



计算机科学

COMPUTER SCIENCE

EGCN-CeDML:一种面向车辆驾驶行为预测的分布式机器学习框架

李可, 杨玲, 赵晏伯, 陈泳龙, 罗寿西

引用本文

李可, 杨玲, 赵晏伯, 陈泳龙, 罗寿西. EGCN-CeDML:一种面向车辆驾驶行为预测的分布式机器学习框架[J]. 计算机科学, 2023, 50(9): 318-330.

LI Ke, YANG Ling, ZHAO Yanbo, CHEN Yonglong, LUO Shouxi. EGCN-CeDML:A Distributed Machine Learning Framework for Vehicle Driving Behavior Prediction [J]. Computer Science, 2023, 50(9): 318-330.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[融合机器阅读理解的中文医学命名实体识别方法](#)

Chinese Medical Named Entity Recognition Method Incorporating Machine Reading Comprehension
计算机科学, 2023, 50(9): 287-294. <https://doi.org/10.11896/jsjcx.220900226>

[基于并行卷积网络信息融合的层级多标签文本分类算法](#)

Hierarchical Multi-label Text Classification Algorithm Based on Parallel Convolutional Network
Information Fusion
计算机科学, 2023, 50(9): 278-286. <https://doi.org/10.11896/jsjcx.221200133>

[基于主成分分析和函数机制的差分隐私线性回归算法](#)

Differential Privacy Linear Regression Algorithm Based on Principal Component Analysis
and Functional Mechanism
计算机科学, 2023, 50(8): 342-351. <https://doi.org/10.11896/jsjcx.220800255>

[基于字符特征的 DGA 域名检测方法研究综述](#)

Survey of DGA Domain Name Detection Based on Character Feature
计算机科学, 2023, 50(8): 251-259. <https://doi.org/10.11896/jsjcx.220700277>

[基于元学习和注意力机制的用户行为识别研究](#)

Human Activity Recognition with Meta-learning and Attention
计算机科学, 2023, 50(8): 193-201. <https://doi.org/10.11896/jsjcx.220900124>

EGCN-CeDML:一种面向车辆驾驶行为预测的分布式机器学习框架

李可^{1,2,3} 杨玲¹ 赵晏伯⁴ 陈泳龙⁴ 罗寿西^{1,3}

1 西南交通大学计算机与人工智能学院 成都 611756

2 教育部可持续智能交通工程研究中心 成都 611756

3 四川省网络通信技术重点实验室 成都 611756

4 西南交通大学利兹学院 成都 611756

摘要 在大规模动态变化的交通场景下,快速准确地预测车辆驾驶行为是智能交通领域极具挑战的问题之一。车辆驾驶行为的预测不仅要考虑通信的有效性,而且要考虑车辆历史行驶轨迹以及车辆之间的相互影响。文中综合考虑了上述因素,提出了一种新的基于边-增强图卷积神经网络的通信有效的分布式机器学习框架 EGCN-CeDML(Edge-enhanced Graph Convolutional Neural Network-Communication-efficient Distributed Machine Learning)。相比面向单一设备的集中式预测框架,EGCN-CeDML是通信有效的分布式机器学习框架,该框架无需将所有原始数据发送到云服务器,而是直接将用户数据在本地边缘设备存储、处理和计算。这种在多个边缘设备训练神经网络的方式缓解了集中训练神经网络的压力,降低了传输数据量和通信延迟,提升了数据处理效率,在一定程度上也保护了用户隐私。各个边缘设备部署的复合图卷积网络(EGCN-LSTM)利用边-增强注意力机制和图卷积神经网络的特征传递机制,当周围车辆数量增长至十几辆时仍能快速提取和传递车辆间的交互信息,保证了较准确的预测性能和较低的时间复杂度。不限于车辆驾驶行为预测,各边缘设备可以根据自身的计算能力和存储能力,在保证神经网络性能的前提下灵活控制神经网络的类型和规模以适用于不同的应用场景。EGCN-CeDML在公开数据集 NG-SIM 上的实验结果表明:无论交通复杂程度如何,EGCN-CeDML 的计算时间和预测性能都优于以往模型,精准率可达 0.9391,召回率可达 0.9557,F1 分数可达 0.9473;预测时长为 1s 时,预测准确率达到 91.21%;即使车辆数目增加,算法也能保持较低的时间复杂度,且稳定在 0.1s 以内。

关键词: 车辆驾驶行为预测;图卷积网络;边增强;注意力机制;分布式机器学习

中图分类号 TP393

EGCN-CeDML: A Distributed Machine Learning Framework for Vehicle Driving Behavior Prediction

LI Ke^{1,2,3}, YANG Ling¹, ZHAO Yanbo⁴, CHEN Yonglong⁴ and LUO Shouxi^{1,3}

1 School of Computer and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China

2 Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, Chengdu 611756, China

3 Sichuan Provincial Key Laboratory of Network and Communication Technology, Chengdu 611756, China

4 Leeds Joint School, Southwest Jiaotong University, Chengdu 611756, China

Abstract In large-scale dynamic traffic scenarios, predicting vehicle driving behavior quickly and accurately is one of the most challenging issues in the field of intelligent traffic driving. The prediction of vehicle driving behavior should consider not only the efficiency of communication, but also the historical vehicle trajectory and the interaction between vehicles. Considering the above factors, this paper proposes a communication-efficient distributed machine learning framework based on edge-enhanced graph convolutional neural networks (EGCN-CeDML). Compared with the centralized prediction framework on a single device, EGCN-CeDML is a communication-efficient distributed machine learning framework, which does not need to transmit all the raw data to the cloud server, and directly stores, processes, and computes user data locally. This way of training neural networks on multiple edge devices relieves the pressure of centralized training neural networks, reduces the amount of transmitted data and communication latency, improves data processing efficiency, and preserves user privacy to a certain extent. EGCN-LSTM deployed on each

到稿日期:2022-10-10 返修日期:2023-05-04

基金项目:国家自然科学基金(62202392,61731017);网络与数据安全四川省重点实验室项目(NDS2022-1);四川省自然科学基金(2023NSFSC0459,2022NSFSC0944)

This work was supported by the National Natural Science Foundation of China(62202392,61731017),Project of Network and Data Security Key Laboratory in Sichuan Province(NDS2022-1) and Natural Science Foundation of Sichuan Province,China(2023NSFSC0459,2022NSFSC0944).

通信作者:李可(keli@swjtu.edu.cn)

edge device utilizes the edge-enhanced attention mechanism and the feature transfer mechanism of the graph convolutional neural network to promptly extract and transfer the interaction information between vehicles when the number of surrounding vehicles increases to more than a dozen, ensuring more accurate prediction performance and lower time complexity. In addition to vehicle driving behavior prediction, each edge device can flexibly control the type and scale of the neural network according to its own computing and storage capabilities, under the premise of ensuring the performance of the neural network, which is suitable for different application scenarios. The experimental results of EGCN-CeDML on public dataset NGSIM show that the amount of data to be transmitted by only accounts for 21.56% of the centralized training. And the calculation time and prediction performance of EGCN-CeDML are better than those of previous models regardless of traffic complexity, with an accuracy rate of 0.9391, a recall rate of 0.9557, and an F1 score of 0.9473. When the prediction time is one second, the prediction accuracy reaches 91.21%. Even if the number of vehicles increases, the algorithm maintains a low time complexity and is stable within 0.1 seconds.

Keywords Vehicle driving behavior prediction, Graph convolutional network, Edge enhancement, Attention mechanism, Distributed machine learning

1 前言

车辆驾驶行为预测是实现自动驾驶的重要组成部分,其通过预测周围车辆的驾驶行为,为车辆提供预警信息,协助车辆做出正确行为决策,从而提高道路行驶的安全性。自动驾驶车辆存在大量需要处理的数据,如车辆数量、行车数据、道路环境、交通状况等,依靠车辆设备处理实时数据,尤其是在复杂动态的交通场景下,容易产生较高延迟,导致预测错误而存在安全隐患。因此,如何在大规模复杂动态交通场景下进行车辆驾驶行为预测是目前智能交通领域极具挑战的问题之一。

目前,单一设备集中式的车辆驾驶行为预测研究已取得一定的进展,但仍有3方面的问题亟需解决。首先,自动驾驶汽车每秒将消耗或产生多个千兆字节(GB)的数据^[1],这些数据来自于配备的高质量摄像头、激光雷达和雷达,但单个边缘设备或车辆的计算能力、存储资源有限,且需要与远端云服务器频繁通信。面对这些海量异构数据,目前的方法仍存在通信效率不高甚至无法实时处理的问题。此外,车辆的行驶数据是用户重要的隐私,许多提供车辆驾驶行为服务的平台会收集用户的行车数据,包括用户住址、出发地、目的地等敏感信息,侵犯了用户个人隐私^[2]。然后,现有的基于一般神经网络的预测方案^[3],仅提取单一车辆的空间特征或时间特征,未同时考虑两方面特征,在大规模动态复杂交通场景下其预测准确性会明显降低,并且计算时间也会随着迭代次数的增多而增加,达不到实时处理的效果;若使用过于复杂的神经网络,针对大规模复杂交通场景,处理节点的算力可能不足,导致神经网络难以完成训练^[4]。最后,现有的框架大多受限于某一个应用场景,缺乏适用于不同的应用场景的通用框架。

本文提出了一种基于边-增强图卷积神经网络的通信有效的分布式机器学习框架(EGCN-CeDML),以分布式训练的方式,实现多设备协同训练,解决上述问题。特别是在大规模复杂动态交通场景下,即使周围车辆数量快速变化,该框架也能以较低的计算时长获得较优的预测性能。本文的主要贡献如下:

