

硕 士 学 位 论 文

基于人-车交互的行人轨迹预测方法

Pedestrian Trajectory Prediction Based on Human-Vehicle Interaction

作 者 姓 名：_____王欣然_____

学 科、 专 业：_____车辆工程_____

学 号：_____21703135_____

指 导 教 师：_____连静_____

完 成 日 期：_____2020.05_____

大连理工大学

Dalian University of Technology

大连理工大学学位论文独创性声明

作者郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用内容和致谢的地方外，本论文不包含其他个人或集体已经发表的研究成果，也不包含其他已申请学位或其他用途使用过的成果。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

学位论文题目：_____基于人-车交互的行人轨迹预测方法_____

作者签名：_____王欣然_____日期：2020年6月2日

摘 要

交通场景中的行人轨迹预测是自动驾驶车辆和智能交通系统的重要组成部分，是缓解交通拥堵，优化智能汽车驾驶决策的重要依据。由于在交通场景中，行人的运动轨迹会受到周围行人，周围车辆的影响，因此研究行人轨迹的社交行为对提高行人轨迹预测的精度有重要意义。针对交通场景中的轨迹预测问题，本文主要工作如下：

首先构建了 DUT 人-车交互数据集，利用相关滤波的方法对行人和车辆的轨迹进行跟踪，得到交通场景中行人和车辆的轨迹时序序列，并使用卡尔曼滤波的方法将得到的数据进行平滑处理，得到了适用于人-车交互的行人轨迹预测数据集。其次，设计了扇形人-人交互邻域和圆形人-车交互邻域来准确捕捉对预测行人有相互作用的行人和车辆。再次，定义了人-人、人-车交互的防碰撞函数和方向注意力函数作为人-车、人-人社交信息的权重，进一步提高了社会信息的精度。然后，将两种交互信息通过两种不同的 LSTM 进行编码后，输入到注意力模块中来筛选对行人影响大的社会信息。最后，将筛选后的社会信息与行人轨迹序列一起输入到 LSTM 预测网络中，实现了轨迹的预测。

通过对比其他现有模型的误差实验来验证本文所提出的轨迹预测模型，实验结果表明，本文提出的方法能够准确地预测出交通场景中，行人未来一段时间内的运动轨迹，有效提高了预测精度，提高了智能驾驶决策的准确性。

关键词：自动驾驶；人车交互；轨迹预测；长短时记忆神经网络

Pedestrian Trajectory Prediction Based on Human-Vehicle Interaction

Abstract

Pedestrian trajectory prediction in traffic scene is an important part of intelligent vehicle and intelligent transportation system. At the same time, it is also an important basis for solving the problems of traffic congestion, intelligent vehicle driving decision and system control. In the traffic scene, the pedestrian's trajectory will be affected by the surrounding pedestrians and vehicles, so it is important to study the social behavior of pedestrian trajectory to improve the accuracy of pedestrian trajectory prediction. The main work of this paper is as follows:

Firstly, the DUT human-vehicle interaction data set was constructed, and the track sequence of pedestrians and vehicles in the traffic scene was obtained by using the correlation filtering method. Then, the Kalman filtering method was used to smooth the obtained data. Secondly, the fan-shaped human-human interaction neighborhood and the circular human-vehicle interaction neighborhood were designed to accurately capture the pedestrians and vehicles that interact with the predicted pedestrians, and the anti-collision function and direction attention function of human-vehicle interaction are defined as the weight of human-vehicle, human-vehicle social information, which further improves the accuracy of social information. Then, three different LSTM coding layers were used to encode the human-human, human vehicle interaction information and the current historical information of the predicted pedestrian. After that the human-human, human-vehicle interaction information was inputted into the attention module to get information that pedestrians pay more attention to. Finally, the filtered social information and the encoded pedestrian history track sequence are input into the LSTM prediction network to predict the pedestrian track.

At last, the DUT human-vehicle interaction data set constructed by our group was used to verify the network proposed in this paper. The experimental results show that the method proposed in this paper can accurately predict the future movement trajectory of pedestrians in traffic and improve the accuracy of intelligent vehicle decision.

Key Words: Automatic driving; Human-vehicle interaction; Trajectory prediction; Long and short-term memory neural network

目 录

摘 要.....	I
Abstract.....	II
1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 自动驾驶与智能交通.....	2
1.3 轨迹预测方法分析.....	4
1.3.1 基于模型的方法.....	5
1.3.2 基于数据的方法.....	6
1.4 本文研究的主要内容.....	8
2 深度学习与神经网络.....	10
2.1 神经网络与神经元模型.....	10
2.2 递归神经网络.....	13
2.2.1 RNN 模型.....	13
2.2.2 LSTM 模型.....	14
2.2.3 Bi-LSTM.....	18
2.2.4 混合 LSTM.....	19
2.3 优化算法.....	19
2.3.1 梯度下降.....	19
2.3.2 AdaGrad 优化算法.....	20
2.3.3 RMSprop 优化算法.....	20
2.4 TensorFlow 平台.....	21
2.5 本章小结.....	22
3 行人及车辆轨迹获取.....	23
3.1 数据集获取.....	23
3.2 轨迹提取.....	24
3.2.1 视频稳像.....	24
3.2.2 轨迹跟踪.....	25
3.2.2 信道可靠性.....	26
3.2.1 空间可靠性.....	26
3.3 卡尔曼滤波.....	28
3.3.1 行人轨迹处理.....	28

3.3.2	车辆轨迹数据处理.....	30
3.4	本章小结.....	31
4	基于人-车交互的行人轨迹预测模型.....	32
4.1	算法流程.....	32
4.2	数据提取及处理.....	33
4.3	网络基本结构.....	34
4.4	人-人、人-车社会关系建模.....	35
4.4.1	人-人交互输入.....	35
4.4.2	人-车交互输入.....	36
4.3.4	注意力模块.....	37
4.5	行人轨迹预测.....	39
4.5.1	预测层结构.....	39
4.5.2	网络训练方法.....	40
4.6	本章小结.....	41
5	轨迹模型预测结果分析.....	42
5.1	实验参数设置.....	42
5.2	实验方案及评价指标.....	42
5.2.1	网络训练分析.....	43
5.2.2	轨迹预测模型参数分析.....	44
5.3	模型对比分析.....	49
5.4	预测轨迹可视化.....	50
5.5	本章小结.....	50
结 论	52
参 考 文 献	53
攻读硕士学位期间发表学术论文情况	57
致 谢	58
大连理工大学学位论文版权使用授权书	59

1 绪论

1.1 研究背景及意义

随着全球汽车保有量的不断增大,道路拥堵问题日益严重,道路交通安全也越来越引起人们的重视。车辆作为道路交通的主要成分之一,其行驶安全问题是研究人员始终关注的重要问题,其中自动驾驶汽车将成为未来道路交通的重要组成部分,其安全问题和通行效率问题更不容忽视^[1]。

随着信息技术的快速发展,为了提高道路交通通行效率、降低交通事故发生概率,世界各国相继致力于智能交通系统(Intelligent Transportation System, ITS)的研究开发^[2]。交通管理部门的监控数据以及车联网系统提供了大量的实时车辆、行人的运行数据,通过分析这些数据,可以获得某一区域当前以及未来一段时间内的交通状况。交通管理部门可以针对不同的交通状况提前做出不同的应对措施,避免交通拥堵现象的发生^[3],也可用于根据车辆轨迹的分布情况来完善城市的基础设施建设等。同时,用户可以根据当前行人的轨迹预测数据做出合理的路径规划,实时调整汽车的决策,一方面节省了自身的出行时间,另一方面也缓解了道路的交通压力。

为了保证行驶安全性,自动驾驶车辆需要预测周围行人的运动轨迹以避免碰撞。为了避免拥堵问题,自动驾驶车辆需要根据行人的运动轨迹来决定是否加速驶过以提高交通效率,而不是遇人即停。对于现阶段的辅助驾驶系统来说,通过轨迹预测,车辆可以将周围车辆、行人的行驶轨迹提前告知驾驶员,以便驾驶员可以针对道路交通情况提前做出反应。在未来的L4、L5自动驾驶车辆中,轨迹预测算法可以把未来一段时间内的交通场景中的各个运动单元的轨迹传递给车辆决策机构加以处理分析,从而制定最佳的路径规划策略,保证车辆的安全高效行驶^[4]。

而随着全球定位技术、5G技术、大数据云计算技术的迅速发展,车辆的行驶轨迹数据采集过程变得越来越简单,基于这些轨迹数据进行位置挖掘和轨迹预测越来越成为当今的研究的热点问题之一^[5]。行人的轨迹数据往往包含了经度、纬度、速度、加速度、时间、行人编码等信息所构成的数据点。轨迹预测技术^[6]是指利用目标当前位置信息、历史移动轨迹信息和其他辅助信息对目标在将来时刻的轨迹进行预先推算和估计。轨迹预测研究在很多领域都有广泛应用,例如自动驾驶、智能交通、路径规划与决策等^[7]。

综上所述,行人的轨迹预测研究不仅是自动驾驶车辆这一未来汽车发展趋势的不可或缺的技术支持,更可以使车路交通系统更加高效安全,给人们带来更好的出行体验,

尤其适用于我国高密度，低速度，人车混杂场景较多的交通场景中。

1.2 自动驾驶与智能交通

随着 21 世纪科学技术与工业技术的飞跃式发展，汽车这种交通工具已经日益成为人们不可或缺的代步工具。据统计，2019 年我国汽车销售量已经上升至 2576.9 万辆，我国汽车保有量已经达到 19327 万辆，并且仍处于稳定增长状态。而随着汽车产业的不断进步发展，一些弊端也逐渐暴露出来，这个行业所带来的负面效果也日益显现，如资源短缺，环境污染，交通拥堵，交通事故等问题。据世界卫生组织统计，交通事故每年造成了约 125 万人死亡以及数千亿美元的经济损失^[8]，如果不及时采取有效的行动，预计到 2030 年交通事故会成为世界第七大死亡原因。在我国，378781 起，造成 89455 人死亡、431139 人受伤，直接财产损失 14.9 亿元，已经连续十余年居世界第一。

统计资料表明，90%以上的交通事故都是由人为因素造成的^[9]，驾驶员不严格遵守交通规则，反应不及时以及错误操作是导致道路交通事故发生的主要原因，交通事故不仅给人员安全带来了威胁，给财产造成了损失，同时也会造成一定时间内的交通堵塞或者局部瘫痪，给城市的交通系统带来了压力。同时在计算机信息技术高速发展的背景下，人们的生活越来越趋向于机械自动化和人工智能化。在智能驾驶领域，智能汽车需要提高其安全性，舒适性，交互性，使得智能汽车具有更高的人性化，智能化，因此智能车辆的研究已成为当前汽车发展的主要热点之一。自谷歌公司（Google Inc.）成功研发出第一辆自动驾驶汽车并向公众推广以来^[10]，世界各大知名车企纷纷创建了自己的自动驾驶研发部门，甚至一些互联网公司“造车新势力”也推动了自身的自动驾驶研究项目，其中特斯拉所研究的智能辅助驾驶汽车^[11]，已真正走进我们的生活，并发挥了其辅助驾驶作用。

智能汽车尤其是 L4 级别的无人驾驶汽车主要包括环境感知技术、智能决策技术、路径规划技术、跟踪控制技术，包含了计算机、通信、人工智能等多个新兴领域，涉及到高精度传感器制造，复杂电子电路设计，自动控制设计等传统技术领域，是一个典型的高新技术集成平台^[12]。在行驶过程中，自动驾驶汽车通过雷达、摄像头等传感器采集环境及路况信息，将数据实时传递至车辆的决策模块，进而控制车辆的加减速、转向等，以实现部分辅助驾驶或完全自动驾驶。随着道路交通状况变得日趋复杂，我国道路交通场景密度越来越高，人-车的混合度越来越高，意味道路交通系统对人们驾驶技术提出了越来越高的要求^[13]。自动驾驶汽车通过提前预警驾驶员或是替代驾驶员操纵车辆，可以有效地提高交通系统效率和降低交通事故发生概率。此外，自动驾驶汽车在军事领域

和商业领域也有着广泛的应用潜力。在军事领域，自动驾驶汽车将成为信息化军事装备体系的重要组成部分，由于无人驾驶汽车不需要驾驶员，在危险任务中能够有效降低人员伤亡比例，从而提高部队的战斗能力。尤其是在战术侦查方面，自动驾驶车辆凭借其灵活性可以深入危险盲区进行侦查，在战术信息获取方面有极大的战略意义。在商用领域，如物流领域、矿区环境等等，自动驾驶车辆可以提高生产效率、节约人力资源，为企业创造更大的商业利润。

无人驾驶汽车在欧美地区起步较早，早在 20 世纪 80 年代初，美国卡内基梅隆大学（Carnegie Mellon University）的相关团队就已经研发出了世界上第一辆无人驾驶汽车 NavLab-1^[14]。1987 年，美国国防高级研究局启动了自主地面车辆的研究^[15]。1998 年，意大利的 Parma 大学研发出了“ARGO”试验车，在意大利的高速公路上进行了 2000 km 的道路实验，实验条件为横向自主控制、纵向人工控制，最终自动驾驶里程占总里程 94% 以上，最高车速为 102 km/h^[16]。自 2004 年开始，美国国防部多次举办了无人驾驶车辆挑战比赛^[17]，引发了人们对无人驾驶技术的广泛关注，大大促进了车辆自动驾驶技术的发展。2015 年，谷歌第一辆原型汽车正式亮相^[18]，并且获得了正式上路测试的法规许可，至今谷歌无人车总行驶里程已超过 30 万公里。2017 年，搭载了 L3 级别的高度自动驾驶功能的第 5 代奥迪 A8 正式对外发布，该车型的下线标志着 L3 级别的自动驾驶汽车实现了工业化量产。2019 年 2 月，美国加州机动车管理局（DMV）发布了《2018 年自动驾驶接管报告》，数据显示，谷歌旗下的 Waymo 无人驾驶汽车继续领跑市场，平均每跑 17846.8 km 才需要人工干预一次，较 2017 年（5596 km/次）成绩有了较大提升。

目前，世界各国都对无人驾驶技术抱有浓厚的兴趣，并且提供了大量人力物力的投入来表示支持，尤其是欧美、日本等发达国家无人驾驶技术的无论是在法律层面，还是在技术层面，无人驾驶的研究都走在世界的前列。美国的 22 个州和特区在 2017 年底之前，已经先后通过了关于自动驾驶汽车的相关法律法规。联合国于 2016 年发布了有关汽车自动驾驶的《国际道路交通公约》修正，改修正案规定，汽车被允许在特定期间，特定道路环境下进行自动驾驶。其中 University of Michigan 专门建立了为无人驾驶汽车测试、V2V/V2I 车联网技术测试的无人驾驶试验区。

在自动驾驶领域，中国的起步较晚，但发展迅速。2001 年，国防科技大学联合中国一汽共同研发出了红旗 CA7460 自动驾驶汽车，在高速公路上试验成功，最高行驶速度可达 130 km/h。2009 年 6 月 4 日，首届“中国智能车未来挑战赛”在西安成功举办，时至今日该赛事已经连续举办了 8 届，极大推动了我国自动驾驶车辆的研发。2011 年，国防

科技大学在中国第一汽车集团资助下，在红旗 HQ3 轿车的基础上实现了无人驾驶^[19]，完成了首次高速无人驾驶实验，其路程为 286 km，平均车速可达 87 km/h。2013 年，百度展开了自动驾驶车辆的研究，逐步开展城市、高速道路及环路等混合工况下的全自动驾驶的研究工作。2018 年 11 月 1 日，在百度世界大会上，百度公司与一汽公司共同发布 L4 级别无人驾驶乘用车，预计在 2020 年投产。此外，国内其他互联网公司和传统车企也纷纷开展了自动驾驶车辆的研究。2016 年，经由工业和信息化部批准，国内第一个“国家智能网联汽车（上海）试点示范区”于上海正式开园运营。

总体而言，我国的自动驾驶相关技术的研究近些年发展迅速，大幅缩小了和传统汽车强国的差距，已经能够实现结构化道路下的自动驾驶，但在传感器的开发、算法的优化等方面和国外还有一定差距^[20]。

智能交通系统是一个集成了计算机技术、信息融合技术、电子通信技术、智能传感技术以及自动控制技术的综合交通管理体系，实现“人、车、路”三者的有机结合，建设成实时、准确、高效的运输和管理系统^[21]。自动驾驶汽车是实现智能交通系统的不可或缺的环节之一。同时，智能交通系统也将为自动驾驶汽车提供大量的数据支持。智能交通系统不仅能够实现车与车之间的通信，更能实现车与道路基础设施之间的通信。例如，现阶段的自动驾驶车辆在识别道路交通标识方面仍存在一定的困难，仅仅依靠车载传感器采集到的信息，无法做到百分百的识别准确率。在智能交通系统的帮助下，当自动驾驶车辆行进到道路交通标识附近的时候，可以通过网络通信将相关信息提前发送给自动驾驶车辆进而保证了车辆对道路交通标识识别的准确度。

在智能交通系统中，自动驾驶车辆的行驶路线在出发前就已在交通控制中心进行了合理的规划^[22]。实际行驶中，车辆的实时地理位置坐标会实时上传到控制中心，同时车辆不断获得加减速、转向等位置移动信息指令。自动驾驶车辆自身只需要通过传感器采集周边物体的距离信息，当发现可能发生碰撞的情况时，优先执行避障操作，之后再继续执行位置移动信息，大大减轻了车载终端的运算量。在多车行驶的环境中，自动驾驶车辆会不断地将车辆 ID、速度、加减速、转向等信息上传至控制中心，接受位置移动信息指令。对每辆车来说，一方面按照控制中心的规划进行调度，另一方面与周围的车辆彼此都了解当前和未来的移动趋势。在智能交通系统的信息支持下，自动驾驶车辆的安全性和通行效率将得到大大提高。

1.3 轨迹预测方法分析

自动驾驶作为一项革新技术，从感知环境到信息处理再到车辆控制，尚存在诸多不

成熟的因素。尤其是在人车发生交互的情景下，由于智能汽车不能与人类驾驶员一样几乎仅依赖视觉信息即可完成车辆操纵的驾驶流程，智能汽车不能够根据“看”到的东西预测行人下一步的轨迹，从而做出决策。因此对于智能汽车，如果能准确预测行人和其他车辆在未来短时间内的行为，并提前调整自身的速度以防止碰撞，则能有效地减少交通事故^[23]。目前常用的轨迹预测方法主要分为基于数据的方法和基于模型的方法。

1.3.1 基于模型的方法

基于模型的方法根据模型的复杂程度可以分为：基于物理模型的轨迹预测方法和基于交互感知模型的轨迹预测方法^[24]。基于物理模型的预测方法将目标表示为满足物理定律的动态实体，如构建目标的运动学或动力学模型，使用物理学参数来预测未来位置，将行人的位置、速度、加速度以及外部条件（如路面的摩擦系数）与车辆的轨迹改变建立联系。其中最简单的是恒速（CV）和恒定加速（CA）模型，它们都假定了目标的直线运动^[25, 26]。但是物理模型一般仅仅考虑了被预测目标的运动学和动力学特性，因此仅适用于短期内的目标轨迹的预测，无法预测由外部环境改变（如前方车辆减速，出现障碍物等）引起的轨迹变化。

基于交互感知模型的轨迹预测方法将车辆或行人表示为彼此交互的机动体，即假设车辆或行人的运动受到场景中其他车辆行人的运动的影响。与基于机动模型的轨迹预测方法相比，考虑车辆与行人之间的相互影响可以更好地解释车辆和的运动轨迹变化^[27]。基于交互感知模型的轨迹预测方法是目前最为全面的模型，但目前相关的研究仍比较少。Gindele T 等^[28]利用动态贝叶斯网络构建了一种过滤器，明确考虑车辆之间的相互作用，能够同时估计交通参与者的行为并预测他们未来的轨迹。在高速公路场景的背景下评估滤波器，即使在非常嘈杂的测量中也表现出良好的性能。Nachiket Deo 等^[29]为车辆的不同机动模式分配置信值，并在此基础上输出未来运动的多模态分布，以此表示车辆间交互的影响，利用长短时记忆网络预测了高速公路上车辆的轨迹分布。

