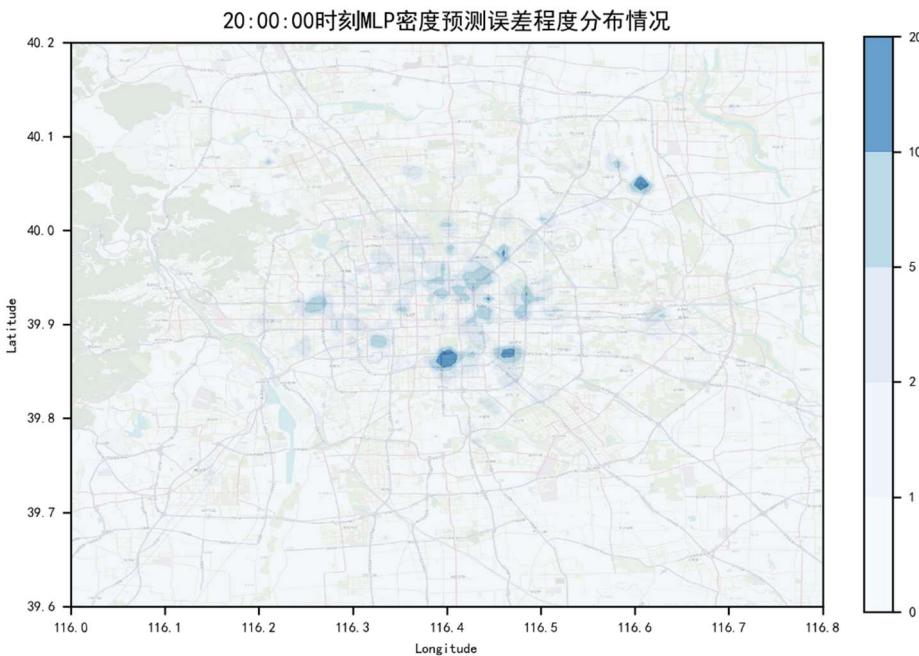


图 5-16 MLP 晚 20 点全局预测误差分布情况

## 5.5 LSTM 热点预测

本节利用长短期记忆模型来实现对车流密度和热点分布的估计，首先配置实验环境，调整模型参数，确定评价标准，最后进行本节的预测实验。

### 5.5.1 实验环境

本文的实验运行环境配置如下：

- (1) 处理器：Intel® Core(TM) i7-9700 CPU @ 3.00 GHz
- (2) 内存(RAM)：16GB
- (3) 操作系统：Windows 64 位
- (4) 编程语言：Python3.7
- (5) 机器学习框架：TensorFlow 2.3.0、Keras 2.4.3
- (6) 工具包与开放库：Pandas、Numpy、Matplotlib 等

### 5.5.2 参数设置

- (1) 模型参数设置

使用 LSTM 神经网络进行回归预测，采用相对简单的网络结构，输入数据采用滑动窗口步长为 3 的训练数据，输入特征为车流密度信息的时间序列，采用 units (隐藏层神经元个数) 为 100 的 LSTM 层，其中 return\_sequences

默认设置为 False (是否返回全部序列), 经过 dropout 为 0.2 (随机剔除 20% 的神经元)的 Dropout 层, 最后经过输出特征为 1 的全连接层得到预测结果。具体网络结果如图 5-17 所示。

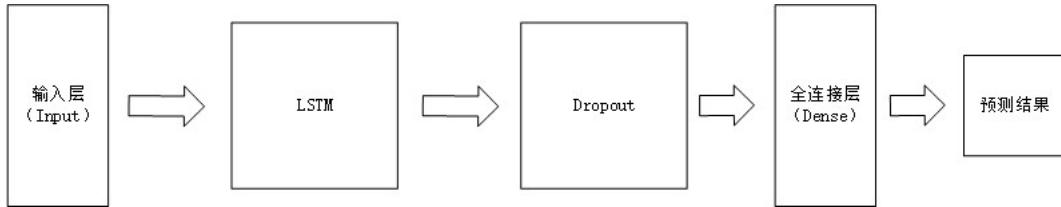


图 5-17 LSTM 预测网络结果概念图

其中, 网络训练参数设置为: batch\_size (每次送入网络中训练的一部分数据大小) 为 64, epoch (所有数据完成训练的迭代次数) 为 50, 模型优化算法采用 Adam 算法。

## (2) 数据集划分及评价指标

训练集和测试集的划分与 4.3.2 节一致, 评价指标与 4.3.3 节一致。

### 5.5.3 实验结果与分析

基于 LSTM 神经网络模型的热点预测实验思路如下: 在基于经典的 MLP 多层神经网络基础上, 使用更适合处理时间序列数据的 RNN 模型进行预测, 采用当前较为先进的模型长短期记忆 (LSTM) 模型对北京市出租车车流密度及热点进行单点局部预测和整体全局预测。其中单点预测与前文相似, 对北京市三个地点的预测效果进行评估, 全局预测即对北京市研究区域内热点分布预测效果进行评估。

采用预测滑动窗口步长为 3 的训练数据, 得到模型的评估参数如表 5-5 所示。

表格 5-5 LSTM 模型指标参数

评价指标	MSE	MAE	$R^2$
测试集	0.5201	0.1667	0.4050

#### (1) LSTM 单点预测效果

利用 5.5.2 节模型的参数设置, 采用预测滑动窗口步长为 3 的训练数据。选取天安门故宫附近 (东经  $116.408^\circ$ , 北纬  $39.916^\circ$ )、北京科技大学西门附近 (东经  $116.36^\circ$ , 北纬  $39.996^\circ$ ) 和首都机场附近 (东经  $116.6^\circ$ , 北纬  $40.054^\circ$ ) 三个地点观察局部的密度预测效果, 如图 5-18 至 5-20 所示。可以