1)在大规模复杂动态的交通场景下,本文提出了通信有效的分布式机器学习框架(EGCN-CeDML),只需将用户数据在本地进行存储,且多个边缘设备对海量数据实时并发处理,

不需要定期地与云端服务器进行参数同步。EGCN-CeDML利用边缘部署的边-增强图卷积神经网络(EGCN)^[5]从自身收集的数据中抽取部分发送到云端的服务器,只需用小部分数据组成公共数据集即可在云端服务器中实现最终训练,因此减少了传输数据量与通信延迟,缓解了单一设备集中训练神经网络的压力,提升了数据处理效率,在一定程度上也保护了用户隐私。

2)考虑到边缘节点的存储能力有限,在每个边缘设备部署EGCN,利用边-增强注意力机制和图卷积神经网络的特征传递机制,以图的方式提取和传递车辆间丰富的交互特征,提升车辆驾驶行为预测的准确性。由于EGCN采用的是图结构,能够同时提取多辆车的交互特征,即使车辆数目增加,计算迭代次数仍相对稳定,具有较低的计算时间复杂度,满足实时应用需求。

3)不限于车辆驾驶行为预测,基于本文提出的EGCN-CeDML框架,各边缘设备还可以根据自身的计算能力和存储能力以及不同应用场景的需求灵活选择神经网络类型和规模,以提供实时预测、分类等性能保证。

2 相关工作

以往关于交通参与者行为建模的研究大致可以分为3个方面:基于物理模型的方法、基于意图的方法和交互感知的方法^[6]。

基于物理模型的方法是根据车辆的动力学特性预测车辆未来一段时间的轨迹。算法普遍考虑了车辆特性(如质量和轮毂)、环境条件(如速度限制和道路类型)和控制输入(如加速度、速度和转向)等,但是物理模型中没有考虑不同交通参与者的明确意图和交互操作^[7]。

针对上述不足,研究者提出了基于意图的方法,将交通参与者的行为建模分为意图识别和行为预测两个阶段。例如,在变道场景中,预先设定的动作为左变道、右变道和保持车道,在第一步识别结果的基础上,建立行为预测模型并生成规划结果。Kumer等^[8]提出了一种结合支持向量机与贝叶斯滤波器的车辆驾驶行为预测框架,能够预测最大预测时间为3.29s的车辆变道意图。He等^[9]建立了一个动态贝叶斯网络模型来推断每个驾驶动作的概率。还有一种方法是将每个动作分解为一系列连续事件,并使用隐马尔可夫模型

(Hidden Markov Model, HMM)表示这一系列事件。不同事件之间的转移概率可以从数据中获得,也可以从观测模型(即不可观测事件和现有观测之间的关系)中获得。对于一个新的观测序列,通过比较每个HMM的观测结果的可能性来估计意图^[10]。但是当输入数据维度或时间序列跨度增加,计算复杂度也会显著增加,而预测准确性明显降低。

针对复杂动态的城市交通场景,研究者又提出了交互感知的方法,该方法考虑了车辆未来的驾驶行为受环境影响的事实,并试图对交互过程建模用于车辆驾驶行为。Ma等^[3]进一步提出了一种基于长短期记忆网络(LSTM)的实时预测框架,该框架通过学习车辆的历史行驶信息来提取车辆行为特征,提高了预测准确度。上述研究工作的不足之处是仅考虑了车辆空间特征或时间特征单一信息,没有综合考虑两者的相互影响,在复杂交通场景下其预测准确性会降低。受卷积社会池的启发,Deo等^[11]利用附近车辆的空间位置信息,根据当前车辆与相邻车辆的局部位置,将LSTM编码的相邻车辆的动力学行为对齐成一个以当前车辆为中心的网格,并应用卷积神经网络(CNN)提取相邻车辆之间的相互作用特征,提供给LSTM解码器来预测当前车辆的未来轨迹。此外,车辆驾驶行为还受道路结构的影响,Gao等^[12]进一步提出了一种分层图神经网络框架,通过卷积神经网络(ConvNets)对移动代理轨迹和高清地图(HD)进行编码,使得该框架能够学习道路结构特征,提升了模型预测的性能,但是该框架对节点(车辆)间随时间变化的交互特征提取不够充分。

上述研究仅从预测性能进行改进,没有考虑计算单元的存储能力对车辆驾驶行为预测的影响。鉴于此,本文提出EGCN-CeDML框架进行车辆驾驶行为预测,该框架综合考虑了车辆的历史行驶特征的影响、车辆之间的相互影响,以及边缘设备的计算、存储能力的影响。EGCN-CeDML利用边缘增强图卷积神经网络EGCN来提取车辆之间的交互特征,同时又能适应周围车辆数目的动态变化;其次,EGCN-CeDML设计了基于知识蒸馏的模型压缩方法,用户数据通过边缘设备在本地存储、处理和计算,无需发送所有原始数据到云服务器,减少了传输数据量和通信延迟,提升了框架的处理性能,在一定程度上保护了隐私;最后,EGCN-CeDML还可以将神经网络放在多个边缘设备中训练,在保证准确预测车辆驾驶行为的同时缓解了训练神经网络的压力,以适应复杂动态交通场景下实时预测的性能要求。

3 EGCN-CeDML 框架

图1是高速公路场景下车辆驾驶行为预测场景示意图,描述了多条车道的车辆行驶情况,其中红色车辆为自我车辆,即发出预测请求的车辆;绿色车辆为周围车辆,即被预测的车辆;灰色车辆为无关车辆,即离自我车辆较远,且不需要进行预测的车辆。图中的云服务器是预测任务的数据中心与计算中心,边缘服务器则是协助云服务器进行存储和计算的边缘设备。

当自我车辆发出预测请求时,需要获取周围车辆的当前行驶信息与历史行驶信息,例如车辆的位置坐标、速度、加速度、航向角等,然后对周围车辆的未来驾驶行为做出预测,

包括车辆保持速度直行、车辆加速驾驶、车辆减速驾驶、车辆左转或者车辆右转等。

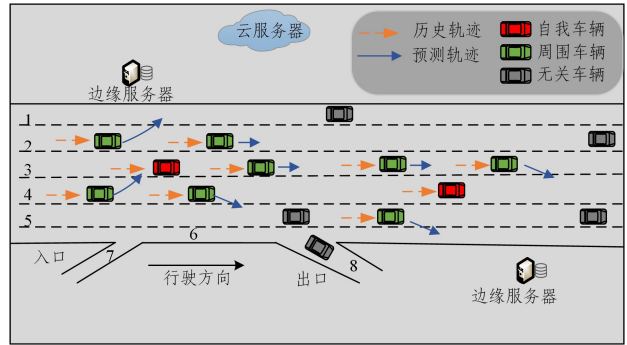


图1 车辆驾驶行为预测场景示例(电子版为彩图)

Fig. 1 Example of vehicle driving behavior prediction

3.1 功能模块

图2给出了EGCN-CeDML的4个功能模块。首先是数据收集模块,该模块为整个EGCN-CeDML框架提供数据支持,主要收集的数据为车辆的位置坐标、速度以及偏向角等信息,并将收集到的信息构建为图信息,为下一步EGCN的扩展复合模型EGCN-LSTM网络训练与公共数据集的传输做好数据准备。接下来是本地训练模块,该模块将每个LNN设置为EGCN-LSTM模型,利用本地收集到的图结构数据进行独立训练。随后是通信模块,此模块需要将本地抽取的公共数据集传输到云端服务器,为云端的EGCN-LSTM模型训练提供数据,并可以将云端的网络模型下载到边缘设备当中。最后是云端训练模块,此模块会利用公共数据集进行云端的EGCN-LSTM网络训练,融合多个边缘设备中的EGCN-LSTM特征,获得具有性能更好的全局模型。

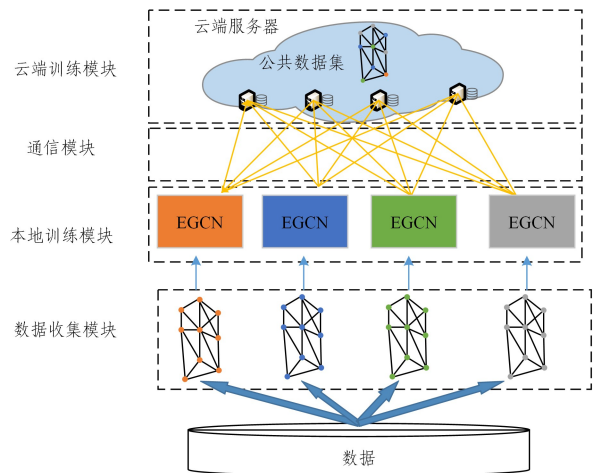


图2 EGCN-CeDML 架构图

Fig. 2 Architecture of EGCN-CeDML

3.2 框架执行流程

EGCN-CeDML的执行分为3个阶段,分别是本地训练阶段、重新标记阶段以及最终训练阶段。在本地训练阶段,每个边缘设备利用本地收集到的车辆坐标信息、车辆速度信息以及偏向角信息来训练一个本地的EGCN-LSTM模型,这些本地模型代表了不同设备数据的特征。在本地训练阶段完成后,将开始EGCN-CeDML最重要的阶段,即重新标记阶段。

