软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2023,34(6):2586−2605 [doi: 10.13328/j.cnki.jos.006851]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于凸优化的无人驾驶汽车转向角安全性验证*

吴慧慧[1,2], 张亚楠[3], 侯 刚[1,2], 渡边政彦[4], 王 洁[1,2], 孔维强[1,2]

[1](大连理工大学 软件学院, 辽宁 大连 116024)

[2](辽宁省泛在网络与服务软件重点实验室, 辽宁 大连 116024)

[3](中汽数据有限公司, 天津 300300)

[4](NTT DATA Automobiligence Research Center, Yokohama 222-0033, Japan)

通信作者: 孔维强, E-mail: wqkong@dlut.edu.cn

摘 要: 无人驾驶汽车系统过大的输入-输出空间(即输入和输出的所有可能组合), 使得为其提供形式化保证变成一项具有挑战性的任务. 提出了一种自动验证技术, 通过结合凸优化和深度学习验证工具DLV来保障无人驾驶汽车的转向角安全. DLV是一个用于自动验证图像分类神经网络安全性的框架. 运用故障安全轨迹规划中的凸优化技术解决预测转向角的判断问题, 然后拓展DLV来实现无人驾驶汽车转向角安全性的验证. 在NVIDIA的端到端无人驾驶架构上说明所提出方法的优势, 该架构是许多现代无人驾驶汽车的关键组成部分. 实验结果表明: 对于给定的区域和操作集, 如果存在对抗性错误分类(即不正确的转向决策), 该技术可以成功地找到, 因此可以实现安全验证(如果在所有DNN层都没有发现错误分类, 在这种情况下, 网络关于转向决策可以说是稳定或可靠的)或证伪(在这种情况下, 这些对抗性反例可以用于后续微调网络).

关键词: 无人驾驶汽车; 转向角; 自动驾驶汽车; 凸优化; 安全性验证

中图法分类号: TP311

中文引用格式: 吴慧慧, 张亚楠, 侯刚, 渡边政彦, 王洁, 孔维强. 基于凸优化的无人驾驶汽车转向角安全性验证. 软件学报, 2023, 34(6): 2586–2605. http://www.jos.org.cn/1000-9825/6851.htm

英文引用格式: Wu HH, Zhang YN, Hou G, Watanabe M, Wang J, Kong WQ. Verification of Steering Angle Safety for Self-driving Cars Using Convex Optimization. Ruan Jian Xue Bao/Journal of Software, 2023, 34(6): 2586–2605 (in Chinese). http://www.jos.org.cn/1000-9825/6851.htm

Verification of Steering Angle Safety for Self-driving Cars Using Convex Optimization

WU Hui-Hui[1,2], ZHANG Ya-Nan[3], HOU Gang[1,2], WATANABE Masahiko[4], WANG Jie[1,2], KONG Wei-Qiang[1,2]

[1](School of Software Technology, Dalian University of Technology, Dalian 116024, China)

[2](Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116024, China)

[3](Automotive Data of China (Tianjin) Co. Ltd., Tianjin 300300, China)

[4](NTT DATA Automobiligence Research Center, Yokohama 222-0033, Japan)

Abstract: Providing formal guarantees for self-driving cars is a challenging task, since input-output space (i.e., all possible combinations of inputs and outputs) is too large to explore exhaustively. This paper presents an automated verification technique ensuring steering angle safety for self-driving cars by incorporating convex optimization and deep learning verification (DLV). DLV is an automated verification framework for safety of image classification neural networks. The DLV is extended by convex optimization technique in fail-safe

* 基金项目: 国家重点研发计划(2020YFB2009500); 中央高校基本科研业务费专项资金(DUT20TD107, DUT22ZD203); NTT DATA 智能汽车研究所

本文由“软件可信性与供应链安全前沿进展”专题特约编辑向剑文教授、郑征教授、申文博研究员、常瑞副教授、田聪教授推荐.

收稿时间: 2022-09-05; 修改时间: 2022-10-10; 采用时间: 2022-12-14; jos在线出版时间: 2023-01-13

trajectory planning to solve the judgement problem of predicted steering angle, and thus, to achieve verification of steering angle safety for self-driving cars. The benefits of the proposed approach are demonstrated on the NVIDIA's end-to-end self-driving architecture, which is a crucial ingredient in many modern self-driving cars. The experimental results indicate that the proposed technique can successfully find adversarial misclassifications (i.e., incorrect steering decisions) within given regions and family of manipulations if they exist. Therefore, the safety verification can be achieved (if no misclassification is found for all DNN layers, in which case the network can be said to be stable or reliable w.r.t. steering decisions) or falsification (in which case the adversarial examples can be used to fine-tune the network).

Key words: self-driving cars; steering angle; autonomous vehicle; convex optimization; safety verification

在实现无人驾驶汽车方面, 如何保证其行为的安全性, 仍然是一个亟待解决的问题. 尽管无人驾驶汽车在过去的十几年中取得了重大进展, 例如, 在没有任何人为干预的情况下行驶了数百万英里, 但在某些危急情况下, 它们仍然可能导致发生交通事故. 一些涉及无人驾驶汽车的交通事故已被报道[1−3], 其中一些甚至引发致命的碰撞. 因此, 迫切需要一种形式化的方法为无人驾驶汽车的行为安全性提供保证.

无人驾驶汽车的深度神经网络(DNN)从不同的传感器获取输入信息, 这些传感器包括摄像机、雷达和红外传感器等, 传感器用于测量驾驶环境, 输入信息经过 DNN 的逐层处理, 最后输出当前条件下安全操纵汽车所需的转向角、制动、速度等指令. 然而, 文献[4,5]中的研究发现: 包括无人驾驶系统在内的基于 DNN 的软件在输入图像发生微小变化时是不稳定的, 并且可能会做出未预见的或不正确的行为预测[5]. 对于这些未预见的错误预测行为, DNN 驱动的无人驾驶汽车可能会引发致命的碰撞. 这显然会给无人驾驶系统等对安全性要求较高的应用程序带来潜在的安全问题, 所以需要自动验证技术来验证其决策的正确性[6]. 但是由于无人驾驶汽车会根据不同的传感器(例如摄像机、红外障碍物检测器等)所测量的驾驶环境来调整汽车本身的行为, 这使得输入-输出空间(即输入和输出的所有可能组合)太大而无法执行彻底的探索. 另一方面, 应用测试为无人驾驶汽车提供安全保障是困难的, 因为测试所需驾驶场景的数量多, 并且测试时间长. 最近一项研究表明, 无人驾驶汽车需要进行长达 4.4 亿公里的测试来证明它们比人类的驾驶具有更好的性能[7]. 这意味着需要 100 辆汽车每天 24 小时不间断地行驶 12.5 年[8].

一类保证无人驾驶汽车行为安全性的方法是定理证明[9−12], 这类方法通过检查表达所需系统性质的逻辑公式的可满足性来加以验证. 尽管定理证明的方法强大而有效, 但是通常需要人工干预才能生成所需系统行为和逻辑公式, 而这些行为和逻辑公式必须经常适应新的场景[13]. 另一类方法是避免碰撞, 具体方法包括可达性分析[14−16]、不可避免的碰撞状态(ICS)[17−19]和控制不变集(CIS)[6,20,21]. 无人驾驶汽车是否会发生碰撞, 可以通过检查其可达集与相关障碍物可达集的交集来识别. 但是, 可达性分析的缺点是: 随着时间的推移, 不安全区域可能会迅速增长, 导致出于潜在的不安全性的考虑而错失安全轨迹. ICS 是无人驾驶汽车发生碰撞的状态[17,18,22−25], 当汽车处于 ICS 中的状态时, 意味着无论车辆沿着怎样的轨迹行驶都必定发生碰撞. 然而, 在任意驾驶场景中确定 ICS 的计算是昂贵的, 并且大多数情况下仅考虑交通参与者的单个轨迹预测来降低计算量[18]. CIS 与 ICS 互补, 对于其他交通参与者的未来行为而言, CIS 的状态意味着至少存在一条无碰撞轨迹, 所以汽车仍然是安全的. 不过, 由于障碍物的未来运动未知, 在动态环境中获得 CIS 具有挑战性.

继我们先前的工作[26]之后, 本文提出了一种新的自动验证技术, 以确保无人驾驶汽车的转向角安全. 该技术基于深度学习验证工具(DLV)[27]和凸优化. DLV 是一个用于自动验证图像分类神经网络安全性的框架, 通过逐层搜索神经网络中的对抗性反例来实现验证. 然而, DLV 不能直接用于验证无人驾驶汽车转向角的安全性, 这是因为正确的转向角通常是在一定范围内的, 而不是像图像分类决策那样是具体的某个值. 在文献[26]中, 我们运用神经元覆盖率[28]和松弛关系将转向角的判断问题转化为可分类问题. 但是, 这种转向角的判断方法仅从统计意义出发, 从避撞的角度来说过于粗糙和不精确. 因此, 本文运用故障安全轨迹规划中的凸优化技术求解横向轨迹以获得转向角, 采用最宽松的横向约束条件求解出能够保障无人驾驶汽车的安全并且满足避撞需求的临界转向角, 然后结合原始图像的预测转向角, 构造出安全转向角的区间, 对经过扰动操作后的预测转向角进行判断. 只要扰动图像的预测转向角落在安全转向角的区间内, 就认为扰动图像的预测转向角与原始图像的预测转向角属于同一类; 否则, 认为两个图像对应的预测转向角的分类不同. 在处理转

向角的分类问题上, 运用凸优化技术构造的安全转向角区间比 SDLV 中依赖神经元覆盖率和松弛关系的方式更准确和可靠. 将转向角的判断问题转化为可分类问题后, 拓展 DLV 以解决无人驾驶汽车转向角安全性的验证问题.

我们在训练好的 NVIDIA 的端到端无人驾驶[29]架构上评估我们的技术, 该架构已被广泛用作像 Rambo 模型[30]等无人驾驶汽车的感知模块或端到端控制器. 现有的实验结果表明: 一个训练有素的端到端系统可以预测转向角, 且其精度能够接近人类驾驶员[29]. 我们的实验结果表明: 在给定区域内, 如果存在对抗性反例(即不正确的转向决策), 我们的技术可以成功地搜索到. 因此, 我们可以实现安全验证(如果在所有 DNN 层都未发现错误分类的情况下, 网络关于转向决策可以说是稳定或可靠的)或证伪(在这种情况下, 对抗性反例可以用于后续微调网络). 我们的实验中所报告的对抗性反例表明, 该类系统或网络可能不稳定并引发危险的行为. 与我们之前的工作 SDLV 相比, 本文方法找到对抗性反例的成功率更高.

本文第 1 节介绍验证无人驾驶汽车安全性的相关方法和研究现状. 第 2 节介绍本文所需背景知识, 包括无人驾驶汽车的 DNN 和抽象的 DLV 算法. 第 3 节介绍本文所提出的无人驾驶汽车转向角安全性的验证方法. 第 4 节通过实验结果和对比实验验证所提方法的有效性. 最后总结全文.