看出，对于局部单点的 LSTM 预测效果不错，不仅整体上符合预测预期，并且在一些车流密度突变点上也能做到很好的预测反应。

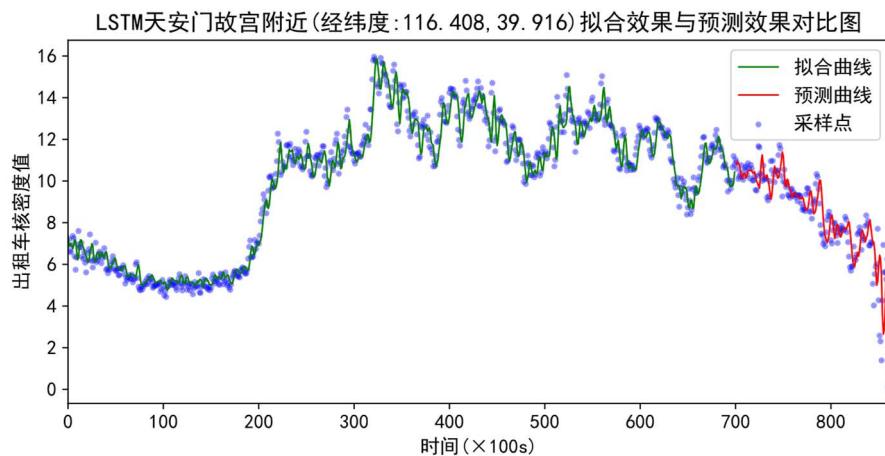


图 5-18 LSTM 预测天安门附近拟合与预测对比

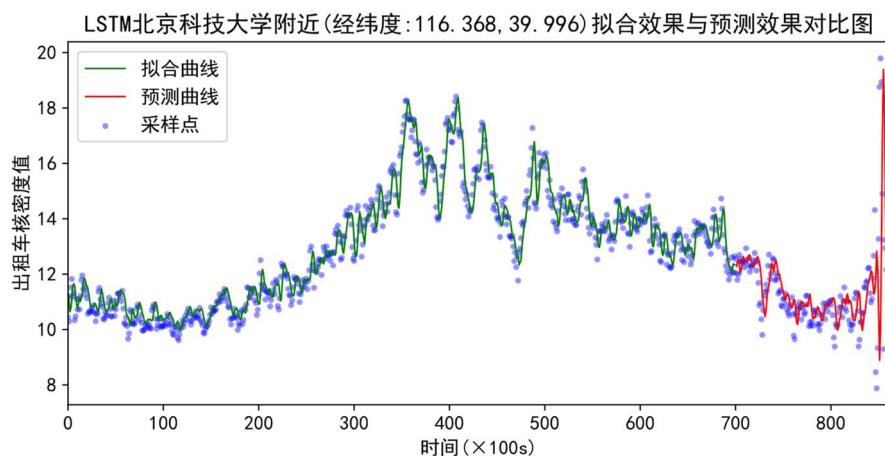


图 5-19 LSTM 预测北京科技大学附近拟合与预测对比

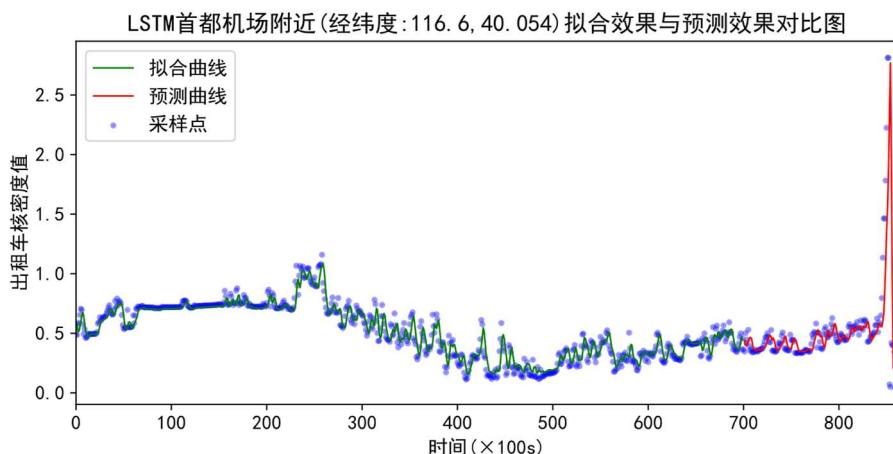


图 5-20 LSTM 预测首都机场附近拟合与预测对比

## (2) LSTM 全局预测效果

对训练出的 LSTM 神经网络预测模型进行若干时间点采样, 取晚上 20:00, 21:00, 22:00 的时间点进行密度分布热力图预测结果与真实情况对比, 如图 5-21 至 5-23 所示。为了更直观的对比预测的误差分布, 通过将预测密度分布与真实密度分布做差值处理, 以 20:00 为例, 如图 5-24 所示。可以看出, 全局的预测效果整体上基本满足预测预期, 在个别热点处边界略有不同, 热点的半径大小有一些差别。