通过对从边缘设备收集的数据进行采样来导出公共数据集,然后复合模型 EGCN-CeDML 分别将公共数据送到经过训练的本地 EGCN-LSTM 模型中,每个本地 EGCN-LSTM 模型都生成公共数据集对应的输出,这些输出相应地带有不同 EGCN-LSTM 的特征。接着利用数据重标记算法(Data Relabeling, DR)对各个 EGCN-LSTM 的输出进行处理,包括筛选出错误的 EGCN-LSTM 输出与聚合各个正确的 EGCN-LSTM 的输出,以此生成新的标签来替换公共数据的原始标签,具有此类新标签的公共数据集称为重标记数据集。最后,在云端服务器中实现最终训练阶段。在这一阶段,云端服务器用重标记数据集将不同的本地 EGCN-LSTM 模型的特征融合,训练得到性能更优秀的全局 EGCN-LSTM 模型,此模型可以被推送到各个设备以提供低延迟的车辆驾驶行为预测服务。为了保持对数据动态地适应,EGCN-CeDML 需要定期更新车辆的信息,构建新的公共数据集,并重复执行上述过程,以保证车辆驾驶行为预测服务的质量。

4 通信有效的设计

本文提出了一个基于边-增强图卷积神经网络的通信有效的分布式机器学习框架(EGCN-CeDML),在保证神经网络性能的前提下灵活控制各个设备的神经网络规模。

通信有效的设计具体包括公共数据集构建、存储以及数据重标记。

4.1 算法设计

EGCN-CeDML 能以较低传输数据量实现较好的神经网络性能的原因在于其使用了特征信息丰富的重标记公共数据集,因此本节将讨论 EGCN-CeDML 如何构建、存储以及重新标记公共数据集。

4.1.1 公共数据集构建与存储

首先是公共数据集的构建,每个边缘设备按照相同的比例从自身收集到的本地数据集中抽取部分数据来构建公共数据集。其次是公共数据集的存储,每个边缘设备与云端服务器都保存公共数据集。

当每个边缘设备与云端服务器都保存公共数据集时,每个边缘设备自行将公共数据集输入到本地模型 LNN(Local Neural Network)当中,再将本地模型 LNN 的输出结果发送到云端服务器。因此需要传输的数据量 l 表示为:

$$l = c * n(s+k) \quad (1)$$

其中,边缘设备数量为 n ,公共数据集中的样本数量为 c ,每个样本的大小为 s 字节,每个样本对应的输出值大小为 k 字节。

4.1.2 数据重标记算法

EGCN-CeDML 通过重标记公共数据集可以融合各个本地模型 LNN 的特征,全局模型(Eventual Neural Network, ENN)利用重标记的公共数据集进行训练可以获得更高的性能。在进行公共数据集的重标记时,需要注意两个问题。首先是如何使用较小规模的训练数据来保证神经网络的训练效果;其次是如何保证重标记数据的准确性。针对上述两个问题,本节介绍数据重标记算法,该算法涉及两个操作,分别是知识蒸馏和协方差计算。DR 算法会对公共数据集进行知识蒸馏,蒸馏后的公共数据集的标签不是只表示一个类,而是

携带所有类别的概率信息,即公共数据集携带更多的本地模型 LNN 特征信息,能一定程度上保证训练效果。DR 算法最后需要将多个边缘设备的输出进行聚合生成重标记的标签,因此在聚合之前应该对多个边缘设备的输出进行筛选,避免错误的输出被聚合。协方差可以衡量各个边缘设备输出之间的相关性,协方差为负说明输出之间为负相关,输出值为正说明输出之间为正相关。DR 采用协方差的目的便是去除负相关的输出,以保证聚合产生的重标记标签的准确性。

DR 算法的第一个重要操作是基于知识蒸馏的标签处理,具体操作如下:为了减小训练数据集规模,Caruana 等提出了一种概率分布抽样的算法,从大规模的数据中进行数据抽样,得到的小规模数据与大规模数据的概率分布保持一致,使得神经网络利用小规模的数据也能得到较好的效果^[13],然而该算法获得的小规模数据集含有的特征信息量有限,使得训练的神经网络模型的性能仍然难以进一步提升。为解决此问题,Hinton 等提出了知识蒸馏的概念,利用知识蒸馏使数据的信息量增多,并利用教师-学生结构将教师网络学习到的知识更好地迁移到学生网络之中^[14]。但此种方法不能直接应用于分布式场景,因为边缘设备处在不同的地理位置,所以很难直接将所有边缘设备的知识同步迁移到云端网络,而部分边缘设备收集到的信息又不够全面,这可能造成云端训练得到的神经网络性能不佳。

基于以上分析,DR 算法的第一个重要操作是分布式知识蒸馏。下面将从知识蒸馏结构、知识表示以及蒸馏方法 3 个关键点进行详细说明。

1)知识蒸馏的结构。传统知识蒸馏是一种基于“教师-学生”的模型,其中教师是“知识”的输出者,学生是“知识”的接受者。而 EGCN-CeDML 改进了传统的知识蒸馏结构,采用“多教师-学生”模型,如图 3 所示,多教师蒸馏到学生可以传递多种知识,提升模型的泛化能力。再者,在 EGCN-CeDML 中,所有模型的规模都可以根据不同设备的计算能力和存储能力进行设置,不严格限定教师模型和学生模型的复杂程度,这样可以使得网络结构更加灵活,更好地适应分布式场景。

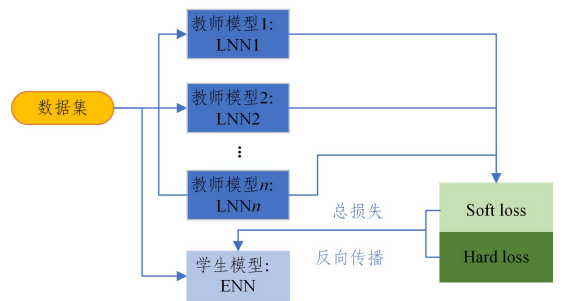


图3 EGCN-CeDML 功能模块示意图

Fig. 3 Function module of EGCN-CeDML

2)知识的迁移。传统的知识蒸馏通常是一个已训练好的教师模型提供知识,学生模型通过蒸馏训练来获取教师的知识。它可以以轻微的性能损失为代价将复杂教师模型的知识迁移到简单的学生模型中。

在 EGCN-CeDML 中,由多个教师模型提供知识。更具体地说,多个边缘设备首先利用自己已有的本地数据训练出

良好的 LNN, 然后使用公共数据集来组合不同 LNN 的知识。

学生模型 ENN 学习知识时, 不需要将所有的原始数据作为输入, 而是将公共数据集输入 ENN 进行模型训练, 以此完成知识迁移。

3) 蒸馏的方法。传统的知识蒸馏方法是单个教师模型利用式(2)将自身学到的原始数据的特征信息提取出来, 生成平滑的概率分布来指导学生模型进行学习。温度的高低改变的是学生模型训练过程中对负标签的关注程度。其中 T 代表温度, 当温度较低时, 对负标签的关注, 尤其是那些显著低于平均值的负标签的关注较少; 而温度较高时, 负标签相关的值会相对增大, 学生模型会更多地关注到负标签。而 EGCN-CeDML 的分布式蒸馏方法聚合了所有本地模型 LNN 的特征, 利用 DR 算法得到重标记的公共数据集, 该数据集包含更平滑的概率分布以及更丰富的特征信息, 进而指导全局模型 ENN 的学习。

$$y_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \quad (2)$$

DR 算法的第二个重要操作是利用协方差去除错误数据, 以保证公共数据集的正确性。不同的边缘设备处于不同的地理位置, LNN 规模也受到其自身计算能力和存储能力的限制, 使得不同的 LNN 具有一定的性能差异, 因而使某些 LNN 存在错误的输出。因此, 在重标记阶段云端服务器将会对各个 LNN 的输出进行判断, 将错误的 LNN 输出去除后, 生成重标记数据集。DR 算法对 LNN 的输出进行判断分为两个步骤。首先是去除掉判断不明确的输出值。以一个分类应用为例, 所有数据样本分为 4 个种类, 分别是 A, B, C, D 类。对于每一个分类, 神经网络的理想输出应该是 $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$, $(0, 0, 0, 1)$ 。假设某个 LNN 的输出值为 $(0.3, 0.2, 0.2, 0.3)$ 或 $(0.5, 0.2, 0.2, 0.1)$, 则该 LNN 不能清楚地判断输入样本属于哪一个类别, 因此 DR 算法将会执行删除操作。第二个步骤是去除与其他 LNN 输出相比异常的输出, 具体体现在针对公共数据集中的某一个样本, 计算各个 LNN 输出的协方差, 组成协方差矩阵 \mathbf{M} , 如下所示:

$$\mathbf{M} = \begin{bmatrix} \sigma(y_1, y_1) & \sigma(y_1, y_2) & \cdots & \sigma(y_1, y_n) \\ \sigma(y_2, y_1) & \sigma(y_2, y_2) & \cdots & \sigma(y_2, y_n) \\ \vdots & \vdots & \sigma(y_i, y_j) & \vdots \\ \sigma(y_n, y_1) & \sigma(y_n, y_2) & \cdots & \sigma(y_n, y_n) \end{bmatrix} \quad (3)$$

协方差矩阵 \mathbf{M} 中元素 $\sigma(y_i, y_j)$ 具体的计算公式如下:

$$\sigma(y_i, y_j) = \frac{1}{v-1} \sum_{k=1}^v (y_{ik} - \bar{y}_i)(y_{jk} - \bar{y}_j) \quad (4)$$

其中, y_i 与 y_j 表示边缘设备 i 与边缘设备 j 的 LNN 输出值, 则 $\sigma(y_i, y_j)$ 表示的是 y_i 与 y_j 的协方差; v 代表输出值的维度, \bar{y}_i 与 \bar{y}_j 则表示输出平均值。协方差 $\sigma(y_i, y_j)$ 的实际含义是两个输出的相关性程度, y_i 与 y_j 的相关性越小, 协方差值就越小, 因此可以利用协方差来观察不同 LNN 输出之间的关系, 并筛选出与多数 LNN 输出不相关的输出。

因此, DR 算法在后续的聚合操作时会忽略错误的输出。最后, DR 会将公共数据集每个样本经过筛选后剩余的输出进行聚合, 并将其作为该样本的新标签, 如此便完成了重标记

