

联合注意力机制与目标点信息的车辆轨迹预测

刘建铭¹, 陈伟侠², 卢仲康²

(1. 华南理工大学 机械与汽车工程学院, 广州 510641;

2. 广州华工机动车检测技术有限公司, 广州 510640)

摘要: 对周围环境中运动物体未来状态的准确预测是影响自动驾驶车辆做出准确决策的重要影响因素, 车辆是最常见也是最需要关注的运动物体之一; 针对结构化道路下周围车辆轨迹预测的多模态输入问题, 提出了基于注意力机制的深度预测网络; 提出交互模块以提取目标车辆与周围车辆及车道线信息存在的交互特征; 结合车道线信息对车辆运动的指引作用, 加入目标点预测模块以预测目标车辆可能到达的目标点, 增加预测准确性; 在 Argoverse 公开数据集上进行实验, 所提轨迹预测网络在 3 秒预测时长实现了 1.45 m 最小平均距离误差及 3.21 m 最小最终距离误差的预测精度, 优于当前主流的预测算法。

关键词: 自动驾驶; 轨迹预测; 深度学习; 门控循环单元; 注意力机制

Vehicle Trajectory Prediction Combined Attention Mechanism with Target Information

LIU Jianming¹, CHEN Weixia², LU Zhongkang²

(1. School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641, China;

2. Guangzhou Huagong Automobile Inspection Technology Co., Ltd., Guangzhou 510640, China)

Abstract: The accurate prediction of the future state of moving objects in the surrounding environment is an important influencing factor for autonomous vehicles to make accurate decisions. A vehicle is one of the most common and demanding moving objects. A depth prediction network based on attention mechanism is proposed for the multi-modal input problem of vehicle trajectory prediction on the structural road. An interactive module is proposed to extract the interactive features of target vehicle, surrounding vehicles and lane line information. Combined with the guidance of lane line information to the vehicle movement, the target point prediction module is added to predict the possible target point of target vehicle to increase the prediction accuracy. The experiments are conducted on Argoverse public dataset, the proposed prediction network achieves the prediction accuracy for the minimum average distance error of 1.45 m and minimum final distance error of 3.21 m in the prediction range of 3 s, which is better than current mainstream methods.

Keywords: autonomous driving; trajectory prediction; deep learning; gated recurrent unit (GRU); attention mechanism

0 引言

对周围环境中车辆及行人的轨迹预测是自动驾驶关键技术之一, 对未来轨迹的准确预测能够避免自动驾驶车辆规划出错误的路径, 增强决策规划模块输出的有效性, 输出更加平滑的规划轨迹。

目前车辆轨迹预测方法主要可以分为两大类型: 基于物理模型以及深度学习方法。基于物理模型的预测方法将车辆表示为受限于某种物理模型控制的运动物体, 该模型一般为运动学模型或者动力学模型。动力学模型需要有准确的车辆物理参数, 因此在预测任务中运动学模型更为常见。常用的运动学模型包括了恒定速度模型、恒定加速度模型、恒定角速度和速度模型^[1]以及恒定角速度和加速度模型^[2]。物理模型方法通常只适用于短期预测, 例如预测一秒内的未来轨迹, 因为这些算法无法对驾驶员的驾驶行

为进行预测^[3]。

深度学习则通过大数据驱动, 使模型能够自主的学习到其他环境因素对目标车辆驾驶行为的影响。轨迹预测作为一个序列预测的问题, 循环神经网络 (RNN, recurrent neural network) 自然的被引入到轨迹预测中。文献 [4] 中将车辆的历史坐标及加速度输入到 LSTM 网络中, 输出车辆在未来时刻位于占据栅格地图每个网格的概率。文献 [5] 基于生成对抗模型进行轨迹预测, 使用 LSTM 作为生成模型的编码器及解码器。

车辆的未来状态除了与自身的历史轨迹相关, 同时也取决于与其他车辆的交互。文献 [6-7] 将被预测车辆周围车辆的信息加入到输入中, 考虑与周围车辆的交互明显加强了预测的准确性。在 Social-LSTM^[8]中提出了社会池化层 (social pooling) 处理行人之间的空间关系, 不同行人的轨迹经过 LSTM 的编码被放入到一个网格图中, 此网格图即

收稿日期: 2023-05-09; 修回日期: 2023-05-29。

作者简介: 刘建铭 (1996-), 男, 硕士。

通讯作者: 卢仲康 (1977-), 男, 大学本科。

引用格式: 刘建铭, 陈伟侠, 卢仲康. 联合注意力机制与目标点信息的车辆轨迹预测[J]. 计算机测量与控制, 2023, 31(11): 106-112, 118.