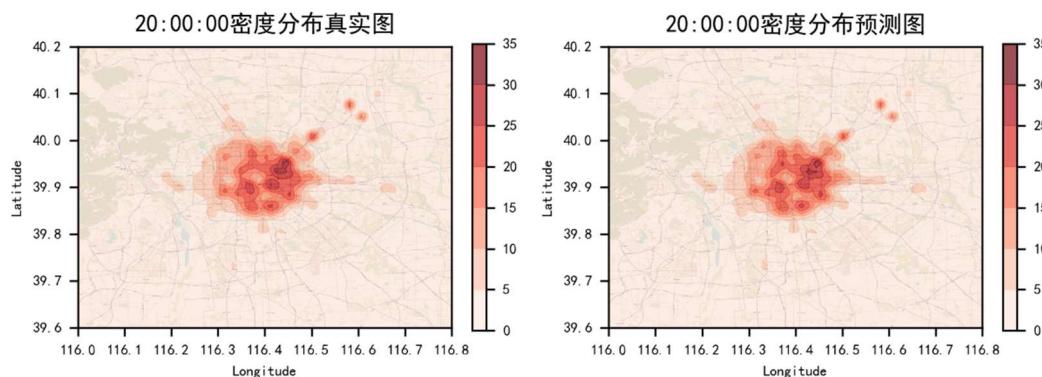


图 5-21 LSTM 晚 20 点全局预测结果与真实情况对比

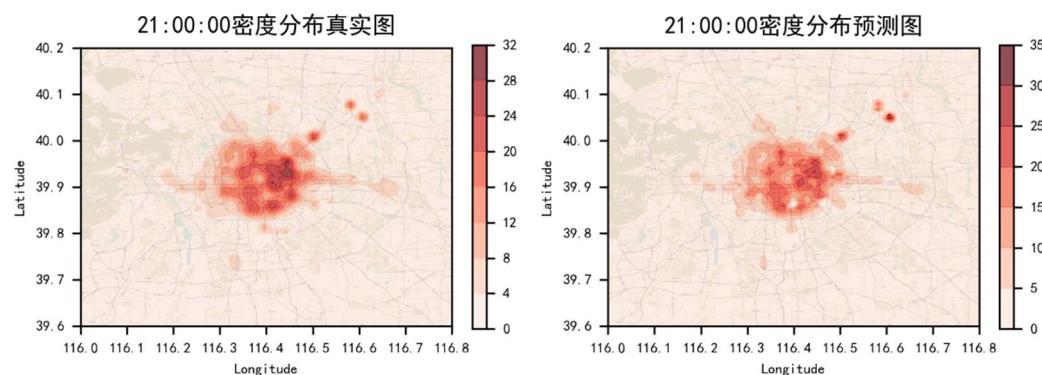


图 5-22 LSTM 晚 21 点全局预测结果与真实情况对比

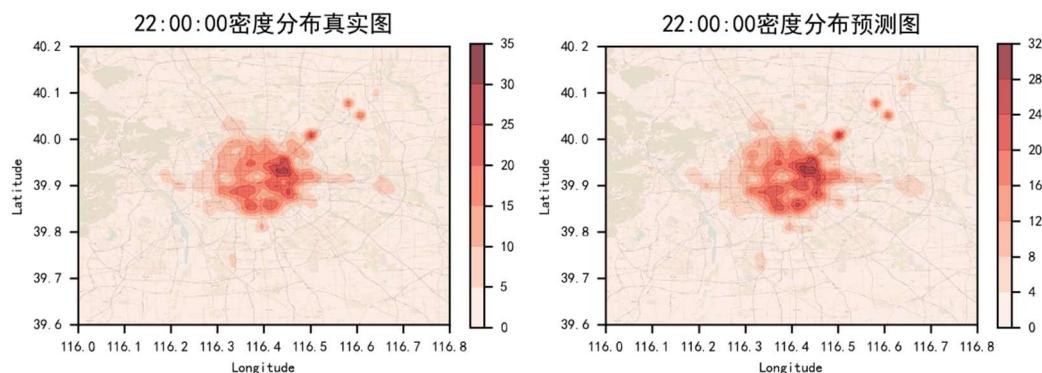


图 5-23 LSTM 晚 22 点全局预测结果与真实情况对比

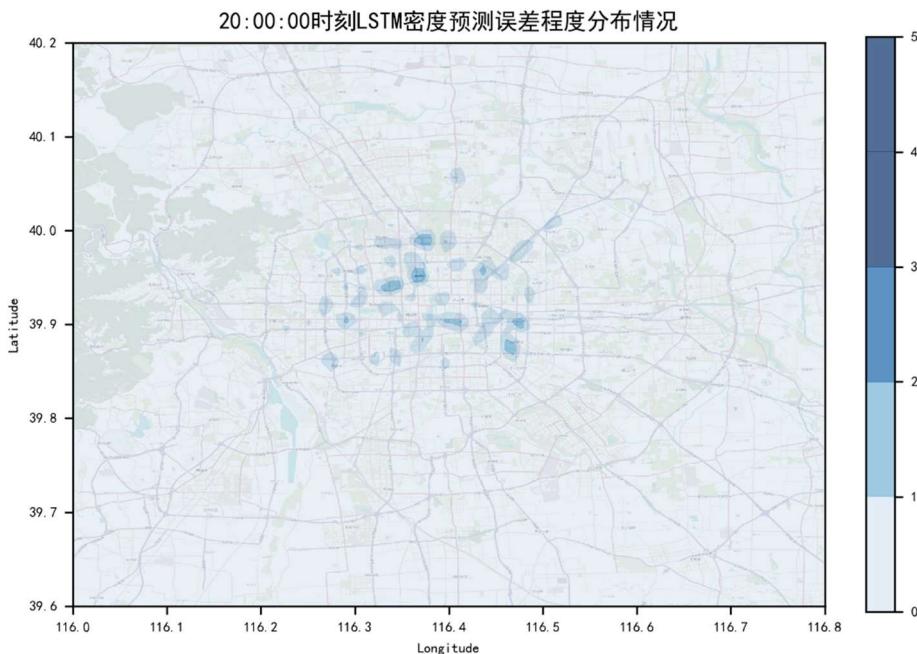


图 5-24 LSTM 晚 20 点全局预测误差分布情况

## 5.6 本章小结

本章是本文探讨的核心部分之一，针对神经网络的构成和循环神经网络模型展开介绍，通过对不同层级的神经网络进行测试，发现滑动窗口步长为 4 的数据集最适合模型训练，同时测试结构中最优的网络结构为单隐藏层神经元个数为 18 的结构。其次，利用较先进的 LSTM 模型进行预测，取得了很不错的效果。本章的研究为机器学习应用于交通领域全局和局部预测提供了新的思路。

## 6 不同时间尺度下模型性能分析

本章主要研究思路：从一个新的视角观察预测效果，前述的实验主要从某一时刻采样时间点或采样地点的角度去观察模型的效果，本章分析不同时间尺度下各种预测模型的性能体现。

### 6.1 不同时间尺度预测模型

#### 6.1.1 短时预测模型

时间序列数据预测采用的是滑动窗口的方法，如前所述，通过将数据集分为训练集和测试集，在使用测试集验证时，属于短时预测模型，即输入的数据是数据集已知的，实际预测的数据是输入数据的下一时间点的预测，即模型训练好后，必须输入预测时间点前窗口步长长度的历史数据，上述预测采用的均为短时预测，也称为实时预测，需要不断向模型输入数据，这样预测未来不同时间尺度上车流密度的精度跟时间尺度的关联性必然不会很强，更多的取决于训练出的模型的性能和输入的窗口数据。为此，提出长时预测模型的概念，为大尺度时间下预测提供一种方式。