阶段, 得到了重标记数据集。该方法可以在一定程度上处理数据异构问题, 保证重标记数据的正确性。

4.2 算法流程

4.1.1 和 4.1.2 小节对 DR 算法中知识蒸馏和协方差两个重要部分进行了详细的介绍, 本节将给出 DR 算法的完整过程。DR 算法在 EGCN-CeDML 的重新标记阶段执行, 主要的目的是为最终训练阶段生成具有更多特征信息、更准确的重标记数据集。如算法 1 所示, DR 需要将公共数据集 S_c 、知识蒸馏温度系数 T 、阈值 γ (协方差为负的个数) 以及边缘设备的个数 n 作为算法的输入。

算法 1 DR 算法

输入: 公共数据集 S_c , 温度参数 T , 阈值 γ , LNN 的数量 n

输出: 重标记数据集 S_r

1. 初始化: $S_r = \emptyset$;
2. for $\forall s \in S_c$ do
3. $Y = \emptyset$;
4. for $i = 1$ to n do
5. $y_i \leftarrow \text{Output}(\text{LNN}_i, s)$;
6. $Y.append(y_i)$;
7. $Y \leftarrow \text{softmax}(Y/T)$;
8. $\mathbf{M} = \text{Get } \mathbf{M}(Y)$;
9. $D = \emptyset$;
10. for $i = 1$ to n do;
11. $\text{count} \leftarrow 0$;
12. for $j = 1$ to n do;
13. if $M_{ij} < 0$ then
14. $\text{count} += 1$;
15. if $\text{count} > \gamma$ then
16. $D.append(i)$;
17. $r \leftarrow 0$;
18. for $k = 1$ to n do;
19. if $k \notin D$ then;
20. $r += Y_k$;
21. $r \leftarrow r / (n - |D|)$;
22. $S_r.append(r)$;
23. return S_r .

DR 会将公共数据集 S_c 的数据输入到每个的 LNN 中, 得到输出集合 Y 。随后利用式(2)对 Y 进行知识蒸馏操作(第 7 行)。接着按照式(3)与式(4)计算得到 LNN 输出 Y 的协方差矩阵 \mathbf{M} (第 8 行), 再对协方差矩阵 \mathbf{M} 中的元素逐行进行筛选, 若某行中元素为负数的个数超过了设置的阈值 γ (第 15 行)则将此行索引记录下来。最后将未被记录的输出进行平均便得到了重标记的标签(第 21 行), 得到最终的重标记数据集 S_r 。

5 本地训练模块部署

为了更好地提取复杂动态场景下车辆之间的连接关系与交互特征, 本文在 EGCN 基础上扩展出 EGCN-LSTM, 该模型具有两个重要机制, 分别是边-增强注意力机制与基于图卷积神经网络的特征传递机制。边-增强注意力机制通过增加边特征的维数, 使得提取的交互特征更加丰富; 图卷积神经

网络的特征传递机制是借助节点特征矩阵与加权邻接矩阵,以动态图结构的数据形式来更新节点之间的交互特征,从而充分刻画自我车辆与周围车辆间的交互关系的变化。

5.1 EGCN-LSTM 神经网络模型

复合神经网络模型 EGCN-LSTM 如图 4 所示,本模型包含 4 个层,分别是 EGCN 层、历史特征层、LSTM 层以及全连接层。EGCN 层包含了多个边-增强图卷积神经网络,用于提取输入数据的特征。历史特征层的作用是生成每辆车的历史特征矩阵 \mathbf{T} ,随后将历史特征层与 EGCN 层的输出进行拼接,输入到 LSTM 层。LSTM 神经网络是一种时间循环神经网络,具有 3 个控制器,分别是遗忘控制器、输入控制器和输出控制器。LSTM 通过记忆细胞学习对过去的隐藏状态进行更新或者是遗忘,因此能较好地解决基于时间序列的相关预测问题。交通场景是典型的时间相关性场景,因此选用 LSTM 网络能够得到较好的预测结果。本文采用具有 LSTM 编解码结构的序列到序列(seq2seq)模型,其主要功能是编码器首先对 EGCN 模块的输出和历史特征模块的输出进行编码,随后通过解码器为车辆生成未来的轨迹序列,最后通过全连接层,获得对车辆驾驶行为的预测结果。

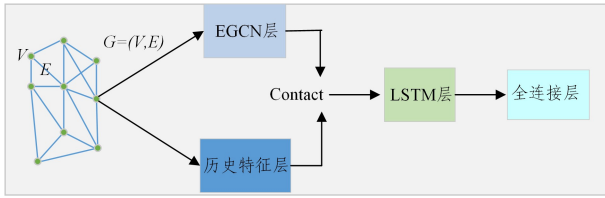


图 4 EGCN-LSTM 网络结构

Fig. 4 Network structure of EGCN-LSTM

5.2 车辆交互模型

在复杂的交通场景下,要获得较好的预测效果,首先需要对车辆所在的交通场景进行建模。传统的利用车辆信息构建一维序列或构建二维图像的方式都不能充分体现各车辆交互的特点,因此选用图结构对车辆交互进行建模,如图 5 所示。图的数学表达式是 $G=(V,E)$,其中 G 表示图,代表车辆之间的交互关系; V 是顶点的集合,代表道路上行驶的车辆; E 是边的集合,代表各车辆之间相互影响的关系,边上可设置权重,表示不同车辆之间有着不同程度的影响。

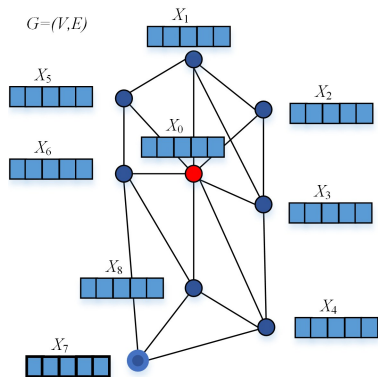


图 5 车辆交互模型

Fig. 5 Interaction model of vehicles

是车辆的位置坐标 (x,y) 、车辆速度 (v_x,v_y) ,以及偏向角 φ 。由此可以通过图中各车辆的特征构建出节点特征矩阵 \mathbf{V} ,如下:

$$V_i = [x_i \ y_i \ v_{xi} \ v_{yi} \ \varphi_i] \quad (5)$$

其中, V_1 与 V_k ($k \in [2,n]$)分别代表自我车辆与周围车辆。此处,边的特征矩阵 \mathbf{E} 代表了车辆之间的交互信息,例如车辆之间是否存在通信连接、通信方向等,其定义如下:

$$E_{ij} = \Delta V'_{ij} = |V'_i - V'_j| \quad (6)$$

边特征矩阵的元素 E_{ij} 的含义是车辆 j 到车辆 i 的交互内容,包含了车辆之间的距离、速度差值、车辆是否建立连接,以及连接方向。 V' 是车辆去除掉偏向角 φ 后的车辆行驶特征,只包含了车辆的位置坐标 (x,y) 以及车辆速度 (v_x,v_y) 。 $\Delta V'$ 是两车之间的状态差值,表示车辆之间的交互信息。考虑到在实际交通场景下,两辆车之间相互影响,那么在图结构中两个顶点应该是双向连接的,但对于自我车辆来说,车辆的驾驶行为是自身可控的,不需要进行预测,因此自我车辆与周围车辆之间的影响是单向的,周围车辆到自我车辆的连接是断开的。在式(6)中表现为当 $i=1$,即车辆为自我车辆时,车辆 j 到车辆 i 不连通,则 $\Delta V'_{ij}$ 的值为 0。又有 E_{ii} 为边特征矩阵的对角线元素,其含义为车辆自身状态对自身的影响,但为了更加符合实际车辆驾驶情况,应当使对角线元素表示为自身状态以及周围所有车的状态对自身车辆的影响,具体表示如下:

$$E_{im} = \sum_{j=1}^n E_{ijm} \quad (7)$$

其中, E_{ij} 具有 4 个特征维度的特征向量; m 代表特征向量的索引, m 的取值为 $m=1,2,3,4$ 。值得注意的是, E_{im} 的值不能为 0,为 0 则表示不产生影响,在此将其设置为 1。

此外,还需要考虑车辆行驶的历史信息,即车辆的坐标位置的变化和车辆速度的变化。在此定义车辆历史矩阵 \mathbf{T} ,其中 \mathbf{T} 的每个元素都会减去车辆起点坐标 (x_0,y_0) ,以表示车辆的位置变化情况,具体如式(8)所示:

$$T_i = [x_i - x_0 \ y_i - y_0] \quad (8)$$

最终得到车辆的整体行驶特征,包括车辆的状态特征、车辆的交互特征以及车辆历史特征。则整体行驶特征矩阵 \mathbf{X} 如下所示:

$$\mathbf{X} = \begin{bmatrix} \mathbf{V} \\ \mathbf{E} \\ \mathbf{T} \end{bmatrix} \quad (9)$$

5.3 边-增强注意力机制

边-增强注意力机制指增加顶点之间注意力的特征维度的机制。边-增强指增加边特征的维度,使得边特征能表达更多信息;注意力机制指给不同的顶点分配不同的权重系数,不同权重具有不同的优先级。顶点的权重越大,表示顶点的信息越丰富,影响程度的值越大。

在实际的交通场景下,道路越复杂,车辆的驾驶行为预测难度就越大,要获得较好的预测效果依赖于充分提取车辆交互的特征。传统的图神经网络模型如图神经网络(Graph Neural Network,GNN)^[15]与图注意力网络(Graph Attention Network,GAT)^[16]都不能很好地满足这个需求,GNN 只