在行人轨迹预测模型的研究中，基于交互感知模型的代表方法为基于社会力模型^[30]的方式主要应用在行人轨迹预测上，它根据引力与斥力方式对行人进行建模，认为行人的目标会对行人产生引力进而吸引他们走向目标，行人之间产生斥力进而防止行人之间发生碰撞，行人所在的场景如边界，障碍物等会对行人的运动轨迹产生影响。该模型具有模型简单直观、复杂性低的优点，但存在模型对模型参数过于敏感，模型学习能力不够强，泛化能力差^[31]的问题。

1.3.2 基于数据的方法

近来基于数据驱动的目标轨迹预测方法主要可以分为有参数模型和无参数模型，有参数模型又分为基于高斯回归过程，基于马尔科夫模型^[32]，基于卡尔曼滤波模型，贝叶斯网络^[33]等。

高斯回归算法与其他回归算法的区别在于，高斯回归的输出值为一个概率分布函数，得到概率最大的多条轨迹而不是单一的轨迹，在轨迹预测方面，Trautman P 等^[34]结合社会力中的势函数来考虑行人之间的交互影响，提出了交互高斯过程 IGP 来在社会力模型的基础上，对行人轨迹进行概率性预测，使得结果为概率性多条轨迹，提高了预测精度。茅晨昊等^[35]提出一种基于高斯过程回归的船舶轨迹预测模型.对已有船舶航行轨迹进行仿真模拟,从而预测出船舶轨迹线路。高建等^[36]针对不同时间道路车流量变化下轨迹预测误差变化大的问题,提出基于概率分布模型的高斯混合-时间序列模型 (GMTSM)，不仅能在正常情况下准确预测车辆轨迹，而且能有效提高道路车流量变化情况下的轨迹预测准确率，适用于现实路况环境。朱坤等^[37]提出了一种高斯混合贝叶斯轨迹预测模型，实验结果表明，GM-BM 模型可以通过调整混合模型中子模型在路段正常交通流下的权重来预测最有可能的轨道。

马尔可夫链为状态空间中经过从一个状态到另一个状态的转换的随机过程，该过程要求具备“无记忆”的性质：下一状态的概率分布只能由当前状态决定，在时间序列中它前面的事件均与之无关。马尔可夫链方法被广泛的应用在时间序列问题中，MORRIS B T 等^[38]为了对不同类别的行人轨迹预测，使用了隐形马尔可夫链的方法，进行时空上的概率建模。在车辆轨迹预测中，李万高等^[39]针对历史轨迹数据不足的问题，构建马尔科夫模型来量化位置的相关性，通过把历史轨迹划分成子轨迹得到了更精确的马尔可夫模型，充分利用了有限的轨迹数据，保证了预测结果较高的精度。乔少杰等^[40]提出了基于隐马尔可夫模型的自适应轨迹预测模型，采用密度聚类的方法将位置映射到不同分区中进行长期轨迹预测，减少了隐马尔可夫模型的状态数量，不仅能提高预测精度还能降低运算时间。KITANI K M 等^[41]使用基于隐含马尔科夫过程的方式对行人进行建模，并使用反向强化学习的方式训练模型参数，进而学习外界静态的物理环境对行人的影响。Lei P R 等^[42]提出了一种时空轨迹模型（STT），对移动目标在时空内进行运动行为建模，该模型能够发现区域之间的顺序遍历关系，有效地捕捉目标的运动行为，并且对于每个区域，STT 导出关于对象出现的时间的相应概率。对于历史轨迹信息数据的不确定性的问题，毛莺池等^[43]重新定义了补全路径算法，弥补了历史轨迹数据的不完整性和不确定性，创新性地提出了不确定性车辆路口轨迹预测算法，该算法基于马尔科夫链模

型对路口的车辆进行未来连续轨迹预测。夏卓群等^[44]将车载移动端实时获得的轨迹数据输入到轨迹预测模型中，实现了轨迹在线实时预测，仿真结果表明该方法能够很好的适应交通环境，降低预测误差，提高实时性。

在滤波模型方面，卡尔曼滤波模型表现优异，它利用多次迭代，对信号进行最优化估计，例如姚智胜^[45]将车速，流量等参数作为卡尔曼滤波算法的状态参数，实现了对交通流的递归预测。李进燕等^[46]使用卡尔曼滤波算法，结合过去，当前的信息数据，预测未来的数据，然后将三组数据进行融合，为路径规划与决策提供实时信息。Hodgson D 等^[47]，将汽车的质量考虑在内，结合遗传算法，来优化滤波算法的协方差矩阵，以估测不同车辆的车速。但是卡尔曼滤波算法只能以当前状态为基准，估计下一时段的状态，仅仅适用于短时估计，不能解决长期预测问题。同时以上有参数模型，都十分依赖于人工设计的模型参数，要求准确的建模，而模型参数的选取全凭设计者的经验。此外行人的轨迹受到的影响因素有很多，有参数模型往往只关注其中一种因素而无法关注于多种影响因素的组合。

随着深度学习和神经网络的飞速发展，基于深度学习的无参数数据驱动算法成为研究热点，深度学习具有数据量大，速度快，鲁棒性强等优点，并且可以自动从样本中学习数据的特征而不是依赖于人工建模的方式，可以避免由于经验不充足造成的参数不准确。其中循环神经网络（Recurrent Neural Network, RNN）具有递归的特性，能够传递时序序列的信息，比较适用于行人轨迹预测^[48, 49]。但 RNN 网络不能够对历史信息进行筛选，而是全部记住，导致数据量过于庞大冗杂，存在梯度爆炸和梯度消失的问题^[50]，优化过程存在一定缺陷。

近年来，LSTM 神经网络的提出很好的解决了以上问题。作为 RNN 神经网络的变体，长短时记忆神经网络（Long-short memory network, LSTM）既具有 RNN 网络输出反馈的特点^[51]，又能够通过自身内部的 cell 结构选择性长期记住需要记住的信息^[52]，可以更好地对行人轨迹进行预测。但是单纯的神经网络方法只将历史轨迹作为预测信息，无法解释行人或车辆的一些社会行为，如结伴，跟随，避让等行为。为了将交互预测的思想与数据驱动预测的思想相结合，让深度学习网络更好的学习行人结伴行走，相互避让，沿路转弯等问题，Alahi A 等^[53]在单纯的 LSTM 网络的基础上提出了社会信息交互池化层（Social-LSTM），该池化层在空间上做了网格化来刻画行人的交互，利用池化结果预测个体进行符合社会规律的轨迹。但是 Social-LSTM 假设每个邻居行人对当前目标的影响大小只有位置远近决定，但是在实际场景中，邻居行人的方向，速度，位置，距离都会对当前预测目标产生影响^[54]，例如预测目标可能会对另一个距离远但是相对而行的

个体产生行动，或对于一个位置远但速度快的个体做出避让，而对于临近的物体，如果行动方向不会发生交互，那么就不会对当前行人产生影响。在此基础上，Xue H^[55]将环境因素考虑在内，建立行人与环境交互的神经网络，能够更好的表达行人的目的性，进而精准的预测。Xu K^[56]建立了 Collision-Free LSTM，这种模型基于智能单元体的防碰撞问题建立神经网络，并且考虑了距离对碰撞的影响。Gupta A 等^[57]结合生成对抗网络，提出了 S-GAN 模型，用一种新的池化策略来描述行人之间的影响，引导轨迹预测模块预测出避免碰撞的轨迹。Zhang P 等人提出了一个 SR-LSTM 的数据驱动状态细化模块，它计算了邻居的当前意图，使得被预测的行人能够更好的判断下一步行为^[58]。Mohamed A 等人提出了 STGCNN 模型^[59]，该模型引入了图网络结构，将行人之间的社交互动嵌入邻接矩阵中，有效地提高了训练速度和预测精度。因此如果能根据 LSTM 神经网络预测到行人和车辆理论上未来一段时间内的轨迹，然后根据社会力的思想建立人车社会交互模型，即可以有效地修正目标预测轨迹。

1.4 本文研究的主要内容

智能驾驶汽车的飞速发展已经成为如今的浪潮趋势，针对于智能驾驶技术的研究也成为当今研究领域的热点问题。由于智能驾驶汽车本质上需要用计算机代替电脑来进行感知决策，结合交通道路对安全性和通行效率的要求，合理地对交通场景中的目标进行轨迹预测是智能驾驶汽车和智能交通的发展重点。提前预知目标的轨迹，不仅能够让汽车及时做出避让以减少交通事故，还能在汽车得知行人与之不发生教会的前提下继续同行而不是遇人则停。因此本文的研究重点是针对人-车交互的交通场景中，对多个行人进行轨迹预测。基于以上对轨迹预测方法的分析，本文基于深度学习的方法，在 LSTM 神经网络的基础上，将历史信息轨迹数据于人-人、人-车等社会交互信息数据结合在一起，构建了基于人-车交互的行人轨迹预测模型，主要内容如下：

（1）研究背景以及研究意义，分析了智能汽车以及目标轨迹预测研究的必然性，重要性，可行性。介绍了国内外智能汽车发展现状及前景，轨迹预测的研究发展的优势不足之处，为后续章节的研究方向提供铺垫。

（2）研究深度学习与神经网络的发展，重点分析本文所需要的递归神经网络模型，以及其变种，分析了深度学习不同的优化算法，并引入了本文所需要的实验平台，

（3）构建 DUT 人-车交互数据集，为轨迹预测模型提供数据支持。分析了无人机设备的基本参数和交通场景的特点，通过视频稳像，目标追踪的方法来提取行人和车辆的轨迹位置。最后使用卡尔曼滤波算法对数据进行处理，得到平滑的，适用于轨迹预测

模型的数据集。

(4) 构建基于人-车交互的轨迹预测模型，对行人轨迹进行预处理，建立人-人交互模型和人-车交互模型，设计防碰撞函数和方向注意力函数为权重函数，引入注意力模块筛选对行人影响较大的行人和车辆数据构建一种结合车辆与行人的多类别社会注意预测模型。

(5) 将人 DUT 人-车交互数据集中的数据输入到第四章所构建的模型中作对比分析实验，通过定量误差实验和定性可视化实验，确定模型的参数值以及衡量本文所构建模型的优越性。

(6) 总结全文，对研究结果进行分析，针对其中的不足提出未来研究工作的建议。

2 深度学习与神经网络

2.1 神经网络与神经元模型

人工神经网络 (Artificial Neural Network, ANN)，也称为神经网络 (Neural Network, NN)，是一种模拟生物神经系统行为特征而人为建立的用于复杂模式和预测问题的数学模型。它以对大脑的生物学研究成果为基础，其目的在于模仿脑部神经系统的某些机制，从而实现所需的特定功能，目前常用的有 BP 神经网络 (Back Propagation Neural Network)、卷积神经网络 (Convolutional Neural Network, CNN) 和时间递归神经网络 (Recurrent Neural Network, RNN) 等。

神经网络是与大脑研究密切相关的，神经网络在结构上由大量模拟生物神经元的人工神经元构成，这种类脑结构使得神经网络有以下优点：

(1) 非线性：在神经网络中人工神经元具有两种不同的状态，分别为激活状态和抑制的状态，这两种不同的状态在数学上就是一种非线性关系。

(2) 分布存储和容错性：在神经网络中，数据按照数据内容的差异，分布存储在整个网络上的不同区域之中，而不是存储在某一特定位置。这种存储数据的方法能够适用于部分信息丢失或损坏的情况，因为神经网络可以恢复出原来的信息，保证系统正常运行。

(3) 并行处理：神经网络的各单元在结构上是并行的，它可以在运算过程中同时处理类似的数据运算过程，大大提高了运算速度。

(4) 自适应性，自学习性和自组织性：和人脑类似，神经网络中神经元之间的连接种类十分的丰富，且可塑性极强，可以按照工程人员的意愿自定义连接。神经网络可以根据设计要求和信息处理的特性，通过学习和训练，构建非线性复杂关系模型。

人工神经元模型是仿生学的一种应用，它模仿了人脑神经元的神经细胞，是人工神经网络中的基本信息处理单元。人工神经元模型如图 2.1 所示。

人工神经元的数学表达式如下：

$$v = \sum_{i=1}^m (W_i x_i) + b \quad (2.1)$$

$$y = f(v) \quad (2.2)$$

其中， x_i 为输入信号； w_i 为与第 i 个神经元相对应的权值； b 为神经网络的偏置； v 为预激活阶段的输入； f 为激活函数； y 为输出。

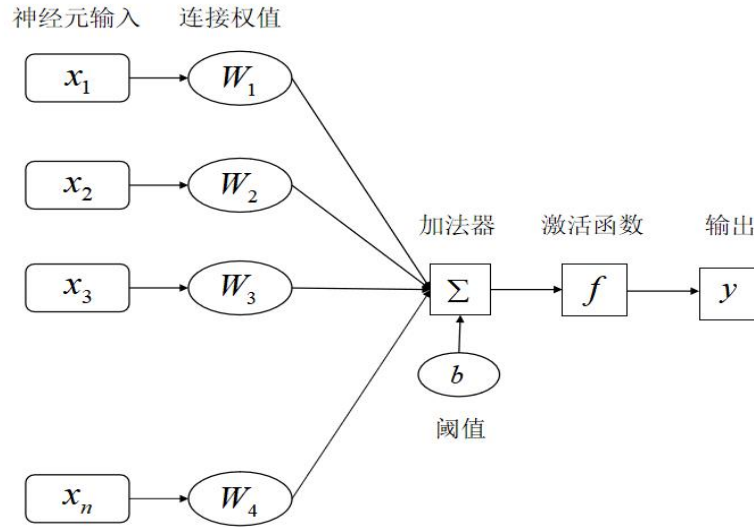


图 2.1 神经元结构图

Figure 2.1 Structure of neurons

人工神经元的数据处理过程可以分为两个步骤：

第一步是预激活阶段，接受来自其他 m 个神经元传递来的输入信息并进行加权求和处理。具体是将这些神经元输入与相对应的连接权重相乘通过加法器传递给下一步，公式如式（2.3）所示。

$$v = \sum_{i=1}^m (W_i x_i) + b \quad (2.3)$$

第二步是激活函数处理阶段，这一步将第一步得到的输出结果传递给激活函数。激活函数会将预激活得到的结果压缩到一个较小的范围内，如 $(-1, 1)$ 或 $(0, 1)$ ，通过这个经压缩的结果的数值大小可以判断神经元处于活跃状态还是抑制状态，然后将输出结果传递给下一层神经元。

$$y = f(v) \quad (2.4)$$

激活函数是神经网络中的一个运算函数，它完成了神经网络中的值从输入端到输出端的映射，对神经网络的运算有着重要的作用，是其核心单元。从生物学的角度上来看，生物大脑中的每一个神经元都有不同的分工，负责处理不同的任务。在进行某个特定任务的时候，一些神经元会处于异常活跃的状态，但与此同时，另一些神经元则呈现出抑制状态。类似地，在神经网络中，人工神经元依旧具有两种状态，分别为激活态和抑制态，神经元通过激活函数来体现自我学习能力。深度学习中，目前较为常用的激活函数主要有以下三种：

(1) 线性函数，`purelin` 函数。常被用于回归神经网络输出，可根据需要将神经元输出放大，表达式如式 (2.5) 所示。

$$y = f(v) = v \quad (2.5)$$

(2) S 型函数，在物理上最接近神经元，是历史上最常用的激活函数。常被用于用于隐层神经元输出，可分为 `logsig` 函数和 `tansig` 函数，表达式如式 (2.6) 所示。

$$y = f(v) = \frac{1}{1+e^{-v}} \quad \text{log sig 函数} \quad (2.6)$$

$$y = f(v) = \frac{1}{1+e^{-2v}} \quad \text{tan sig 函数} \quad (2.7)$$

(3) ReLU 激活函数，弥补了 S 型函数的梯度消失问题。常被用于隐层神经元输出，其表达式如式 (2.8) 所示。

$$y = f(v) = \max(0, v) \quad (2.8)$$

其函数曲线特点如图 2.2 所示：

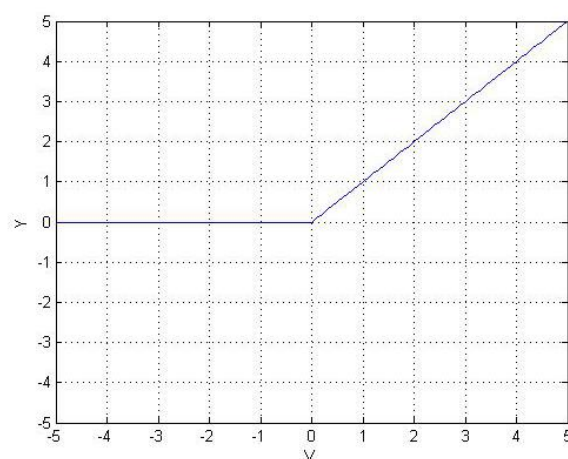


图 2.2 ReLU 激活函数图像

Figure 2.2 Image of RELU activation function

ReLU 激活函数有着近似生物神经元的特性，在大多数网络和应用中，取得了非常好的效果，如今正变得越来越流行。与传统的 s 型函数相比，ReLU 函数有以下优点：

(1) 单侧抑制。当输入值 $n < 0$ 时，神经元处于被抑制的状态，相反，当输入值 $n > 0$ 时，神经元处于被激活的状态。

(2) 稀疏激活性。相比其他的激活函数，ReLU 有着稀疏性的优势，使得其在实际的应用中其收敛速度更快。

(3) ReLU 激活函数具有宽阔的激活边界。不管是 logsig 函数还是 tansig 函数，激活态大都集中在中间较小的范围内，而 ReLU 则具有相对宽阔的激活边界，只要输入值 $n>0$ ，神经元都将被激活。

2.2 递归神经网络

2.2.1 RNN 模型

在经典的神经网络中，通过对模型参数的合理设置，在一定的训练轮数过后，可以得到较好的预测结果。但是，传统的神经网络如 CNN 神经网络在结构每层之间没有能够链接的节点，对数据在时间相关性上的记忆与分析能力较差^[60]。因此传统的神经网络不能很好的适用于行人轨迹这种时间序列特征比较强的时序数据。而 RNN 在处理和预测序列数据上有着显著优势，递归神经网络具有前馈传播和反馈传播，能够将之前输入得信息存储下来，并利用记住的信息影响后面节点的输出。图 2.3 为 RNN 神经网络的基本结构。

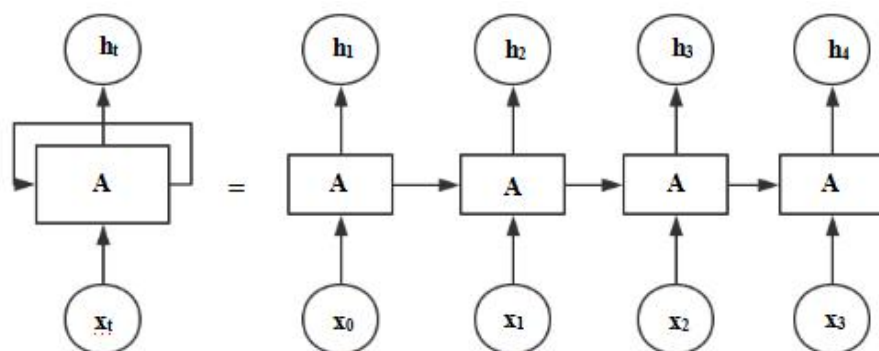


图 2.3 时序序列上的 RNN 结构展开图

Figure 2.3 RNN structure and sequence expansion

从图中可以看出，递归神经网络的主体结构 A 的输入除了输入的 x_t 之外，还有上一时刻的隐藏状态 h_{t-1} 。由于模块 A 中的权重和运算在各个时刻均是相同的，因此递归神经网络可以看作是同一神经网络结构不断复制的结果。理论上网络模型可以扩展到无限维，也就是无限长度的序列。RNN 具有如下两个特点：