1 验证无人驾驶安全性的相关工作

如何验证无人驾驶汽车的行为安全性, 是一个备受关注的重要问题[31]. Pei 等人[28]提出了一种用于测试深度学习系统的方法, 他们提出了神经元覆盖率这一新概念以自动识别错误行为, 而无需人工标记. 随后, Tian 等人[32]提出了一个名为 DeepTest 的系统测试工具, 用于自动检测 DNN 驱动的无人驾驶汽车的错误行为. 他们通过对一组种子图像应用不同的变换生成合成测试图像, 从而最大化 DNN 的神经元覆盖率, 并通过生成测试输入, 系统地探索 DNN 逻辑的不同部分. 他们的方法在测试用例中发现了数千个错误行为, 其中一些可能导致致命的碰撞. 类似地, Zhang 等人[33]提出了 DeepRoad 来合成真实的驾驶场景, DeepRoad 是一种用于测试基于 DNN 的无人驾驶系统不一致行为的无监督学习框架. 该方法在 3 个真实的 Udacity 无人驾驶模型上进行了演示, 并检测了数千种不一致的行为. 与 DeepTest 相比, DeepRoad 可以在不使用图像变换规则(如缩放、剪切、旋转)的情况下, 自动合成大量不同的驾驶场景.

为了保证无人驾驶汽车的行为不会造成交通事故, 在线验证方法[34−36]引起了大家的关注. 与基于测试的方法相比, 在线验证的方法在任何情况下都能够执行在线安全分析, 所以绝不会错过对某个场景的验证而导致安全危机情况的发生[13]. Pek 等人[8]提出了一个安全框架以实时验证每条规划轨迹的安全性, 该框架由基于集合的预测、故障安全轨迹生成和在线验证模块组成, 使用形式化方法来处理不确定测量和交通参与者的未来行为以及对车辆自身的干扰等. 当汽车运行期间无法获得新的轨迹时, 通过执行存储的、验证过的故障安全轨迹, 仍能够保证无人驾驶汽车的安全. 之后, Pek 等人[13]又提出了一种在线验证技术, 通过使用故障安全轨迹来确保无人驾驶汽车不会引发碰撞, 只有在经过安全验证的情况下才能执行预期轨迹. Wu 等人[37]提出了一种用于验证无人驾驶汽车行为安全性的在线验证框架, 该框架基于 5 个有关汽车行为安全的考虑: 新的纵向和横向安全距离、变道、超车以及如何面对新的交通参与者. 与之前的验证不同, 该验证框架允许无人驾驶汽车的实际行为与目前流行的严格安全距离暂时不一致.

在离线情况下, 离线测试方法[38]可以保证无人驾驶汽车的安全性, 但这些方法不能适应频繁更新的软件和在线机器学习方法. 在线监测系统[39]使用概率指标[40−42]来确定碰撞或经验性能指标[43]以对测试车辆的安全性进行分级, 但一些经过训练的模型可能会扩展到复杂的实现, 使得它们本身不能被通过[44]. 为了提供基于强制要求的保证, 提出了一些形式化或确定性的方法, 包括可达集[9,34,45]、运行时验证[46]和基于度量的方法[47,48]. 但是, 为特定软件量身定制的一些技术可能不够灵活, 而且不容易与其他软件组件或技术集成.

通过在几种操作模式之间切换来提供保证, 也是一种有效的方式. Idriz 等人[49]提出了一种高速公路驾驶场景下的自动转向与自适应巡航控制集成控制策略. 通过适当创建纵向和横向控制器之间的协同和互联, 可以实现横向稳定性和先进的自适应巡航控制功能, 并和谐共存. 然而, 对于实时计算环境下纵向和横向综合

解决方案的安全性和性能, 以及纵向自适应巡航控制策略和横向控制策略的系统集成, 在很大程度上还缺乏验证. 在这种情况下, Rachman 等人[50]提出了一种集成控制策略的实时验证, 目的是在纵向和横向控制器之间建立起安全的交互.

基于仿真的方法经常被用来检查系统的性能. 刘斌斌等人[51]提出了验证驱动的基于代码自动生成的无人车决策系统开发框架, 该框架利用模型检验技术对无人车决策系统进行环境建模, 验证无人车决策系统的设计过程并发现其中缺陷, 解决安全性问题. 最近, Cho 等人[52]提出了一种基于功能模拟接口和云计算的自动驾驶系统(ADS)验证框架, 该框架节省了仿真成本, 可以高效地验证 ADS. 该框架协同仿真了 4 个模块进行精确的功能验证, 每个模块支持基于对象管理组数据分发服务的分布式通信. 进一步地, 虚实交互的方式被引入以用于检查系统的性能. 朱宇等人[53]利用实验场 LTE-V 和 EUHT 网络建立传输链路, 结合基于 PreScan 的虚拟测试环境, 构建了基于封闭实验场的虚实交互系统; 通过在实验场内对信息交互中的数据包投递率、传输时延和吞吐量等指标进行动静态实车测试, 以验证虚实交互系统信息传输的性能. 此外, 运用开放式的汽车系统架提升性能的方式也引起研究者的关注. 龙翔等人[54]提出了一种功能可拓展、硬件可升级的自动驾驶汽车系统架构. 该架构采用模块化、层次化、封装化的设计思想, 在不改动车辆原有结构的基础上, 设计了开放式的硬件接入结构以及模块化、层次化的系统软件架构, 使得功能拓展和部件升级更为便利, 硬件更新和软件迭代易于实现.

2 基础知识

2.1 无人驾驶汽车的深度神经网络

无人驾驶汽车的关键组成部分是由底层 DNN 控制的感知模块[32]. 无人驾驶汽车的 DNN 从摄像头、光探测和测距传感器(LiDAR)以及红外传感器(IR)等用于测量驾驶环境的传感器接收输入信息. DNN 的每一层逐步抽象输入信息, 并且在最后一层输出当前条件下安全操纵汽车所必需的命令, 如转向角、制动等. 如图 1 所示, 在本文中, 我们主要研究摄像机输入和转向角输出.

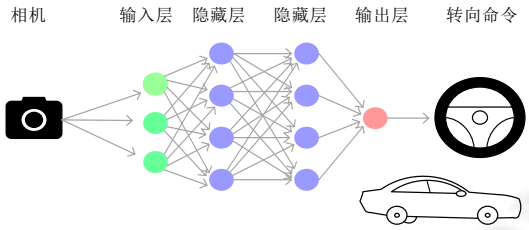

图 1 相机输入和转向角输出的无人驾驶汽车的 DNN

DNN 的每一层都由多个被称为神经元的独立计算单元组成. 神经网络中的神经元排列在不相交的层, 每一层中的每个神经元通过边与下一层连接, 但同一层中的神经元之间没有连接. 每条边都有一个对应的权重, 权重是在 DNN 的训练过程中根据训练数据计算出来的. 现有的 DNN 大多数都使用反向传播[55]的梯度下降法进行训练. 一旦训练完成, DNN 就可以用于预测, 而不需要对权重进行任何进一步的改变. 本文主要研究摄像机的输入和转向角的输出, 一个训练有素的无人驾驶汽车 DNN 可以根据输入图像预测转向角.

网络中每个神经元的值是通过计算前一层神经元的值和对应边的权重参数的线性组合, 然后应用非线性激活函数[56]来确定的. 在这里, 我们主要应用修正线性单元(ReLU)激活函数[57]. 当对一个神经元应用 ReLU 激活函数时, 该神经元的值被计算为前一层神经元的线性组合与 0 之间的最大值.

2.2 深度学习验证DLV

DLV[27]是一个用于验证前馈深度神经网络分类决策安全性的自动验证工具, 该技术基于对数据点周围区域的系统探索, 以搜索给定操作类型的对抗性反例, 并将分析传播到更深的层. 对于更全面的描述, 读者可以参考文献[27]. 本文算法扩展了 DLV, 因此我们从 DLV 验证算法中图像分类神经网络分类决策的安全性的概

念开始逐一加以介绍.

对一个 DNN 我们记为 N, 网络的每一层用 L_k 表示, $k\in\{0,\ldots,n\}$, L_0 是输入层, L_n 是输出层. 点 x 在第 k 层的激活表示为 $\alpha_{x,k}$. 输入点的向量空间用 $\mathbb{R}^{n_k}$ 表示. 网络的每一层 L_k 都与一个 n_k 维向量空间 $D_{L_k}\subseteq\mathbb{R}^{n_k}$ 相关联, 其中, 每一维对应一个神经元. 分类决定由图像分类神经网络的输出层确定, 这里用 $\alpha_{x,n}$ 表示关于输入 x 的分类. 因此, 如果有 $\alpha_{y,n}=\alpha_{x,n}$, 则表示输入 x 和 y 有相同的分类.

DLV 假设在给定点 x 周围存在一个或无限的区域, 该区域内的点与 x 的分类相同, 其中, 该区域内的所有点必须具有相同的类别. 该区域由用户指定, 可以由一个小直径约束, 也可以是一类具有相同特征点的集合. 例如, 一个区域 $\eta_k=\{z\in D_{L_k}\mid\|z-x\|\leqslant d\}$ 是通过其直径 d 关于某些用户指定的范数来确定的, 这种方式直观地衡量出区域内的点与点 x 的接近程度.

假设 1. 对于点 x 在 L_k 层的每个激活 $\alpha_{x,k}$, 其与区域 $\eta_k(\alpha_{x,k})$ 内包含的激活之间的距离非常小, 以至于人类观察者将 x 和区域 $\eta_k(\alpha_{x,k})$ 内的点归为同一类.

安全性是针对单个分类决策而定义的. 网络 N 在点 x 的安全性, 意味着分类决策在 x 处对区域 $\eta_k(\alpha_{x,k})$ 内的扰动是稳健的. 如果在区域 η 内存在一个点 y, 其与 x 的距离很小, 使得 $\alpha_{x,n}\neq\alpha_{y,n}$, 则 y 就是一个对抗性反例.

定义 1(一般安全性定义). 设 $\eta_k(\alpha_{x,k})$ 是神经网络 N 在 L_k 层的一个区域, 并且有 $\alpha_{x,k}\in\eta_k(\alpha_{x,k})$. 如果对 $\eta_k(\alpha_{x,k})$ 内的所有激活 $\alpha_{y,k}$ 有 $\alpha_{y,n}=\alpha_{x,n}$, 则认为 N 关于输入 x 和区域 $\eta_k(\alpha_{x,k})$ 是安全的, 记为 $N,\eta_k\vDash x$.

安全性由操作集 Δ 和给定图像的区域参数化. 一个操作是对图像进行特定修改的运算符, 在该操作下, 区域 η 内的点所代表的图像的分类决策应保持不变. 这类操作可以模拟图像扰动, 例如相机不精确、相机角度的变化或特征的替换. 设 Δ_k 为层 L_k 中所有可能操作的集合. 如果在 x 上应用操作不会导致分类改变, 则认为 DLV 对于输入 x 关于区域 η 和操作集 Δ 的网络决策是安全的.

定义 2(关于操作的安全性). 给定一个神经网络 N、一个输入 x 和一个操作集合 Δ_k, 如果从 $\alpha_{x,k}$ 开始生成一棵完备树, 树的每个节点与 $\alpha_{x,k}$ 具有相同的分类, 并且相邻节点生成的有限个超矩形可以覆盖邻域 $\eta_k(\alpha_{x,k})$, 那么就说 N 对于输入 x 关于区域 η_k 和操作 Δ_k 是安全的, 记为 $N,\eta_k,\Delta_k\vDash x$.