表示了当前时刻行人的空间关系。Deo 等人^[9]在社交池化层后使用卷积层代替原本的全连接层, 利用卷积层的等价性能提取网格图中的局部有用信息。Lin^[10]等人在社交池化层增加了额外的注意力层, 在每个时间步 T 计算目标车辆对其他网格内车辆的注意力权重, 该权重与 LSTM 输出的隐变量相乘。

在结构化道路下行驶的车辆一般会沿着车道线行驶, 如果能有效地提取道路信息将提升网络预测性能。Khandelwal 等人提出 WIMP^[11], 在轨迹预测中加入了路网信息, 将路网分段离散并表示成有向图。LanGCN^[12]同样将路网表示为有向图, 并定义了 LaneConv 算子以充分利用路网的连接关系。VectorNet^[13]则将道路和轨迹向量化, 通过聚合函数将向量聚合在一个高维特征中, 并使用注意力机制提取各个高维特征之间的信息。文献 [14] 延续了 VectorNet 对道路的编码形式, 但是增加了路段和路点作为道路信息的额外特征并拼接在原来的高维特征中。文献 [15-16] 中道路信息被表示为图像的形式, 各种道路信息按照一定的分辨率绘制在图像的不同通道中, 使用卷积神经网络处理该图像以提取道路信息。

对驾驶员的目标点意图预测也是提升模型预测精度的方式, 在 TPNet^[16]中提出了预测目标点的方式以增加预测准确性。TNT^[17]模型使用分层图神经网络 VectorNet 来编码场景的特征向量, 该特征向量被用于预测目标点及生成最终轨迹。

综上, 车辆轨迹预测算法中依然存在以下问题:

1) 如何更高效编码道路信息以及历史轨迹信息, 并建立对应的交互模型以充分利用环境信息是预测模块提供准确预测轨迹的关键问题。

2) 目标点信息对未来轨迹预测精度是重要影响因素之一, 如何准确的预测目标点并利用目标点信息增强预测精

度同样也是一项问题。

针对以上问题, 本文以道路信息、车辆历史轨迹信息作为输入, 提出了基于注意力机制的实时轨迹预测网络。本文的主要贡献包括:

1) 对预测网络的多模态输入问题, 提出了基于注意力机制的交互模块处理车道线信息和历史轨迹信息, 该模块将提取影响车辆未来状态的 3 个重要交互信息: 车道线对于车辆的指引、车流信息以及车辆间的交互。

2) 增加目标点预测模块, 从道路中心线中采样目标点, 在轨迹预测模块中加入目标点信心进一步增强轨迹预测模块的准确性。

1 轨迹预测问题描述

轨迹预测模块的任务是给定自动驾驶车辆周围环境若干时间步的历史状态, 预测周围车辆未来几个时间步的状态或某个状态的转移概率。使用数学语言描述即为, 给定历史状态 X_{hs} :

$$X_{hs} = \{x_{t-n}^i, x_{t-(n-1)}^i, \dots, x_t^i\}_{i=1}^N \quad (1)$$

式中, X_t^i 表示周围环境中第 i 辆车在 t 时刻时的状态 (如位置、速度、加速度等), N 是周围车辆的数目, n 为考虑的历史时间的步长。根据以上历史状态 X_{hs} 计算第目标车辆的分布 $P(X_{pi} | O_{EV})$, 式 O_{EV} 中表示本车的对周围物体的观测值, $X_{pi} = \{x_t^i, x_{t+1}^i, \dots, x_{t+m}^i\}$ 表示第 i 辆车未来第 m 个时间步的状态。