(1) 持续性：在具有典型序列特征的数据中，前后数据之间并非相互独立的，而是相互关联的。当前阶段的输出结果会受到过去的决策的影响，同理当前阶段的输出结果也会影响到后续阶段的决策。

(2) 记忆性：RNN 可以保留序列的“记忆”信息。在处理序列数据时，RNN 处理信息的机制有着明显的循环性，这一点与人类大脑处理记忆的过程非常相似。人类的记忆会在大脑中不断地循环更新，称为日常生活中的经验知识。

由于 RNN 的这些特点，使得其在处理与时间序列相关的问题上准确率较高。但 RNN 也有着明显的缺陷，如果输入信息在时间上的跨度较长，会出现梯度消失现象，这种现象会导致距离当前时刻较远时刻的信息丢失，即标准 RNN 只能记忆较短时间周期内的数据信息，在长序列数据上效果并不好。例如在自然语言处理中，当一个单词能表达多种不同的含义时，为了确定其准确的意义，这时候不仅需要考察周围的单词，很可能还需要考察上下文的语境，比如更早的句子信息，甚至前一段的信息。在这种序列跨度较大的应用中，标准 RNN 的效果并不理想。

2.2.2 LSTM 模型

长短时记忆网络（Long Short-Term Memory, LSTM）的设计是为了解决标准 RNN 网络的长期依赖问题。LSTM 模型是由 Sepp Hochreiter 和 Jurgen Schmidhuber 于 1997 年提出的，它是一种特殊的循环体结构。标准 RNN 的重复模块非常简单，例如只有一个 \tanh 层，如图 2.4 所示。LSTM 的结构与标准 RNN 基本一致，但隐藏层的设计更加复杂。与标准 RNN 简单的重复结构不同，LSTM 的隐含层具有一个核心结构单元——细胞单元（cell），cell 由三个交互的“门”结构组成，通过着三个门，cell 将记住神经元需要它记住的信息，并将这些信息传递下去。图 2.5 所示为 LSTM 的结构。

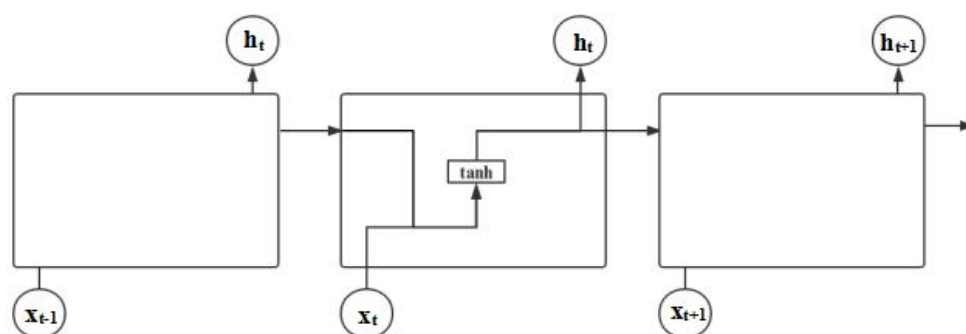


图 2.4 标准 RNN 的重复模块
Figure 2.4 Repetitive modules of RNN

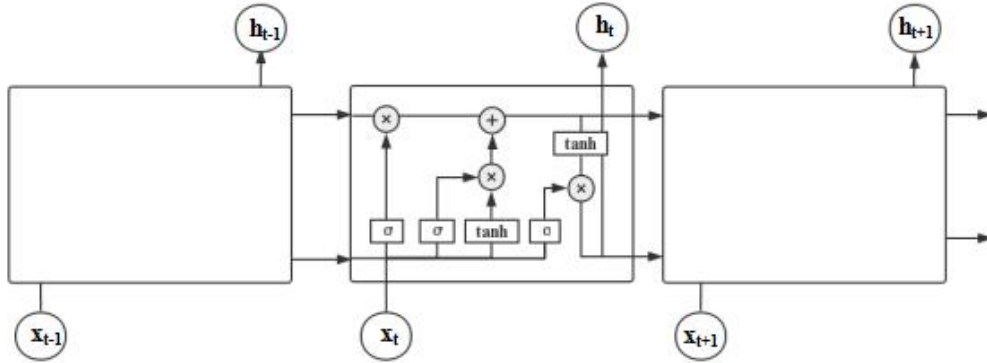


图 2.5 LSTM 的结构

Figure 2.5 Structure of LSTM

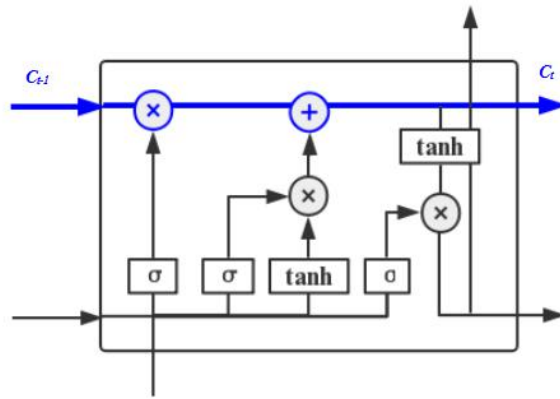


图 2.6 细胞单元

Figure 2.6 Structure of cell

LSTM 的隐含层由三个“门”结构组成，其结构是由一个 sigmoid 层和一个按位做乘法的操作组成，分别为遗忘门(Forget Gate)、输入门(Input Gate)和输出门(Output Gate)。这些“门”结构让有用的信息被留下来，无用的信息被遗忘掉，能够做到有选择性地需要的信息作为当前细胞的状态 C_t 的输出。LSTM 的工作过程可分为四个步骤。

(1) 遗弃部分信息。首先通过遗忘门来确定需要被遗弃的信息，如图 2.7 蓝线部分所示。遗忘门会根据当前输入 x_t 和上一时刻的输出 h_{t-1} 进行线性组合后，利用激活函数将函数值压缩得到 f_t ， f_t 的公式如式 (2.9) 所示，其取值大小在 0 和 1 之间。当其取值接近于 0 时，表示当前信息会被丢弃；当其取值接近于 1 时，表示当前信息会被保留。

$$f_t = \delta(W_f[h_{t-1}, x_t] + b_f) \quad (2.9)$$

其中， δ 表示 Sigmoid 激活函数， W_f 为 LSTM 的权重参数， b_f 为相对应的偏置。

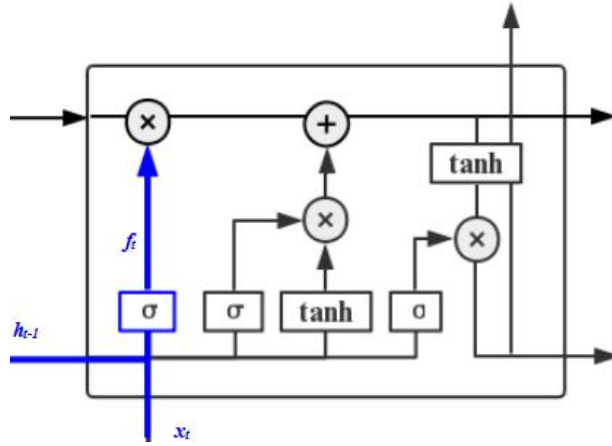


图 2.7 遗弃部分信息

Figure 2.7 Abandoned part information

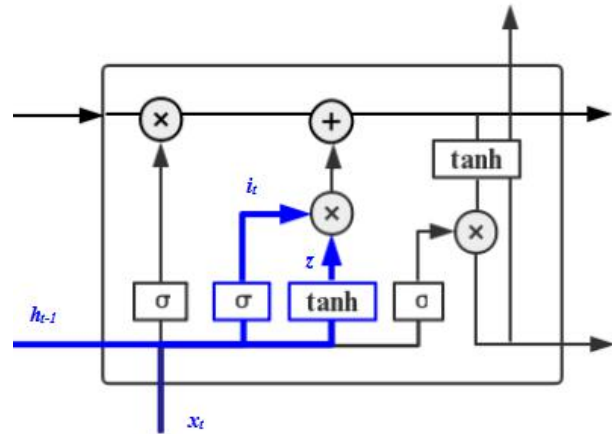


图 2.8 产生新的信息

Figure 2.8 Generating new information

(2) 产生新的信息。通过输入门决定哪些信息被添加，如图 2.8 蓝线部分所示。输入门会根据当前输入 x_t 和上一时刻的输出 h_{t-1} 计算得到 i_t ， i_t 的公式如式 (2.10) 所示，决定哪些信息加入到状态 c_{t-1} 中生成新的状态 c_t 。同时，当前输入 x_t 和上一时刻的输出 h_{t-1} 通过一个 \tanh 函数，产生一个输入状态值 z ， z 的公式如式 2.11 所示。

$$i_t = \delta(W_i[h_{t-1}, x_t] + b_i) \quad (2.10)$$

$$z = \tanh(W_z[h_{t-1}, x_t] + b_z) \quad (2.11)$$

其中， δ 表示 Sigmoid 激活函数， W_i 、 W_z 为 LSTM 的权重参数， b_i 、 b_z 为相对应的

偏置。

(3) 更新细胞单元的状态。如图 2.9 蓝线部分所示，根据上两步的输出结果将细胞状态从 c_{t-1} 更新到 c_t ，具体公式如式 (2.12) 所示。

$$c_t = f_t \cdot c_{t-1} + i_t \cdot z \quad (2.12)$$

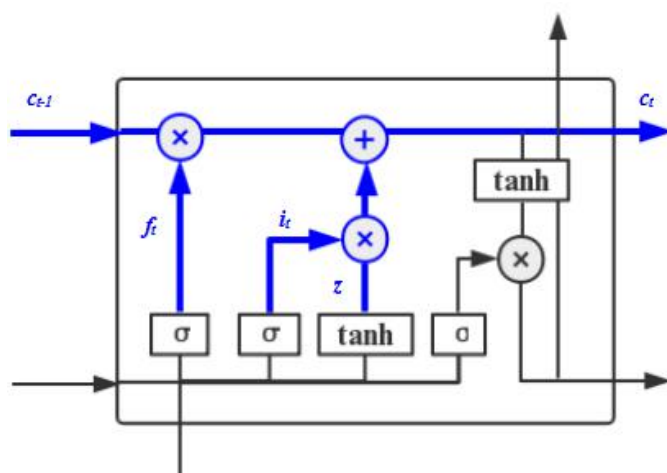


图 2.9 更新细胞的状态

Figure 2.9 State of new cells

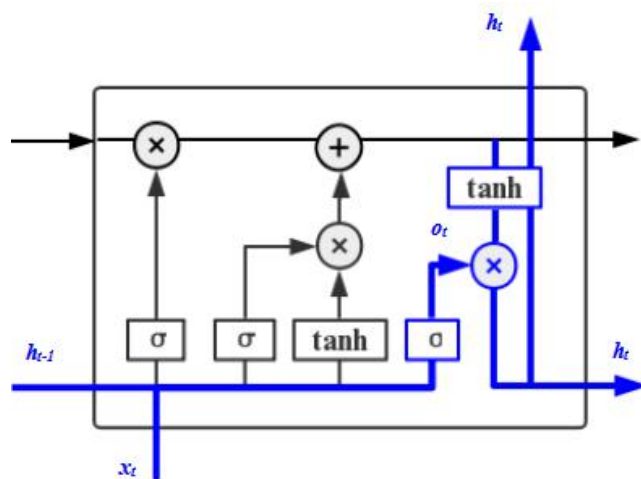


图 2.10 输出信息

Figure 2.10 Output information

(4) 输出信息。通过输出门决定哪些信息被输出，如图 2.10 蓝线部分所示。输出门会根据当前输入 x_t 和上一时刻的输出 h_{t-1} 计算得到 O_t ，具体公式如式 (2.13) 所示，决定细胞状态的哪些部分可以被输出。之后，把细胞状态通过 \tanh 函数进行处理并将

它和 O_t 相乘，得到最终输出，具体公式如式 (2.14) 所示。

$$o_t = \delta(W_o[h_{t-1}, x_t] + b_o) \quad (2.13)$$

$$h_t = o \cdot \tanh c_t \quad (2.14)$$

其中， δ 表示 Sigmoid 激活函数， W_o 为 LSTM 的权重参数， b_o 为相对应的偏置。

上述的每一步就是 LSTM 重复模块的计算过程，LSTM 是 RNN 的一个优秀的变种模型，继承了绝大部分 RNN 模型的特性，同时避免了标准 RNN 网络的梯度消失现象，解决了其长期依赖问题。LSTM 非常适合处理与时间序列高度相关的问题，在图像识别、语音识别及机器翻译等领域取得了不错的效果。

2.2.3 Bi-LSTM

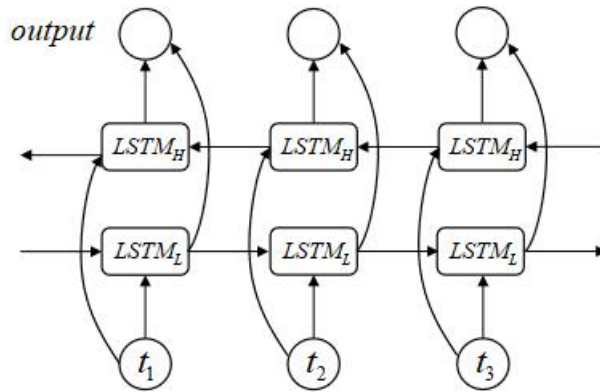


图 2.11 Bi-LSTM 结构图

Figure 2.11 Structure of Bi-LSTM

单向的 LSTM 网络根据接受序列的不同顺序可以训练不同的网络。对于序列 S 而言，单向的 LSTM 如果接收正向序列的 S ，则训练得到正向的 LSTM，其网络中的每个单元参考的是序列 S 中的上文信息，并传递给其后的单元；当接收的是反向序列 S 中的信息，则得到的是反向 LSTM，其网络中每个单元仅仅参考序列 S 中的下文信息，然后传递给其后的单元。因此单向的 LSTM 存在无法从后向前编码的问题，在更细粒度的分类中，无法捕捉双向的信息。Bi-LSTM 即 Bidirectional LSTM，Bi-LSTM^[61] 网络以单项 LSTM 网络为基础，将训练序列分为前后两个方向（即正向 LSTM 编码和反向 LSTM 编码），两个训练序列连接着同一个输出层，使得 LSTM 编码层能够获知前后时间序列的信息。其结构如图 2.11 所示。

2.2.4 混合 LSTM

递归神经网络拥有提取时序序列特征的能力，相比于 RNN 神经网络，LSTM 能够选择性地记住重要信息，忽略无用信息，防止产生梯度消失或梯度爆炸的现象，适用于目标轨迹预测模型。同时 Bi-LSTM 相比于 LSTM 能够传递双向信息，在提取社会信息特征时具有较好的表现。因此本文将两种网络模型结合起来，Bi-LSTM 作为社交信息编码层，以传达精度更高的社交信息，使用 LSTM 为轨迹预测层，提高轨迹预测的精度。

2.3 优化算法

2.3.1 梯度下降

梯度下降（Gradient descent）是神经网络中最常应用的优化算法之一，主要用于在神经网络训练中进行参数更新，使模型收敛。具体原理是在某一个方向上，不断更新整权重的参数，使得损失函数最小。梯度下降的迭代方法由函数的一阶导数（梯度）决定，假设现有 n 元函数 $f(x_1, x_2, \dots, x_n)$ ，函数在某点 $x^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ 处的梯度是一个向量：

$$d = \left(\frac{\partial f}{\partial x_1} \Big|_{x_1=x_1^0}, \frac{\partial f}{\partial x_2} \Big|_{x_2=x_2^0}, \dots, \frac{\partial f}{\partial x_n} \Big|_{x_n=x_n^0} \right)^T \quad (2.15)$$

该向量的方向就是函数 $f(x_1, x_2, \dots, x_n)$ 在该点 $x^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ 时函数值增长最快的方向。基于梯度的这个性质，如果能够把迭代的每一步沿着当前点梯度方向的反方向进行迭代，那么就能得到一个逐步减小的序列，最后就能到达局部的最小值。按照每次迭代所使用的数据集范围不同，可以把梯度下降算法分为：批量梯度下降（BGD）、随机梯度下降（SGD）和小批量梯度下降（MGBD）。

批量梯度下降训练时使用全部的数据集进行运算，这导致当数据量很大的时候，速度会非常慢。且不能以在线的方式更新模型，即当有新的训练数据加入时，需要对全量的数据进行更新，效率非常低，目前使用的比较少。随机梯度下降是对批量梯度下降的更新，SGD 的每一次更新，只考虑一个样本数据的误差损失，因此运算速度更快。更重要的是，它可以在线进行参数更新。小批量梯度下降结合了 BGD 和 SGD 的优点，克服了它们各自的缺点。MGBD 的更新策略是每次的参数更新，优化的目标函数由 n 个样本数据构成， n 取值一般在 10 到 500 之间。当前的梯度下降算法普遍采用了 MGBD 策略。

但这些优化方法仍有着一些缺点，它们都是针对迭代方向进行优化，且步长是固定的，所有的参数共享相同的学习率，在优化高度非凸的误差函数的函数时很可能被卡在鞍点的位置。

2.3.2 AdaGrad 优化算法

在以上基于梯度的优化算法中，所有的参数共享相同的学习率。但在神经网络模型中，不同的参数的重要程度是显著不同的，因此针对不同参数采取不同的学习率，能使得网络训练速度更快。针对简单的 MGBD 策略等方法存在的这类问题，2011 年 John Duch 等^[62]提出了自适应梯度策略（Adaptive Gradient, AdaGrad）。AdaGrad 是一种自适应的调整学习率的方法，可以针对不同的参数进行学习率的调整。通常情况下，AdaGrad 优化算法在训练初期的时候会激励收敛，随着训练的不断进行会惩罚收敛，速度越来越慢。以下是 AdaGrad 的每一次的迭代公式：

$$\text{sum}g_i = \text{sum}g_{i-1} + (g(\theta_{i-1}))^2 \quad (2.16)$$

$$\theta_i = \theta_{i-1} - \frac{lr}{\varepsilon + \sqrt{\text{sum}g_i}} \times g(\theta_{i-1}) \quad (2.17)$$

其中， $\text{sum}g_i$ 是累加的梯度， $g(\theta_{i-1})$ 是 θ 参数在 $i-1$ 时间步的梯度， lr 是全局学习率， ε 是一个非常小的常数，目的是为了防止分母为 0。

当迭代的时间愈长，累加的梯度会越大，使得学习率会随着时间减小，在接近目标函数的最小值时，不会因为学习率过大而越界。如果当前参数的 $\text{sum}g_i$ 值比较小，则采用较大学习率，即使用更大步长进行参数迭代；相反，如果参数的 $\text{sum}g_i$ 值比较大，则适当采用较小的学习率，在进行参数迭代的时候使用小步长，因此 AdaGrad 优化算法更适合处理稀疏矩阵。因为对于稀疏权重参数，对模型影响较小的参数其 $\text{sum}g_i$ 值会比较小，在训练时可以采用大学习率；而对于对模型影响十分大的参数，其 $\text{sum}g_i$ 值会比较大，对于这些参数的训练要选取较小的学习率。

2.3.3 RMSprop 优化算法

AdaGrad 优化算法用于凸函数时能快速收敛，而当应用于非凸问题时，梯度下降过程中可能穿过不同的区域，学习率不断下降，以至于在找到全局最小值区域之前学习率就过小，导致最终收敛缓慢。不同于 AdaGrad 算法里累加梯度 $\text{sum}g_i$ 是截至时间步 i 所有小批量随机梯度 $g(\theta)$ 取平方和累加，RMSProp 算法将这些梯度在取平方和后做指数衰减平均。具体来说，RMSProp 算法在给定时间步 i 的累加梯度如式（2.18）所示：

$$\text{sum}g_i = \rho \times \text{sum}g_{i-1} + (1 - \rho) \times (g(\theta_{i-1}))^2 \quad (2.18)$$