#### 6.1.2 长时预测模型

短时预测能够充分体现模型通过喂入数据展现出的优越性能，长时预测则需要模型具有很强的预知能力，以及预测偏差后的纠正能力。长时预测即需要利用训练集训练出的模型来不停预测下一时刻的预测值，然后将预测值作为下一时刻预测的输入数据，即后续的预测数据都取决于已预测出来的历史数据，这样就不需要像实时预测那样不停的喂入历史数据。长时预测和短时预测的对比如图 6-1 所示。

本小节对长时预测模型的概念进行解释，从原理上来看，长时预测通过预测出的数据作为下一次预测的输入数据，每次预测会产生一定的预测偏差，导致下一次输入数据与真实数据产生细微偏差，不断累积，随着时间尺度增加，预测的误差也会累积，导致时间尺度过大时预测误差较大。长时间的预测会导致预测失效的现象。而针对这一问题可以提出两个解决办法：（1）通过增加数据集时间线长度，将一天的数据扩充到一周甚至一个月，然后相应增加输入数据采样时间点的间隔，本文数据采用 100 秒的间隔，可以通过增加采样间隔，将长时预测转化为预测精准的短时预测，但是要求原始数据的

时间线足够长; (2) 通过一些非线性的方法, 实现模型对长时间跨度预测的优化, 克服或减小误差累计的影响。

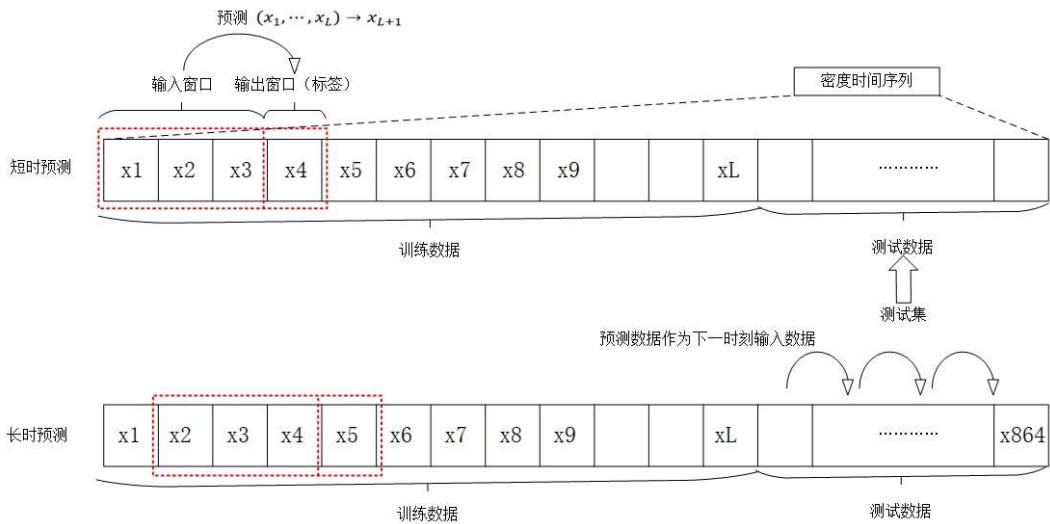


图 6-1 长时预测和短时预测的对比

## 6.2 不同时间尺度预测模型性能对比

### 6.2.1 实验参数设置

本次实验采用短时预测模型观察不同时间尺度下模型预测的性能。使用第四章中最优的 SVR 模型, 第五章中最优 MLP 模型和 LSTM 模型对未来不同时间尺度预测进行分析。