不同的顶点代表不同的车辆,每个车辆有 5 维特征,分别

考虑了两个顶点之间是否存在连接关系,但不能表示不同顶点之间不同的影响;GAT在GNN的基础上添加了注意力机制,能够表示不同顶点的影响力大小,但GAT中两个顶点间的影响力是用单个权重值表示的,单一维度的权重值所包含的特征信息不够丰富,以上两种传统的图神经网络模型不能充分地表达边的特征,所以需要一种注意力机制来有效地提取出车辆交互的特征。

注意力机制体现在车辆之间的相对状态,例如某车辆距离自我车辆的距离很近,容易发生碰撞,因此将此车辆与自我车辆的边上赋予较高的权重,使得此车的影响变大。边增强注意力机制的改进在于边的特征可以是多维的,而不仅仅是一个实数,这就使得边包含的信息更多,权重系数分配更准确。如图6所示,顶点 p 与周围顶点的边特征向量用于计算周围车辆的权重值,此步骤最终的目的是生成一个带权值的邻接矩阵,以表示不同车辆之间的影响大小,邻接矩阵具体的表示如式(10)和式(11)所示:

$$\mathbf{A} = \sum_{i=1}^n \mathbf{E}'_i \mathbf{W}_a \mathbf{W}_a^T \mathbf{E}'_i^T \quad (10)$$

$$\mathbf{A}' = \text{softmax}(\mathbf{A}) \quad (11)$$

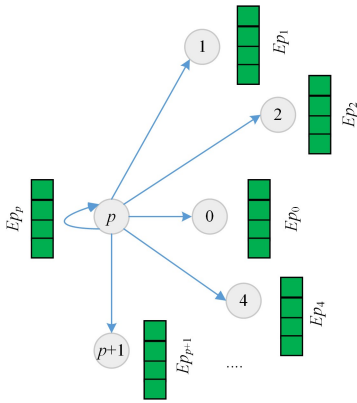


图6 边增强注意力机制

Fig. 6 Edge-enhanced attention mechanism

式(10)和式(11)描述了注意力矩阵 \mathbf{A}' 的过程,首先需要让边特征矩阵 \mathbf{E} 经过归一化得到 \mathbf{E}' ,然后让边特征矩阵与可训练的注意力参数矩阵 \mathbf{W}_a 相乘,得到注意力矩阵 \mathbf{A} ,随后将矩阵 \mathbf{A} 进行softmax计算得到 \mathbf{A}' ,使得 \mathbf{A}' 中元素的值在0到1的范围内,以便表示不同的权重。最后得到加权邻接矩阵 \mathbf{A}_{adj} :

$$\mathbf{A}_{\text{adj}} = \mathbf{E}' \mathbf{A}' \quad (12)$$

5.4 图卷积神经网络的特征传递机制

车辆间的邻接关系与交互特征以图的形式作为神经网络的输入,利用图卷积神经网络的特征传递方式进行信息更新,使得网络能更加充分地提取车辆之间的交互关系。由5.2节和5.3节可以得到节点特征矩阵 \mathbf{V} 和加权邻接矩阵 \mathbf{A}_{adj} ,这两个矩阵作为更新信息在构建的交互模型中进行传播,具体更新过程如式(13)所示:

$$\mathbf{H}^{k+1} = \tanh(\mathbf{A}_{\text{adj}} \mathbf{H}^k \mathbf{W}_g^k) \quad (13)$$

其中, \mathbf{H}^k 为隐藏矩阵,当 $k=0$ 时, $\mathbf{H}^0 = \mathbf{V}$,即为输入的节点特征矩阵; k 代表在当前EGCN的第 k 层进行计算, \mathbf{W}_g^k 代表可训练的权重矩阵,在训练时进行更新。最后,三者相乘的结果

经过激活函数后得到 \mathbf{H}^{k+1} 。因此,利用图卷积神经网络进行特征传递,可以较好地捕捉输入的车辆特征,使得预测更准确。

6 实验设置

6.1 实验环境设置

实验软件与硬件环境设置如表1和表2所列。

表1 硬件环境设置

硬件	参数
CPU	Intel(R) Core(TM) i7-10700F CPU@2.90 GHz
GPU	NVIDIA GeForce RTX 3070
内存	32GB DDR5 5200

表2 软件环境设置

软件	参数
操作系统	Windows 11 64 位
编程语言	Python 3.6
集成开发环境	Visual Studio Code
CUDA	11.2
深度学习框架	Tensorflow-GPU 2.6.0

6.2 数据集处理

本节对数据集进行介绍,首先获取有变道行为的中心车辆,然后获取中心车辆的周围车辆,最后获取车辆的边特征矩阵。

NGSIM^[17](Next Generation Simulation,NGSIM)是美国联邦高速公路管理局(Federal Highway Administration, FHWA)在2004年进行的研究项目,此项目通过在两组高速公路的检测路段上空安装高清摄影机的方式采集车辆的信息,后期利用图像处理技术对汽车的驾驶轨迹数据进行重建以还原车辆的高精度轨迹数据。

对于数据集的选择,本文采用NGSIM,即下一代交通仿真数据,进行车辆驾驶行为的研究。

表3 NGSIM的公路信息

采集公路	I-80	US101
道路类型	高速公路	高速公路
道路长度	503 m	640 m
采集时常	45 min	45 min
采集时段	04:00 p. m. -04:15 p. m.	07:50 a. m. -08:05 a. m.
	05:00 p. m. -05:15 p. m.	08:05 a. m. -08:20 a. m.
	05:15 p. m. -05:30 p. m.	08:20 a. m. -08:35 a. m.

NGSIM采集公路的信息如表3所列,本文选用I-80公路对车辆驾驶行为进行研究。I-80公路位于加利福尼亚州80号洲际公路,车辆的驾驶方向为由南向北。I-80公路包含了8条车道,分别是主路车道1-5号车道,车道的编号由左向右依次递增,车道7为进口匝道,车道8为出口匝道,I-80检测路段的公路示意图如图7所示。在车辆数较少的情况下,即道路不拥堵的情况下,大多数车辆会保持车道不变,导致车辆驾驶行为单一,不能满足本文的研究要求,因此本文选择I-80的第三个采集时间段,此时间段内公路上的车辆更多,交通情况更加复杂,车辆会频繁变换车道,

车辆的驾驶行为信息更加丰富。

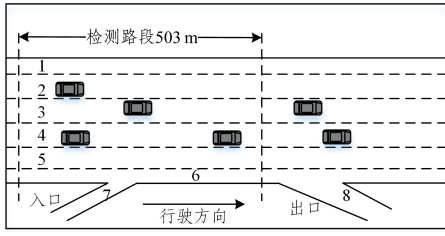


图7 I-80公路检测路段示例图

Fig.7 Example of the I-80 freeway section

表4是本文需要的NGSIM相关字段描述。本文规定每辆车每隔0.5s进行一次数据采集,而Frame_ID则是每隔0.1s生成一次,因此本文采集的Frame_ID的间隔为5。Lane_ID代表公路上不同的车道编号,因此车辆的变道信息可以由Lane_ID处理得到。

表4 NGSIM轨迹数据字段描述

Table 4 Data field description in NGSIM dataset

数据名称	数据描述	单位
Vehicle_ID	按车辆进入顺序编号	—
Frame_ID	该条数据在某一时刻的帧	—
Global_Time	时间戳	ms
Local_X	车头中心与道路左侧的距离	Feet
Local_Y	车头中心与道路起点的距离	Feet
v_Vel	车辆瞬时速度	Feet/s
Lane_ID	车辆在某一时刻所在车道编号	—

6.2.1 获取有变道行为的中心车辆

由于车辆在一般情况下都倾向于保持原车道驾驶,变道行为相较而言较少发生,因此在选择自我车辆时需要选择有变道行为的车辆,这样对周围车辆的驾驶行为进行预测才有意义。因此,从原始数据中筛选出变道车辆的数据的操作流程如算法2所示。

算法2 变道车辆选择

输入:原始行驶数据S,变道的车辆集合F,变道车辆数c

输出:F,c

1. 初始化: $L = \emptyset$, $F = \emptyset$, $c = 0$;
2. for ID in set(S.Vehicle_ID) do;
3. $T = \text{sort}(S.Vehicle_ID = ID) \text{ as } Global_Time$;
4. for i to len(T)-1 do
5. if $T[i+1] - T[i] \neq 100$ then
6. $L.append(i+1)$;
7. for j to len(T)-1 do
8. $M = T[L[j]:L[j+1]]$
9. if Lane_ID changed then
10. $F.append(M)$;
11. $c += 1$;
12. return F,c.