其中， $0 \leq \rho < 1$ ，其他的参数和 AdaGrad 算法相同。

RMSProp 算法与 AdaGrad 算法相同，它按元素运算重新调整目标函数自变量中的

每个元素的学习率，然后更新自变量。

$$\theta_i = \theta_{i-1} - \frac{lr}{\varepsilon + \sqrt{\text{sum}g_i}} \times g(\theta_{i-1}) \quad (2.19)$$

因为 RMSProp 算法的累加梯度 $\text{sum}g_i$ 是对平方项 $(g(\theta_{i-1}))^2$ 的指数衰减平均，这样的做法使得优化过程中梯度摆动幅度大大减小，提高了收敛速度。

2.4 TensorFlow 平台

TensorFlow 是由 Google 人工智能团队在 Google 自身第一代深度学习系统 DisBelief 的基础上改进而来的通用计算平台^[63]。在谷歌内部，DisBelief 取得了巨大的成功，但由于其过于依赖谷歌内部的系统框架，很难对外开源。谷歌大脑团队对 DisBelief 进行了改进，并在 2015 年 11 月开源的第二代人工智能学习系统——TensorFlow。相比于前作，TensorFlow 可移植性更好即计算模型更加通用；在性能上也有显著改进，计算速度更快、系统稳定性更高；构架灵活性也得到增强，能够支持更多的平台和算法。因为 TensorFlow 的这些优势，使得其受到了科研学术界和工业界的广泛关注。在各种开源社区当中，TensorFlow 的活跃程度都要远远超过大部分其他工具。到 2019 年 5 月为止，GitHub 上已经有 59000 多个代码库中提到了 TensorFlow，其中只有很少的一部分是谷歌官方提供的。TensorFlow 的优点可以简单地概括如下：

(1) 高度的灵活性。用户在使用 TensorFlow 时，不仅可以实现并训练各种神经网络的模型，还可以按照个人所需完成其他的计算任务。开发人员只需要将自己设计的数学模型转换成计算图的形式就可以在 TensorFlow 上完成运算。

(2) 可移植性强。TensorFlow 支持的设备类型十分广泛，既可以在 CPU 上运行也可以在 GPU 上运行。此外其还支持多种操作系统，如 Linux、Mac OS X、Android、IOS 等等。因此其可以在个人 PC、服务器和手机等多种设备上使用。

(3) 提供了大量神经网络模型。在 TensorFlow 的帮助下，企业开发人员和科研工作者可以省去写模型底层实现的繁琐工作，节约了时间成本，提高了研发效率。

(4) 自动求微分。大量神经网络的训练都采用了梯度下降的方法。在使用 TensorFlow 训练网络的时候，开发人员只需要选择损失函数、定义网络中哪些权重参数需要训练，TensorFlow 就能够自动求微分利用梯度下降的方法训练模型。

(5) 性能优化。深度学习往往需要大量数据的运算，这对运算设备的性能要求很高。当用户使用一些多 CPU 和 GPU 的运算平台时，无论是多线程、队列或是异步操作等等，TensorFlow 都能很好地实现。

(6) 多语言支持。用户在使用 TensorFlow 时，可以灵活地选择使用 Python 或是 C++ 的代码来编写程序。

2.5 本章小结

本章首先分析了神经网络的理论基础，其次分析了本文所使用的编码网络的基本原理，然后分析了神经网络常用的优化算法，最后介绍了本文所使用的 Tensorflow 框架，为第四章的预测模型的搭建提供了技术支持。具体结论如下：

(1) 神经网络具有非线性，分布储存和容错性，并行处理，自学习和自适应性的优点，适合于应用在基于数据驱动的轨迹模型预测问题中；

(2) 在递归神经网络中，相比于 RNN 神经网络，LSTM 能够选择性地记住重要信息，忽略无用信息，防止产生梯度消失或梯度爆炸的现象。同时 Bi-LSTM 相比于 LSTM 能够传递双向信息，可以提高轨迹预测的精度，因此本文采用 Bi-LSTM 作为社交信息编码层，LSTM 为轨迹预测层；

(3) RMSProp 算法将梯度在取平方和后做指数衰减平均，能使梯度快速收敛进而显著提高计算效率，因此本文选择 RMSProp 算法作为神经网络优化算法。

3 行人及车辆轨迹获取

本文提出的轨迹预测模型针对于人-车交互运动，而现有的公开数据集 ETH, UCY 都只包含行人的数据，缺乏车辆轨迹信息，因此无法满足本文的数据需求，因此本章构建了适用于人车交互的大连理工大学（DUT）数据集。

3.1 数据集获取

DUT 数据集是在中国大连理工大学（DUT）校园的两个拥挤位置收集的。一个位置包括在没有交通信号的交叉口处的人行横道区域。由于没有交通信号灯，当行人与车辆交互时，理论上汽车和行人没有通行的优先权，因此二者的交互行为可以体现人车交互的“社会性”。另一个位置是环形交叉口附近，这个路口有相对较大的行动空间，行人和车辆可以在其中自由移动，而不受制于路面，可以得到多个方向上的预测结果。本章用一架 DJI-Mavic-Pro 无人机作为俯拍设备在感兴趣的区域上空的一个行人和车辆察觉不到的地方拍摄，以达到不会因为无人机的出现影响行人运动的目的。视频分辨率为 1920×1080 ，fps 为 23.98。行人主要是刚下课离开教室的大学生，车辆是穿过校园的车辆。无人机设备的硬件参数如表 3.1 所示。

表 3.1 无人机设备硬件参数表

Table 3.1 Hardware parameter of DJI-Mavic-Pro

参数名称	参数值
无人机续航时间	31 min
无人机续航里程	18 km(无风环境)
无人机飞行高度	60 m
传感器类型	CMOS
镜头	FOV78.8°28 mm
视频分辨率	4096×2160
帧率	23.98 fps
输出图像格式	JPEG, DNG
输出视频格式	MP4, MOV

视频拍摄完成之后，在所拍的视频中剪辑出包含人-车交互的视频片段，经过处理后，共有 17 个人行横道场景片段和 11 个圆弧空间场景片段，包含了 1793 个轨迹，一些剪辑中甚至包含两辆以上的车辆同时与行人交互，如图 3.1 所示。

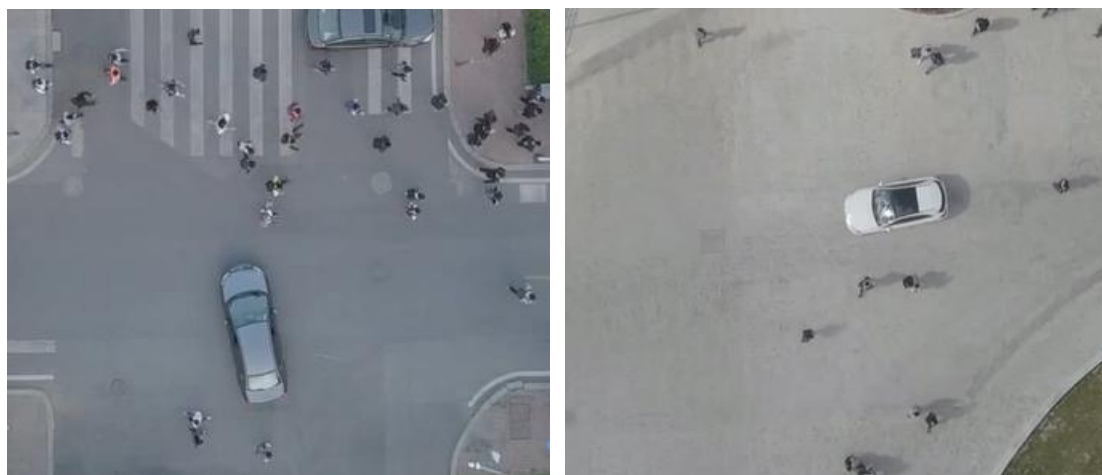


图 3.1 DUT 数据集俯拍图

Figure 3.1 Top view of DUT data set

3.2 轨迹提取

3.2.1 视频稳像

由于无人机受风速，自身电动机的运动的影响能够产生不稳定的运动，进一步形成噪声，所以本章需要对原始视频进行稳定处理。该过程包括三个步骤，首先设置第一帧 F_1 作为参考帧，对于接下来的每一帧 F_i ， $i=2,3,4...$ ，将使用尺度不变特征变换（Scale Invariant Feature Transform, SIFT）算法寻找图像中的关键特征点，然后用 K 最近邻算法（KNN）算法寻找两帧之间相匹配的关键特征点，最后使用随机抽样一致算法（RANDOM Sample Consensus, RANSAC）每个视频帧和第一个视频帧（参考帧）之间的透视变换计算当前帧和参考帧之间的透视变换矩阵，通过透视变换固定视角，得到稳定的视频。

SIFT 算法具有多量性、高速性、可拓展性的特点，能够适用于数据量庞大，关键点清晰的数据，其工作是在图像的尺度空间上通过高斯微分函数来捕捉一些对于尺度和旋转不变的关键兴趣点，这些特征点一般为像素变化比较大的点如边缘点等，这些点不会因为外界影响如光照，拍摄角度，噪音等发生变化。SIFT 算法识别这些特征点之后，拟合出关键点的尺度以及位置，再通过计算局部图像的梯度方向，计算关键点的方向。

找出关键点之后，本章采用 KNN 算法寻找特征点之间的关系进行匹配。KNN 是一种通过测量每个样本之间的特征的距离来进行样本分类的算法。该算法是计算一个样本特征空间内与训练集样本的特征空间内特征点的距离（通常为欧氏距离和马氏距离），

距离最短的 K 个训练集样本中的大多数所在的类别为该样本的类别。通常情况下 $K \leq 20$ ，且 K 为正整数。

通过 KNN 匹配法，可以得到当前帧和参考帧的互相匹配的关键点，理论上可以计算不同帧之间关键点变化的透视矩阵。为了消除匹配不佳甚至匹配错误的关键点，本章使用 RANSAC 算法排除掉这些“局外点”来提高匹配精度，消除错误匹配。

最后计算 F_i 到 F_{ref} 的透视变换矩阵 M_i ，利用透视变换将 F_i 帧校准为稳定 F_{cal_i} ， $i=2,3,4,\dots$ ，达到视频稳像的目的。

3.2.2 轨迹跟踪

视频稳定后，本章将使用有信道和空间可靠性 (CSR-DCF) [64] 的判别相关滤波器对行人和车辆进行自动跟踪。在跟踪过程中，稳像后的视频将会被切割成很多小片段，并划分出固定的感兴趣区域，一旦行人或车辆出现在感兴趣中，就会手动给出初始位置。其中行人的位置为以行人的头部正中心的矩形框，车辆的位置约定为汽车四个角点的中心，四个角点构成车辆的矩形框，如图 3.2 所示，当他们推出感兴趣区域，停止追踪。

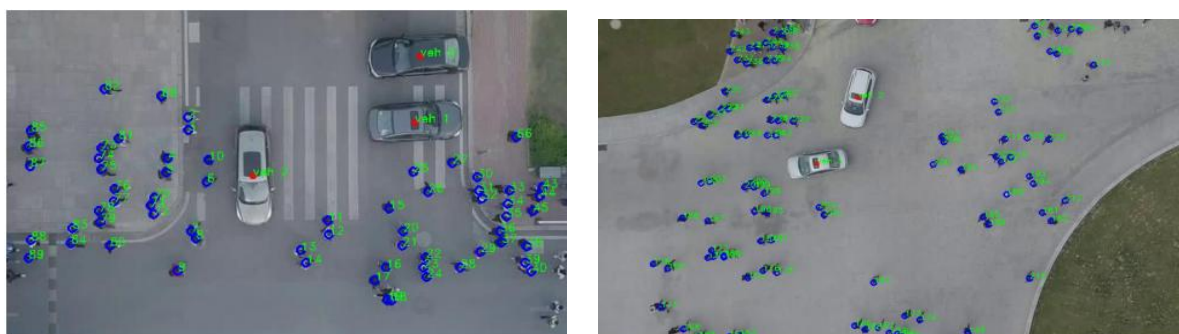


图 3.2 手动目标检测图

Figure 3.2 Manual target detection results

轨迹跟踪实际上就是一个目标追踪问题，上文中的初始位置确定代表行人和车辆的矩形框之后的每一帧需要跟踪算法紧跟住这个框，轨迹跟踪算法主要分为生成式跟踪和判别式跟踪。生成式跟踪算法的基本原理是在当前帧对目标区域进行建模处理，在接下来的每一帧全地图搜索与前一帧建好的模型最相似的区域，这一区域即为下一帧跟踪的目标，这种算法效果较差，应用不如判别法广泛。判别法利用分类器原理，在当前帧将目标区域和不包含目标的背景区域设置为正训练样本和负训练样本，利用机器学习来训练分类器，接下来的每一帧用训练好的分类器寻找图像中的最优区域。在判别类算法中，

相关滤波算法（discriminative correlation filter, DCF）算法有着出色的表现，因此本节采用 DCF 算法追踪目标轨迹。

DCF 算法最早应用于信号处理，用来描述两个信号之间的相关程度。DCF 的基本思路是通过一段跟踪好的目标区域，提取其中的样本特征，通过特征训练出一个滤波器的模板，模板特征为： $h = \{h_d\}_{d=1:N_d}$ 。在接下来待跟踪的新帧中，提取上一帧的目标区域位置的特征 $f = \{f_d\}_{d=1:N_d}$ ，与滤波器模板作相关，相关最强的地方为新帧中目标的中心位置 x ，计算公式如式（3.1），然后以该位置为中心提取出特征，反过来重新对滤波器模型进行训练，在下一帧重复以上步骤，并进行模型训练与目标跟踪。通过这种方法，可达到实时在线目标跟踪的目的。

在多通道情况下，目标函数为

$$\begin{aligned} \arg \min_h \sum_{d=1}^{N_d} \|f_d * h_d - g\|^2 + \lambda \sum_{d=1}^{N_d} \|h_d\|^2 \\ = \arg \min_h \sum_{d=1}^{N_d} \left(\left\| \hat{h}_d^H \text{diag}(\hat{f}_d) - \hat{g}_d \right\|^2 + \left\| \hat{h}_d \right\|^2 \right) \end{aligned} \quad (3.1)$$

其中， g 表示期望的响应输出， λ 表示正则项用于防止过拟合。

3.2.2 信道可靠性

上文设计的算法中，各个通道之间是独立的，因此在检测计算中每个通道的应该做加权叠加，CSR-DCF 算法中将权重设计为学习信道可靠性度量和检测可靠性度量的乘积。学习信道可靠性的响应计算很简单，即 f_d 和 h_d 的点乘，即响应越大可靠性越大，因此可靠性权重如下所示：

$$\omega_d = \varsigma \max(f_d * h_d) \quad (3.2)$$

其中 ς 是一个标准化函数，它保证了所有信道的权重之和为 1。

检测可靠性是当前检测帧的特征和 h 进行点成，得到响应 map 图，这里计算第二主振型和第一主振型的峰值比，比值越小，跟踪效果越好，可靠性越高，因此检测可靠性可以由式（3.3）计算：

$$\omega_d^{(\text{det})} = 1 - \min(\rho_{\max 2} / \rho_{\max 1}, 0.5) \quad (3.3)$$

3.2.1 空间可靠性

为了抑制边界效应，CSR-DCF 提出了空间可靠性图，设计了与每一个特征点信息 x 和目标外观相关的不规则 mask ，根据贝叶斯原理，其展开式如式（3.4）所示：

$$p(m=1|y, x) \propto p(y|m=1, x)p(x|m=1)p(m=1) \quad (3.4)$$

第一项 $p(y|m=1, x)$ 为外观似然模型，根据贝叶斯原理从目标的前景和背景的颜色直方图中计算。第二项表示高可靠性区域的位置的概率，即约靠近目标中间部分的概率越大，越远离目标中心的概率越小。第三项 $p(m=1)$ 为先验概率，是提取前景/背景直方图的区域大小之比。

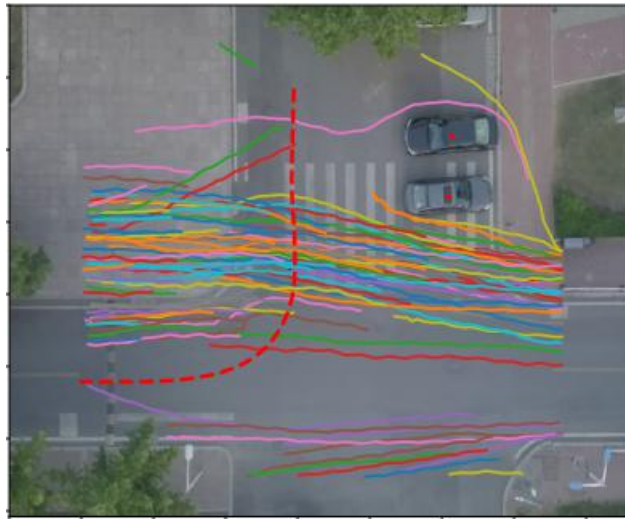


图 3.3 十字路口车辆（红色虚线）和行人（彩色实线）的轨迹

Figure 3.3 Trajectories of vehicles (red dashed line) and pedestrians (colorful solid lines) in a clip of the intersection scenario.

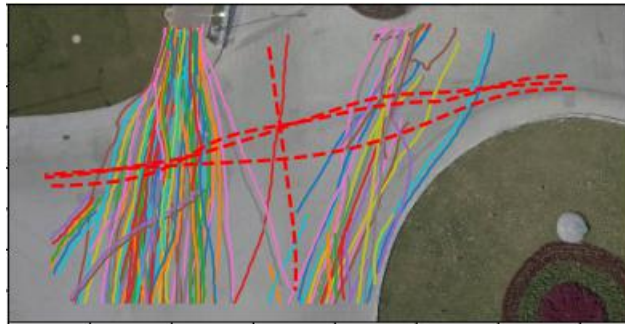


图 3.4 圆弧路口车辆（红色虚线）和行人（彩色实线）的轨迹

Figure 3.4 Trajectories of vehicles (red dashed line) and pedestrians (colorful solid lines) in a clip of the shared space scenario.