对于一个给定的神经网络 N 和输入 x, 高维区域 η 被离散化以实现对对抗性误分类的有限穷举搜索. 具体地, 对来自某层 L_k 的 $\alpha_{x,k}$ 实施操作集 Δ_k 中的一系列连续操作, 包含 $\alpha_{x,k}$ 的区域 $\eta_k(\alpha_{x,k})$ 被划分为有限个可以覆盖区域 η_k 的小区域. 对于每个小区域, 如果在所有小区域内都没有找到对抗性反例, 则在第 L_{k+1} 层执行细化操作. 细化操作的定义见定义 4. DLV 可以从任何层开始验证, 逐层传播分析, 将区域和操作映射到更深层. 在操作极小化的附加假设下, 传播分析是完整的, 验证的细化框架如图 2 所示. 箭头表示安全概念之间的推导关系, 并在需要约束的情况下标明所需条件. 细化的目标是找到一系列关系式, 推导出 $N,\eta_k,\Delta_k\vDash x$.

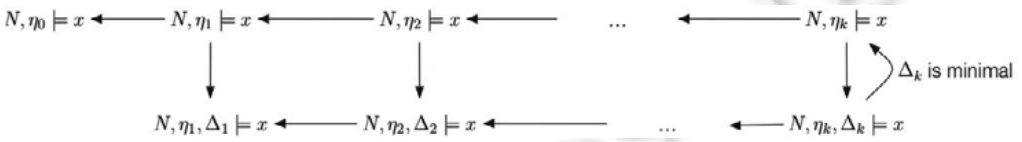

图 2 DLV 的细化框架

分析的层层传播涉及隐藏层中的细化操作. 为了推动分析, 除了设置激活函数 $\phi_k: D_{L_{k-1}}\to D_{L_k}$ 外, 还设置了反向映射 $\psi_k: D_{L_k}\to D_{L_{k-1}}$, 用来表示 L_k 层的被操作的激活如何影响 L_{k-1} 层的激活. 为了将点 x 处的区域 $\eta_k(\alpha_{x,k})$ 的安全性传播到更深的层, 假设函数 η_k 的存在将激活映射到区域, 并对函数 ϕ_k 和函数 ψ_k 施加如下约束.

定义 3. 函数 $\{\eta_0,\eta_1,\ldots,\eta_n\}$ 和 $\{\psi_1,\ldots,\psi_n\}$ 映射激活到区域满足以下条件.

(1) 对于 $k=0,\ldots,n$, $\eta_k(\alpha_{x,k})\subseteq D_{L_k}$;

(2) 对于 $k=0,\ldots,n$, $\alpha_{x,k}\in\eta_k(\alpha_{x,k})$;

(3) 对于所有 $k=0,\ldots,n$, $\eta_{k-1}(\alpha_{x,k-1})\in\psi_{\eta_k(\alpha_{x,k})}$.

上述定义的前两个条件规定每个函数 η_k 在激活 $\alpha_{x,k}$ 附近分配一个区域, 最后一个条件通过 ψ_k 将区域 η_k 从 L_k 层映射到 L_{k-1} 层的区域, 并且该区域应覆盖区域 η_{k-1}, 目的是基于 η_k 和神经网络计算函数 $\eta_{k+1},\ldots,\eta_n$.

Huang 等人[27]定义了层之间细化操作的概念: 如果 L_k 层中存在实现 L_{k-1} 层操作的操作序列, 那么就说 L_k 层中的操作是可细化的.

定义 4. 如果在 L_k 层存在激活和有效操作可以表达出 $\delta_{k-1}(\alpha_{y,k-1})$, 那么就说这个操作 $\delta_{k-1}(\alpha_{y,k-1})$ 在第 L_k 层是可细化的. 其中, 有效操作 Δ_k 意味着相应的激活 $\alpha_{y,k}$ 是多面体的一个内点, 该多面体包含 $\alpha_{y,k}$ 的所有超矩形. 给定一个神经网络 N 和一个输入 x, 如果对于所有 $\alpha_{y,k-1}\in\eta_{k-1}(\alpha_{z,k-1})$, 其所有有效操作 $\delta_{k-1}(\alpha_{y,k-1})$ 在 L_k 层都是可细化的, 则认为操作 Δ_k 是对 η_{k-1}、Δ_{k-1} 和 η_k 的一个细化.

Huang 等人[27]总结了下面基于搜索的递归验证过程, 该过程由给定点周围的区域 η_k 和一系列操作 Δ_k 参数化. 验证算法可以从任意层开始, 并将分析传播到更深的层, 直至搜索到对抗性反例终止, 并将对抗性反例映射回输入层. 这里, 用于确定区域的向量范数可以由用户指定并随层变化.

Huang 等人[27]将安全性验证简化为搜索对抗性反例, 对抗性反例的搜索过程如算法 1 所示. 如果在所有 DNN 层都未发现错误的分类决定, 那么网络可以说是稳定或可靠的; 如果发现对抗性反例, 那么这些对抗性反例可以用于后续改善网络.

算法 1. 给定一个神经网络 N 和一个输入 x, 从某一层 $l\geqslant 0$ 开始, 递归地执行下面的步骤. 设 $k\geqslant l$ 是正在考虑的当前层.

(1) 确定一个区域 η_k, 使得当 $k>l$ 时, η_k 和 η_{k-1} 满足定义 3;

(2) 确定一个操作集合 Δ_k, 使得当 $k>l$ 时, Δ_k 是由 η_{k-1}、Δ_{k-1} 和 η_k 根据定义 4 执行的层细化;

(3) 验证 $N,\eta_k,\Delta_k\vDash x$ 是否成立:

(a) 如果 $N,\eta_k,\Delta_k\vDash x$, 那么:

i. 报告 N 对于 x 关于 $\eta_k(\alpha_{x,k})$ 和 Δ_k 是安全的; 并且

ii. 继续第 k+1 层验证;

(b) 如果 $N,\eta_k,\Delta_k\nvDash x$, 则报告一个对抗性反例.

3 基于凸优化和 DLV 验证无人驾驶汽车转向角的安全性

第 2.2 节中描述的 DLV 算法是一种用于验证图像分类神经网络的分类决策安全性的高效自动验证技术. 虽然无人驾驶系统也是基于 DNN 的, 但 DNN 驱动的无人驾驶汽车无法直接运用 DLV 算法进行验证.

图像分类决策的判断是唯一的. 例如: 对于图 3 中的 3 幅扰动后的汽车图像, 如果预测的输出不是汽车, 则图像分类决策一定是错误的. 但是, 基于 DNN 的无人驾驶汽车的预测转向角的决策并不是唯一的. 由于每个人的驾驶习惯不同, 在同一驾驶场景下, 所做出的驾驶决策也不同. 如图 4 所示, DeepTest[32]曾报告一些关于转向角的假阳性的例子. 每组图像中: 左边的图像是原始图像, 转向角(蓝色箭头)是手动标签; 右边的图像是扰动后的合成图像, 预测转向角(红色箭头)被 DeepTest 报告是不正确的, 但这两组例子的转向角在人类驾驶员的判断下正确的.

(a) 汽车被错分为青蛙 (b) 汽车被错分为猫 (c) 汽车被错分为飞机

图 3 原始图像及其扰动图像

(a) 平移(10,10) (b) 平移(50,50)

图 4 两组假阳性例子

DLV 在逐层搜索对抗性反例的过程中, 需要比较原始图像的预测转向角和扰动图像的预测转向角是否具有相同的分类. 然而, 由于安全的转向角并不唯一, 所以即使扰动图像的预测转向角与原始预测转向角不相等, 扰动图像的预测转向角也可能是安全的. 因此, DLV 不能直接用于验证无人驾驶汽车转向角的安全性. 为了验证无人驾驶汽车转向角的安全性, 必须解决无人驾驶模型的预测转向角的判断问题.

然而, 转向角的判断是一个复杂且困难的问题. 基于 DNN 的无人驾驶汽车经常表现出错误或意外的行为, 这些行为可能带来致命碰撞等危险后果. 关于无人驾驶汽车的交通事故已有相关报道[1−3], 其中一起甚至引发致命的事故. 这显然为像无人驾驶系统这样对安全性要求等级较高的应用程序带来了潜在的安全问题, 因此需要自动验证技术来验证其预测行为的正确性. 由于安全的转向角是在一个连续的范围内, 我们可以利用一些条件来帮助判断扰动后图像的预测转向角是否安全.

在我们之前的研究[26]中, 运用神经元覆盖率[28]和松弛关系将转向角判断问题转化为可分类问题. 神经元覆盖率是给定的输入图像所激活的神经元数与 DNN 中神经元总数的比值. 正的统计相关性表明: 随着神经元覆盖率的增加, 转向角增加; 反之亦然[32]. 利用神经元覆盖率与转向角之间的统计相关性, 引入神经元覆盖率作为相同转向角的参考. 松弛关系是基于为无人驾驶汽车定义的变质关系, 即: 当输入图像发生微小变化时, 转向角不应该发生明显变化. 如果扰动后的图像的神经元覆盖率与原始图像的神经元覆盖率相等, 并且扰动后的图像的预测转向角满足两种松弛关系, 则认为扰动后的图像预测转向角与原始图像的预测转向角是同一类. 只要不满足这 3 个条件中的任意一个, 则认为两个图像的预测转向角分类不同.

但是, 这种转向角的判断方法仅从统计的角度出发, 从避免碰撞的角度考虑显得过于粗糙和不精确. 因此, 本文运用故障安全轨迹规划中所应用的凸优化技术求解包含转向角的横向轨迹以获得转向角时, 采用最宽松的横向约束条件, 求解出能够保障无人驾驶汽车的安全并且满足避撞需求的临界转向角, 将该转向角与无人驾驶系统对原始图像的预测转向角相结合, 构造出关于安全转向角的区间, 再对扰动后图像的预测转向角进行判断. 只要扰动后图像的预测转向角落在安全转向角区间内, 则认为扰动后的图像的预测转向角与原始图像的预测转向角属于同一类; 否则, 认为两个图像对应的转向角属于不同的类. 将转向角的判断问题转化为可分类问题后, 拓展了 DLV 解决无人驾驶汽车的转向角安全性的验证问题.

3.1 故障安全轨迹

保证无人驾驶汽车的行驶安全是一项具有挑战性的任务, 因为驾驶环境中的其他交通参与者随时可能出现偏离预测的行为. 一种解决方案是, 确保车辆在任何时候都能执行无碰撞规避轨迹. 在文献[58]中, 作者提出了一种运用凸优化技术在任意交通场景下生成故障安全轨迹的方法. 通过在规划器中集成安全验证, 这种规划能够实时生成故障安全轨迹, 保证无人驾驶汽车的行为安全. 无人驾驶的运动规划被分离成纵向和横向两部分, 在规划路径时, 优化抖动参数还可以提高驾驶舒适度. 在本文中, 我们利用该故障安全轨迹生成方法计算出安全的转向角, 以帮助判断扰动操作后的图像的预测转向角. 因此, 接下来, 我们简要介绍其主要技术, 对这一方法有兴趣的读者可以参考文献[58], 以进一步了解方法的细节.