2 轨迹预测网络结构设计

所提轨迹预测网络整体架构如图 1 所示, 轨迹预测网络主要包括了 3 个模块: 历史轨迹与道路的编码模块将低维输入转换为高维的特征向量; 交互模块负责提取道路与车辆、车辆与车辆之间的交互, 输出目标车辆与环境的交互特征; 预测模块包括了两个预测任务——目标点及轨迹

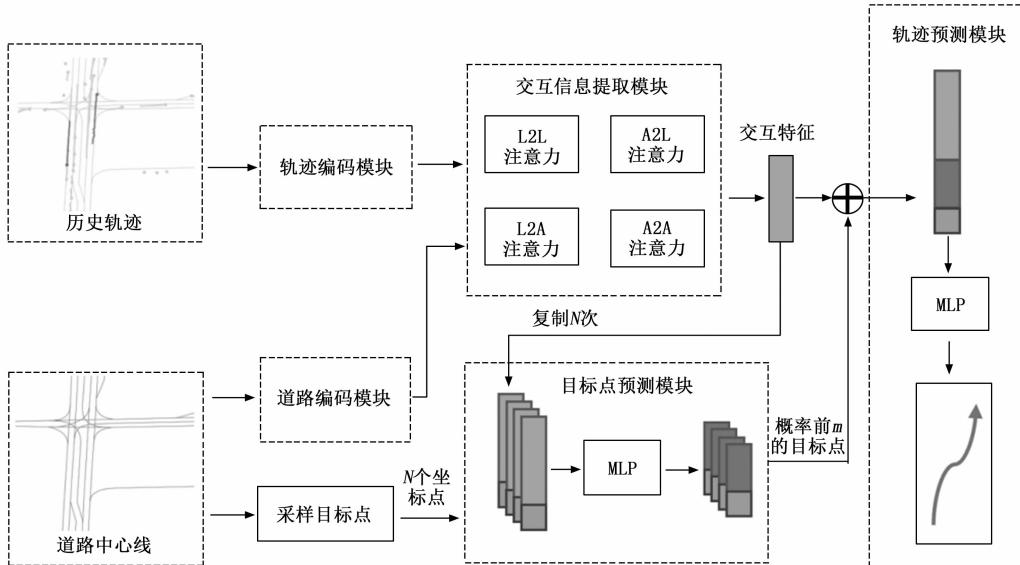


图 1 轨迹预测网络结构图

预测, 交互特征向量、预测目标点坐标及对应的概率作为该模块的输入。网络使用了简洁统一的架构对多模态的输入进行编码并提取不同输入特征之间的交互信息, 下面对每个模块进行详细介绍。

2.1 编码模块

2.1.1 轨迹编码模块

对于车辆轨迹信息, 预测网络的输入首先应当包括各个车辆的二维坐标, 由于传感器的采样频率并不能保证在全部分场景都是稳定的, 即在部分场景中轨迹的帧时间间隔大于设定的采样间隔, 所以将时间戳信息也加入到网络输入中。在部分时刻由于障碍物遮挡会导致观测信息为空值, 因此附加一列指示位表示该时刻是否有观测值。根据上述处理方法, 最后将得到一个 $m \times t_{obs} \times 4$ 的历史轨迹输入 $P_{his} = \{p_1, p_2, \dots, p_m\}$, 其中 m 为该场景中的相关车辆数目, t_{obs} 为可观测的时间长度。

从输入的特征可以看出车辆的历史信息是按照一定的时间间隔记录的车辆状态, 对时间序列进行编码的常用方式是循环神经网络 (RNN)。轨迹编码模块选用了 RNN 中的一个变体 UGRU^[18], 所有的轨迹将使用同一个 U-GRU 模块进行编码, 编码后的特征为 τ , 其定义如下:

$$\tau = \{t_1, t_2, \dots, t_m\} \quad (2)$$

$$t_i = UGRU(P_{his}) \quad (3)$$

式 (3) 中 U-GRU 的计算公式如下:

$$h_{t-1}^b = GRU(h_t^b, x_t, W_b) \quad (4)$$

$$h_{t+1}^f = GRU(h_t^f, concat[x_t, h_t^b], W_f) \quad (5)$$

式 (4) 与式 (5) 中, x_t 表示 t 时刻的输入, W_b 与 W_f 表示 UGRU 中可学习的参数, h_{t-1}^b 为 $t-1$ 时刻 GRU 模块的隐藏状态, $concat[\cdot, \cdot]$ 表示拼接操作, 最后的隐藏状态 $h_{t_m}^f$ 将作为 UGRU 模块的最终输出。