具体参数设置如表 6-1 所示。

表格 6-1 不同时间尺度下不同模型参数设置

模型	预测窗口步长	模型独有参数
SVR	3	$C=10$
MLP	4	单隐藏层, 神经元 18 个
LSTM	3	Unit=100, Dropout=0.2

### 6.2.2 模型对比与分析

针对不同时间尺度下的未来车流密度预测, 本次实验的思路是: 首先采用短时预测模型, 使用前述实验中最优的 SVR 模型, MLP 模型和 LSTM 模型进行实时预测, 观察不同时间尺度下的预测精度。

根据实验参数设置实验的模型, 根据不同时间尺度下研究区域内所有采

样点的预测精度均值进行观察，如图 6-2 所示。短时预测的性能并不受时间尺度的增大而表现出明显下降的趋势，而是模型的预测性能稳定在某一数值附近。其实也不难理解，短时预测不断多输入已知数据，预测下一时刻的结果，一旦模型训练完毕，预测的性能完全取决于模型短时的性能和输入的测试数据，本身和时间尺度没有任何关联，时间尺度在这里成为了需要预测多少个下一时刻车流密度信息。体现的是模型对短期内下一时刻的预测精度，没有出现误差累计的过程，因此短时预测的效果取决于模型训练出的性能，可以看出，LSTM 模型的预测效果最佳，MLP 效果最差，可能与模型参数的调整和测试结构数量限制有关，SVR 预测效果仅次于 LSTM。

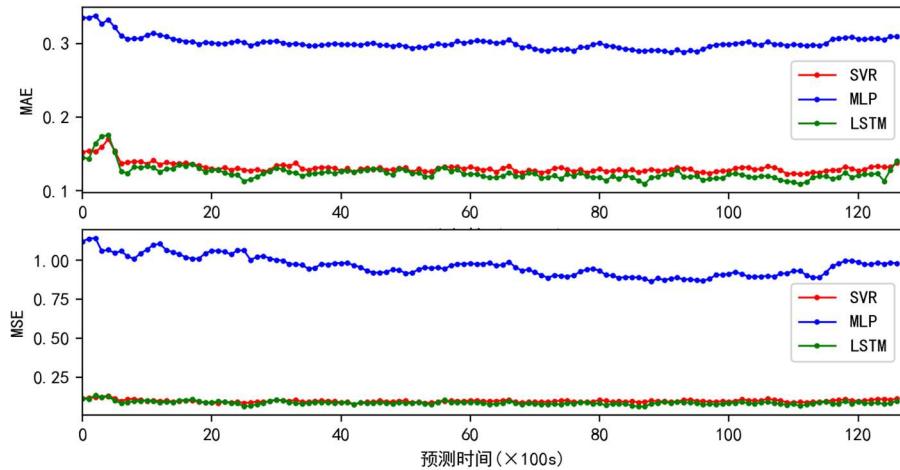


图 6-2 短时预测 SVR, MLP 与 LSTM 不同尺度预测对比

### 6.3 本章小结

本章是本文研究中的新思路新角度的部分，针对前述利用各种不同的预测模型，测试短时预测效果下参数对模型的性能影响。对比短时预测模式下 SVR, MLP 和 LSTM 在不同时间尺度下的预测效果，给予了一些特定结果的分析。引入长时预测模型的概念，为后续工作提出一个新的方向。本章为研究交通预测提出了新的角度。

## 7 结论与展望

本文以北京市一周内出租车车辆轨迹数据为研究对象，提取车流密度信息；以利用历史车流密度信息来预测未来车流密度信息及热点分布为目标，主要在核密度估计算法提取密度信息和机器学习方法预测热点密度信息方面进行深入研究。

本文的主要成果如下：

(1) 使用核密度估计方法进行出租车车流密度特征提取。首先对给出的出租车数据集进行预处理工作，提出时空数据车辆密度挖掘的流程方法，将车辆散点数据转化为有价值的车流密度信息矩阵。在此过程中，测试了核密度估计带宽参数的对研究数据分布的影响，选择最合适的带宽参数进行核密度估计，构建时间采样序列和空间采样矩阵来对时空数据的时间和空间维度进行处理；生成时空密度信息数据集，并将该方法处理一周内七天的北京市出租车数据，并利用爬虫技术获取开源地图 Open Street Map 北京市街道背景影像，实现数据点与地图的匹配，完成车流密度分布可视化。同时，这也是本文应用于交通预测领域提出的最具特色的设计，传统的研究方法普遍直接用车辆定位点的速度信息和设定地点的车流信息来研究，本文从另一个角度——车辆密度的角度去分析，让交通预测更具全局的管控信息，是本文的创新性工作之一。

(2) 使用支持向量回归和不同结构的神经网络对车辆密度信息进行预测。首先将核密度估计算法提取的车流密度信息矩阵进行进一步处理，引入滑动窗口模型，完成对模型训练提供的数据集构造，且通过设置不同的预测滑动窗口步长来构建相应的数据集，形成训练数据集的集合集。归一化处理好的训练数据集合进行 Min-Max 方法处理，利用支持向量回归测试不同步长滑动窗口下的预测效果，找到最合适的步长参数进行全局热点预测和局部单点车流密度预测。同时，使用神经网络模型进行预测，构造不同结构的神经网络，主要体现在神经网络的层数和隐藏层神经元的个数上，本文测试了单隐藏层网络结构和双隐藏层网络结构，找出测试结构中最优模型的性能，并测试不同滑动窗口步长参数的找到最优参数，完成全局和局部的预测。最后，利用较先进的循环神经网络中的长短期记忆模型来处理本文的车流密度时空数据，构建简单的网络结构进行预测，发现其优越的预测性能。该成果内容是本文研究的核心，也是实现预测的最主要內容，完成不同模型处理核密度估计信息预测的性能分析，为后续研究提供一定的基础性参考。