对于一辆无人驾驶汽车, 引入配置空间 $X\subset\mathbb{R}^n$ 作为所有可能的状态 x 的集合, 输入集合 $U\subset\mathbb{R}^m$ 作为允许控制输入 u 的集合, 集合 $Z\subset\mathbb{R}^q$ 作为作用于无人驾驶车辆的可能扰动 z, 无人驾驶汽车的运动由微分方程控制:

$$\dot{x}(t)=f(x(t),u(t),z(t)) \tag{1}$$

我们用 $x^{(i)}$, $i\in\mathbb{N}_0$ 来描述状态变量 x 的第 i 个分量. 不失一般性, 假设初始时间为 t_0=0, 并且用符号 $x([t_0,t_h])$ 来表示时间间隔$[t_0,t_h]$的状态轨迹, 这里 $t_0<t_h$.

考虑一个基于车道的故障安全规划环境, 它被建模为欧几里德空间$\mathbb{R}^k$ 的一个子集. 从配置空间引入一个关系 occ, 这个关系将 X 映射到世界坐标中基于车道的环境.

定义 5(状态占有率). 算子 $occ(x)$将状态向量 x 映射到系统所占据的环境中的点集 $occ(x):X\to P(\mathbb{R}^k)$, 其中, $P(\mathbb{R}^k)$是$\mathbb{R}^k$ 的幂集. 给定一个集合 X, 定义 $occ(X):=\{occ(x)|x\in X\}$.

在文献[58]中, 有一个预测的假设, 它使用诸如文献[59]中介绍的工具来考虑障碍物未来任何可行的运

动, 以保证规划运动的安全性. 在给定的时间点上, 将可能被占用的点的集合表示为一个占用集.

定义 6(占用集). 占用集 $O(t)\subseteq\mathcal{N}^k$ 描述了在时间 t 可能被障碍物占用的环境中的点集. 对于时间间隔$[t_1,t_2]$, $t_1<t_2$, 定义 $O([t_1,t_2])=\bigcup_{t_1\leqslant t\leqslant t_2}O(t)$.

如图 5 所示, 曲线坐标系用于运动规划并与给定的参考路径Γ对齐. 这意味着所有欧几里德位置都将根据沿Γ的弧长 s 和与Γ正交的偏差 d 来表示.

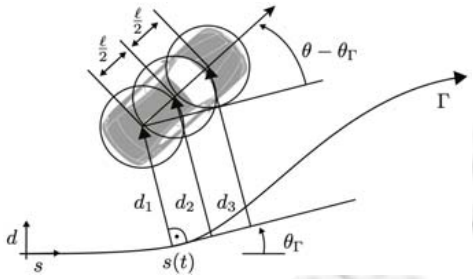

图 5 曲线坐标系下的运动模型[58]

为了降低计算成本, 使用了运动规划问题[60]的凸近似. 这是通过将运动分离为纵向和横向组成部分而实现的, 同时保证了计算出的运动规划的可驾驶性.

将无人驾驶汽车的纵向运动的状态建模为 $x_{lon}=(s,v,a,j)$, 其中, s 是纵向位置, v 是速度, a 是加速度, j 是沿给定参考路径Γ的抖动参数. 运用输入 $u(t)=\dddot{a}(t)$, 车辆的纵向运动由下面的线性时不变系统[58]表示:

$$\frac{d^4}{dt^4}s(t)=u(t) \tag{2}$$

为了确保轨迹在运动学上可行, 应用下面的时不变状态约束:

$$a_{\min}\leqslant a(t)\leqslant a_{\max} \tag{3}$$

$$v_{\min}\leqslant v(t)\leqslant v_{\max} \tag{4}$$

当无人驾驶所在车道内前后都有障碍物时, 其纵向位置由沿着参考路径Γ的最大和最小碰撞约束来限制.

$$s_{\min}(t)\leqslant s(t)\leqslant s_{\max}(t) \tag{5}$$

二次成本函数 J_{lon} 分别通过权重 $w_a\in\mathbb{N}_+$和 $w_j\in\mathbb{N}_+$惩罚高加速度和颠簸程度来生成舒适的轨迹[58], 并定义为

$$J_{lon}(x_{lon}(t))=\int_0^{t_h}w_a x_{lon}^{(2)}(t)^2+w_j x_{lon}^{(3)}(t)^2\,\mathrm{d}t \tag{6}$$

我们用 d 表示到参考路径Γ的横向距离, θ表示方向, κ是曲率, $\dot{\kappa}$ 是曲率变化. 汽车横向运动的状态建模为 $x_{lat}=(d,\theta,\kappa,\dot{\kappa})^{\mathrm{T}}$. 在图 5 中, 参考路径方向表示为$\theta_\Gamma$. 由于车辆应沿着预定义的参考路径Γ移动, 假设当前方向θ和参考路径方向θ_Γ之间的差异$\Delta=\theta-\theta_\Gamma$小到可以忽略不计. 在这个假设下, 三角函数可以近似为 $\sin(\Delta)\approx\Delta$和 $\cos(\Delta)\approx1$. 在不引入新的状态变量的情况下, 有效地集成避撞, 参考路径Γ的方向θ_Γ被建模为扰动 $z(t)=\theta_\Gamma(s(t))$. 运用输入 $u(t)=\ddot{\kappa}$, 汽车的纵向运动由下面的线性时不变系统[58]表示:

$$\dot{x}_{lat}=\begin{bmatrix}0&v(t)&0&0\\0&0&v(t)&0\\0&0&0&1\\0&0&0&0\end{bmatrix}x_{lat}(t)+\begin{bmatrix}0\\0\\0\\1\end{bmatrix}u(t)+\begin{bmatrix}-v(t)\\0\\0\\0\end{bmatrix}z(t) \tag{7}$$

为了检查碰撞, 引入 3 个具有相等半径 r 的圆来近似车辆的形状. 不失一般性, 选择第 1 个和第 3 个圆的中心分别与后轴和前轴重合, 两圆中心点之间的距离为 l. 第 2 个圆的中心与另外两个圆的中心的距离相等. 因此, 从第 $i\in1,2,3$ 个圆的中心到参考路径Γ的横向距离 d_i 为

$$d_i=d+\frac{i-1}{2}l\sin(\theta-\theta_\Gamma)\approx d+\frac{i-1}{2}l(\theta-\theta_\Gamma) \tag{8}$$

运用上述表达式, 将横向运动的约束值定义为 $x_{lat,constr}=(d_1,d_2,d_2,\kappa,\dot{\kappa})^{\mathrm{T}}$, 计算方式为

$$x_{constr}=\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & \frac{1}{2}l & 0 & 0\\ 1 & l & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}x_{lat}(t)+\begin{bmatrix}0\\ -\frac{1}{2}l\\ -l\\ 0\\ 0\end{bmatrix}z(t) \tag{9}$$

通过计算沿参考路径Γ的每个圆 $i\in1,2,3$ 的最小和最大横向位移来避免碰撞. 结合转向系统施加的物理约束, 应用下面的时变约束[58]来获得可行驶轨迹:

$$\dot{x}_{\min}(t)=\begin{bmatrix}d_{1,\min}(t)\\ d_{1,\min}(t)\\ d_{1,\min}(t)\\ \kappa_{\min}(t)\\ \dot{\kappa}_{\min}(t)\end{bmatrix}\leqslant x_{constr(t)}\leqslant\begin{bmatrix}d_{1,\max}(t)\\ d_{1,\max}(t)\\ d_{1,\max}(t)\\ \kappa_{\max}(t)\\ \dot{\kappa}_{\max}(t)\end{bmatrix}=x_{\max}(t) \tag{10}$$

其中, $d_{i,\min}$ 和 $d_{i,\max}$ 分别是圆 $i\in1,2,3$ 允许的最小和最大横向偏差, $\kappa_{\min}$ 和 $\kappa_{\max}$ 分别是允许的最小和最大曲率, $\dot{\kappa}_{\min}$ 和 $\dot{\kappa}_{\max}$ 分别是允许的最小和最大曲率变化, 从转向系统的技术规范中获得.

带有权重 $w_d,w_\theta,w_\kappa,w_{\dot{\kappa}}\in\mathbb{R}_+$ 的二次成本函数 J_{lat} 最小化横向距离和沿着参考路径Γ的方向偏差, 并惩罚高曲率以获得平滑的横向轨迹[58]:

$$J_{lat}(x_{lat}(t))=\int_0^{t_h}w_d x_{lat}^{(0)}(t)^2+w_\theta(x_{lat}^{(1)}(t)-\theta_\Gamma(t))^2+w_\kappa x_{lat}^{(2)}(t)^2+w_{\dot{\kappa}}x_{lat}^{(3)}(t)^2\,\mathrm{d}t \tag{11}$$

图 6 描述了运用解耦将运动规划问题分离为纵向和横向运动问题来计算故障安全轨迹的过程. 这里假设故障安全轨迹的初始状态 x_0 和参考路径Γ是已知的, 对Γ的计算有兴趣的读者可以参考文献[61].

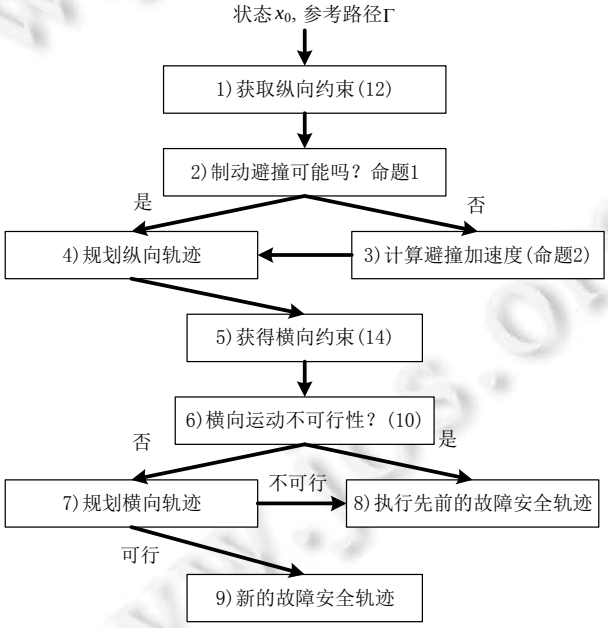

图 6 故障安全轨迹计算步骤[58]

在图 6 的第 1 步中, 首先计算纵向碰撞约束. 在曲线坐标系中沿着Γ, 转化每个与安全相关的障碍物 $b\in\mathcal{B}$ 的预测占用集$\mathcal{O}_b(t)$, 得到$\mathcal{O}_{b,cls}(t)$. 基于车辆的纵向位置 s_{ego}, 位置约束的计算方式[58]如下:

$$s_{\max}(t)=\inf\{s>s_{ego}|(s,d)^{\mathrm{T}}\in\mathcal{O}_{b,cls}(t),b\in\mathcal{B}\} \tag{12}$$

最小位置约束 $s(t)\geqslant s_{\min}(t)$的计算方法[58]类似:

$$s_{\min}(t)=\sup\{s>s_{ego}|(s,d)^{\mathrm{T}}\in\mathcal{O}_{b,cls}(t),b\in\mathcal{B}\} \tag{13}$$

这里, 仅当无人驾驶汽车进行变道时才使用 $s_{\min}(t)$. 图 7 解释了纵向约束的计算问题.