2.1.2 道路信息编码模块

结构化道路具有明确的交通标志如车道线、人行道等, 位于结构化道路的车辆在大部分情况下将沿着道路中心线行驶或移动至另一条道路中心线, 道路中心线是车辆驾驶行为的重要影响因素之一, 因此选用了道路中心线作为输入。

自动驾驶系统中的道路信息一般来自于高精地图 (HD map, high definition map), 高精地图在表示道路信息采用点、线段、区域 3 个层级的方式。点是最基本的构成单元, 可以用来表示路点或者交通指示灯。线段由离散的路点组成, 一般用于表示道路中心线。区域同样由点所组成, 线段与区域的区别是后者是封闭的, 一般用于表示道路的交互区域, 如人行横道等。

假设道路信息输入为 $L = \{L_0, L_1, \dots, L_n\}$, 其中 L_j 表示第 j 条道路中心线。输入采用文献 [13] 对道路信息的表示方法, 如图 2 所示, 将高精地图中的元素如道路中心线、人行道向量化。

向量化过程如下: 对于一段由 m 个路点组成的道路中心线 L_j , 第 i 个向量 v_i 由第 i 个路点作为起点与第 $i+1$ 个路点作为终点连接形成, 则这段道路中心线 L_j 将由 $m-1$ 个向

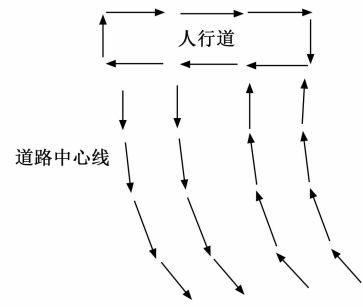


图 2 道路元素向量化与道路编码

量 v 组成, 向量的特征被定义为:

$$v_i = [X_i^s, X_i^e, A_i, j] \quad (6)$$

式 (6) 中 X_i^s, X_i^e 分别表示向量起点和终点的坐标, A_i 表示路点的额外信息如该路点的限速、是否处在交互区域等, j 表示了该段道路中心线的编号, 所有同一路段下的所有向量有相同的编号。

对向量化后道路中心线的编码方式, 同样使用文献 [13] 使用的方法。对于若干向量 v 组成的一段路段 L_j , 其编码特征 p_j 计算公式如下:

$$p_j = \varphi_{agg}(\{v_i^l\}) \quad (7)$$

$$v_i^{l+1} = \varphi_{rel}(g_{enc}(v_i^l), \varphi_{agg}(g_{enc}(v_i^l))) \quad (8)$$

式中 v_i^l 表示第 l 层第 i 个向量的编码特征; φ_{agg} 表示聚合函数, 具体实现为最大池化 (maxpooling); g_{enc} 为单个向量的编码函数, 具体实现为多层感知器 (MLP, Multi-Layer Perception), 每个向量将共享同一个 MLP 的权重参数; φ_{rel} 为关联函数, 将同一个路段内的向量聚合信息与向量的编码信息相关联, 具体实现为两个向量的拼接操作, 即

$$\varphi_{rel}(x, y) = concat[x, y] \quad (9)$$

将以上的编码方式应用在道路中心线的每个路段中得到最终的道路中心线编码特征。

2.2 交互模块

如前所述, 在结构化道路中行驶的车辆, 其驾驶行为除了自身驾驶意图外主要受到两个因素的制约, 首先是与同一场景中其他车辆的交互, 其次驾驶员会受到道路信息的指引, 自身驾驶意图主要由轨迹信息编码模块从历史轨迹中提取, 车辆之间、车辆与道路之间的交互信息将由交互模块获取。