算法2的主要功能是从NGSIM的I-80数据S里选择具有变道行为的车辆数据F。其中的关键点在于需要将同一Vehicle_ID的车辆按时间顺序排列,将Lane_ID发生变化的车辆的数据加入到F中,之后的步骤将基于此数据进行操作。

6.2.2 获取中心车辆的周围车辆

车辆上所部署的传感器有一定距离限制,同时,车辆与

车辆之间的相互影响也有距离限制,车辆之间距离越远,影响越小,因此需要选择离中心车辆一定距离内的车辆作为周围车辆。如算法3所示,根据车辆的Vehicle_ID和Frame_ID确定中心车辆与当前帧内存在的所有车辆,并计算每台车辆与中心车辆的距离,选择在距离dis内的车辆作为有效的周围车辆集合C中的元素。

算法3 选择中心车辆的周围车辆

输入:原始形式数据S,当前车辆编号Vehicle_ID,当前帧编号Frame_ID,周围车辆集合C,车辆间距dis

输出:C

1. 初始化: $C = \emptyset$
2. $get(\text{Vehicle_ID's coordinates } x, y \text{ at } \text{Frame_ID})$;
3. if S.Frame_ID == Frame_ID then
4. $T = S$;
5. $T.add(d)$;
6. $T[d] = 0.3048 \times \sqrt{(T['local_x'] - x)^2 + (T['local_y'] - y)^2}$;
7. if $T[d] < dis$ then
8. $C.append(T)$;
9. return C

6.2.3 获取边特征矩阵

由5.2节可知本文设计的车辆交互模型为图结构,具有顶点特征V与边特征E。顶点特征V矩阵的获取较为简单,可以直接根据式(5)得到,边的特征矩阵的获取较为复杂,算法4对边特征矩阵的获取进行了描述。关键的步骤为处理对角线元素,使边特征矩阵的每个元素不为0。

算法4 计算边特征矩阵

输入:顶点特征矩阵V

输出:边特征矩阵E

1. 初始化: $V = V[:, : - 1]$;
2. for $V_1 \in V$ do
3. for $V_2 \in V$ do
4. $E = |V_1 - V_2|$;
5. $E = E.sum(1)$; // 在E的维度1上进行求和计算
6. for i to len(V)-1 do
7. for j to 4 do
8. if $E[i][j] > 0$
9. $E[i][j] = E[i][j]$;
10. else
11. $E[i][j] = 1$;
12. return E

6.3 实验评估指标

本文的任务为预测自我车辆的周围车辆的驾驶行为,而车辆的驾驶行为是有限的离散值,因此可以将车辆的驾驶行为预测看作分类任务,本文采用的评价指标为分类准确率。

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} = \frac{N_{right}}{N_{all}} \quad (14)$$

其中,TP(True Positives)表示实际为正例且被分类器判定为正例的样本数;FP(False Positives)表示实际为负例且被分类器判定为正例的样本数;FN(False Negatives)表示实际为正例但被分类器判定为负例的样本数;TN(True Negatives)表示实际为负例且被分类器判定为负例的样本数^[18]。总的来说,就是预测正确的样本数量 N_{right} 与总样本数量 N_{all} 的比值。

精准率 Pre 表示预测为正的样本数量与实际为正的样本数量的比值,具体公式如式(15)所示:

$$Pre = \frac{TP}{TP + FP} \quad (15)$$

召回率 Re 表示在所有实际为正的样本中被预测为正确样本的比值,具体公式如式(16)所示:

$$Re = \frac{TP}{TP + FN} \quad (16)$$

$F1$ 分数是一种综合了精准率与召回率的指标,如式(17)所示:

$$F1 = 2 * \frac{Pre * Re}{Pre + Re} \quad (17)$$

除此之外,还可以采用混淆矩阵对模型的性能进行评估,混淆矩阵能够直接看出模型对每种驾驶行为的分类效果。混淆矩阵的具体表示如下:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & h_{ij} & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{nm} \end{bmatrix} \quad (18)$$

混淆矩阵 \mathbf{H} 中的元素 h_{ij} 表示类别 i 被识别为 j 的样本数量,即对角线元素代表预测正确的样本数量。

另外,在多分类任务中还可选用宏平均指标对模型进行性能评估,宏平均的计算公式如下:

$$Macro_Pre = \frac{1}{N} \sum_i P_i \quad (19)$$

$$Macro_Re = \frac{1}{N} \sum_i R_i \quad (20)$$

$$Macro_F1 = \frac{1}{N} \sum_i F1_i \quad (21)$$

其中, N 表示类别的总数; P_i , R_i 以及 $F1_i$ 分别代表第 i 类的精准率、召回率以及 $F1$ 分数。宏平均会对每一个类进行算术平均,表示类中每一个种类都同等重要。

最后,为了测试网络模型的稳定性与计算速度,需要将模型的计算时间作为模型的性能评价指标。即:

$$t = t_{end} - t_{start} \quad (22)$$

其中, t 是模型计算单个输入时所花费的时间,是衡量模型任务计算快慢的重要指标; t_{start} 是模型接收到输入的时间; t_{end} 是模型将输入数据计算完毕得到输出的时间。此外,模型的稳定性体现在模型的计算时间是否会根据不同的输入数据而产生较大的波动。

6.4 对比算法

为了验证本文提出框架的效果,选取了如下 4 种具有代表性的模型进行对比。

1) CNN^[19]: 卷积神经网络在图像领域发挥着巨大作用,它主要利用卷积层来提取输入值特征,再配合池化层和全连接层来得到输出。在交通场景下,可以将车辆的特征以二维数组的形式进行表示,可以近似地看作一张图片,再利用 CNN 来进行特征的提取,最后经过全连接层得到车辆驾驶行为的预测结果。

2) LSTM^[19]: 长短期记忆网络在具有时间相关性的任务中有着出色的表现,它主要利用输入门、输出门、遗忘门这 3 个控制门来对数据的操作进行控制,使得数据之间的时间

相关性得以被发掘。在交通场景下,可以将车辆的特征按照时间顺序组合为序列交给 LSTM 进行处理,根据提取的特征预测得到未来车辆的轨迹序列,从而得到对车辆驾驶行为的预测。

3) CNN-LSTM^[19]: 使用 CNN 和 LSTM 的复合模型可以结合两个模型的优点,既考虑了车辆的空间信息,又利用了车辆之间的时间相关性。

4) BP^[20]: 一种按误差进行反向传播训练的多层前馈网络。BP 网络能学习给定输入与输出之间的映射关系,不需要提前指出输入输出之间具体的函数关系。BP 网络通常使用梯度下降的方法,通过反向传播来不断调整网络的参数,使网络的损失值不断减小。

7 实验设置结果与分析

7.1 通信有效性评估

7.1.1 传输数据量评估

传输数据量评估是将所提模型与参数同步 DDNN^[21]、集中式、EGCN-CeDML 这 3 种方式进行比较。

1) 参数同步 DDNN: DDNN 在训练时需要每个边缘设备在每次迭代中与云端服务器进行神经网络参数同步。每个边缘设备中的神经网络大小都为 $(784 \times 100 + 100 \times 10 + 100 + 10) \times 4 \text{ Bytes} / 1024 = 310.59 \text{ KB}$, 并且每一次迭代都需要向云端服务器发送模型参数。在云端融合了不同边缘设备的参数后,再下发到每个边缘设备中,因此一次迭代所产生的数据量为 $2 \times 310.59 \text{ KB} = 621.18 \text{ KB}$ 。经过完整的 DDNN 训练后,需要传输的数据总量为 $500 \times 10 \times 621.18 \text{ KB} = 2.96 \text{ GB}$ 。

2) 集中式: 集中式训练需要将所有原始数据发送到云端服务器进行训练。集中式训练所需要传输的原始训练数据大小为 $50 \text{ KB} \times (784 + 10) \times 4 \text{ Bytes} / 1024^2 = 151.44 \text{ MB}$, 当云端服务器完成神经网络的训练后需要将网络模型下发到各个边缘设备,则需要多传输 $10 \times 310.59 \text{ KB} / 1024 = 3.03 \text{ MB}$ 的数据,因此集中式训练产生的数据量为 $(151.44 + 3.03) \text{ MB} = 154.47 \text{ MB}$ 。

3) EGCN-CeDML: EGCN-CeDML 不需要发送所有原始数据到云端服务器,仅需构建较小的公共数据集就可以完成训练。在 EGCN-CeDML 中需要传输的公共数据集大小为 $10 \times 1000 \times 784 \times 4 \text{ Bytes} / 1024^2 = 29.91 \text{ MB}$, 并且将公共数据集输入到本地模型 LNN 中还将产生 $10 \times 1000 \times 10 \times 4 \text{ Bytes} / 1024^2 = 0.38 \text{ MB}$ 的数据,这些数据用于重新标记阶段的新标签制作。当云端的全局模型 ENN 训练完毕后,需要将神经网络发送到各个边缘设备使用,因此 EGCN-CeDML 产生的总的传输数据量为 $(29.91 + 0.38 + 3.03) \text{ MB} = 33.32 \text{ MB}$ 。

基于以上讨论,通过计算发现 EGCN-CeDML 需要传输的数据量仅为集中训练的 21.57%, 为参数同步方式训练的 1.10%。

如表 5 所列,对较为简单的 MNIST 数据集来说,EGCN-CeDML 仅需要从原始数据集中抽取 10% 的数据组成公共数据集便可以接近使用所有数据训练得到的精度,两者的精度差值小于 1%。这说明 EGCN-CeDML 仅需要传输数据量

较小的公共数据集便能达到较高的分类准确性。

表5 不同的数据集的性能表现

Table 5 Performance of different datasets

数据集	训练方案	神经网络模型	样本数量/%	分类精度/%
MNIST	EGCN-CeDML	BP	10	91.60
MNIST	集中式	BP	100	92.12

7.1.2 边缘设备数量对准确性的影响

本节将探究边缘设备数量对EGCN-CeDML准确性的影响,采用MNIST数据分别在边缘设备数量为5,10,20,30,40,50的情况下进行实验验证,结果如图8所示。

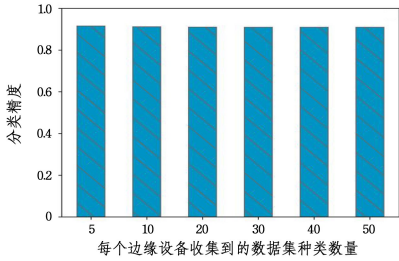


图8 边缘设备数量对准确性的影响

Fig. 8 Effect of the number of edge devices on accuracy

由图8可知,随着边缘设备的数量增加,EGCN-CeDML对MNIST的分类准确性基本保持稳定,这是由于最终训练阶段所使用的公共数据集没有改变,因此全局模型ENN学习到的特征基本不变。由此可知,EGCN-CeDML能在不同的边缘设备数量下保持稳定的准确性。