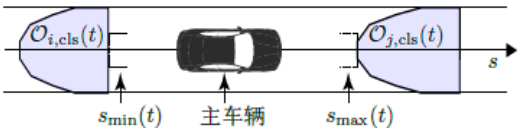

图 7 对于同车道的给定占有集计算纵向约束[58]

在第 2 步中, 检查仅采取制动的策略是否足以避免碰撞. 由于占用集包含障碍物随时间变化的信息, 所以运用公式(12)检查避撞.

命题 1(通过制动避免碰撞). 一次与障碍物的碰撞表示为一个碰撞约束 $s(t)\leqslant s_{\max}(t)$, $t\in[0,t_h]$, 对于初始位置 s_0、速度 v_0 和反应时间 δ_{brake}, 以$|a_{\max}|$执行紧急制动的车辆, 如果有下面的条件成立, 则碰撞可以避免:

$$\forall t\in[0,t_h]: s_0+v_0(\tau+\delta_{brake})-\frac{1}{2}|a_{\max}|\tau^2\leqslant s_{\max}(t),\ \tau:=\min\left(t,\frac{v_0}{|a_{\max}|}\right).$$

如果主车辆可以通过制动避免潜在的碰撞, 则使用前面描述的轨迹规划计算纵向轨迹; 否则, 可以通过转向另一条车道或在不离开当前车道的情况下避开障碍物来避免碰撞.

对于这些情况, 首先引入保证碰撞时间.

定义 7(保证碰撞时间). 关于车辆初始纵向位置 s_0 和速度 v_0 以及允许的最大纵向位置 $s_{\max}(t)$, $t\in[0,t_h]$的保证碰撞时间(GTTC)定义为

$$GTTC:=\arg\min_{t\in[0,t_h]}|(s_0+v_0t)-s_{\max}t|.$$

由于车辆运动模型中的纵向和横向动力学解耦, 必须确保在整个操作过程中规避所需的最大横向加速度 a_{eva} 是可行的. 最坏情况是, 规避操作不再允许制动. 假设没有减速, 并且完全到达相邻车道的横向距离为 $d_{eva}>0$, 则将规避操作的持续时间称为 GTTC.

命题 2(规避加速度). 对于初始横向速度 $v_{lat}\geqslant 0$, 横向距离 d_{eva}, 持续时间 $GTTC$, 转向反应时间 $\delta_{steer}<GTTC$, 一个规避操作所需的横向加速度 a_{eva} 为

$$a_{eva}=\frac{2(d_{eva}-v_{lat}(GTTC-\delta_{steer}))}{(GTTC-\delta_{steer})^2}.$$

在图 5 的第 5 步中, 计算了对车辆横向运动的约束. 因此, 关于先前规划的纵向运动预测车辆沿Γ的位置. 基于前面用 3 个等圆近似车辆的形状, 在不与障碍物发生碰撞的约束下, 计算每个圆允许的最大横向偏移量.

设 $circ_i(d,t)$表示圆 $i\in\{1,2,3\}$从无人驾驶车辆在时间 t 的位置沿法线方向移动了 d 的占有率. 最大横向偏移约束[58]为

$$d_{i,\max}(t)=\sup\left\{d\geqslant 0\middle|circ_i(d,t)\cap\left(\bigcup_{b\in\mathcal{B}}\mathcal{O}_b(t)\right)=\varnothing\right\} \tag{14}$$

类似地, 获得最小横向偏移约束[58]$d_{i,\min}(t)$:

$$d_{i,\min}(t)=\sup\left\{d\leqslant 0\middle|circ_i(d,t)\cap\left(\bigcup_{b\in\mathcal{B}}\mathcal{O}_b(t)\right)=\varnothing\right\} \tag{15}$$

图 8 解释了横向约束的计算问题.

在第 6 步中, 检查是否存在 $t\in[0,t_h]$, 使得 $d_{\min}(t)>d_{\max}(t)$. 这意味着不再存在可行的解决方案, 因为公式(10)已被违反. 如果横向规划问题变得不可行, 直接切换到之前计算的故障安全轨迹, 该轨迹仍然有效. 然而, 如果规避操作选项是可行的, 则规划无人驾驶车辆的横向运动并获得新的有效故障安全轨迹. 与现有的验证方法相比, 如果验证失败, 现有方法则不会返回可替代的无碰撞轨迹, 这一方法通过使用过近似的占用集作

为约束, 将验证合并到规划器中. 如果所使用的求解器在数值上是稳定的, 则可以保证生成的故障安全轨迹对于障碍物的任何物理上可行的未来运动都是安全的.

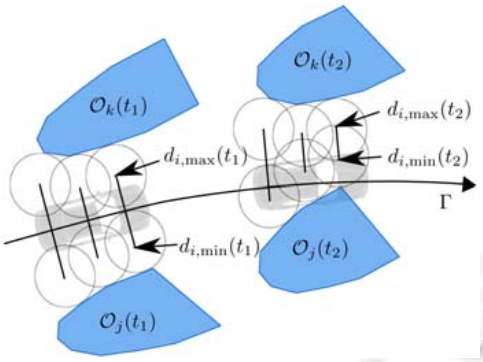

图 8 计算关于时间间隔 t_1 和 t_2 的横向约束 $d_{i,\min}$ 和 $d_{i,\min}$[58]

由于本文主要研究转向角的安全性, 所以将图 6 中的故障安全轨迹简化为: 首先, 运用基于集合的预测工具 SPOT[59]计算出每个障碍物的占有率; 然后, 获得纵向位置约束(12)和横向位置约束(14); 最后, 运用解决凸优化问题的工具 CVXPY[62]计算二次函数 J_{lon} 和 J_{lat}, 获得车辆的纵向轨迹和横向轨迹. 通过求解横向轨迹获得能够避撞的转向角, 用于构造安全的转向角区间.

3.2 验 证

在本节中, 利用第 3.1 节描述的故障安全轨迹规划中运用凸优化所计算出的避免碰撞转向角, 将转向角判断问题转化为可分类问题. 在 DLV 的理论分析中, 假设 DNN 关于输入是局部鲁棒的, 关于原始图像的预测输出是正确的, 那么扰动操作后的图像的预测输出应与原始图像的预测输出是同一类, 如果搜索到对抗性反例, 则证明该 DNN 是不安全的. 关于一个输入图像 x, 如果在搜索过程中, x 与Δ上所有操作过的图像 y 具有相同的分类, 则认为基于 DNN 的无人驾驶系统 N 关于 x 是安全的. 接下来, 我们将 DLV 理论扩展为验证无人驾驶系统的转向角安全性的理论. 在引入算法之前, 先引入下面的定义.

定义 8. 对于输入图像 x 和扰动后的图像 y, 将根据简化的图 6 中的步骤计算出的能够保障无人驾驶汽车的安全并且满足避撞需求的临界转向角和原始输入图像 x 的预测转向角中的最大值和最小值分别记为$\alpha_{x,\max}$和$\alpha_{x,\min}$. 如果扰动后的预测转向角$\alpha_{y,n}$位于转向角的安全区间$[\alpha_{x,\min},\alpha_{x,\max}]$内, 即$\alpha_{y,n}\in[\alpha_{x,\min},\alpha_{x,\max}]$, 则说 x 与 y 具有相同的转向角分类.

将无人驾驶的转向角预测问题转化为可分类问题后, 将 DLV 应用于无人驾驶汽车转向角的安全性验证. 我们的算法分为两部分: 第 1 部分, 利用故障安全轨迹规划中的凸优化方法获得避碰转向角; 第 2 部分, 将计算出的避碰转向角和原始图像的预测转向角构建安全的转向角区间作为判断转向角分类的条件, 并拓展 DLV 理论对无人驾驶系统进行转向角的安全性验证.

我们的工作主要研究关于无人驾驶汽车的神经网络预测转向角输出的局部鲁棒性验证. 在验证过程中, 对于任何给定的输入, 训练后的神经网络都可以得到一个转向角输出. 然后, 算法 2 从神经网络的某一层开始扰动操作, 判断操作后的预测转向角是否与原始预测转向角一致. 在所有层均未找到对抗性反例的情况下, 该神经网络可以实现转向角的安全验证; 如果发现对抗性反例, 那么神经网络对关于这个输入是不安全的.

算法 2. 给定一个神经网络 N 和一个输入 x, 从某一层 $l\geqslant 0$ 开始, 递归地执行下面的步骤. 设 $k\geqslant l$ 是正在考虑的当前层.

(1) 根据定义 8 确定避撞的转向角与原始预测转向角形成的安全转向角的区间;

(2) 确定一个区域η_k, 使得当 $k>l$ 时, η_k 和 η_{k-1} 满足定义 3;

(3) 确定一个操作集合Δ_k, 使得当 $k>l$ 时, Δ_k 是由 η_{k-1}、Δ_{k-1} 和 η_k 根据定义 4 执行的层细化;

(4) 对于所有$\delta\in\Delta_k$, 如果输入图像x和操作后的图像$\delta(x)$满足定义8, 则确定x和$\delta(x)$具有相同的分类. 这一步将无人驾驶汽车预测转向角的判断问题转化为像图像分类神经网络的分类决定一样的可分类问题;

(5) 验证$N,\eta_k,\Delta_k\vDash x$是否成立:

(a) 如果$N,\eta_k,\Delta_k\vDash x$, 那么:

i. 报告N对于x关于$\eta_k(\alpha_{x,k})$和Δ_k是安全的; 并且

ii. 继续第k+1层验证;

(b) 如果$N,\eta_k,\Delta_k\nvDash x$, 则报告一个对抗性反例.

4 实验分析

4.1 实验数据

我们以NVIDIA的端到端无人驾驶系统[29]为例, 对本文提出的验证算法进行评估; 同时, 这个系统也是无人驾驶汽车的关键组成部分. 无人驾驶汽车的端到端系统意味着从传感器获取的输入直接决定汽车的行为, 比如油门、刹车和方向等. 本文主要研究图像的输入和转向角的输出.

端到端无人驾驶系统的网络结构如图9所示, 该网络结构由9层组成, 包括1个归一化层、5个卷积层和3个全连接层. 全连接层的神经元与上一层的所有神经元连接, 不同神经元之间的多个连接权重不同; 而卷积层的神经元只与下一层的部分神经元连接, 不同神经元之间的多个连接共享相同的权值. 这里, 前3个卷积层中使用的跨步卷积步幅为2×2, 内核为5×5; 在后两个卷积层中使用非跨步卷积, 卷积核为3×3. 最后, 在输出层得到预测控制值, 即转向角命令.

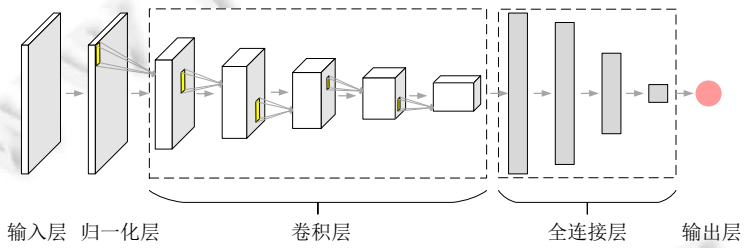

图9 NVIDIA的端到端无人驾驶系统的网络架构

4.2 实验结果

我们在NVIDIA的端到端无人驾驶系统[29]上实现了本文所提出的验证转向角安全性的方法. 该系统已在驾驶数据集[63]上进行了超过6个小时的训练, 训练后的网络拥有超过100万个实值参数. 输入网络的图像大小为135×240, 有3个通道. 对每幅图像进行预处理, 使得每个像素位于区间[0,1]之间.