交互模块的整体流程如图 3 所示, 场景中每个车辆历史轨迹与道路中心线的路段将被视为一个节点, 交互模块将使用注意力机制^[19]提取路段与车辆, 车辆与车辆之间的交互信息。该目标由图 3 中的 4 个子模块完成: 路段对于车辆的注意力模块 (L2A, lane to actors)、路段之间的注意力模块 (L2L, lane to lane)、车辆对于道路的注意力模块 (A2L, actors to lane) 以及车辆之间交互的注意力模块 (A2A, actors to actors), 其中前 3 个模块用于提取车辆与车道中心线之间的交互, 最后一个模块用于提取车辆之间的交互。

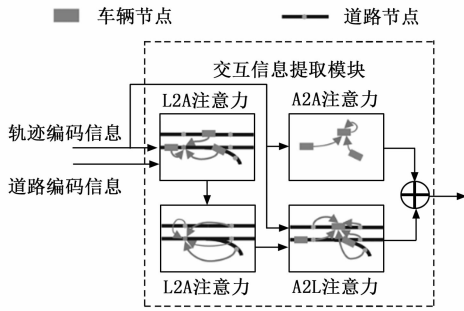


图 3 交互信息提取模块

L2A 模块以道路中心线编码特征 L 作为 query 的输入向量, 轨迹编码特征 τ 作为 key 与 value 的输入向量, 输出为每个路段对于周围车辆的注意力信息 A_{L2A} 。在轨迹编码特征 τ 中包含了每个车辆的历史轨迹, 该模块通过每个路段和车辆之间的空间信息从位于路段周围车辆信息中更新路段的交通状况。

L2L 模块以路段的注意力信息 A_{L2A} 作为 query、key 与 value 的输入向量, 输出为路段与路段之间的注意力信息 A_{L2L} 。经过自注意力后每个路段将自身周围路段的交通状况以不同权重聚合到该路段。

A2L 模块以轨迹编码特征 τ 作为 query 的输入向量, 路段与路段之间的注意力信息 A_{L2L} 作为 key 与 value 的输入向量, A2L 模块输出为 A_{A2L} 。该模块将最终将车道相关的信息以不同权重聚合到每个车辆节点上, 车辆节点对路段的不同权重也表示了各个路段节点对于车辆的指引作用。

A2A 模块负责提取车辆之间的交互信息, 以该场景中的每辆车的轨迹编码特征 τ 作为 query、key 与 value 的输入向量, 因此该模块的注意力同样是自注意力, 该模块输出为 A_{A2A} 。A2A 模块表示了车辆间的交互作用, 虽然经过 L2A、L2L 与 A2L 的交互信息流在一定程度上通过车流信息表示车辆间的交互, 但 A2A 模块以更直接的方式提取这种交互。

随后将 A2L 模块与 A2A 模块的输出相加得到交互特征 $I = A_{A2L} + A_{A2A}$, 交互特征 I 包含了场景中所有车辆对于车道线及其他车辆的交互信息, 本文研究的预测任务仅限于单个目标车辆, 交互模块最终仅输出目标车辆与其他车辆的交互信息 I_{agent} 。

2.3 轨迹预测

2.3.1 目标点预测

人类驾驶员另一个常见的驾驶习惯是确定一个目标点, 然后规划到这个目标的路径, 如果能准确预测目标车辆在未来几秒后的目标点就能增强对未来轨迹预测的准确性。

在道路信息编码模块中使用了道路中心线的路点, 在目标点预测任务中对离散路点进行重新采样以产生待选的目标点集合。对于目标点预测分为两个部分的预测, 对离散点的分类预测和偏离离散点距离的回归预测, 分类任务的目标是预测最靠近预测轨迹真实目标点的路点, 而回归任务是预测每个路点距离真实目标点的偏移量。即最终预