对于每个独立的信道，空间可靠性映射图 m 标识在学习过程中应该忽略的滤波器中的像素，引入对偶变量 h_c ，约束 $hc \equiv m \odot h$ ，其中 m 为 mask 矩阵，得到增广拉格朗日方程如式 (3.5) 所示

$$L(\hat{h}_c, h, \hat{l} | m) = \left\| \hat{h}_c^H \text{diag}(\hat{f}) - \hat{g} \right\|^2 + \frac{\lambda}{2} \|h_m\|^2 + \left[\hat{l}^H (\hat{h}_c - \hat{h}_m) + \overline{\hat{l}^H (\hat{h}_c - \hat{h}_m)} \right] + \mu \|\hat{h}_c - \hat{h}_m\|^2 \quad (3.5)$$

其中，其中， $hm=(m \odot h)$ ， h 为滤波器， h_c 的主要作用是让算法能够收敛 \hat{l} 是一个复数拉格朗日乘子， $\mu > 0$ ，字母上面的横杠表示共轭矩阵，字母右上方的 H 表示共轭转置矩阵。

式 (3.5) 每次迭代时可以变为：

$$\hat{h}_c^{i+1} = \underset{h_c}{\operatorname{argmin}} L(\hat{h}_c, h^i, \hat{l}^i | m) \quad (3.6)$$

$$h^{i+1} = \underset{h}{\operatorname{argmin}} L(\hat{h}_c^{i+1}, h, \hat{l}^i | m) \quad (3.7)$$

拉格朗日因子更新方程为：

$$\hat{l}^i + \mu (\hat{h}_c^{i+1} - \hat{h}^{i+1}) \quad (3.8)$$

式 (3.6)，式 (3.7) 中的最小封闭解为：

$$\hat{h}_c^{i+1} = (\hat{f} \odot \bar{\hat{g}} + (\mu \hat{h}_c^{i+1}, -\hat{l})) \odot^{-1} (\hat{f} \odot \bar{\hat{f}} + \mu^i) \quad (3.9)$$

$$h^{i+1} = m \odot F^{-1} [\hat{l}^i + \mu^i \hat{h}_c^{i+1}] / \left(\frac{\lambda}{2D} + \mu^i \right) \quad (3.10)$$

经 CSR-DCF 方法对行人和车辆进行目标追踪后的结果如图 3.3，3.4 所示（其中虚线代表车辆的轨迹，不同颜色的实线代表行人的轨迹。

3.3 卡尔曼滤波

卡尔曼滤波算法能够将信号中干扰信号滤除，对信号进行最优化估计，本节利用卡尔曼滤波算法对上节目标追踪得到的轨迹进行去除噪声，细化轨迹的处理。卡尔曼滤波算法的思想是将新的数据即当前时刻观测值和前一时刻的最佳估计值，依托目标的状态转移方程，进行线性优化，得到当前时刻的最优估计值。由此可见卡尔曼滤波不必考虑过去时刻的所有的测量数据，只需要递推计算，即可得到后续的数据的最优估计值。

3.3.1 行人轨迹处理

在行人轨迹问题上，可以将行人简化成一个质点，卡尔曼滤波方程为：

$$\hat{x}_t = A x_{t-1} + w \quad (3.11)$$

其中， x_{t-1} 为 $t-1$ 时刻的系统的最佳估计值， \hat{x}_t 为通过上一时刻最优估计值计算出的当前系统估计值， A 是状态转移矩阵， w 是状态转换噪声。

由运动学方程可知，本文的卡尔曼滤波问题中的状态转移方程为：

$$\hat{y}_t = y_{t-1} + v_t + w_1 \quad (3.12)$$

$$\hat{v}_t = v_{t-1} + a_t + w_2 \quad (3.13)$$

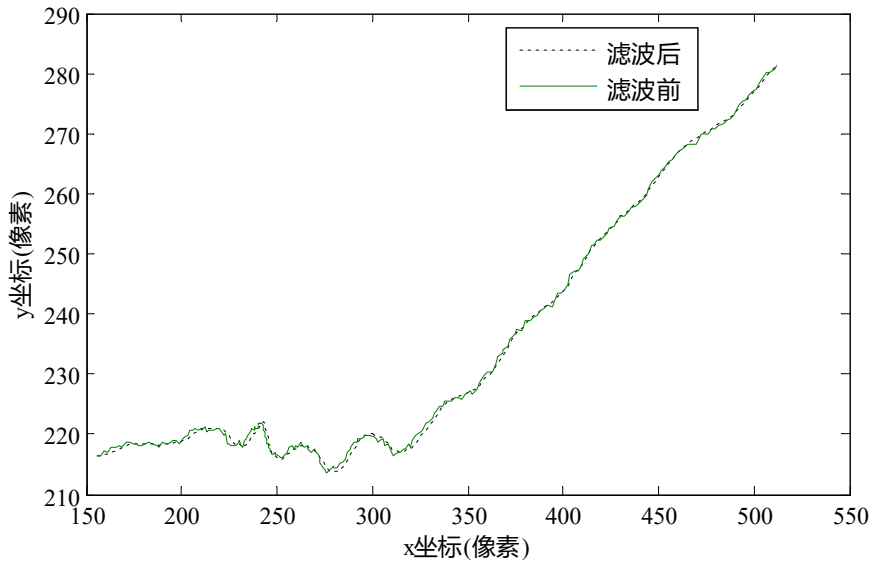


图 3.5 卡尔曼滤波前后行人位置对比图

Fig3.5 Comparison of pedestrian position after Kalman filter

其中 $y_t \in R^2$ 是行人位置坐标， $v_t \in R^2$ 是行人的速度， $w = [w_1, w_2] \sim N(0, Q)$ ，由于行人速度比较慢，因此假设行人是匀速运动的，即 $a_t = 0$ ，则可以得到系统最优估计值和状态转移矩阵的表达式如下：

$$x_k = [y_k, v_k] \quad (3.14)$$

$$A = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \quad (3.15)$$

当前状态的测量方程为：

$$z_t = Hx_t + r_t \quad (3.16)$$

其中 z_k 是系统测量值，即目标追踪后得到的行人的轨迹， $H = [1 \ 0]$ 为系统测量矩阵， $r \sim (0, R)$ 为测量噪声。

则 t 时刻，系统的最优估计值方程为：

$$x_t = \hat{x}_t + K_t(z_t - H\hat{x}_t) \quad (3.17)$$

其中 K_t 是卡尔曼增益。

经卡尔曼滤波处理后的行人轨迹对比图如图 3.5 所示。

3.3.2 车辆轨迹数据处理

受到车辆本身为刚体的性质的影响, 汽车的运动受到一定限制, 例如汽车的横向运动速度不能突然改变, 因此在车辆运动轨迹卡尔曼滤波模型研究中, 采用非线性运动的自行车模型, 其建模方式如下:

$$\hat{y}_x^t = y_x^{t-1} + v_t \cos(\theta + \beta) + w_1 \quad (3.18)$$

$$\hat{y}_y^t = y_y^{t-1} + v_t \sin(\theta + \beta) + w_2 \quad (3.19)$$

$$a_t = a_{t-1} \quad (3.20)$$

$$\hat{\theta}_t = \frac{v_t}{l_r} \sin \beta + w_3 \quad (3.21)$$

$$\hat{v}_t = v_{t-1} + a_t + w_4 \quad (3.22)$$

$$\beta = \arctan\left(\frac{l_r}{l_f + l_r} \tan \delta_f\right) \quad (3.23)$$

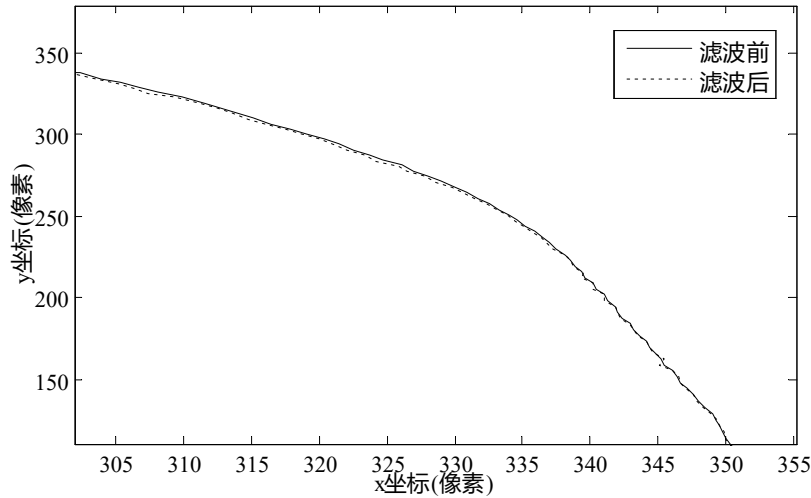


图 3.6 卡尔曼滤波前后车辆位置对比图

Figure 3.6 Comparison of vehicle position after Kalman filter

因此在车辆轨迹问题中, 系统状态变量 $x_t = [y_x^t \ y_y^t \ v_t \ a_t]$, y_x^t , y_y^t 为车辆在 t 时刻, x, y 方向的位移, v_t 是汽车速度, β 是相对于汽车重心的车速夹角, l_f 和 l_r 是汽车重心距离前轮和后轮的距离, a 是汽车加速度, δ_f 是汽车的前轮的转向角,

$w = [w_{t1}, w_{t2}, w_{t3}, w_{t4}] \sim N(0, Q)$ ，这里假设输入的前轮转向角为 0° 。

对车辆数据进行卡尔曼滤波处理后的结果如图 3.6 所示。

3.4 本章小结

本章主要构建了 DUT 人-车交互轨迹数据集。首先分析了无人机设备的基本参数和交通场景的特点，获取数据集。然后通过视频稳像，目标追踪的方法来提取行人和车辆的轨迹位置。最后使用卡尔曼滤波算法对得到的行人和车辆的轨迹进行处理，为下一章预测模型提供数据支持。主要工作如下：

(1) 使用无人机在放学时间的人员和车辆稠密的交通环境下，选取交叉路口和环形路口两个交通场景，进行俯拍，采集视频数据；

(2) 通过视频稳像，相关滤波算法进行目标追踪，得到行人和车辆在一段时间内的位置轨迹；

(3) 将处理后的算法经过卡尔曼滤波算法对轨迹进行平滑处理，将处理后的数据作为神经网络训练和测试的数据支持。

4 基于人-车交互的行人轨迹预测模型

行人在复杂交通环境中行驶时，会根据周围其他车辆和行人的行为来调整自身的行为。例如行人之间的结伴行走，避免碰撞，例如车辆加速时，行人会减速或停步躲避车辆，来保证交通安全，车辆加速时，行人会加速快步同行来保证交通效率。因此行人的轨迹变化与周围车辆，周围行人的轨迹密不可分。这种轨迹变化不能单独地通过观察当前行人的历史轨迹来预测，还需要将周围行人和车辆的轨迹考虑在内。

本文的算法流程可以大体分为两个模块，数据提取及处理以及递归神经网络预测模型。数据提取及处理的目的是对历史轨迹数据进行预处理。原始数据集中的行人和车辆轨迹信息有诸多的特征，如车辆位置、车辆速度、车辆的偏向角、车辆编号、行人编号、行人位置等。当输入特征过多的时候，会使得网络结构过于复杂，难以收敛，因此要进行特征提取的操作。之后根据本文采用的训练方法，进行数据集的划分，便于后续工作的展开。由于原始数据的范围过大，会对模型的预测结果造成较大影响，因此采取归一化处理。递归神经网络预测模型的作用是产生行人的未来轨迹。基于处理过后的数据集，提取出自身行人的历史轨迹以及相邻车辆和行人的相关信息，得到输入数据序列，将数据序列输入到轨迹预测网络模型中，对网络进行训练优化网络参数。网络训练完成后，基于训练好的模型得到未来时刻行人的轨迹位置。

4.1 算法流程

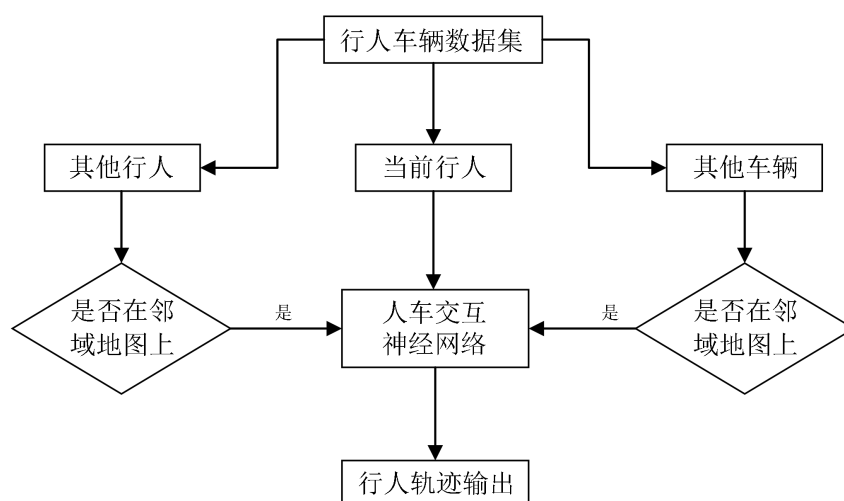


图 4.1 算法流程图

Fig. 4.1. Flow diagram for the algorithm

本文的算法流程可以大体分为两个模块，数据提取及处理以及递归神经网络预测模型，具体如图 4.1 所示。

(1) 数据提取及处理，该部分目的是对历史轨迹数据进行预处理。原始数据集中的行人和车辆轨迹信息有诸多的特征，如车辆位置、车辆速度、车辆的偏向角、车辆编号、行人编号、行人位置等。当输入特征过多的时候，会使得网络结构过于复杂，难以收敛，因此要进行特征提取的操作。之后根据本文采用的训练方法，进行数据集的划分，便于后续工作的展开。由于原始数据的范围过大，会对模型的预测结果造成较大影响，因此采取归一化处理。

(2) 递归神经网络预测模型，该部分的作用是产生行人的未来轨迹。基于处理后的数据集，提取出自身行人的历史轨迹以及相邻车辆和行人的相关信息，得到输入数据序列，将数据序列输入到轨迹预测网络模型中，对网络进行训练优化网络参数。网络训练完成后，基于训练好的模型得到未来时刻行人的轨迹位置。

4.2 数据提取及处理

本文第二章构建的数据集包含了大量车辆，行人的轨迹信息，但过多的输入数据会使得网络结构复杂、规模过大，导致网络不易收敛。此外，若各个输入数据之间彼此有高度的相关性，会造成数据的重复使用，导致网络计算时间过长。为方便模型的训练和测试，本文仅留下时间、行人或车辆 id、x 坐标、y 坐标，车辆车速等数据。由于下课期间，学生人流量很大，会导致网络运算过慢，因此选择车辆和行人发生交互的片段的时间内的轨迹。

行人行走过程中，若没有其他障碍物阻挡，会沿自己的运动方向一直前进，若遇到其他车辆或行人，会修正自己的运动轨迹，这种行为最直观的体现在行人前进的方向和速度上。行人会依据周边环境，修正自己的方向和速度，进而改变自己未来的轨迹。因此在本文的行人轨迹预测问题中，将行人轨迹预测问题作为时间序列模型。

首先假定每个场景中的行人以当前场景为标准坐标系，定义每个时间（以帧为单位）的所有行人坐标。在任意时间 t ，定义第 i 个行人的 x, y 坐标为 (x_t^i, y_t^i) 。将一段行人时间序列分为观察帧和预测帧，取 $t \sim t + obs$ 为观察帧， $t + obs + 1 \sim t + obs + pred$ 为预测帧，其中 $obs, pred$ 为观察帧和预测帧的长度，单位为帧数。则预测问题可以被分为分段预测问题，其输入数据是行人在过去一段时间，每一帧相对于上一帧，行人的转向角和行人的速度。具体公式如下：

$$person_{obs}^i = \{x_{t+obs}^i, y_{t+obs}^i\} \quad (4.1)$$

$$x_{pred}^i = \{x_{t+obs+pred}^i, y_{t+obs+pred}^i\} \quad (4.2)$$

$$v_{obs}^i = \{vx_{t+obs}^i, vy_{t+obs}^i\} \quad (4.3)$$

$$angle_{obs}^i = \left\{ \arctan \left(\frac{y_{t+n}^i - y_{t+n}^i}{x_{t+n}^i - x_{t+n}^i} \right) \right\} \quad (4.4)$$

$$input_t^i = \{angle_{obs}^i, v_{obs}^i\} \quad (4.5)$$

$$output_t^i = \{angle_{pred}^i, v_{pred}^i\} \quad (4.6)$$

其中, $person_{obs}^i$ 是观察帧序列中的行人 i 位置, v_{obs}^i 是观察帧序列中行人 i 的速度, vx_t^i 是 t 时刻, 行人 i 在 x 轴方向的速度, 单位为 m/s ; vy_t^i 是 t 时刻, 行人 i 在 y 轴方向的速度, 单位为 m/s ; $angle_{obs}^i$ 是观察帧序列中, 行人的方向角, 单位为 rad ; $input_t^i$ 是行人 i 的输入序列, $output_t^i$ 是行人 i 的输出序列。

4.3 网络基本结构

如图 4.2 所示, 为本文的技术路线图, 轨迹预测模型的输入包含行人轨迹和汽车轨迹。

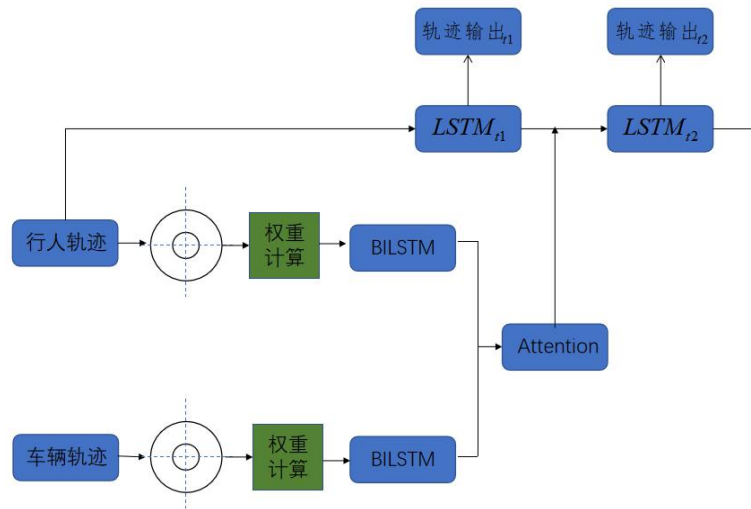


图 4.2 网络总体结构图

Figure 4.2 Structure of network

其中行人轨迹分为当前预测行人的轨迹和邻居行人的轨迹, 将行人和车辆的轨迹按照不同的邻居地图进行筛选, 得到对当前预测行人有影响的“邻居”们, 接着将这些邻居们的信息转化为栅格地图信息。其次将行人栅格地图信息和车辆栅格地图信息通过防碰撞函数和方向注意力函数进行计算, 得到栅格地图信息的权重。然后将处理后的栅格地

图信息通过两个 Bi-LSTM 编码层进行编码。人、车栅格地图信息经编码后,通过 attention 注意力模块筛选行人重点关注的社会信息,输入到当前时刻被预测行人的 LSTM 预测层中,与 t_l 时刻的隐藏状态一起输入到 t_2 时刻中,预测下一时刻的行人轨迹。

4.4 人-人、人-车社会关系建模

4.4.1 人-人交互输入

行人在道路上行走的过程中,会通过邻居的动作修正自己的走路路径(如结伴,防止碰撞等行为)。这些邻居行人的轨迹反过来又受到周围环境中其他人的影响,并可能随着时间的推移而改变他们的路径。因此神经网络的输出不仅包含行人的历史轨迹信息,还应包含行人历史时刻的周围环境的信息数据,本文采用栅格地图的方式来衡量当前预测行人周围的行人状况,如图 4.3 所示。

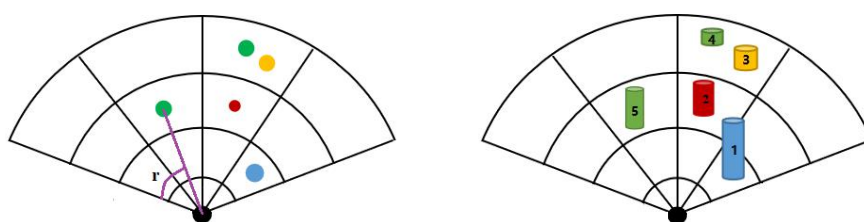


图 4.3 扇形邻域图

Figure 4.3 Sector neighborhood

在行人与行人的交互过程,被预测的行人往往会关心视野前方的人群,而不是身后的人,因此选择以当前行人作为圆心, r 为半径的扇形区域为人-人社交感兴趣区域,在这个区域中的行人称为当前行人的“邻居”,将扇形区域划分为如图 4.3 所示的 16 个网格,构成栅格地图。若将栅格地图写成 4×4 矩阵的形式,每个数字代表网格中的人数,如图 4.4 所示。

0	0	2	0
0	1	1	0
0	0	0	1
0	0	0	0

图 4.4 行人栅格地图

Figure 4.4 Pedestrian grid map

由于邻居行人与被测行人的距离不同，邻居行人对当前行人的影响力也不同（即越靠近，影响力越大），本文在栅格地图的基础上，进一步考虑更精细的位置变化，为不同行人在栅格地图上赋予不同的与距离相关的权重。参考文献^[56]建立排斥函数来计算不同距离的邻居的排斥权重，以表示来自对应于一个感兴趣区域的不同行人的影响，如式所示：

$$R_t^{i,j} = a \exp(-(s-r)) \quad (4.7)$$

其中参数 a 是相“邻居行人”的排斥幅度， s 是当前行人 i 与“邻居”行人 j 的距离， r 是行人避免碰撞的距离半径。因此行“邻居”输入矩阵 $P_t^i(a,b)$ 变为：

$$P_t^i(a,b) = \sum_{j \in N^i} (1_{ab}(s_t^{i,j}, r_t^{i,j}) \times R_t^{i,j}) \quad (4.8)$$