(3) 使用短时预测模型对各类预测模型不同时间尺度下的性能分析。首

先对短时预测和长时预测进行定义，对比两者的差别和特点，对比分析不同预测模型在这短时预测模式下的性能效果，并分析给出预测结果的原理分析。同时，提出的长时预测模型，是本文提出的新的预测思路，将模型推广到非实时预测的长时间尺度的预测场景。

当然，本文的研究仍存在着一些不足，具体体现在以下几个方面：

(1) 本文由于实际计算设备的局限，北京市一周内的数据集并未全部使用，仅使用了周一一天的全部出租车数据进行处理预测，尚未形成以一周为周期的预测模型，有待后续进行算力分布处理或使用更高性能设备进行处理和模型训练。

(2) 本文在研究各类预测模型的性能时，并没有顾全测试所有参数对预测性能的影响，例如：SVR 预测的核函数的对比实验，神经网络正则化参数，优化方法和激活函数的对比实验。仅对主要参数进行了对比测试，最优模型理解为局部最优测试模型，并非全局最优测试模型。

(3) 本文验证了短时预测的性能，并提出的长时预测模型概念，实验没有给出明确的长时预测方法，仅在短时预测上完成预期，并达到预期的预测效果，不同时间尺度上的长时预测有待进一步研究。

综合来看，本文通过引入车流密度信息提取模型来从一种新的角度预测交通状况，提供了一种全新的研究思路，并通过目前常见的机器学习方法验证了预测模型的性能，提供了预测的基础性研究参考，为后续使用更先进网络预测提供基础性对比。也通过短时预测场景下不同模型的对比，为短时实时预测提供一些实验数据，并为长时预测建立模型。

随着机器学习在交通领域的蓬勃发展，不同领域的关联也不断紧密。相信在不久的将来，交通研究工作人员可以利用更先进的机器学习方法与技术，对交通的信息预测更加精准和多元，能对交通管控和人们出行提供更合理、更准确的建议，最终实现交通预测的自动化和多元化，促进交通预测科学领域的发展。



## 参考文献

- [1] X. Zhan, Y. Zheng, X. Yi and S. V. Ukkusuri. Citywide Traffic Volume Estimation Using Trajectory Data[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 29(2): 272-285.
- [2] T. Kyaw, N. N. Oo and W. Zaw. Estimating Travel Speed of Yangon Road Network Using GPS Data and Machine Learning Techniques[C]. 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). 2018, Chiang Rai, Thailand.
- [3] L. Huang and L. Xu. Research on Taxi Travel Time Prediction Based on GBDT Machine Learning Method[C]. 2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), Harbin, China, 2018.
- [4] 羊琰琰. 基于出租车 GPS 数据的热点区域识别及寻客推荐模型研究 [D]. 北京交通大学, 2020.
- [5] W. Liu, Y. Watanabe and Y. Shoji. Vehicle-Assisted Data Delivery in Smart City: A Deep Learning Approach[J]. IEEE Transactions on Vehicular Technology, 2020, 69(11): 13849-13860.
- [6] Vlahogianni E I, Karlaftis M G, Golias J C. Short-term traffic forecasting: Where we are and where we're going[J]. Transportation Research Part C: Emerging Technologies, 2014, 43: 3-19.
- [7] Parzen E. On estimation of a probability density function and mode[J]. The annals of mathematical statistics, 1962, 33(3): 1065-1076.
- [8] Yang J, Zhu J, Sun Y, et al. Delimitating urban commercial central districts by combining kernel density estimation and road intersections: a case study in nanjing city, china[J]. ISPRS International Journal of Geo-Information, 2019, 8(2): 93.
- [9] 李淑娟, 高琳. 山东省乡村旅游景点空间结构及影响因素研究[J]. 中国生态农业学报(中英文), 2019, 27(10): 1492-1501.
- [10] Shi X, Li M, Hunter O, et al. Estimation of environmental exposure: interpolation, kernel density estimation or snapshotting[J]. Annals of GIS, 2019, 25(1): 1-8.
- [11] 李洋, 王浩, 张佳京, 左玉婷, 熊月琳, 罗华堂, 周业华, 徐明星. 基于 GIS 的 2017 年武汉市血吸虫病疫情分析[J]. 中国血吸虫病防治杂志,