测的目标点集合 τ_p 为

$$\tau_p = \{(x_n, y_n) + (\Delta x_n, \Delta y_n)\}_{n=1}^N \quad (10)$$

式中, (x_n, y_n) 为离散路点的坐标, $(\Delta x_n, \Delta y_n)$ 为真实目标点与离散路点的偏移量, N 为候选目标点的个数。

该阶段的具体实现方式是: 假定预测场景中采样 N 个候选目标点, 将交互模块中属于预测目标车辆的交互向量 I_{agent} 复制 N 次, 并将每个候选目标点的坐标 (x_n, y_n) 与复制 N 次的向量 I_{agent} 拼接形成新特征 $I_{\text{target}} = \{i_{\text{target}}^n\}_{n=1}^N$ 。新的向量随后送入两个如图 4 所示的 MLP 中, 其中一个 MLP 输出结果是每个路点的分类概率 p_n , 另一个 MLP 则输出每个路点与真实目标点的偏移量 $(\Delta x_n, \Delta y_n)$ 。

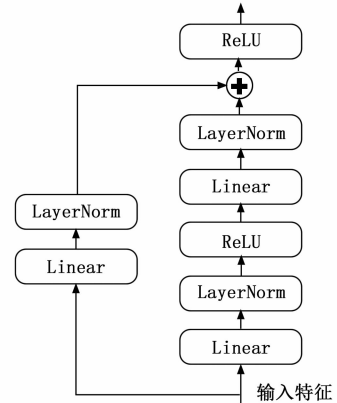


图 4 目标点预测多层感知器结构

在训练中该阶段的损失函数被定义为两部分:

$$L_{\text{target}} = L_{\text{cls}}(P, P_t) + L_{\text{reg}}(\Delta x, \Delta y, \Delta x_t, \Delta y_t) \quad (11)$$

式中, L_{cls} 为分类损失, 此处选择使用交叉熵损失函数; P 为预测值, P_t 为路点真实标签, 最靠近真实目标点的候选路点标签为 1, 其余路点为 0; L_{reg} 为回归损失, 此处使用 Huber 损失函数; $\Delta x, \Delta y$ 为偏移量的预测值, $\Delta x_t, \Delta y_t$ 为偏移量的真值。

2.3.2 轨迹生成

轨迹生成阶段的输入为目标车辆的交互特征 I_{agent} 与目标点预测阶段概率最大的前 m 个点, 这 m 个点的坐标将与其预测概率排列后拼接至 I_{agent} 后形成新的特征 I_{tar} 。在该阶段不直接输出一整条轨迹而是输出每个时间步的位移。

该阶段的具体实现是一个两层的 MLP 以及一个 Linear 层, 将特征 I_{tar} 转换为 $1 \times 2T_{\text{predict}}$ 大小的向量 P_{pre} , 定义如下:

$$P_{\text{pre}} = \{\Delta x_0, \Delta y_0, \Delta x_1, \Delta y_1, \dots, \Delta x_{T_{\text{pre}}}, \Delta y_{T_{\text{pre}}}\} \quad (12)$$

式中, T_{pre} 表示预测时间的步长, $\Delta x_{T_i}, \Delta y_{T_i}$ 表示 T_i 时刻到 T_{i+1} 时刻的位移。该阶段的损失函数 L_{tra} 同样使用 Huber 损失函数。

2.4 损失函数

所提预测网络的任务有两项, 对车辆行驶目标点的预测以及未来轨迹的预测, 总的损失函数被定义为:

$$L = \lambda_1 * L_{\text{target}} + \lambda_2 * L_{\text{tra}} \quad (13)$$

L_{target} 、 L_{tra} 分别为目标点预测和轨迹生成阶段的损失函

数, λ_1, λ_2 为两项任务的权重系数, 用来平衡网络对两项任务的侧重部分, 权重越大表示该项任务的准确性对总损失函数的影响越大。

3 实验结果与分析

3.1 轨迹预测数据集介绍

Argoverse 运动预测数据集^[20]是由 Argo AI 自动驾驶公司、卡内基梅隆大学以及佐治亚理工学院开源的大规模数据集。使用安装在车辆上的摄像头、激光雷达以及 GPS 采集数据, 经过处理后该数据集不仅包含了车辆轨迹的鸟瞰图, 同时提供了与每个场景下的车道线信息。图 5 展示了数据集中的一个场景, 图中实线表示了车辆的历史轨迹, 圆点为历史轨迹最后一个时刻的位置, 虚线表示道路中心线。