其中， $1_{ab}(\cdot)$ 是判定 j 行人是否在感兴趣区域中网格 (a, b) 上的判别函数。若在网格 (a, b) 中则为 1； N 是半径为 r 的扇形感兴趣区域， $s_t^{i,j}$ 是行人 j 与行人 i 之间的欧氏距离， $r_t^{i,j}$ 是行人 j 与行人 i 位置连线与扇形左侧边的夹角，如图 4.3 中 r 所示。

4.4.2 人-车交互输入

实际交通场景中，行人不仅会与行人交互，更会受到车辆的影响，最典型的交互行为即为避免碰撞行为。由于汽车与行人的属性不同，“邻居车辆”对当前行人的轨迹的影响与“邻居行人”对当前行人轨迹的影响区别很大。相对于与行人交互，当前行人在与车辆发生交互的时候，无论车辆是否在行人的前方，行人都会进行避让或加速以防止碰撞，因此本文采用圆形区域作为车辆的感兴趣区域。另外由于汽车体积比较大，因此每辆汽车应该占据多个网格，如图 4.5 所示。

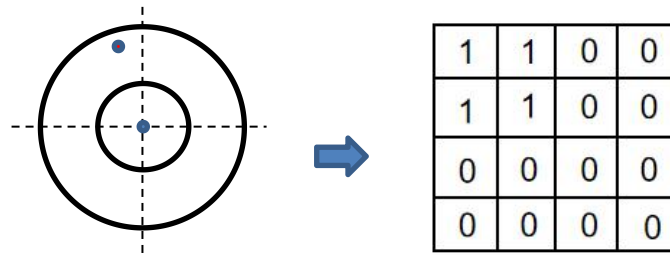


图 4.5 汽车邻域地图

Figure 4.5 Vehicle neighborhood

由于汽车行驶方向相对固定，行人往往更关注于即将与自己相撞的车辆，若车辆与行人相对而行，行人大概率会修正自己的轨迹以避让或加速前进，若车辆与行人背道而

行，车辆的运动状态对行人的影响相对较小。若汽车与行人之间的距离变小，“邻居车辆”的池化层比重将会变大，若汽车与行人之间距离变远，“邻居车辆”的池化层比重将会变小。为了量化行人与车辆的方向对行人轨迹的影响，本文设计方向注意系数来确定行人在不同方向上的注意力，以改变不同“邻居”池化层的权重，如式（4.9）所示：

$$I_s = b \exp(s_b - s_i) \quad (4.9)$$

因此人-车交互输入 $V_t^i(a, b)$ 为：

$$V_t^i(a, b) = \sum_{j \in N^i} (I_{ab}(s_t^{i,j}, r_t^{i,j}) \times R_t^{i,j} \times I_t^{i,j}) \quad (4.10)$$

4.3.4 注意力模块

本文建立注意力模块来对人-人交互输入和人-车交互输入进行编码，进一步选择有用的社会信息。注意力模块由一个全连接层和软注意力机制^[57]组成。图 4.6 是计算 Attention 值的过程图：

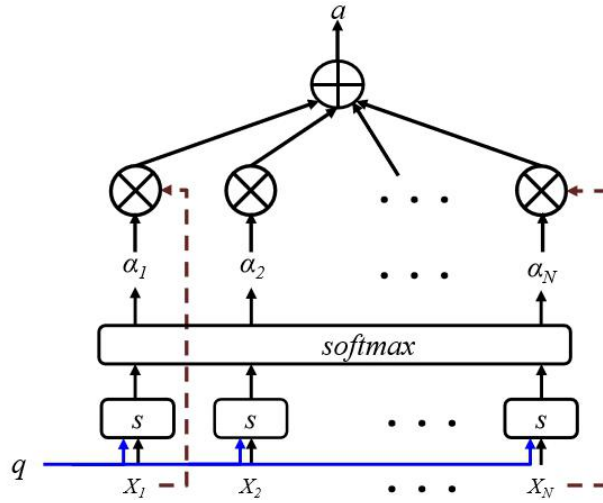


图 4.6 Attention 值计算过程

Figure 4.6 Calculation process of attention value

具体公式如下：

$$N_t^i = \text{BilSt}(P_t^i) \oplus \text{BilSt}(V_t^i) \quad (4.11)$$

$$C_t^i = \text{FC}(N_t^i, W_{fc}) \quad (4.12)$$

$$H_t^i = f^{att}(C_t^i, h_i^{t-1}) \quad (4.13)$$

首先将行人信息和车辆信息链接在一起获得社会信息 N_i^t ，将社会信息通过全连接网络 $FC(\bullet)$ 进行编码， W_{ft} 为全连接层的权重。然后利用软注意力机制 f^{att} 根据行人之前的状态 h_i^{t-1} 和编码后的社会信息 C_i^t 选择出行人更关注的社会信息 H_i^t 。

软注意力机制的原理是把输入信息向量 X 看做是一个信息存储器，现在给定一个查询向量 q ，用来查找并选择 X 中的某些信息，采取“软性”选择机制，不是从存储的多个信息中只挑出一条信息来，而是从所有的信息中都抽取一些，最相关的信息抽取得就多一些。定义一个注意力变量 $z \in [1, N]$ 来表示被选择信息的索引位置，即 $z=i$ 来表示选择了第 i 个输入信息，然后计算在给出了 q 和 X 的情况下，选择第 i 个输入信息的概率 α_i ：

$$\begin{aligned}\alpha_i &= p(z=i | X, q) \\ &= \text{softmax}(s(X_i, q)) \\ &= \frac{\exp(s(X_i, q))}{\sum_{j=1}^N \exp(s(X_j, q))}\end{aligned}\quad (4.14)$$

其中 σ_i 构成的概率向量就称为注意力分布 (Attention Distribution)。 $s(X_i, q)$ 是注意力打分函数，有以下几种形式：

加性模型	$s(X_i, q) = v^T \tanh(WX_i + Uq)$
点积模型	$s(X_i, q) = x_i^T q$
缩放点积模型	$s(X_i, q) = \frac{x_i^T q}{\sqrt{d}}$
双线性模型	$s(X_i, q) = x_i^T Wq$

其中 W 、 U 和 v 是可学习的网络参数， d 是输入信息的维度。理论上，加性模型和点积模型的复杂度差不多，但是点积模型在实现上可以更好地利用矩阵乘积，从而计算效率更高。但当输入向量的维度 d 比较高的时候，点积模型的值通常有较大的方差，从而导致 softmax 函数的梯度比较小，而缩放点积模型可以很好地解决这个问题。双线性模型可以看做是一种泛化的点积模型。相比点积模型，双线性模型在计算相似度时引入了非对称性。由于本文的输入向量维度不高，输入信息又具有高度的时变性和非对称性，因此本文使用双线性模型作为打分函数。

注意力分布 α_i 表示在给定查询 q 时，输入信息向量 X 中第 i 个信息与查询 q 的相关程度。采用“软性”信息选择机制给出查询所得的结果，就是用加权平均的方式对输入信息进行汇总，得到 Attention 值：

$$att(X, q) = \sum_{i=1}^N \alpha_i X_i \quad (4.15)$$

4.5 行人轨迹预测

4.5.1 预测层结构

本文的位置预测是通过将 LSTM 的隐藏层状态经过输出层编码成二维高斯分布 (Bivariate Gaussian Distribution) 的参数来进行的, 具体的预测结构如图 4.7 所示。

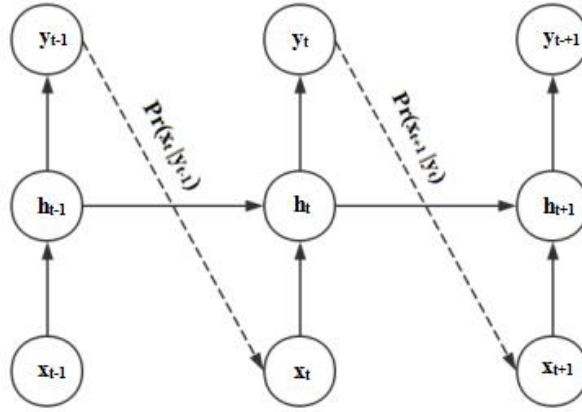


图 4.7 递归神经网络预测结构

Figure 4.7 Prediction Structure of Recurrent Neural Network

在进行轨迹预测的时候, 需要用到一段时间内的车辆轨迹数据。对一段时间内每个时刻车辆的轨迹数据提取出自身车辆的信息、相邻车辆的信息, 之后经过 Embedding 层处理过后得到输入向量序列。输入向量序列 $x = (x_1, x_2, \dots, x_T)$ 传递到递归神经网络中, 通过运算可以得到隐藏层的向量序列 $h = (h_1, h_2, \dots, h_T)$, 然后把隐藏层的向量序列通过输出层可以得到输出向量序列 $y = (y_1, y_2, \dots, y_T)$ 。每个输出向量 y_t 用于在可能的下一个输入 x_{t+1} 上参数化预测分布 $\Pr(x_{t+1} | y_t)$ 。每个输入向量序列的第一个元素 x_1 通常是一个空的向量, 因为在时间序列上没有比它更早的输入数据。因此在实际数据预测过程中, 网络的第一个输入是 x_2 向量。对于一个长度为 T 的输入向量序列, 隐藏层的计算可以通过迭代公式 (4.16) 得到。

$$h_t = H(W_{ih}x_t + W_{hh}h_{t-1} + b_h) \quad (4.16)$$

其中, W 表示权重矩阵 (W_{ih} 是连接输入层到隐藏层的权重矩阵, W_{hh} 是隐藏层的循

环连接)， b_h 项表示偏置矢量， H 是隐藏层激活函数。

给出隐藏层输出序列之后，通过输出层，得到输出向量序列，计算公式如下：

$$y_t = Y(x_t) \quad (4.17)$$

其中 Y 是输出层函数，输出向量 y_t 用于参数化下一输入的预测分布 $\Pr(x_{t+1}|y_t)$ 。必须仔细选择 $\Pr(x_{t+1}|y_t)$ 的形式以匹配输入数据，Alahi A 等^[53]在进行预测时选择二元高斯分布，并取得了较好的效果。因此本文也选择二元高斯分布作为未来轨迹的分布。

假设一个二元高斯分布的参数如下所示：

$$\text{期望: } \mu_t^i = (\mu_x, \mu_y)_t^i \quad (4.18)$$

$$\text{标准差: } \sigma_t^i = (\sigma_x, \sigma_y)_t^i \quad (4.19)$$

$$\text{相关系数: } \rho_t^i \quad (4.20)$$

因此，输出的向量 y_t 需要在形式上需要满足二元高斯分布的参数，因此输出层的公式可以写成如式（4.21）所示：

$$y_t = Y(h_t) = W_p h_t = [\mu_t^i, \sigma_t^i, \rho_t^i] \quad (4.21)$$

其中， W_p 为输出层的权重矩阵， W_p 大小为[hidden_size,5]，hidden_size 为 LSTM 隐藏层状态维度。

总的来说，本文的模型基于输入的车辆轨迹数据序列，通过 LSTM 模型给出未来轨迹位置的二元高斯分布，预测到的车辆未来位置坐标 $(\hat{x}_t^i, \hat{y}_t^i)$ 的分布通过式（4.22）得到：

$$(\hat{x}_t^i, \hat{y}_t^i) \sim N(\mu_t^i, \sigma_t^i, \rho_t^i) \quad (4.22)$$

4.5.2 网络训练方法

在创建基于 LSTM 的车辆轨迹预测模型之后，需要通过训练网络来调整权值矩阵 W_c 、 W_a 、 W_l 、 W_p 以及相对应的偏置项等参数的值，提高模型预测的准确性。

在训练开始之前，首先需要对网络的各个权值矩阵以及偏置项进行初始化。对于偏置项，采用常量初始化(Constant Initialization)的方法初始化为 0.1。权值矩阵的初始化采用了使用较多的截断高斯分布初始化(Truncated Normal Initialization)的方法。截断高斯分布初始化的数据生成基于截断高斯分布，截断高斯分布就是在高斯分布的基础上加以限制，使得生成的数据在一定范围之内。高斯分布生成的数据范围 $(-\infty, +\infty)$ ，截断式正

态分布生成的数据在 $(\mu-2\sigma, \mu+2\sigma)$ 这个范围内（其中 μ 为均值， σ 为标准差）。

之后将轨迹数据输入进网络，经过权值矩阵运算后得到预测坐标 $(\hat{x}_t^i, \hat{y}_t^i)$ 。采用负对数似然损失函数作为误差函数，具体函数如式 4.23 所示：

$$\text{loss} = - \sum_{i=1}^n \log(P(x_t^i, y_t^i | \mu_t^i, \sigma_t^i, \rho_t^i)) \quad (4.23)$$

其中 n 是 sequence length, (x_t^i, y_t^i) 为 t 时刻车辆轨迹的实际坐标, $P(x_t^i, y_t^i | \mu_t^i, \sigma_t^i, \rho_t^i)$ 的含义是在当前二元高斯分布的基础上，对于样本 $(\mu_t^i, \sigma_t^i, \rho_t^i)$ ，其预测值为 (x_t^i, y_t^i) 的概率，即预测正确的概率。为了将概率之间的乘法运算转化为加法，将其取对数。由于是损失函数，需要满足模型预测正确的概率越高损失值越小，因此再对函数添加负号。

为了避免训练过程中的过拟合，在损失函数的基础上加上了 L2 正则化项，最终的损失函数如式（4.24）所示：

$$\text{loss} = - \sum_{i=1}^n \log(P(x_t^i, y_t^i | \mu_t^i, \sigma_t^i, \rho_t^i)) + \frac{\lambda}{2} W^2 \quad (4.24)$$

其中 λ 为正则化系数， W 为需要求解的模型参数。

通过最小化负对数似然损失函数可以获得 LSTM 模型的参数，对于权值矩阵 W_e 、 W_a 、 W_l 、 W_p 以及相对应的偏置项等参数分别使用损失函数计算梯度，利用 RMSprop 优化算法反复迭代降低损失直至收敛。

4.6 本章小结

本章主要分析了轨迹预测的算法流程，设计了轨迹预测模型的基本结构，描述了轨迹模型的各个组成部分以及网络训练方法，为下一章的实验做铺垫，主要工作有：

（1）分析了轨迹模型预测算法的流程，然后对轨迹数据进行了提取处理。为了更直观的描述行人的动作，并且避免归一化带来的数据误差，将历史信息轨迹序列转化为方向角和速度；

（2）建立了基于 LSTM 的人-车交互轨迹预测模型，引入了行人与车辆之间的交互信息，设计了方向注意力函数和防碰撞函数来提高社会交互信息的精度。将交互信息通过不同的 Bi-LSTM 层进行编码；

（3）建立注意力模块，对编码后的社会交互信息数据进行筛选，筛选出对行人影响更大的社会信息以提高预测精度。

5 轨迹模型预测结果分析

5.1 实验参数设置

深度学习中参数的设置对最后的预测结果起着至关重要的作用，本文将模型的初始参数值设定如下，再经过多次实验分析最终选定参数。

- **embedding size:** 车辆自身的信息和相邻车辆的信息的嵌入维度的大小，本文使用 128，即将车辆的空间坐标信息转化为 128 维度的向量再输入 LSTM 模型。
- **num layers:** LSTM 模型的隐藏层层数，设置为 1。
- **Bi-LSTM hidden size:** Bi-LSTM 隐藏层状态维度，设置为 64
- **hidden size:** LSTM 隐藏层状态维度，设置为 128。
- **neighborhood size:** 扇形行人占用地图半径，设置为 4，单位为米
- **grid size:** 行人占用地图网格大小，设置为 1，单位为米
- **vehicle neighborhood size:** 车辆占用地图半径，设置为 10
- **vehicle grid size:** 车辆占用地图网格大小，设置为 2
- **output size:** 输出层节点的个数，设置为 5。本文将 LSTM 的输出转换成二维高斯分布的 5 个参数。
- **num epochs:** 训练次数，设置为 1000。
- **batch size:** 批尺寸，即每批数据的输入规模，设置为 128。
- **learning rate:** 学习速率，设置为 0.0003。
- **decay rate:** 学习率衰减，设置为 0.95。
- **seq length:** 输入数据序列的长度，设置为 12。
- **lambda_param:** L2 正则化参数，设置为 0.0005。
- **dropout:** 防止过拟合的参数值，设置为 0.2
- **激活函数:** Relu

5.2 实验方案及评价指标

为了在训练模型时充分利用数据集，本文采用留一交叉验证的方法。如前文所述，处理过后的数据集一共有五个，在每次训练的时候，本文在 4 组数据上训练和验证模型（训练集：验证集 = 7：3），利用剩余的一组数据测试模型。本文对所有的五组数据集都重复这一操作，用这五次测试结果的平均值来衡量模型的好坏。

在测试模型期间，本文观察到目标车辆 7 s 内的轨迹并预测接下来 5 s 内的轨迹，当

帧速率为 1 s 时，这对应于观察 7 帧并预测接下来的 5 帧，总轨迹的长度为 12 帧。因此，在训练模型的时候，输入 LSTM 模型的轨迹长度设置为 12。

使用以下指标对每个行人的轨迹进行预测评估：

1. 平均位移误差（Average Displacement Error, ADE）：实际轨迹与预测轨迹序列在每一个时间步长内的 L2（欧几里得）距离。

$$ADE = \frac{1}{n} \sum_{i=1}^n \frac{1}{pred} \sum_{t=t_{obs}+1}^{t_{obs}+t_{pred}} \sqrt{(x_i^t - \hat{x}_i^t)^2 + (y_i^t - \hat{y}_i^t)^2} \quad (5.1)$$

最终位移误差（Final Displacement Error, FDE）：实际轨迹与预测轨迹序列在最终位置的 L2（欧几里得）距离。

$$FDE = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i^{t_{pred}} - \hat{x}_i^{t_{pred}})^2 + (y_i^{t_{pred}} - \hat{y}_i^{t_{pred}})^2} \quad (5.2)$$

其中， $pred$ 是预测序列长度， obs 是观察序列长度， x_i^t ， y_i^t 是行人 i 在时刻 t 的真实位置， \hat{x}_i^t ， \hat{y}_i^t 是行人 i 在 t 时刻的预测位置。

5.2.1 网络训练分析

将上文设计好的模型进行训练，为了在每次迭代中都使用最大可能数目的样本来训练，本文训练模型时采用留一交叉验证的方法。选取三组交叉路口数据集和两组环形路口数据集，在其中 4 组数据上进行训练和验证，在剩余的一组数据上进行测试。对所有数据集重复这一操作，如图 5.1 所示，为交叉路口和环形路口数据集的训练误差曲线：