类似于SDLV和DLV, 本文的验证算法也是用Python实现的. 我们使用的SMT求解器是Z3[64], 它有Python api. NVIDIA的端到端无人驾驶系统的神经网络由神经网络库Keras[65]构建, 并以深度学习包Theano[66]为后端. 实验在3.2 GHz Intel Core i7 CPU并且内存为32 GB的台式计算机上进行.

在文献[26]中, 验证无人驾驶汽车转向角安全性的方法SDLV在搜索方式上有单路径搜索和多路径搜索两种: 如果按照预先指定的顺序对根据特征划分的区域内的点进行检查, 这种方式称为单路径搜索; 如果通过穷尽搜索所有可能的顺序对根据特征划分的区域内的点进行检查, 这种方式称为多路径搜索. 因此, 本文也运用单路径搜索和多路径搜索来验证端到端无人驾驶系统的转向角安全性.

给定一个输入图像x, 我们的算法从k=1层开始执行验证, 每层最大扰动操作维数设置为150维(L_1层共有1 824维). 关于给定的一个无人驾驶系统的神经网络, 对于输入图像x, 假设预测输出是正确的转向角. 对于邻域$\eta_k(\alpha_{x,k})$, 每个选定维度的激活值在区间[–1,1]内, 并且在执行扰动操作时允许更改. 集合Δ_k包含了可以

通过将每个维度的值递增或递减 1 来改变 150 维度的子集的激活值的操作. 逐层检查扰动操作后的预测转向角是否位于凸优化所计算出的避撞转向角与原始预测转向角构建的安全转向角的区间内: 如果落在安全转向角的区间内, 则认为扰动后的预测转向角与原始预测转向角是同一分类, 即为正确的转向角; 若不在该区间内, 则认为扰动后的预测转向角是不正确的转向角.

状态 x_0 和参考路径Γ由时间步长δ_t=0.25 s 和训练数据集中与所选输入图像 x 相邻位置的数据来确定. 本节实验中相关参数的设置与文献[13]的参数设置一样, 具体见表 1. 实验结果表明: 通过比较发生变化的像素个数, 在 L_1 层大部分对抗性反例的变化维度可以在 50 维内发现转向角误分类的变化, 其中一些对抗性反例的变化维度甚至小于 10 维. 在本文随后的图中, 展示了由本节算法报告出的一些对抗性反例. 每组图像的左侧图像为原始图像, 右侧图像为扰动后的图像, 并在每张图像下方标记出相应的预测转向角.

表 1 驾驶实验参数

参数	参数值
时间步长	Δt=0.25 s
速度范围	v_{ego}∈[0 m/s,15 m/s]
纵向加速度范围	$a_{ego,lon}$∈[−4 m/s^2,2 m/s^2]
横向加速度范围	$a_{ego,lat}$∈[−8 m/s^2,8 m/s^2]
抖动参数	j_{ego}∈[−10 m/s^3,10 m/s^3]
目标速度	v_{des}=13.9m/s
曲率变化	κ∈[−0.2m/s,0.2m/s]
曲率变化范围	$\dot{\kappa}$ ∈[−0.2m/s,0.2m/s]
汽车长和宽	长=5.238 m, 宽=2.169 m
近似汽车的圆	l=3.5 m, r=1.4 m
反应制动时间	δ_{brake}=0.3 s
反应转向时间	δ_{steer}=0.3 s
车道宽	3.5 m
规避距离	d_{eva}=3.5 m
J_{lon} 中的权重	w_a=1, w_j=2
J_{lat} 中的权重	w_d=0.2, w_θ=2, w_κ=20, $w_{\dot{\kappa}}$=20

我们从驾驶数据集中随机选择 100 张图像. 表 2 给出了运用本文方法报告出的对抗性反例的统计数据. 实验设置最大改变维数 *dim* 为{50,150,300}, 从第 1 层开始单路径搜索对抗性反例.

表 2 本文方法报告的对抗性反例统计

dim	50	150	300
对抗性反例的数目	91	94	98

在图 10 中给出了两组原始(正确预测)和扰动操作后获得的(错误预测)图像. 图 10(a)在第 1 层搜索中, 当变化的维数达到 20 时报告一个不安全转向角的对抗性反例, 层 L_1 有 1 824 维, 该数据占比为 0.011%. 图 10(c)在第 1 层搜索中, 当第 1 个维度变化时就立即报告有不安全转向角的对抗性反例, 该数据占层 L_1 的总维度的 0.00055%. 这两个对抗性例子表明: 即使扰动操作涉及很少的维度, 运用单路径的搜索方式也可以快速找到对抗性反例. 然而, 搜索对抗性反例并不总能成功, 图 11 所示的图像在 L_1 层中, 当变化的维数达到 150 时被报告仍是安全的, 并未发现对抗性反例. 在这种情况下, 需要涉及更多维度和更复杂的操作来搜索转向角决策的改变. 操作改变原始图像部分维度的激活值. 每幅图像都是从第 1 个隐藏层反向映射得到的, 并且表示靠近对应区域边界的一个点. 选择的维数越大, 意味着应用操作的区域就越大, 并且, 当操作移动到该区域的边界时, 表明由激活所处位置会发生更大的变化. 例如, 图 11 中, 两个示例变化维数达到 150, 两幅扰动图像的上方有明显的块状显示, 而图 10 中两个示例的变化维数只有 20 和 1, 因而扰动后的图像与原始图像的区别并不明显.

图 12 给出了两组多路径搜索的结果. 两组例子在第 1 层的第 1 维发生改变时就报告有不安全转向角, L_1 层共有 1 824 维, 这个维数变化占到第 1 层维数的 0.00055%. 这两个对抗性例子表明: 即使扰动操作涉及很少的维度, 运用多路径的搜索方式也可以快速找到对抗性反例.

(a) −14.52° (b) −15.4° (c) −11.89° (d) −12.14°

图 10 单路径搜索到的对抗性反例

(a) −17.38° (b) −14.32° (c) −16.43° (d) −14.29°

图 11 单路径搜索到最大维数 150 未发现对抗性反例

(a) −13.35° (b) −13.94° (c) −16.03° (d) −16.64°

图 12 多路径搜索到的对抗性反例

此外, 我们还补充了 3 个示例, 其中, 图 13(b)和图 13(d)是运用单路径搜索到的对抗性反例, 图 13(f)是运用多路径搜索到的对抗性反例. 图 13(a)在第 1 层的搜索中, 当变化的维数达到 9 时报告不安全转向角, L_1 层共有 1 824 维, 这个维数变化占到第 1 层维数的 0.00493%. 图 13(c)和图 13(f)在第 1 层的第 1 维发生改变时报告不安全转向角, 这个维数变化占到第 1 层维数的 0.00055%.

(a) −7.05° (b) −6.62° (c) −14.9° (d) −13.6° (e) 2.57° (f) 1.21°

图 13 补充示例

对于场景图 13(c)和图 13(e), 我们假设了真值数据集, 具体场景参数见表 3.

表 3 场景参数

参数	参数值
图 14 中的主车辆	$(x,y,\theta,v)^{\mathrm{T}}_{ego}$ =(0 m,0 m,−16.84°,33 m/s)$^{\mathrm{T}}$
图 14 中的障碍物 1	$(x,y,\theta,v)^{\mathrm{T}}_{b1}$ =(7.4 m,3.7 m,−15.63°,35 m/s)$^{\mathrm{T}}$
图 14 中的障碍物 2	$(x,y,\theta,v)^{\mathrm{T}}_{b2}$ =(26.7 m,0.7 m,−14.92°,27 m/s)$^{\mathrm{T}}$
图 15 中的主车辆	$(x,y,\theta,v)^{\mathrm{T}}_{ego}$ =(0 m,0 m,4.54°,33 m/s)$^{\mathrm{T}}$
图 15 中的障碍物 1	$(x,y,\theta,v)^{\mathrm{T}}_{b1}$ =(21.4 m,3.7 m,2.62°,35 m/s)$^{\mathrm{T}}$
图 15 中的障碍物 2	$(x,y,\theta,v)^{\mathrm{T}}_{b2}$ =(36 m,−0.3 m,2.22°,27 m/s)$^{\mathrm{T}}$
初始加速度 a_0	0 m/s^2
障碍物长度	4.8 m
障碍物宽度	2 m
占有率计算步长	Δt=0.1 s
占有率计算时长	1.5 s

我们假设场景图 13(c)和图 13(e)在一条两车道的高速公路上, 主车辆位于靠右边的车道. 在场景图 13(c)

中, 障碍物 1 位于主车辆行驶方向左侧, 与主车辆距离较近, 并且速度快于主车辆, 而与主车辆同车道的障碍物 2 的速度慢于主车辆. 在场景图 13(e)中, 障碍物 1 和障碍物 2 都位于主车辆的前方, 障碍物 1 的速度快于主车辆的速度, 障碍物 2 的速度慢于主车辆. 我们运用工具 SPOT[59]绘制了两个障碍物在时刻 $t\in[0\ \text{s},1.5\ \text{s}]$的预测占有率, 得到图 14 和图 15: 图 14 对应于场景图 13(c)的占有率, 图 15 对应于场景图 13(e)的占有率. 根据预测占有率计算出车辆的纵向位置约束和横向位置约束, 然后计算出横向和纵向轨迹.

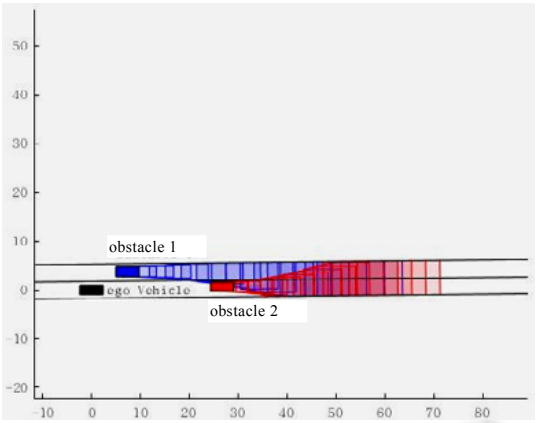

图 14 场景图 13(c)的占有率

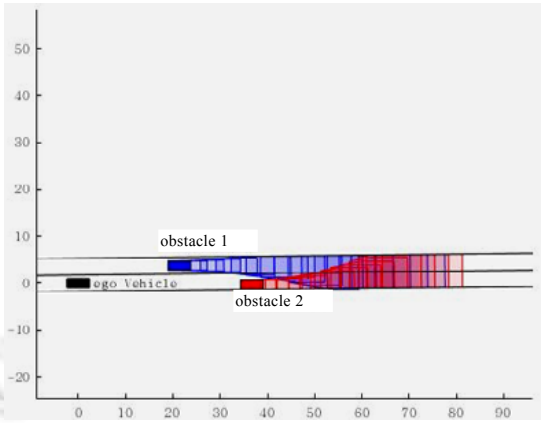

图 15 场景图 13(e)的占有率

4.3 对比实验

我们将所提出的验证方法与现有的验证无人驾驶汽车转向角安全性的方法 SDLV[26]进行比较. 在搜索对抗性反例时, 我们的方法与 SDLV 在输入层或隐藏层的特征空间中探索了一定比例的维数, 而两种方法在将转向角判断问题转化为可分类问题的依据不同. 从直观上看, 本文方法与 SDLV 对转向角的判断差异在于: 本文方法依赖于轨迹规划中避撞; 而 SDLV 是基于统计意义的, 即神经元覆盖和松弛关系.