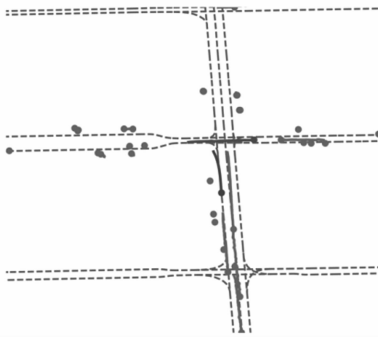


图 5 数据集场景示例

在数据集中提供车辆轨迹部分的数据类型见表 1。在 Argoverse 提供的每个场景中, 有一辆车类型被标记为 AGENT, 该车辆即为需要预测未来轨迹的目标车辆。车辆轨迹以 10 Hz 的频率采样, 总长度为 5 s, 其中 (0, 2] s 被用做可观测到的轨迹信息, 而 (2, 5] s 被当做预测轨迹地面真值。

表 1 Argoverse 数据类型

字段	说明	单位
TIMESTAMP	时间戳	s
TRACK_ID	轨迹编号	—
OBJECT_YPE	车辆类型	—
X	轨迹的 X 坐标	m
Y	轨迹的 Y 坐标	m
CITY_NAME	采集场景所在城市	—

3.2 数据预处理

根据预测网络的输入, 需要处理的数据有三项: 轨迹信息、车道线信息以及候选目标点。

3.2.1 轨迹信息归一化

选取在最后一可观测时刻 T_{obs} 距离目标车辆 30 m 内的其他 n 辆车辆轨迹与目标车辆的轨迹作为该场景的所有历史轨迹 $T_{his} = \{t_{his}^i\}_{i=1}^M$, 其中 $t_{his}^i = \{x_{-n}^i, y_{-n}^i, \dots, x_0^i, y_0^i\}$ 。并对历史轨迹 T_{his} 进行归一化处理, 使用目标车辆历史轨迹倒数第 4 个点与最后一个点计算目标车辆的航向角, 归一化处理将目标车辆的航向角指向 Y 轴正方向, 归一化处理减少了不

同场景下差异, 使得网络训练收敛更加快速、稳定。归一化处理将同样使用在接下来的车道线信息与目标点中。

3.2.2 车道线信息重采样

车道线信息首先通过目标车辆历史轨迹最后一点 $X_{T_{obs}}$ 获取目标车辆周围给定曼哈顿距离内的所有路段及其路点, 随后使用样条曲线插值的方式拟合原本的路点, 并以一定间隔 d 重新采样曲线上的点作为该路段中新的路点。

重新采样后路点内的路点数量可能会与原本的数量有较大的差距, 若某一路段内的路点数量超过设定阈值, 该路段将被分割为几个路点数目更少的几个路段。该分割方法保证了每个路段内包含的向量数目在设定阈值之下, 防止存在过长的特征被压缩, 导致信息丢失。

3.2.3 目标点采样

目标点采样过于稀疏会导致最终预测的目标点与真值相差过远, 反之如果采样过多则可能因目标点数过多使训练难以进行, 且最终预测的目标点会集中在某个区域, 不利于多模态的预测。最终选择的采样方式是首先确定与历史轨迹的最后一点最接近的路段 L_i , 使用深度搜索的方式前向搜索 L_i 的后续路段, 使用这种搜索方式将去除掉目标车辆不可能行驶到的路段。同时合理选择前述的样条曲线插值的间隔, 使得候选目标点数量在一定的范围内。

3.3 轨迹预测网络训练

轨迹预测网络训练使用的计算机软、硬件配置如表 2 所示。预测网络具体参数如下: 在历史轨迹编码模块中, UGRU 前向传播与后向传播输出维数均为 64。道路信息编码中的 MLP 模块中间层及输出的维数为 64, 聚合层数 l 为 3 层。交互模块中注意力模块的输出特征长度为 64。目标点预测及轨迹预测使用的 MLP 中间层维数为 64。场景的采样点个数 N 最大数量为 1 000, 选择目标点个数 m 为 6。