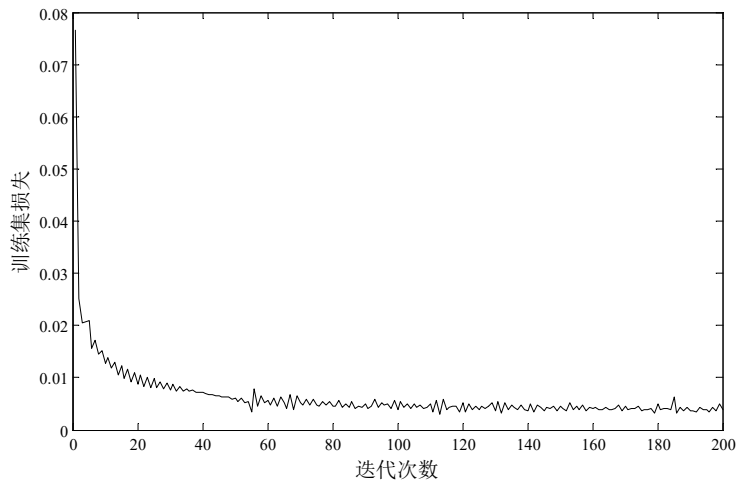


图 5.1(a) 交叉路口数据集训练损失曲线

Figure 5.1 (a) Intersection data set training loss curve

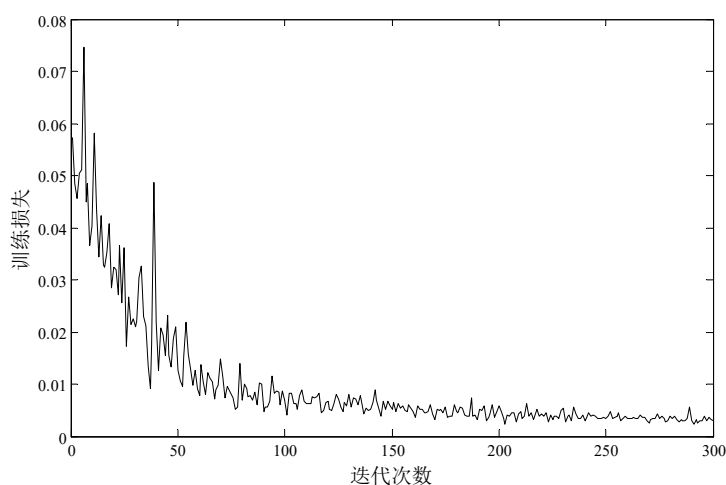


图 5.1(b) 弧形路口训练误差曲线

Figure 5.1 Training error curve in curve

由以上图像可知，基于 LSTM 模型的车辆交互轨迹预测模型训练效果较好，在误差允许范围内达到预期的效果，可以对复杂交通中的车辆进行轨迹预测。

5.2.2 轨迹预测模型参数分析

为了验证注意力模块对轨迹预测模型的影响，本文设置了对比实验，含有注意力模块的模型和不含注意力模块的误差结果在 ADE 和 FDE 上的对比结果如表 5.1 所示。所示误差单位为米（m），ADE 和 FDE 的数值越小，表示模型预测结果与真实结果的偏差越小，也就意味着模型预测精度越高。

表 5.1 attention 模块对比实验

Table 5.1 Contrast experiment of attention module

数据集	交叉路口(ADE)	环形路口(ADE)	交叉路口(FDE)	环形路口(FDE)
VP-LSTM (含 attention)	0.10	0.16	0.10	0.18
VP-LSTM(不含 attention)	0.11	0.19	0.12	0.20

如表 5.1 中所示，增加注意力模块之后，行人轨迹预测平均位移误差下降了 9.1%和 18.7%，最终位移误差下降了 16.7%和 11.1%，证明注意力模块可以有效地筛选对行人有影响的信息，使得预测模型更精确。

为了测试邻域大小对预测误差的影响，本文对比了不同行人、车辆的邻域范围预测模型的 ADE 和 FDE 值。在行人邻域范围不变的条件下，将汽车邻域的大小设置为 4, 8, 12, 16, 20 米，意味着行人将关注 4~20 米范围内的汽车并作出相应的动作。图 5.2，图 5.3 分别为交叉路口数据集上，不同汽车邻域大小的 FDE 和 ADE 结果图。

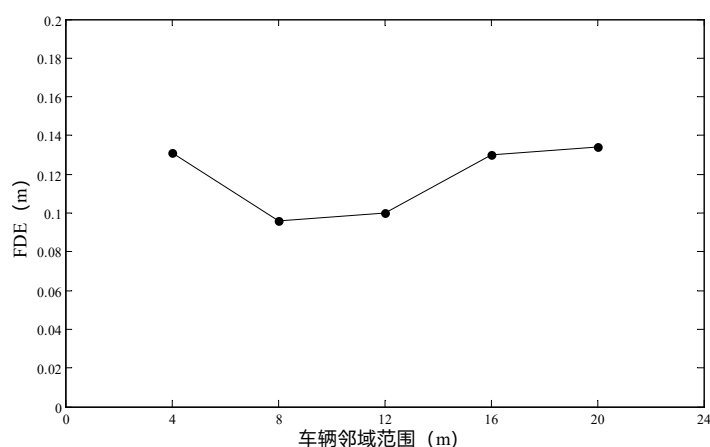


图 5.2 不同汽车邻域下 FDE 误差

Figure 5.2 FDE in different vehicle neighborhoods

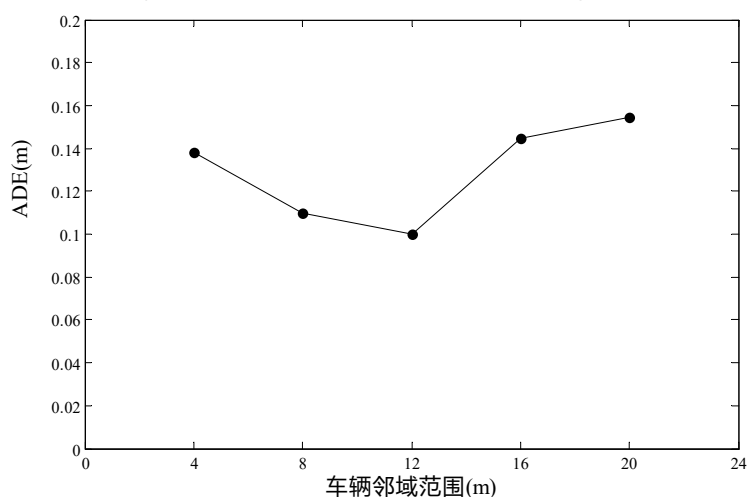


图 5.3 不同汽车邻域下 ADE 误差

Figure 5.3 ADE in different vehicle neighborhoods

从图 5.2, 图 5.3 中可以看出, 当汽车邻域范围从 4m 增加到 20m, 也就是行人关注 4~20 米范围内的汽车时, ADE 和 FDE 均先下降后升高。如图 5.2, 图 5.3 所示, 随着邻域范围的增加, 行人将更早的将汽车位置对下一步轨迹的影响考虑在内, 使得模型拥有更多的人-车交互信息, 进而更早的进行躲避动作, 所以在 4-12 米范围内, 轨迹预测的误差逐渐减少, 轨迹预测模型能够模拟行人的行为对汽车进行避让。而随着邻域范围增加的过大, 由于场景中的车速较慢, 行人没有关注更远范围内的汽车, 模型关注了过多的车辆信息, 并且这些车辆信息没有与行人产生交互动作, 进而对预测结果产生干扰,

使得误差反而增大。从表中可以看出当车辆邻域范围大小为 8 米和 12 米时，获得了最小的最终位移误差，大小分别为 0.095 和 0.104，从图 5.3 中可以看出，当汽车邻域大小为 8 米和 12 米时，取得了最小的平均位移误差，分别为 0.112 和 0.101。因此综合考虑精度大小，本文设置汽车邻域范围为 12m。

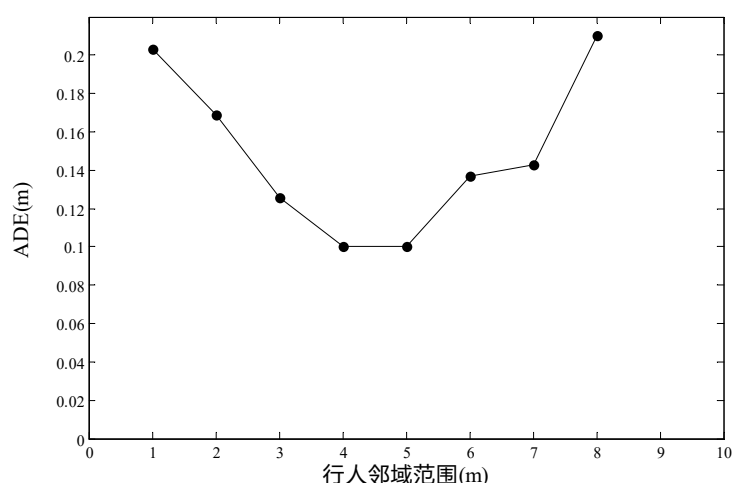


图 5.4 不同行人邻域下 ADE 误差

Figure 5.4 ADE in different pedestrain neighborhoods

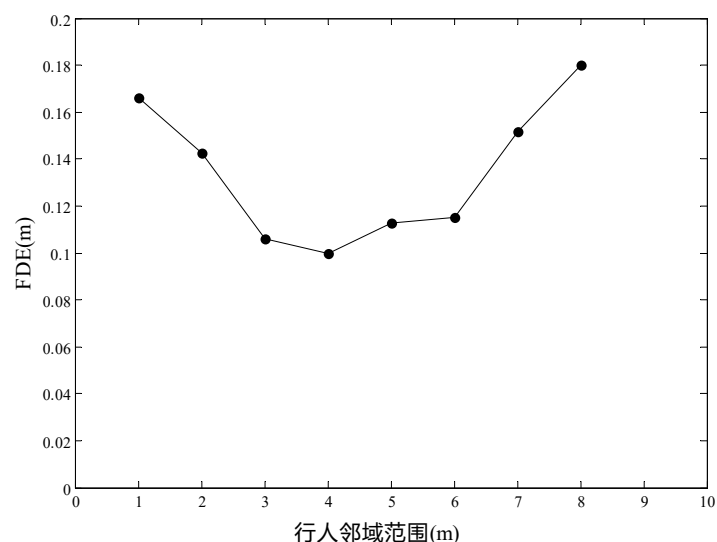


图 5.5 不同行人邻域下 FDE 误差

Figure 5.5 FDE in different pedestrain neighborhoods

在汽车邻域范围不变的条件下，将行人邻域大小设置为 1，2，3，4，5，6，7，8 米，意味着行人将关注 1~8 米范围内的其他行人并作出相应的动作，不同行人邻域大小

中的 FDE 和 ADE 对比图如图 5.4, 图 5.5 所示。

从图 5.4, 5.5 中可以看出, 当行人邻域范围从 1 m 增加到 8 m, 也就是行人关注 1~8 m 范围内的汽车时, ADE 和 FDE 均先下降后升高。如图 5.4, 图 5.5 所示, 随着邻域范围的增加, 行人拥有更多的人-人交互信息, 进而对下一步轨迹做更精确的预测, 所以在 1-5 米范围内, 轨迹预测的误差逐渐减少, 轨迹预测模型能够模拟行人的行为对邻居进行避让或跟随。而随着邻域范围增加的过大, 模型获取了过多的邻居行人的位置信息, 这些信息对当前行人轨迹预测没有影响, 而过多的信息在训练过程中会预测结果产生干扰, 使得误差反而增大。从图 5.4 中可以看出当行人邻域范围大小为 4 米和 5 米时, 获得了最小的最终位移误差, 大小分别为 0.11 和 0.10, 从图 5.5 中可以看出, 当行人邻域大小为 4 米和 5 米时, 取得了最小的平均位移误差, 分别为 0.10 和 0.10。因此综合考虑精度大小, 本文设置行人邻域范围为 5m。

为对比不同行人网格划分对轨迹预测模型精度的影响, 本文将行人邻居网格地图大小设置为 $n*n$, 如图 5.6 所示, 为 $n=4$ 的情形:

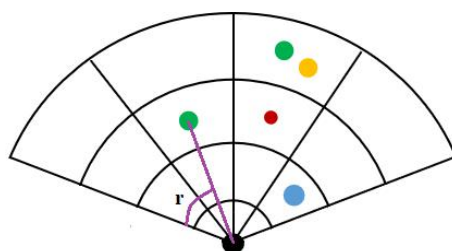


图 5.6 $n=4$ 时行人网格地图

Fig.5.6 Pedestrian grid map when $n=4$

不同网格划分方式后的 ADE 和 FDE 的变化曲线如图 5.7 和 5.8 所示。从图中可以看出当 $n \leq 4$ 时, 随着网格划分的数目的增加, ADE 和 FDE 均有所下降, 说明网格划分的越细, 越能精确的反应汽车周围的邻居们的位置信息, 能够为轨迹预测模型提供更精确的信息以方便模型学习。但是当 $n > 4$ 时, 网格数目的增加不仅不会让预测结果变得更好, 精度反而下降, 是因为网格划分过于精细, 对于预测结果不会有太大的提升, 反而会导致信息量冗余, 增加计算难度, 使得预测精度下降。因此本文选择行人网格大小为 $4*4$ 。

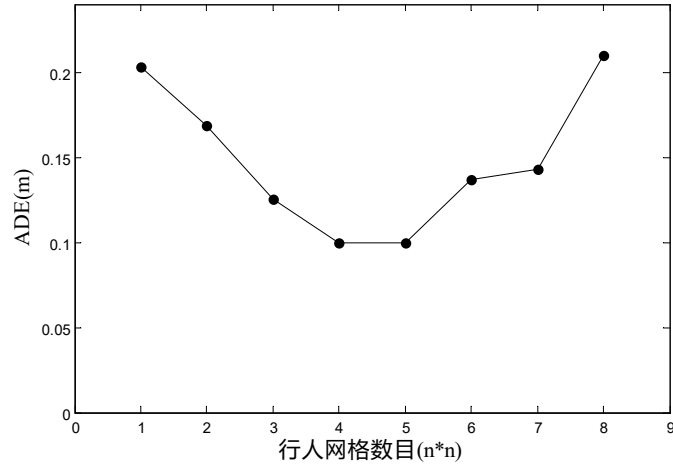


图 5.7 不同行人网格下 ADE 误差

Figure 5.7 ADE under different pedestrian grids

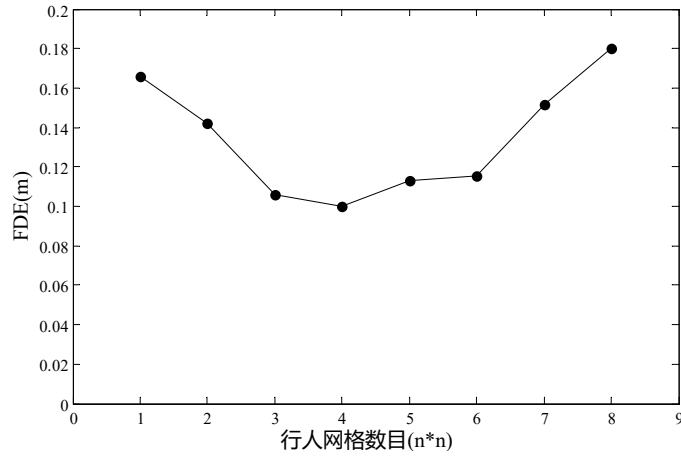


图 5.8 不同行人网格下 FDE 误差

Figure 5.8 FDE under different pedestrian grids

为了验证预测时长对轨迹预测结果的影响，本文对比了不同预测时长下的轨迹预测模型的精度误差。当输入时长固定时，预测时长 $pred$ 取 6~12 帧，相应的 ADE，FDE 结果如图 5.9，5.10 所示。从图 5.9，5.10 中可以看出，在 6~12 帧的预测条件下，本文搭建的预测模型都表现良好，误差精度在 0.2m 以内，证明了本文提出的 VP-LSTM 模型能够获得良好的精度，验证了有效性。从图中可以看出，当预测时长为 6 帧时，模型所得到的最终位移误差和平均唯一误差均为最小值，预测时长为 12 帧的时候，所得到的两种误差均取最大值。随着预测时长的增加，两种预测误差逐渐增加，符合轨迹预测中，预测时间越长，精度越差的基本规律。

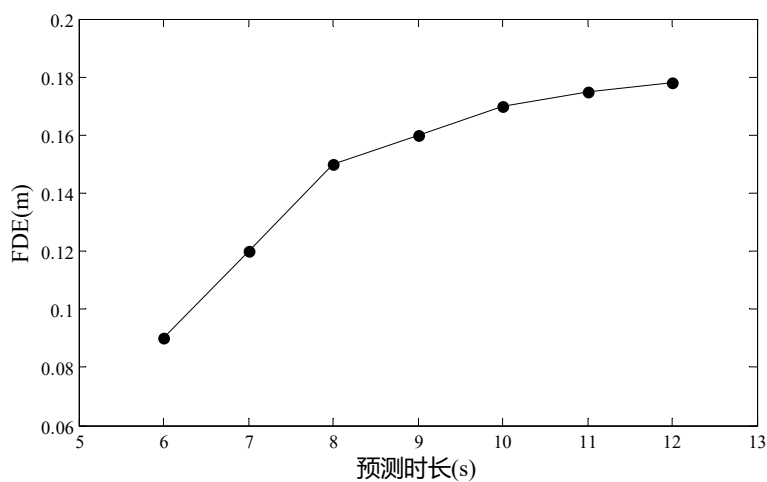


图 5.9 不同预测时长下 FDE 误差

Figure 5.9 FDE under different predicted length

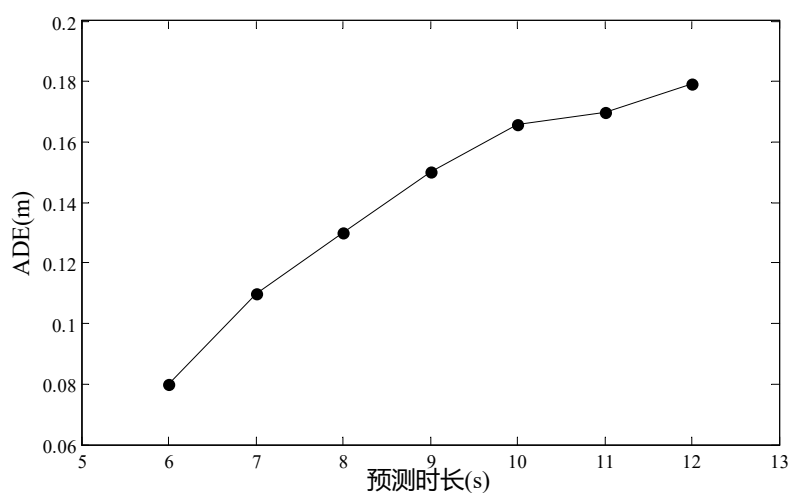


图 5.10 不同预测时长下 ADE 误差

Figure 5.10 ADE under different predicted length

5.3 模型对比分析

本文和不同文献轨迹预测模型的误差对比如表 5.2 所示，分别为 LSTM 模型（移除本文提出的轨迹预测模型中的社会关系模块和注意力模块，直接通过 LSTM 网络预测行人轨迹），S-LSTM 模型，SGAN 模型，STGCNN 模型和本文所提出的基于人车交互的 LSTM 模型（VP-LSTM）。

从表 5.2 中可以看出，无论是交叉路口或环形路口的场景上，本文提出的 VP-LSTM 网路在精度上都有所提升。其中 LSTM 模型精度最低，原因是本文所使用的数据集为下

课时间段行人和汽车流量比较大的数据集，行人的轨迹受外界干扰较大，单纯的 LSTM 网络不能很好的描述行人运动期间的社会关系，因此精度有所下降。相对于 LSTM 网络，ADE 与 FDE 在交叉路口场景中降低了 28.5%，33.3%，在环形路口场景中降低了 23.8%，28%。说明本文提出的 VP-LSTM 能够考虑行人与的社交状态，进而准确的预测轨迹。相比于 S-GAN 模型和 S-LSTM 模型，本文增加考虑了汽车的影响，因此精度也有所提高，相对于 S-GAN 模型，在交叉路口场景中 ADE 与 FDE 降低了 23.1%，44.4%，在环形路口中 ADE 与 FDE 降低了 11%，33%，相对于 S-LSTM 网络，ADE 和 FDE 分别降低了 16%，16%和 20%，18.1%，说明可以看出基于人-车交互的轨迹预测模型可以根据人车交互的躲避动作来合理的预测行人的轨迹，提高轨迹预测精度。特别地，对于环形路口场景，由于环形路口存在地下停车场，车辆较多，行人的轨迹受汽车的干扰比较大，本文提出的 VP-LSTM 模型相对于 S-LSTM 模型，误差下降较多，进一步验证了本文提出的预测模型在交通场景中的优越性。