在我们的实验中, 一些运用 SDLV 未能发现对抗性反例的图像, 运用本文的方法能够成功搜索到. 在图 16 中, 我们给出了两个运用 SDLV 搜索 300 维都未能找到对抗性反例的例子. 运用本文方法, 两组图像都在第 1 层搜索中, 当第 1 个维度变化时就报告有不安全转向角的对抗性反例, 该数据占层 L_1 的总维度的 0.00055%.

(a) −13.12° (b) −13.06° (c) −4.47° (d) −4.58°

图 16 SDLV 中未找到对抗性反例, 运用本文方法单路径搜索到

另外, 将本文的验证方法与两种现有的自动检测 DNN 驱动的无人驾驶汽车错误转向的方法 SDLV 和 DeepTest[32]进行比较. DeepTest 应用图像转换自动生成测试用例来检测错误的转向行为, 而本文方法和 SDLV 探索输入或隐藏层的特征空间中一定比例的维数.

我们从驾驶数据集中随机选择 100 张图像. 表 4 给出了 3 种方法在驾驶数据集上的鲁棒性评估比较. 对于 DeepTest, 根据文献[32]中的实验结果, 我们选择 5 作为输入参数λ来平衡假阳性和假阴性. 在 SDLV 的实验中, 将 5 作为可配置参数μ^2和ξ, 这里, μ^2 相当于文献[32]中的λ. 类似地, 我们的实验参数也这样设置, 设置最大改变维数 dim 为{50,150,300}, 选择单路径搜索从第 1 层开始. 验证过程中, 对每个输入图像进行小于最大维度的扰动操作, 如果找到对抗性反例, 则返回该示例并终止程序; 而当搜索达到最大维数时, 程序报告

失败并返回最后一个扰动示例.

表 4 DeepTest vs. SDLV vs. 本文方法

	DeepTest	SDLV (*dim*=50)	SDLV (*dim*=150)	SDLV (*dim*=300)	本文方法 (*dim*=50)	本文方法 (*dim*=150)	本文方法 (*dim*=300)
L^1	25 731	10 246.25	16 531	38 359.6	5 253	16 468	30 963.9
L^2	159.6	97.08	112.55	199.66	46.73	78.44	87.76
成功率(%)	19	37	52	73	91	94	98

表 4 收集了 3 个统计数据, 包括输入图像与返回的扰动图像之间的平均 L^1 距离和平均 L^2 距离, 以及找到对抗性反例的成功率. 3 个统计数据的计算公式分别为公式(16)和公式(17):

$$L^d=\frac{\sum_{x\in C} diff(x,\delta(x))\times L^d(x,\delta(x))}{\sum_{x\in C} diff(x,\delta(x))} \tag{16}$$

$$\text{成功率}(\%)=\frac{\sum_{x\in C} diff(x,\delta(x))}{M} \tag{17}$$

我们用 C 和 M 分别表示测试集和测试集的图像数量(本文实验随机选取 100 张图像作为测试集, 即 M=100), 用 $L^d(x,\delta(x))$表示输入 x 和报告的扰动图像$\delta(x)$的距离, 其中, d 取值 1 或 2 分别对应 L^1 距离和 L^2 距离. 设置 $diff(x,\delta(x))\in\{0,1\}$是布尔值, 用来表示 x 和$\delta(x)$的转向角是否有相同分类. 对于成功率非常高的情况, 即 λ=5、DeepTest 为 19%、*dim*=300 的 SDLV 为 73%, 而本文的验证方法在 *dim*=300 时为 98%. 此时, 本文的验证方法与 DeepTest 在 L^1 和 L^2 距离上的平均距离都小于 SDLV. 当 *dim*=50 或 *dim*=150 时, 本文的验证方法与 SDLV 在 L^1 和 L^2 距离上的平均距离都小于 DeepTest. 显然, 本文的验证方法发现对抗性反例的成功率远高于另外两种, 这是因为本文验证方法所求解出的安全转向角区间的判别方法比 SDLV 中根据神经元覆盖率和松弛条件判断转向角更精确和严格.

误分类距离越小, 就越有可能导致较低的可转移率[67], 这意味着在相同或一小部分数据集上训练的另一个模型上可能更难发现误分类[27]. 事实上, DeepTest 对整个输入图像进行图像转换, 而本文的验证方法与 SDLV 只是根据网络拓扑对可能导致误分类的部分维度进行操作. 因此, DeepTest 的平均 L^1 和 L^2 距离大于 *dim*=50 或 *dim*=150 时本文的验证方法以及 SDLV 的平均距离. 但是, 当变化维度 *dim* 增加到 300 时, SDLV 和本文的验证方法均优于 DeepTest.

SDLV 的转向角判断方法仅从统计的角度出发, 与避撞相比显得过于粗糙和不精确. 为了解决这个问题, 本文运用故障安全轨迹中的凸优化方法求解出能够满足避撞需求的转向角, 结合原始图像的预测转向角, 构造出安全的转向角区间, 作为转向角的判断. 图 16 展示了两个运用 SDLV 搜索 300 维都未找到对抗性反例, 而运用本文方法却能找到的例子. 另一方面, 从表 4 所示搜索到的对抗性反例的成功率能够发现: 无论 *dim* 取值如何, 本文方法搜索到对抗性反例的成功率比 SDLV 的成功率要高. 上述两方面综合加以考虑, 说明基于凸优化的转向角判断方式比基于统计的方式更严格和精确.

5 总 结

本文提出了一个自动验证无人驾驶汽车的转向角安全性的方法. 利用故障安全轨迹规划中的凸优化技术, 构造了一个关于安全转向角的区间, 将预测转向角的判断问题转化为可分类问题, 拓展了 DLV 执行无人驾驶汽车转向角的安全性验证. 本文在 NVIDIA 的端到端无人驾驶架构上进行实验, 以说明所提出验证方法的优势. 实验结果表明: 在某些情况下, 即使扰动操作涉及很少的维度, 也可以在几秒内找到对抗性反例. 与我们之前的工作 SDLV 相比, 该方法找到对抗性反例的成功率更高. 此外, 在处理转向角的分类问题时, 运用凸优化技术构造的安全转向角区间比 SDLV 中依赖神经元覆盖率和松弛关系的方式更准确和可靠.

下一步, 我们计划提高本文技术的可扩展性. 具体来说, 我们将纳入更多关于轨迹规划和避免碰撞的考虑, 以进行更加精准的转向角安全性验证, 以及涉及更多无人驾驶预测输出项的安全性. 对于涉及“固定障碍

物”“变道车辆”和“鬼探头行人”的行为安全性, 我们将基于转向角的安全性验证, 结合雷达、红外灯等传感器的输入数据, 进一步考虑涉及无人驾驶汽车另外两个预测输出加速度和制动的行为安全性验证. 除了改进我们技术中涉及的安全考虑之外, 还计划通过探索扰动操作和转向角之间的关系来提供更好的可解释性, 我们认为, 这也是一项有趣且具有挑战性的工作.

References:

[1] Google’s self-driving car caused its first crash. 2016. https://www.wired.com/2016/02/googles-self-driving-car-may-caused-first-crash/

[2] Uber driver charged in fatal 2018 autonomous car crash. 2018. https://www.autoweek.com/news/technology/a34039541/uber-driver-charged-in-fatal-2018-autonomous-car-crash/

[3] NHTSA probing 16 deadly Tesla highway crashes. 2023. https://www.kron4.com/news/bay-area/nhtsa-probing-16-deadly-tesla-autopilot-highway-crashes/

[4] Biggio B, Corona I, Maiorca D, Nelson B, Srndic N, Laskov P, Giacinto G, Roli F. Evasion attacks against machine learning at test time. In: Proc. of the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. Berlin, Heidelberg: Springer, 2013. 387−402.

[5] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. In: Proc. of the 2nd Int’l Conf. on Learning Representations. 2014. 14−16.

[6] Althoff D, Althoff M, Scherer S. Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets. In: Proc. of the 2015 IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems (IROS). IEEE, 2015. 3470−3477.

[7] Kalra N, Paddock SM. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability. Transportation Research Part A: Policy and Practice, 2016, 94: 182−193.

[8] Pek C, Koschi M, Althoff M. An online verification framework for motion planning of self-driving vehicles with safety guarantees. In: Proc. of the AAET-Automatisiertes und vernetztes Fahren. Braunschweig, 2019.

[9] Damm W, Peter HJ, Rakow J, Westphal B. Can we build it: Formal synthesis of control strategies for cooperative driver assistance systems. Mathematical Structures in Computer Science, 2013, 23(4): 676−725.

[10] Hilscher M, Linker S, Olderog ER. Proving safety of traffic manoeuvres on country roads. In: Proc. of the Theories of Programming and Formal Methods. Berlin, Heidelberg: Springer, 2013. 196−212.

[11] Loos SM, Platzer A, Nistor L. Adaptive cruise control: Hybrid, distributed, and now formally verified. In: Proc. of the Int’l Symp. on Formal Methods. Berlin, Heidelberg: Springer, 2011. 42−56.

[12] Mitsch S, Loos SM, Platzer A. Towards formal verification of freeway traffic control. In: Proc. of the IEEE/ACM 3rd Int’l Conf. on Cyber-physical Systems. IEEE, 2012. 171−180.

[13] Pek C, Althoff M. Fail-safe motion planning for online verification of autonomous vehicles using convex optimization. IEEE Trans. on Robotics, 2020, 37(3): 798−814.

[14] Althoff M. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In: Proc. of the 16th Int’l Conf. on Hybrid Systems: Computation and Control. 2013. 173−182. [doi: 10.1145/2461328.2461358]

[15] Herbert SL, Chen M, Han SJ, Bansal S, Fisac JF, Tomlin CJ. FaSTrack: A modular framework for fast and guaranteed safe motion planning. In: Proc. of the 56th IEEE Annual Conf. on Decision and Control (CDC). IEEE, 2017. 1517−1522.

[16] Mitchell IM. Comparing forward and backward reachability as tools for safety analysis. In: Proc. of the 10th Int’l Workshop on Hybrid Systems: Computation and Control. Pisa, 2007. 428−443.

[17] Althoff D, Buss M, Lawitzky A, Werling M, Wollherr D. On-line trajectory generation for safe and optimal vehicle motion planning. In: Proc. of the Autonomous Mobile Systems 2012. Berlin, Heidelberg: Springer, 2012. 99−107. [doi: 10.1007/978-3-642-32217-4_11]

[18] Martinez-Gomez L, Fraichard T. An efficient and generic 2D inevitable collision state-checker. In: Proc. of the 2008 IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems. IEEE, 2008. 234−241.

[19] Petti S, Fraichard T. Safe motion planning in dynamic environments. In: Proc. of the 2005 IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems. IEEE, 2005. 2210−2215.