- 2019, 31(04): 410-413.
- [12] Laasasenaho K, Lensu A, Lauhanen R, et al. GIS-data related route optimization, hierarchical clustering, location optimization, and kernel density methods are useful for promoting distributed bioenergy plant planning in rural areas[J]. Sustainable Energy Technologies and Assessments, 2019, 32: 47-57.
- [13] 杨军, 李波. 结合核密度估计理论的 ICM 遥感影像分割算法[J]. 测绘科学, 2020, 45(05): 63-71+87.
- [14] 吴嘉逸, 席唱白, 苑振宇, 芮一康, 王结臣. 核密度法的南京苏果超市分布热点探测[J]. 测绘科学, 2017, 42(11): 68-73.
- [15] 卢敏, 杨柳, 王金茵, 黄煌, 王结臣. 基于核密度估计的点群密度制图应用研究[J]. 测绘工程, 2017, 26(04): 70-74+80.
- [16] Lee K, Eo M, Jung E, et al. Short-term traffic prediction with deep neural networks: A survey[J]. IEEE Access, 2021, 9: 54739-54756.
- [17] Ahmed M S, Cook A R. Analysis of freeway traffic time-series data by using Box-Jenkins techniques[M]. 1979.
- [18] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003, 50: 159-175.
- [19] 闵盈盈. 基于 ARIMA 模型的时间序列数据挖掘方法改进[J]. 哈尔滨商业大学学报(自然科学版), 2014, 6: 675-676.
- [20] 邹进贵, 肖扬宣, 张士勇. 马尔科夫链改进的 ARIMA-BP 神经网络模型研究[J]. 测绘地理信息, 2016, 41(04): 32-36.
- [21] Liu J, Guan W. A summary of traffic flow forecasting methods [J]. Journal of Highway and Transportation Research and Development, 2004, 3: 82-85.
- [22] Lütkepohl H. Vector autoregressive models[M]. Handbook of Research Methods and Applications in Empirical Macroeconomics. Edward Elgar Publishing, 2013.
- [23] Vlahogianni E I, Golias J C, Karlaftis M G. Short-term traffic forecasting: Overview of objectives and methods[J]. Transport reviews, 2004, 24(5): 533-557.
- [24] Polson N G, Sokolov V O. Deep learning for short-term traffic flow prediction[J]. Transportation Research Part C: Emerging Technologies, 2017, 79: 1-17.
- [25] Guo G, Wang H, Bell D, et al. KNN model-based approach in classification[C]. OTM Confederated International Conferences " On the Move to Meaningful Internet Systems". Springer, Berlin, Heidelberg, 2003:

- 986-996.
- [26] Xu D, Wang Y, Peng P, et al. Real-time road traffic state prediction based on kernel-KNN[J]. *Transportmetrica A: Transport Science*, 2020, 16(1): 104-118.
  - [27] Bernaś M, Płaczek B, Porwik P, et al. Segmentation of vehicle detector data for improved k-nearest neighbours-based traffic flow prediction[J]. *IET intelligent transport systems*, 2015, 9(3): 264-274.
  - [28] Castro-Neto M, Jeong Y S, Jeong M K, et al. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions[J]. *Expert systems with applications*, 2009, 36(3): 6164-6173.
  - [29] Sun Y, Leng B, Guan W. A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system[J]. *Neurocomputing*, 2015, 166: 109-121.
  - [30] Zeng D, Xu J, Gu J, et al. Short term traffic flow prediction based on online learning SVR[C]. *2008 Workshop on Power Electronics and Intelligent Transportation System*. IEEE, 2008: 616-620.
  - [31] Pinkus A. Approximation theory of the MLP model[J]. *Acta Numerica* 1999: Volume 8, 1999, 8: 143-195.
  - [32] Zargari S A, Siabil S Z, Alavi A H, et al. A computational intelligence-based approach for short-term traffic flow prediction[J]. *Expert Systems*, 2012, 29(2): 124-142.
  - [33] Wang W, Bai Y, Yu C, et al. A network traffic flow prediction with deep learning approach for large-scale metropolitan area network[C]. *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018: 1-9.
  - [34] Han X. Automatic liver lesion segmentation using a deep convolutional neural network method[J]. *arXiv preprint arXiv:1704.07239*, 2017.
  - [35] Yin X, Wu G, Wei J, et al. Deep Learning on Traffic Prediction: Methods, Analysis and Future Directions[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021.
  - [36] Ranjan N, Bhandari S, Zhao H P, et al. City-Wide Traffic Congestion Prediction Based on CNN, LSTM and Transpose CNN[J]. *IEEE Access*, 2020, 8: 81606-81620.
  - [37] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. *arXiv preprint arXiv:1406.1078*, 2014.
  - [38] Madan R, Mangipudi P S. Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN[C]. *2018 Eleventh*

- International Conference on Contemporary Computing (IC3). IEEE, 2018: 1-5.
- [39] Zhene Z, Hao P, Lin L, et al. Deep convolutional mesh RNN for urban traffic passenger flows prediction[C]. 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2018: 1305-1310.
- [40] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM[J]. Neural computation, 2000, 12(10): 2451-2471.
- [41] Tian Y, Pan L. Predicting short-term traffic flow by long short-term memory recurrent neural network[C]. 2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity). IEEE, 2015: 153-158.
- [42] Essien A E, Chukwelu G, Giannetti C. A Scalable Deep Convolutional LSTM Neural Network for Large-Scale Urban Traffic Flow Prediction using Recurrence Plots[C]. 2019 IEEE AFRICON. IEEE, 2019: 1-7.
- [43] Li Y, Wu X. Traffic flow prediction based on long short term memory network[C]. 2018 Chinese Automation Congress (CAC). IEEE, 2018: 1157-1162.
- [44] Chen D, Xiong C, Zhong M. Improved LSTM Based on Attention Mechanism for Short-term Traffic Flow Prediction[C]. 2020 10th International Conference on Information Science and Technology (ICIST). IEEE, 2020: 71-76.
- [45] Bennett J. OpenStreetMap[M]. Packt Publishing Ltd, 2010.
- [46] ArcGIS10.2 在线帮助文档中核密度估计函数的默认带宽设定 [EB/OL].[http://resources.arcgis.com/en/help/main/10.2/index.html#/How\\_Kernel\\_Density\\_works/009z00000011000000/.](http://resources.arcgis.com/en/help/main/10.2/index.html#/How_Kernel_Density_works/009z00000011000000/.(2014,10).)(2014,10).
- [47] BAILEY T C, GATRELL A C. Interactive spatial data analysis[M]. Longman Scientific&Technical Essex, 1995.
- [48] 王远飞, 何洪林. 空间数据分析方法[M]. 北京:科学出版社, 2007.
- [49] 柴进. 基于聚类和 LSTM 算法的车辆轨迹预测模型研究[D]. 北京交通大学, 2020.
- [50] 周越. 基于深度学习的城市出租车流量预测模型[D]. 电子科技大学, 2020.
- [51] 郑义彬, 赖伟伟. 基于支持向量机的高速短时交通流量预测[J]. 工程与建设, 2020, 34(02): 201-204.
- [52] Jiang Y, Han F. A hybrid algorithm of adaptive particle swarm optimization