表 5.2 不同轨迹预测模型误差对比
Table 5.2 comparison of different trajectory prediction models

方法	交叉路口 (ADE)	环形路口(ADE)	交叉路口 (FDE)	环形路口(FDE)
LSTM	0.14	0.21	0.15	0.25
S-LSTM	0.12	0.20	0.13	0.22
S-GAN	0.13	0.18	0.18	0.27
VP-LSTM	0.10	0.16	0.10	0.18

5.4 预测轨迹可视化

为了更直观的分析实验结果，本文将模型的预测结果进行了可视化。图 5.11 中展示了本文提出的模型的部分预测结果。其中 GT 为行人的真实轨迹，VP 为行人的预测轨迹。从四幅图中可以看出，预测轨迹与真实轨迹十分相近，本文提出的模型可以准确的进行行人轨迹预测。从图 5.11(a)中可以看出，编号 95 号行人和编号 94 号行人距离过近，因此会产生避让动作，本文提出的预测模型可以很好的预测出避让的轨迹。从图 5.11(b)中可以看出，本文的预测模型能够捕捉到前方汽车位置，并做出避让轨迹，符合交通场景中人车交互防碰撞的规律。从图 5.11(c)中可以看出本文的预测模型能够准确的预测出行人转弯的轨迹。因此，本文提出的预测轨迹适用于多种场景，多目标交互的交通环境，可以准确的预测行人轨迹，为智能汽车的决策提供依据。

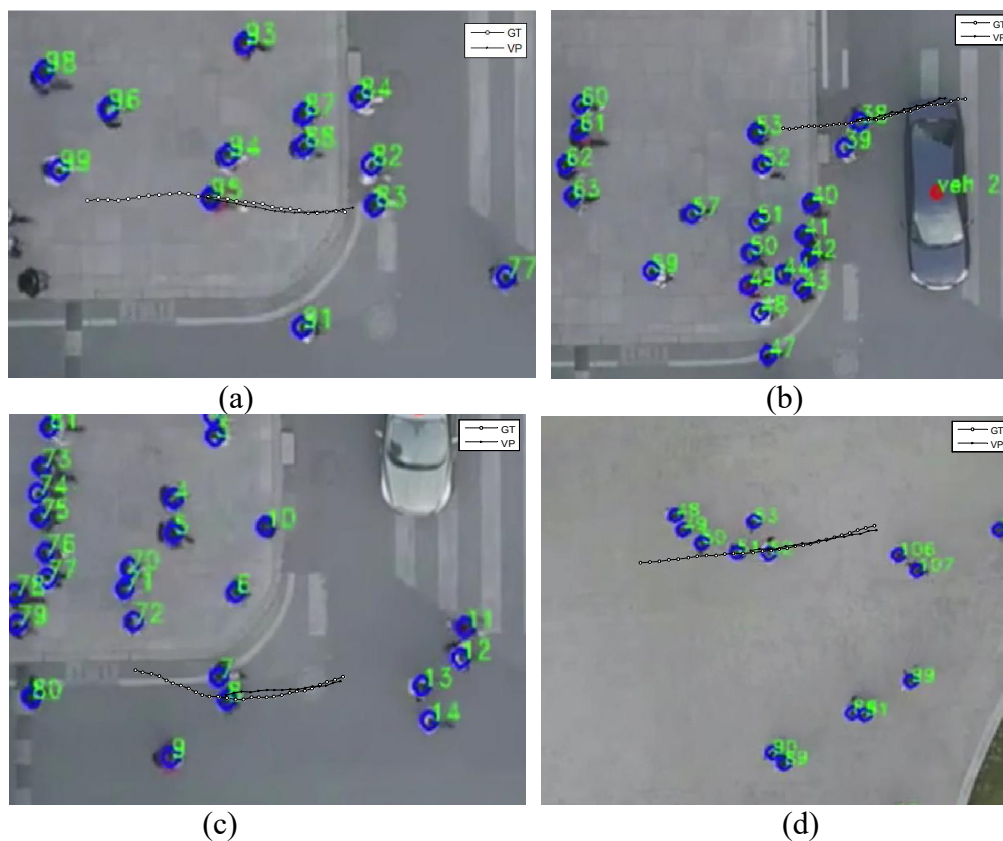


图 5.11 预测结果图

Figure 5.11 Prediction results

5.5 本章小结

本章是利用上章构建的轨迹预测模型对第三章构建的人-车交互数据集进行轨迹预测实验，并对参数选取进行分析。主要内容有：

(1) 分析了网络误差训练曲线，根据实验结果可以看出训练误差能够快速下降并收敛，证明了本文提出的算法的有效性。

(2) 分析了邻居距离和网格划分方法对轨迹预测模型精度的影响，为轨迹预测模型确定参数。

(3) 分析了注意力模块的作用和与传统轨迹预测模型的对比实验，验证了本文所提出的模型的优越性。

(4) 将预测的轨迹进行可视化，直观的分析行人运动过程中与邻居行人、汽车的交互行为。

结 论

针对交通场景中智能驾驶汽车所面临的行人轨迹预测问题，基于行人轨迹预测问题中行人的非独立性和社会的交互性，本文基于行人与行人、行人与汽车的交互，构建了人-车-人交互预测模型（VP-LSTM），为智能驾驶汽车的决策，减少交通事故和交通拥堵提供可靠的依据。论文主要研究成果包括：

（1）构建 DUT 人-车交互数据集，为轨迹预测模型提供数据支持。分析了无人机设备的基本参数和人-车交通场景的特点，使用 CSR-DCF 算法来提取行人和车辆的轨迹位置。最后使用卡尔曼滤波算法对得到数据集进行平滑处理，适用于轨迹预测模型的数据集。

（2）根据目标轨迹预测中的交互问题，本文基于长短期记忆网络与深度学习的方法，构建人-车-人社会交互预测模型（VP-LSTM）。设计了人-人、人-车交互邻域来捕捉对被预测行人有影响的目标。根据行人、汽车的性质不同，设计不同形状的交互邻域。

（3）为了进一步精确计算社交信息，本文设计了交互的防碰撞函数和方向注意力函数作为人-车，人-人社交信息的权重，计算更加精确。同时设计了注意力模块将人-车，人-人交互信息进行集成并选择对行人影响更大的社会信息。

（4）将处理好的社会信息输入到轨迹预测层中，得到预测结果。以第三章构建的 DUT 人-车交互数据集为例，对本文提出的 VP-LSTM 模型进行训练，验证和测试，以分析本文提出的模型的可行性，并确定模型参数。实验结果表明，本文提出的 VP-LSTM 能够准确的预测出交通场景中行人未来一段时间内的轨迹，有效提高了预测精度，为智能驾驶决策提供更可靠的依据。

尽管本文提出的轨迹预测模型能够很好的提高轨迹预测精度，但是由于时间的限制，仍有一些不足之处，主要在于以下几个方面：

本文在采集数据集时的主要场景为大连理工大学内放学时间内的一段路程，这个过程中车辆相对较少，很少发生行人与多辆车的轨迹交互，在未来的工作中，将会考虑扩充校外数据集，以便对轨迹预测中的车-车交互，一人对多车的交互进行深入研究。

本文主要关注于交通场景中的人-车-人等动态目标的交互运动。而交通场景时由人-车-环境构成的。未来的研究中可以结合语义分割等方式将道路等环境系统也作为行人轨迹预测的关键影响因素，将人-环境交互的信息作为输入，以进一步提高轨迹预测精度。

参 考 文 献

- [1] 李威. 基于历史轨迹的车辆类别预测[D]. 山东大学, 2018.
- [2] 成松. 移动电子警察系统关键功能模块的设计与实现[D].2011.
- [3] 管家意. 自动驾驶汽车轨迹跟踪控制方法研究[D]. 2017.
- [4] 张迎亚.基于隐马尔可夫模型的车辆轨迹预测算法的研究[D].南京邮电大学,2017.
- [5] 蒋益娟. 基于历史车辆轨迹数据的出行时间预测分析[D]. 华东师范大学, 2013.
- [6] 李茹昕. 面向自动驾驶车辆的交通管理研究[D]. 2019.
- [7] 孙浩. 考虑交通车辆运动不确定性的轨迹规划方法研究[D].吉林大学.2017.
- [8] 杨先碧. 如何有效减少交通事故[J]. 生命与灾害, 2010(3):20-23.
- [9] 陈鑫欣. 谷歌正在研发全自动驾驶汽车[J]. 发现, 2010(11):47-47.
- [10] 赵锴. 汽车自动驾驶多模型自适应控制研究[D]. 2017.
- [11] 陈耕艺. 特斯拉与自动驾驶汽车[J]. 中国信息化, 2016, 000(007):16-19.
- [12] 蔡自兴.智能驾驶技术研究的重要进展[J].机器人技术与应用,2019(03):37-39.
- [13] 王健. 汽车在两种转向工况下的路径规划与路径跟踪研究[D]. 南京航空航天大学, 2015.
- [14] 王也. 人工智能时代无人驾驶汽车交通事故责任问题探讨[J]. 安庆师范大学学报:社会科学版, 2019, 038(001):72-76.
- [15] 远帆. 颠覆性的装甲技术革命——美国国防部开启“地面 X 战车技术”研究项目[J]. 兵器, 2014(11):6-9.
- [16] 陈彗岩,熊光明,龚建伟.无人驾驶车辆理论与设计[M].北京:北京理工大学出版社出版社,2018.
- [17] Joseph Ogando. 机器人的角逐美国国防部高级研究计划署 DARPA 比赛用遥控汽车同时让军用和民用车受益[J]. 工业设计, 2008(03):51-53.
- [18] 邓学, 陈平, 郑宏达,等. 无人驾驶已悄然上路——汽车颠覆时代无人驾驶热血而来[J]. 机器人产业, 2016(4):44-56.
- [19] 王心钰, 张长青. 无人驾驶汽车的法律冲突探究[J]. 科技导报, 2017(19):13.
- [20] 徐志刚, 张宇琴, 王羽,等. 我国自动驾驶汽车行业发展现状及存在问题的探讨[J]. 汽车实用技术, 2019, 280(01):21-29.
- [21] 周斌. 基于车联网仿真平台的城市交通信号控制[D].浙江大学,2016.
- [22] 冉斌,谭华春,张健,等.智能网联交通技术发展现状及趋势[J].汽车安全与节能学报,2018,9(02):119-130.
- [23] Huang L, Wu J, You F, et al. Cyclist Social Force Model at Unsignalized Intersections With Heterogeneous Traffic[J]. 2017, 13(2):782-792.
- [24] Zhang J F, Zhu Z, Zhang X P, et al. Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas[J]. Journal of Hydrology, 2018, 561:918-929.

- [25] Ammoun S, Nashashibi F. Real time trajectory prediction for collision risk estimation between vehicles[C]//2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing. IEEE, 2009: 417-422.
- [26] Polychronopoulos A, Tsogas M, Amditis A J, et al. Sensor fusion for predicting vehicles' path for collision avoidance systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2007, 8(3): 549-562.
- [27] Zhang R, Cao L, Bao S, et al. A method for connected vehicle trajectory prediction and collision warning algorithm based on V2V communication[J]. International Journal of Crashworthiness, 2017, 22(1): 15-25.
- [28] Gindele T, Brechtel S, Dillmann R. A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments[C]//13th International IEEE Conference on Intelligent Transportation Systems. IEEE, 2010: 1625-1631.
- [29] Deo N, Trivedi M M. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1179-1184.
- [30] Liu B , Liu H , Zhang H , et al. A social force evacuation model driven by video data[J]. Simulation Modelling Practice & Theory, 2018, 84:190-203.
- [31] Huang L, Wu J, You F, et al. Cyclist Social Force Model at Unsignalized Intersections With Heterogeneous Traffic[J]. 2017, 13(2):782-792.
- [32] Schreier M, Willert V, Adamy J. Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems[C]//17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014: 334-341.
- [33] 吴子珺, 于重重, 孙利民,等. 一种针对特定车辆潜在群体的行驶轨迹预测方法[J]. 计算机应用研究, 2014(07):37-41.
- [34] Trautman P, Krause A. Unfreezing the robot: Navigation in dense, interacting crowds[C]//IEEE/RSJ. International Conference on Intelligent Robots & Systems. New York:IEEE, 2010:797-803.
- [35] 茅晨昊, 潘晨, 尹波,等. 基于高斯过程回归的船舶航行轨迹预测[J]. 科技创新与应用, 2017, 000(031):28-29,31.
- [36] 高建, 毛莺池, 李志涛. 基于高斯混合时间序列模型的轨迹预测[J]. 计算机应用, 2019, 39(8):2261-2270.
- [37] 朱坤. 基于高斯混合-贝叶斯模型的轨迹预测[J]. 计算机与现代化, 2019, 000(002):72-81.
- [38] MORRIS B T, TRIVEDI M M. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2011, 33(11):2287-2301
- [39] 李万高, 赵雪梅, 孙德广. 基于改进贝叶斯方法的轨迹预测算法研究[J]. 计算机应用, 2013, 33(7):1960-1963.
- [40] 乔少杰, 李天瑞, 韩楠, 等. 大数据环境下移动对象自适应轨迹预测模型[J]. 软件学报, 2015, 26(11):2869-2883.
- [41] KITANI K M, ZIEBART B D, BAGNELL J A, et al. Activity forecasting[C]// European

- Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012:201-214.
- [42] Lei P R, Shen T J, Peng W C, et al. Exploring spatial-temporal trajectory model for location prediction[C]//2011 IEEE 12th International Conference on Mobile Data Management. IEEE, 2011,1: 58-67.
- [43] 毛莺池,陈杨.不确定性车辆路口的轨迹预测[J].计算机科学,2018,45(03):237-242.
- [44] 夏卓群,胡珍珍,罗君鹏.EAVTP:一种环境自适应车辆轨迹预测方法[J].小型微型计算机系统,2016,37(10):2375-2379.
- [45] 姚智胜. 基于实时数据的道路网短时交通流预测理论与方法研究[D]. 北京交通大学, 2007.
- [46] 李进燕, 朱征宇, 刘琳,等. 基于行程时间多步预测的实时路径导航算法[J]. 计算机应用研究, 2013(02):32-35
- [47] Hodgson D , Barrass P G , Mecrow B C , et al. Effect of vehicle mass changes on the accuracy of Kalman filter estimation of electric vehicle speed[J]. Iet Electrical Systems in Transportation, 2013, 3(3):67-78.
- [48] Kumar J, Goomer R, Singh A K. Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters[J]. Procedia Computer Science, 2018, 125:676-682.
- [49] Lin Z , Huang Y , Wang J . RNN-SM: Fast Steganalysis of VoIP Streams Using Recurrent Neural Network[J]. IEEE Transactions on Information Forensics & Security, 2018:1-1.
- [50] Liu J, Wang Gang, Duan L Y, et al. Skeleton-Based Human Action Recognition with Global Context-Aware Attention LSTM Networks[J]. IEEE Transactions on Image Processing, 2017, (99):1-1.
- [51] Xu K, Qin Z, Wang G, et al. Collision-Free LSTM for Human Trajectory Prediction[C]//Springer International Conference on Multimedia Modeling. Cham: Springer, 2018:106-116.
- [52] 朱自升. 基于神经网络的车辆轨迹预测算法的研究与实现[D].西安电子科技大学,2018.
- [53] Alahi A, Goel K, Ramanathan V, et al. Social lstm: Human trajectory prediction in crowded spaces[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 961-971.
- [54] Shi X, Shao X, Guo Z, et al. Pedestrian trajectory prediction in extremely crowded scenarios[J]. Sensors, 2019, 19(5): 1223.
- [55] Xue H, Huynh D Q, Reynolds M. SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction[C]//IEEE. Winter Conference on Applications of Computer Vision (WACV). New York: IEEE, 2018:1186-1194.
- [56] Xu K, Qin Z, Wang G, et al. Collision-Free LSTM for Human Trajectory Prediction[C]//Springer International Conference on Multimedia Modeling. Cham: Springer, 2018:106-116.
- [57] Gupta A, Johnson J, Li F F, et al. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks[C]//IEEE/CVF. Conference on Computer Vision and Pattern Recognition (CVPR). New York: IEEE, 2018:2255-2264.
- [58] Zhang P, Ouyang W, Zhang P, et al. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern

- Recognition. 2019: 12085-12094.
- [59] Mohamed A, Qian K, Elhoseiny M, et al. Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction[C].//IEEE/Conference on Computer Vision and Pattern Recognition(CVPR). New York:IEEE,2020:accepted.
- [60] Sreelekshmy S, Vinayakumar R, Gopalakrishnan E A, et al. Stock price prediction using LSTM, RNN and CNN-sliding window model[C]//IEEE. International Conference on Advances in Computing, Communications and Informatics (ICACCI). New York: IEEE, 2017:1643-1647.
- [61] Rui Z , Ruqiang Y , Jinjiang W , et al. Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks[J]. Sensors, 2017, 17(2):273.
- [62] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. Journal of Machine Learning Research, 2011, 12(Jul): 2121-2159.
- [63] 章敏敏,徐和平,王晓洁,等.谷歌 TensorFlow 机器学习框架及应用[J].微型机与应用,2017,36(10):58-60.
- [64] Lukezic A, Vojir T, ˇCehovin Zajc L, et al. Discriminative correlation filter with channel and spatial reliability[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 6309-6318.

攻读硕士学位期间发表学术论文情况

- 1 连静, **王欣然**, 李琳辉, 周雅夫, 周彬, 基于人-车交互的行人轨迹预测, 中国公路学报, 录用待发, EI 源刊 (硕士学位论文第四章, 第五章)。
- 2 Lian Jing, **Wang Xinran**, Li Linhui, Zhou Yafu, Liu Xiujie. Plug-in HEV Energy Management Strategy based on SOC trajectory. International Journal of Vehicle Design, 录用待发, SCI 源刊.
- 3 连静, **王欣然**, 李琳辉, 周雅夫, 刘秀杰, 基于 SOC 参考轨迹的插电式混合动力汽车能量管理方法, 中国, 发明专利, 专利号: ZL 2019 1 0266714.X。

致 谢

不知不觉研究生得三年时光匆匆走过，在大连理工学习的时间里我学到的不仅仅是理论上的知识，更有很多实践上的锻炼，让我们每个走出校门的学生真正做到融会贯通。车辆工程这个专业具有实用性，商业性，广泛性的特点，学习车辆工程不仅要理论扎实，头脑清晰，更要动手实践，行知合一，这是三年里我最重要的收获。

研究生期间以来首先要感谢我的指导教师连静老师，入学之初的入门工作开始，连老师就非常耐心的给我进行指导，定期督促我完成工作，在我需要解答疑问的时候提供帮助，当我感到灰心的时候给我鼓励，使我在毕业论文期间不仅仅是写出一篇论文，也是对智能驾驶领域得以了解熟知，为今后的工作生活打下基础。

其次要感谢我的父母，父母一直是最坚实的后盾，无论是物质生活上的提供还是精神上的鼓励安慰，都让我在这段时间度过一次又一次难关，让我能够在学业上认真对待，努力学习。

同时也要感谢我的同学们，感谢教研组内每一位同学和我的室友，谢谢你们在我研究生期间对我在学习和生活上的帮助，大家从陌生到相知，结下了宝贵的友谊，感谢你们陪我走过三年快乐又充实得日子。最后要感谢培养了我七年的学校，学院，祝愿大连理工大学科研水平再创新高，更上一层楼。

大连理工大学学位论文版权使用授权书

本人完全了解学校有关学位论文知识产权的规定，在校攻读学位期间论文工作的知识产权属于大连理工大学，允许论文被查阅和借阅。学校有权保留论文并向国家有关部门或机构送交论文的复印件和电子版，可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印、或扫描等复制手段保存和汇编本学位论文。

学位论文题目： 基于人-车交互的行人轨迹预测方法

作者签名： 王欣然 日期： 2020 年 6 月 2 日

导师签名： 连静 日期： 2020 年 6 月 2 日