[20] Berntorp K, Weiss A, Danielson C, Kolmanovsky IV, Cairano SD. Automated driving: Safe motion planning using positively invariant sets. In: Proc. of the 20th IEEE Int'l Conf. on Intelligent Transportation Systems. 2017. 1−6.

[21] Jalalmaab M, Fidan B, Jeon S, Falcone P. Guaranteeing persistent feasibility of model predictive motion planning for autonomous vehicles. In: Proc. of the 2017 IEEE Intelligent Vehicles Symp. (IV). IEEE, 2017. 843−848.

[22] Fraichard T. A short paper about motion safety. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation. IEEE, 2007. 1140−1145.

[23] Althoff D, Kuffner JJ, Wollherr D, Buss M. Safety assessment of robot trajectories for navigation in uncertain and dynamic environments. Autonomous Robots, 2012, 32(3): 285−302. [doi: 10.1007/s10514-011-9257-9]

[24] Söntges S, Althoff M. Determining the nonexistence of evasive trajectories for collision avoidance systems. In: Proc. of the 18th Int'l Conf. on Intelligent Transportation Systems. IEEE, 2015. 956−961. [doi: 10.1109/ITSC.2015.160]

[25] Söntges S, Althoff M. Computing the drivable area of autonomous road vehicles in dynamic road scenes. IEEE Trans. on Intelligent Transportation Systems, 2018, 19(6): 1855−1866.

[26] Wu H, Lv D, Cui T, Hou G, Watanabe M, Kong W. SDLV: Verification of steering angle safety for self-driving cars. Formal Aspects of Computing, 2021, 33(3): 325−341. [doi: 10.1007/s00165-021-00539-2]

[27] Huang X, Kwiatkowska M, Wang S, Wu M. Safety verification of deep neural networks. In: Proc. of the Int'l Conf. on Computer Aided Verification. Cham: Springer, 2017. 3−29. [doi: 10.1007/978-3-319-63387-91]

[28] Pei K, Cao Y, Yang J, Jana S. Deepxplore: Automated whitebox testing of deep learning systems. In: Proc. of the 26th Symp. on Operating Systems Principles. 2017. 1−18. [doi: 10.1145/3132747.3132785]

[29] Bojarski M, Testa DD, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang JK, Zhang X, Zhao J, Zieba K. End to end learning for self-driving cars. arXiv:1604.07316, 2016.

[30] Rambo model. 2017. https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/rambo

[31] Koopman P, Wagner M. Challenges in autonomous vehicle testing and validation. SAE Int'l Journal of Transportation Safety, 2016, 4(1): 15−24. [doi: 10.4271/2016-01-0128]

[32] Tian Y, Pei K, Jana S, Ray B. DeepTest: Automated testing of deep-neural-network-driven autonomous cars. In: Proc. of the 40th Int'l Conf. on Software Engineering. 2018. 303−314. [doi: 10.1145/3180155.3180220]

[33] Zhang M, Zhang Y, Zhang L, Liu C, Khurshid S. DeepRoad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In: Proc. of the 33rd ACM/IEEE Int'l Conf. on Automated Software Engineering (ASE 2018). Montpellier, 2018. 132−142. [doi: 10.1145/3238147.3238187]

[34] Althoff M, Dolan JM. Online verification of automated road vehicles using reachability analysis. IEEE Trans. on Robotics, 2014, 30(4): 903−918.

[35] Majumdar A, Tedrake R. Funnel libraries for real-time robust feedback motion planning. The Int'l Journal of Robotics Research, 2017, 36(8): 947−982. [doi: 10.1177/0278364917712421]

[36] Vaskov S, Kousik S, Larson H, Bu F, Ward J, Worrall S, Johnson-Roberson M, Vasudevan R. Towards provably not-at-fault control of autonomous robots in arbitrary dynamic environments. In: Bicchi A, Kress-Gazit H, Hutchinson S, eds. Proc. of the Robotics: Science and Systems XV. Freiburg im Breisgau: University of Freiburg, 2019.

[37] Wu H, Lyu D, Zhang Y, Hou G, Watanabe M, Wang J, Kong W. A verification framework for behavioral safety of self-driving cars. IET Intelligent Transport Systems, 2022, 16(5): 630−647.

[38] Luckcuck M, Farrell M, Dennis LA, Dixon C, Fisher M. Formal specification and verification of autonomous robotic systems: A survey. ACM Computing Surveys (CSUR), 2019, 52(5): 1−41. [doi: 10.1145/3342355]

[39] Lefèvre S, Vasquez D, Laugier C. A survey on motion prediction and risk assessment for intelligent vehicles. ROBOMECH Journal, 2014, 1(1): 1−14.

[40] Kim B, Park K, Yi K. Probabilistic threat assessment with environment description and rule-based multi-traffic prediction for integrated risk management system. IEEE Intelligent Transportation Systems Magazine, 2017, 9(3): 8−22.

[41] Annell S, Gratner A, Svensson L. Probabilistic collision estimation system for autonomous vehicles. In: Proc. of the 19th IEEE Int'l Conf. on Intelligent Transportation Systems (ITSC). IEEE, 2016. 473−478.

[42] Lambert A, Gruyer D, Pierre GS, Ndjeng AN. Collision probability assessment for speed control. In: Proc. of the 11th Int'l IEEE Conf. on Intelligent Transportation Systems. IEEE, 2008. 1043−1048.

[43] Reschka A, Böhmer JR, Nothdurft T, Hecker P, Lichte B, Maurer M. A surveillance and safety system based on performance criteria and functional degradation for an autonomous vehicle. In: Proc. of the 15th Int'l IEEE Conf. on Intelligent Transportation Systems. IEEE, 2012. 237−242.

[44] Stahl T, Eicher M, Betz J, Nothdurft T, Diermeyer F. Online verification concept for autonomous vehicles–illustrative study for a trajectory planning module. In: Proc. of the IEEE 23rd Int'l Conf. on Intelligent Transportation Systems (ITSC). IEEE, 2020. 1−7. [doi: 10.1109/ITSC45102.2020.9294703]

[45] Schürmann B, Heß D, Eilbrecht J, Stursberg O, Koster F, Althoff M. Ensuring drivability of planned motions using formal methods. In: Proc. of the IEEE 20th Int'l Conf. on Intelligent Transportation Systems (ITSC). IEEE, 2017. 1−8.

[46] Kane A, Chowdhury O, Datta A, Koopman P. A case study on runtime monitoring of an autonomous research vehicle (ARV) system. In: Proc. of the Runtime Verification. Cham: Springer, 2015. 102−117. [doi: 10.1007/978-3-319-23820-3_7]

[47] Feth P, Schneider D, Adler R. A conceptual safety supervisor definition and evaluation framework for autonomous systems. In: Proc. of the 36th Int'l Conf. on Computer Safety, Reliability, and Security (SAFECOMP 2017). Trento, 2017. 135−148. [doi: 10.1007/978-3-319-66266-4_9]

[48] Shalev-Shwartz S, Shammah S, Shashua A. On a formal model of safe and scalable self-driving cars. arXiv:1708.06374, 2017.

[49] Idriz AF, Abdul Rachman AS, Baldi S. Integration of auto-steering with adaptive cruise control for improved cornering behaviour. IET Intelligent Transport Systems, 2017, 11(10): 667−675. [doi: 10.1049/iet-its.2017.0089]

[50] Rachman ASA, Idriz AF, Li S, Baldi S. Real-Time performance and safety validation of an integrated vehicle dynamic control strategy. IFAC-PapersOnLine, 2017, 50(1): 13854−13859.

[51] Liu BB, Liu WW, Mao XG, Dong W. Correctness verification of rules for unmanned vehicles's decision system. Computer Science, 2017, 44(04): 72−74, 113 (in Chinese with English abstract).

[52] Cho DS, Yun S, Kim H, Kwon J, Kim W. Autonomous driving system verification framework with FMI co-simulation based on OMG DDS. In: Proc. of the IEEE Int'l Conf. on Consumer Electronics (ICCE). IEEE, 2020. 1−6.

[53] Zhu Y, Zhao XM, Li CY, Li Y, Wang RM. Design and verification of virtual-real interaction system for automatic driving virtual-real combined test. In: Proc. of the 2022 World Transportation Conf. (Transportation Planning and Interdisciplinary) (WTC 2022). 2022. 910−918 (in Chinese). [doi: 10.26914/c.cnkihy.2022.019368]

[54] Long X, Gao JB, Wei HB. Development and test validation of a systematic architecture for autonomous vehicle. Journal of Chongqing University of Technology (Natural Science), 2019, 33(12): 45−54 (in Chinese with English abstract).

[55] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature, 1986, 323(6088): 533−536.

[56] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press, 2016.

[57] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: Proc. of the ICML. 2010.

[58] Pek C, Althoff M. Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization. In: Proc. of the 21st Int'l Conf. on Intelligent Transportation Systems (ITSC). IEEE, 2018. 1447−1454.

[59] Koschi M, Althoff M. SPOT: A tool for set-based prediction of traffic participants. In: Proc. of the IEEE Intelligent Vehicles Symp. (IV). IEEE, 2017. 1686−1693.

[60] Boyd S, Boyd SP, Vandenberghe L. Convex Optimization. Cambridge University Press, 2004.

[61] Paden B, Čáp M, Yong SZ, Yershov D, Frazzoli E. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Trans. on Intelligent Vehicles, 2016, 1(1): 33−55.

[62] Diamond S, Boyd S. CVXPY: A Python-embedded modeling language for convex optimization. The Journal of Machine Learning Research, 2016, 17(1): 2909−2913.

[63] SullyChen. Driving dataset.

[64] Z3. 2019. http://rise4fun.com/z3

[65] Keras. 2019. https://keras.io

[66] Theano. http://deeplearning.net/software/theano/

[67] Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A. Practical black-box attacks against deep learning systems using adversarial examples. arXiv:1602.02697, 2016.

附中文参考文献:

[51] 刘斌斌, 刘万伟, 毛晓光, 董威. 无人驾驶汽车决策系统的规则正确性验证. 计算机科学, 2017, 44(4): 72–74+113.

[53] 朱宇, 赵祥模, 李春银, 李妍, 王润民. 面向自动驾驶虚实结合测试的虚实交互系统设计与验证. 见: 2022 世界交通运输大会(WTC 2022)论文集(运输规划与交叉学科篇). 2022. 910–918. [doi: 10.26914/c.cnkihy.2022.019368]

[54] 龙翔, 高建博, 隗寒冰. 一种自动驾驶汽车系统架构开发与测实验证. 重庆理工大学学报(自然科学版), 2019, 33(12): 45–54.

吴慧慧(1992－), 女, 博士生, CCF 学生会员, 主要研究领域为无人驾驶, 安全性验证.

张亚楠(1987－), 女, 硕士, 主要研究领域为智能网联汽车网络安全, 数据安全等信息安全技术与政策研究.

侯刚(1982－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为智能系统, 对抗机器学习, 信息物理系统, 可信软件, 形式化方法, 模型检测.

渡边政彦(1962－), 男, 博士, 主要研究领域为嵌入式软件设计, 软件工程, 形式化验证, 自动驾驶算法及仿真技术.

王洁(1979－), 男, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为并行体系结构, 异构计算, 演化硬件与容错计算, 网络信息安全.

孔维强(1978－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为软件工程, 模型检测, 形式化方法.