7.1.3 通信时延评估

通信时延在实际应用中有着十分重要的作用,本节将对EGCN-CeDML的通信时延进行计算,并与集中式和参数同步式的时延进行比较。其中,集中式训练的通信时延包括各个边缘设备发送所有原始数据到云端服务器所消耗的时间以及云端服务器向边缘设备发送神经网络所消耗的时间,DDNN的通信时延包括各个边缘设备在多次训练迭代中与云端服务器进行参数同步所消耗的时间总和。

表6 各方案的传输时延

Table 6 Transmission delay of each scheme

方案	数据量/MB	调制方式	传输时延/ms
集中式	154.47	QPSK	85799
DDNN	3031.04	QPSK	1680656
EGCN-CeDML	33.32	QPSK	18504
集中式	154.47	64-QAM	15924
DDNN	3031.04	64-QAM	311906
EGCN-CeDML	33.32	64-QAM	3436

假设带宽为10MHz,调制方式如果选择QPSK,则 $n_{nbpm}^{NR} = 1.03$;如果选择64-QAM,则 $n_{nbpm}^{NR} = 5.55$,可以计算得到时延,各个方案的时延情况如表6所列。从表6可以看出,EGCN-CeDML的传输时延仅为DDNN的1.10%,集中式训练法的21.57%,因此EGCN-CeDML框架具有较低的延迟。

上述实验结果表明,EGCN-CeDML能以较低的通信量得到较好的神经网络性能,并且时延较低,说明了EGCN-CeDML具有良好的通信性能。

7.1.4 数据重标记算法的分类准确度评估

EGCN-CeDML的另一个优势在于其可以处理异构数据分布问题。由于边缘设备分布在不同的位置,收集到的数据也有差异,比如边缘设备只能收集到一半类型的数据时,联合多个边缘设备,便可以得到全部类型的数据。因此本节将研究每个边缘设备收集到的不同样本种类数量与使用DR算法对准确性的影响。这里选用了图像识别领域的数据集MNIST^[22]。

MNIST是手写数字数据集,数据集包含了60000个训练数据和10000个测试数据。每个样本是一张手写数字图片,每张图片的标签对应0-9的一个数字。样本的像素大小为 28×28 ,在数据集中表现为一个784维向量。

图9给出了MNIST在BP上的准确性与每个边缘设备收集到的样本种类的关系。其中Y轴代表分类精度,X轴代表每一个边缘设备上收集到的样本种类的数量(MNIST数据集是一个十分类的数据集,样本种类总数为10),其中绿色的柱状图表示没有经过DR算法处理过的结果,蓝色的柱状图表示经过了DR算法处理的结果。从整体上看,无论DR算法是否处理过,分类精度都随着边缘设备收集到的样本种类数量的增加而提高。这是由于边缘设备收集到了更多种类的样本使得网络可以获取更加丰富的图像特征,学习到更多的信息,从而提升了分类准确性。其次,从图中可以看出,利用DR算法处理数据后,大多数情况下可以使得神经网络的分类准确性更高,这是由于DR利用协方差矩阵去除了错误的样本使得网络获得的样本是正确的,去除了干扰,提升了网络的性能。最后由图9可知,当每个边缘设备收集到的样本种类数在5以上,网络的分类准确性的增幅变缓,这是由于每个边缘设备都能收集到一半以上的样本种类,并且收集到的样本种类是不相同的,因此聚合多个网络后,便可以接近收集所有种类样本的效果,达到较高的分类准确性,分类准确性稳定在91.60%左右。

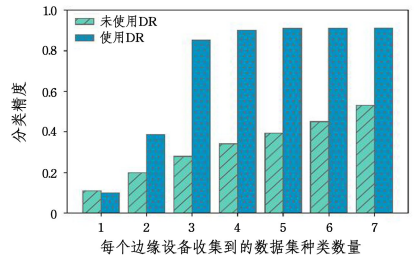


图9 MNIST在BP上的准确性与样本种类数量的关系

(电子版为彩图)

Fig. 9 Accuracy vs. the number of sample types of MNIST on BP

通过本节的实验可知,EGCN-CeDML拥有较好的准确性,当每个边缘设备能收集到的样本种类的数量增加时,神经网络的性能也会随之提升。同时,使用了DR算法后,EGCN-CeDML仅需收集一半种类的数据就可以达到较高的准确率。

7.2 模型预测性能评估

本文利用不同的性能指标对模型的预测性能进行评估,各个模型的混淆矩阵如图10所示。KL表示保持直行,TL表示左转,TR表示右转。以图10(d)为例,第一行表示总共

有 113 个保持直行的样本,其中模型预测为保持直行的样本数为 108,左转弯的样本数为 4,右转弯的样本数为 1。从混淆矩阵可以看出,各个模型对保持直行的样本预测正确的数量最多,右转弯最难以判断,这是因为在实际驾驶过程中,车辆大多保持直行,而车辆在变道时大多向左变道,右转弯的行为相对较少。并且对比 4 个模型的混淆矩阵可知,本文所使用的 EGCN-LSTM 模型在各个行为的预测上都能体现出一定的优势。

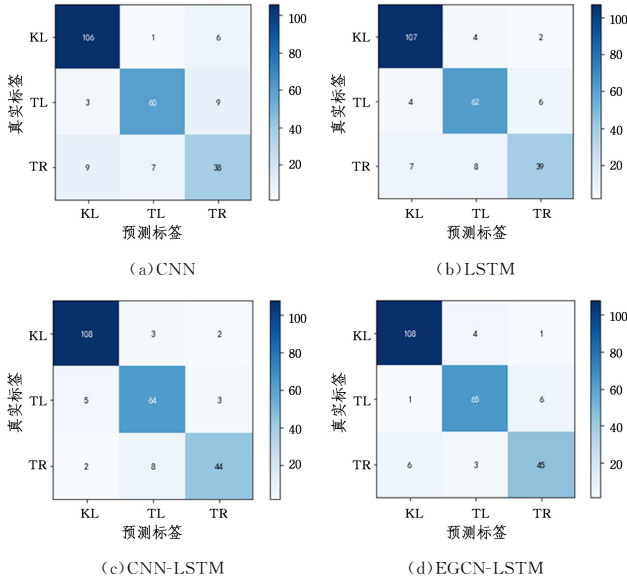


图 10 各个模型的混淆矩阵

Fig. 10 Confusion matrix of each models

此外还计算了每种行为上的精准率、召回率以及 F1 分数,如表 7—表 10 所列。

表 7 CNN 模型在测试集的性能

Table 7 Performance of CNN on test set

	保持直行	左转	右转
精准率	0.8983	0.8823	0.7169
召回率	0.9380	0.8333	0.7037
F1 分数	0.9177	0.8571	0.7102

表 8 LSTM 模型在测试集的性能

Table 8 Performance of LSTM on test set

	保持直行	左转	右转
精准率	0.8991	0.8611	0.8125
召回率	0.9469	0.8611	0.7222
F1 分数	0.9224	0.8611	0.7647

表 9 CNN-LSTM 模型在测试集的性能

Table 9 Performance of CNN-LSTM on test set

	保持直行	左转	右转
精准率	0.9391	0.8533	0.8979
召回率	0.9557	0.8888	0.8148
F1 分数	0.9473	0.8707	0.8543

表 10 EGCN-LSTM 模型在测试集的性能

Table 10 Performance of EGCN-LSTM on test set

	保持直行	左转	右转
精准率	0.9391	0.9027	0.8653
召回率	0.9557	0.9027	0.8333
F1 分数	0.9473	0.9027	0.8490

由表 7—表 10 可知,保持直行的 F1 分数(F1 分数兼顾了精准率与召回率)最大,说明 EGCN-LSTM 模型对保持直行的行为预测效果最好;右转的 F1 分数最小,说明对右转的行为预测效果较差。对比可知,EGCN-LSTM 模型对车辆的各种行为都有较好的预测效果。

最后,对各个模型的综合表现进行评估,具体指标如表 11 所列。从表 11 可知,EGCN-LSTM 的准确率达到 91.21%,与其他对比模型相比体现出了一定的优势。其中,由于 CNN 与 LSTM 模型分别仅考虑空间特征或时间特征,使得整体的准确率偏低;CNN-LSTM 在一定程度上综合考虑了车辆的空间与时间特征,所以能一定程度上提升对车辆行为预测的准确率。本文采用的 EGCN-LSTM 除了考虑了时空信息外,还更多地考虑了车辆间的邻接关系与交互信息,使得车辆间的特征信息更加丰富,准确率更高。

表 11 模型测试结果

Table 11 Prediction accuracy comparison with baselines

	Acc	Macro_pre	Macro_Re	Macro_F1
CNN	0.8539	0.8325	0.8205	0.8283
LSTM	0.8702	0.8575	0.8434	0.8434
CNN-LSTM	0.9037	0.8908	0.8864	0.8908
EGCN-LSTM	0.9121	0.9024	0.8972	0.8997

7.2.1 预测时长对准确性的影响

不同预测时间的长度对预测的效果(本节对车辆进行预测的动作为直行、左转与右转)可能会产生不同的影响,其中预测时间长度指基于历史轨迹记录,对未来行为进行预测的一个时间范围。例如:基于相同的车辆历史数据对未来 1s 或者 2s 的行为做出的预测可能都会有不同的预测准确性,因此本节将探究预测时间长度对预测准确性的影响。