based on adaptive moment estimation method[C]. International Conference on Intelligent Computing. Springer, Cham, 2017: 658-667.



## 在学取得成果

### 一、 在学期间所获的奖励

2018年11月 “鑫台华杯”华北五省及港澳台大学生计算机应用大赛本科组一等奖 北京市教育委员会

2018年11月 第十届全国大学生数学竞赛（非数学类）二等奖 中国数学会普及工作委员会

2018年11月 国家奖学金 中华人民共和国教育部

2018年11月 优秀三好学生 北京科技大学

2018年11月 优秀学生干部 北京科技大学

2018年12月 第三十五届全国部分地区大学生物理竞赛非物理类A组二等奖 北京物理学会

2019年08月 第十二届全国大学生计算机设计大赛一等奖 中国大学生计算机设计大赛组织委员会

2019年10月 第十三届iCAN国际创新创业大赛北京赛区选拔赛三等奖 iCAN国际创新创业大赛中国组委会

2019年11月 “远洋航海杯”华北五省及港澳台大学生计算机应用大赛本科组二等奖 北京市教育委员会

2019年11月 优秀三好学生 北京科技大学

2019年12月 北京科技大学工程训练综合能力竞赛校内赛一等奖 共青团北京科技大学委员会

2019年12月 国家奖学金 中华人民共和国教育部

2020年07月 第十四届iCAN国际创新创业大赛北京科技大学校内选拔赛二等奖 北京科技大学教务处

2020年07月 第二十一届北京科技大学“摇篮杯”学生创业竞赛二等奖 共青团北京科技大学组委会

2020年08月 第十三届全国大学生计算机设计大赛三等奖 中国大学生计算机设计大赛组织委员会

2020年11月 百优志愿者 共青团北京科技大学委员会

2020年11月 优秀三好学生 北京科技大学

2020年12月 国家奖学金 中华人民共和国教育部

二、 在学期间发表的论文

三、 在学期间取得的科技成果

## 致 谢

光阴荏苒，岁月如歌。在北京科技大学的四年里，从曾经的懵懂稚嫩的自己到如今即将毕业成熟稳重的自己，这一路上磕磕绊绊，少不了每一位老师、同学、朋友的帮助，因为你们，我的大学生活为此丰富多彩、充满阳光。我想，四年来的自己，每一年都有不一样的体会，从大一时的迷茫和探索兴趣，到大二时专心学业和扩展视野，到大三时的深入专业和动手实践，再到大四实习学习和读研深造。这条路走来，难免会遇到种种挫折与羁绊，在岔路口犹豫不决，是你们的帮助与建议，让我在这条坎坷颠簸的路上，给予我果断坚定选择和勇气与信心！

我要感谢我的导师和实验室的帮助我的师姐们。本科的四年里，感谢我的本科生导师隆克平为我在专业道路上指明方向，对未来的规划有了新的理解，感谢为我选题、一路给我指导的皇甫伟老师，让我在机器学习这个研究方向上收获很多。感谢实验室的秦运慧师姐和高涵师姐为我的毕业设计提供指点和建议，感谢你们悉心的关照，让本篇论文得以完成！

我要感谢我的父母。大学四年里为我付出最多的人是你们，你们的支持让我学习到自己热爱的知识，感谢你们的牵挂与付出。我一定不辜负你们的期望，尽自己所能回报社会，感恩父母！

我要感谢教导过我的老师们。这四年我学到了很多，从基础课程到专业课程吧，我遇到了很多可敬可亲的老师，你们的认真负责让我储备了扎实的知识。老师的敦敦教诲，我都会铭记在心中。

我要感谢我的辅导员。日常的学习与生活上，会有很多的琐事和杂事，都是您帮我们梳理解决，感谢我的导员郑智予这一路无微不至的关照，帮助我解决生活上遇到的种种难题，您为我们操心，在这里向您道一声辛苦啦！

我要感谢我的朋友们。四年的生活学习中，遇到了很多可爱的小伙伴，感谢通信 1701 班的每一位同学，让我在这里快乐成长；感谢社团和学生工作中的伙伴们，你们给予我大学全新的体验；感谢和我一起打拼在竞赛赛场的战友们，你们让我看到了提升的另一种可能；感谢每一个和我一起走过的朋友，认识你们是我人生的宝贵财富！

我要感谢我的母校，感谢培养我的北京科技大学。这里经历的美好，是我这辈子忘不了的回忆。

最后，感谢美好时光，感谢自己的坚持，追忆最美好的青春年华！