在图 11 中,横轴代表的是预测时间的长度,单位为 s;纵轴代表的是对行为预测的准确性,不同颜色的线条代表了不同的神经网络模型。由图 11 可以看出,随着预测时间的增加,行为预测的准确性逐步降低,这是因为随着预测时间的增加,车辆驾驶行为的不确定性增加,车辆的历史信息作用减弱,使得整体的预测准确性降低。其中,使用 EGCN 能够保持较好的预测效果,在 1s 时预测效果达到了 91.21%,相比其他对比算法体现出了一定的优势。CNN 与 LSTM 模型分别只考虑了车辆的空间信息与时间信息,因此预测效果较差。而 CNN-LSTM 复合模型考虑了车辆的时间与空间信息,预测效果比单独的 CNN 与 LSTM 模型更好。但 CNN-LSTM 也未考虑车辆之间的交互关系,预测的准确性低于 EGCN 模型。

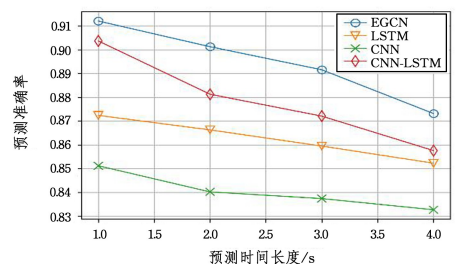


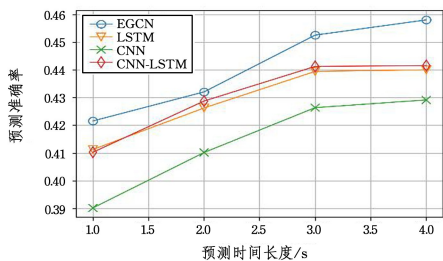
图 11 驾驶行为预测准确性与预测时间长度的关系

Fig. 11 Prediction accuracy vs. prediction time of driving behaviors

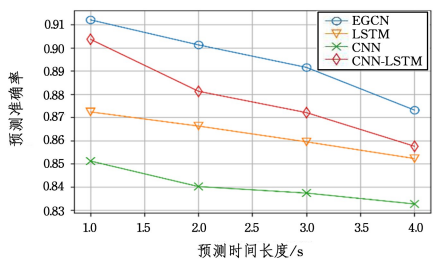
7.2.2 驾驶行为对准确性的影响

车辆的驾驶行为具有不同的复杂程度,如车辆的驾驶行为可分为直行、左转、右转,或者是匀速直行、加速直行、减速直行、左转、右转。不同的驾驶行为复杂程度可能对预测的准确性有着不同的影响。因此本节将探究驾驶行为对预测准确性的影响。

如图 12 所示,不同的预测动作对预测的结果有着重要影响。如图 12(a)所示,当需要对 5 种车辆行为进行预测时,准确率为 45.81%,车辆对加速直行、减速直行等行为预测的效果不佳。导致此结果的原因在于车辆行驶数据中包含的特征信息不够,因此不能对带有速度信息的行为进行更好的预测。随着预测时间的增加,左转或右转的车辆增加,加速直行、减速直行的车辆数减少,整体的预测准确性上升。由图 12(b)可知,在不需要对加速直行、减速直行等行为进行预测时,准确率达到了 91.21%,说明车辆行驶的特征信息对于直行、左转、右转等动作的预测是足够的。



(a) 预测动作为匀速直行、加速直行、减速直行、左转、右转



(b) 预测动作为直行、左转、右转

图 12 驾驶行为预测准确性与驾驶行为的关系

Fig. 12 Prediction accuracy vs. driving behaviors

7.2.3 周围车辆数对模型计算时间的影响

模型计算的快慢与稳定性在实际的道路场景中有着重要的意义。由于实际道路中需要进行预测的车辆数量是不确定的,模型需要在周围车辆数量变化的情况下,保持较快的计算速度与稳定性,因此本节将探究周围车辆数对模型计算时间的影响。

图 13 给出了模型计算时间与周围车辆数目的关系。从图中可以看出,当周围车辆的数目增加时,模型的计算时间会增加。其中 EGCN 采用的是图结构,它可以灵活增加车辆数目而不增加计算迭代次数,使得模型在计算时能保持较少的计算时间,并且随着车辆数目增加能保持计算时间的相对稳定。LSTM 会随着车辆数目增加而增加迭代计算次数,从而导致计算时间增加。CNN 需要规定输入车辆的数目,当超出规定的车辆数目时,则会使得迭代次数增加,计算时间随之增加。CNN-LSTM 与 CNN 输入结构类似,但 CNN-LSTM 增加了 LSTM 模块,额外的计算会使得计算时间增加。

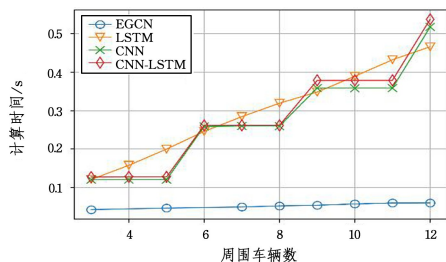


图 13 计算时间与周围车辆数目的关系

Fig. 13 Computing time vs. the number of vehicles around

结束语 本文提出了一种基于边-增强图卷积神经网络的通信有效的分布式机器学习框架 EGCN-CeDML 以解决复杂交通场景下车辆驾驶行为预测的问题。EGCN-CeDML 是一种通信有效的分布式机器学习框架,边缘设备直接对用户数据进行本地存储并基于知识蒸馏与协方差计算进行处理,无须将所有原始数据发送到云服务器,降低了传输数据量和通信延迟,缓解了集中训练神经网络的压力。其中边缘设备部署的边-增强的图卷积神经网络综合利用了边-增强注意力机制和图卷积神经网络的特征传递机制,捕获了丰富的车辆交互特征,充分表示了车辆之间的交互关系,提高了车辆驾驶行为的准确性。由于边-增强的图卷积神经网络采用的是图结构,可以灵活增加车辆数目而不需要增加计算迭代次数,使得模型在计算时能保持稳定的计算时间,从而协助车辆做出快速准确的行为决策,提高道路行驶的安全性。在 EGCN-CeDML 框架中,各边缘设备可以根据自身的计算能力和存储能力,在保证神经网络性能的前提下灵活选择神经网络类型和控制其规模。不限于车辆驾驶行为预测,基于本文提出的分布式机器学习框架还可以为不同的交通应用提供分布式机器学习设计思路与解决方案,如轨迹预测、交通流量预测等。本文在智能交通系统中,通过预测周围车辆的驾驶行为,可以为车辆提供警告信息,协助车辆提前做出行为决策,提高道路行驶的安全性。

参考文献

- [1] XING E P, HO Q, DAI W, et al. Petuum: A new platform for distributed machine learning on big data[C]//Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2015:1335-1344.
- [2] ANDREW G. Bring HPC techniques to deep learning [EB/OL]. <https://xueshu.baidu.com/bringing-hpc-techniques-deep-learning>.
- [3] MA Y, ZHU X, ZHANG S, et al. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019:6120-6127.
- [4] AMDAHL G M. Validity of the single processor approach to achieving large scale computing capabilities[C]//Spring Joint Computer Conference. 1967:483-485.
- [5] GONG L, CHENG Q. Exploiting edge features for graph neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019:9211-9219.
- [6] LEFÈVRE S, VASQUEZ D, LAUGIER C. A survey on motion

- prediction and risk assessment for intelligent vehicles [J]. *ROBOMECH Journal*, 2014, 1(1):1-14.
- [7] LI Z, GONG J, LU C, et al. Interactive Behavior Prediction for Heterogeneous Traffic Participants in the Urban Road: A Graph-Neural-Network-Based Multitask Learning Framework [J]. *IEEE/ASME Transactions on Mechatronics*, 2021, 26(3): 1339-1349.
- [8] KUMAR P, PERROLLAZ M, LEFEVRE S, et al. Learning-based approach for online lane change intention prediction[C]// 2013 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2013: 797-802.
- [9] HE G, LI X, LV Y, et al. Probabilistic intention prediction and trajectory generation based on dynamic bayesian networks[C]// 2019 Chinese Automation Congress (CAC). IEEE, 2019: 2646-2651.
- [10] AOUBE G S, DESARAJU V R, STEPHENS L H, et al. Driver behavior classification at intersections and validation on large naturalistic data set [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2012, 13(2):724-736.
- [11] DEO N, TRIVEDI M M. Convolutional social pooling for vehicle trajectory prediction[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018: 1468-1476.
- [12] GAO J, SUN C, ZHAO H, et al. Vectornet: Encoding hd maps and agent dynamics from vectorized representation[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020:11525-11533.
- [13] BUCILA C, CARUANA R, NICULESCU-MIZIL A. Model Compression[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06). 2006.
- [14] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. arXiv:1503.02531, 2015.
- [15] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model[J]. *IEEE Transactions on Neural Networks*, 2008, 20(1):61-80.
- [16] WANG X, JI H, SHI C, et al. Heterogeneous graph attention network[C]// The World Wide Web Conference. 2019:2022-2032.
- [17] KRAJEWSKI R, BOCK J, KLOEKER L, et al. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems[C]// 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018:2118-2125.
- [18] GUO J. Research on Precise Hybrid Recommendation Method Based on Feature Engineering and Efficient Gradient Boosting Tree Algorithm[D]. Wuhan: Wuhan University of Science and Technology, 2020.
- [19] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. *Communications of the ACM*, 2017, 60(6):84-90.
- [20] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning Representations by Back Propagating Errors[J]. *Nature*, 1986, 323(6088):533-536.
- [21] TEERAPITTAYANON S, MCDANEL B, KUNG H T. Distributed Deep Neural Networks Over the Cloud, the Edge and End Devices[C]// 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017:328-339.
- [22] LECUN Y, BOTTOU L. Gradient-based learning applied to document recognition[C]// Proceedings of the IEEE. 1998:2278-2324.



LI Ke, born in 1985, Ph.D, lecturer, is a member of China Computer Federation. Her main research interests include intelligent transportation system and Internet of vehicles.

(责任编辑:何杨)