表 2 软硬件配置清

单软、硬件属性	参数
中央处理器(CPU)	Intel Xeno E5-2620v3 2.4GHz
物理内存	96 GB
图形处理器(GPU)	NVIDIA GeForce 2080Ti
操作系统 Ubuntu	18.04
深度学习框架	Pytorch 1.8.0

网络训练的各项参数如下: 训练批次大小 (batch size) 设置为 128; 训练次数 (epoch) 设置为 50; 学习率 (learning rate) 在开始训练时设置为 $1e-3$, 在经过 15 轮训练后, 每 5 轮训练学习率将随指数下降, 下降率设置为 0.9; 训练的优化器为 Adma, 其中 Adam 优化器的 $\beta_1 = 0.9, \beta_2 = 0.999$, 权重衰减率为 $1e-2$; 损失函数的两项权重分别选择为 $\lambda_1 = 0.5, \lambda_2 = 1$ 。

3.4 评价指标

目前对于预测轨迹质量常用的评价指标主要有以下两个: 最小平均位移误差 (minADE, minimum average distance error)、最小最终距离误差 (minFDE, minimum final

distance error)。

3.4.1 最小平均位移误差

minADE 为最佳预测轨迹按照每一个时间步与真值差值的二范数:

$$\min ADE = \min(\frac{1}{T} \sum_{i=1}^n \|Y_i - \hat{Y}_i\|_2), j = 1, \dots, N \quad (14)$$

式中, Y_i 表示 i 时刻的真实坐标, \hat{Y}_i 表示第 j 条预测轨迹在 i 时刻的坐标。该项指标在总体上评价了预测轨迹的准确性, 越小表示整体预测轨迹越接近真实值, 但由于最终结果对所有时间步取平均值, 无法反应某些预测点过于偏离真实值。

3.4.2 最小最终距离误差

minFDE 为最佳预测轨迹终点与真值的终点之间的二范数距离:

$$\min FDE = \min(\|Y^{T_j} - Y_j^{T_j}\|), j = 1, \dots, N \quad (15)$$

式中, Y^{T_j} 表示 T_j 时刻的真实坐标, $Y_j^{T_j}$ 表示第 j 条预测轨迹在 T_j 时刻的坐标。该项指标表示了对预测目标点的准确性, 越小表示预测的终点与真值越接近, 由于只计算了轨迹最终点的距离误差, 因此无法反应整条轨迹的准确性。

3.5 对比实验

将所提网络在 argoverse 数据集中的验证集于其他方法进行对比以验证所提算法在单模态预测精度上的提升, 试验结果如表 3 所示, 选择对比的方法有以下几种。

1) 恒速模型^[20]: 使用历史轨迹 1 秒至 2 秒间速度观测值的平均值作为预测时间段的速度, 利用运动学模型计算未来轨迹。

2) LSTM-social^[20]: 使用 LSTM 作为编码器和解码器, 在输入特征中加入了车辆间的交互特征。

3) LSTM-map^[20]: 同样使用 LSTM 作为编码器和解码器, 与方法 (2) 的区别在于输入特征加入道路信息。

4) VectorNet^[13]: 将道路信息和历史轨迹信息都视为向量, 使用自注意力机制同时获取道路与车辆、车辆与车辆之间的交互信息。

5) SGANAM^[21]: 以社会对抗生成网络为基础, 加入使用注意力机制的社会注意力模块以及历史注意力模块学习车辆间的交互作用, 使用车道卷积操作^[12]提取道路特征。

6) GA-MTP^[22]: 将道路信息建模为图模型, 使用车道卷积操作提取道路特征, 对历史轨迹信息使用 LSTM 进行编码, 随后使用注意力机制获取车辆以及道路交互。

表 3 验证集模型性能对比

方法	minADE/m	minFDE/m
恒速模型 ^[20]	3.11	7.02
LSTM-social ^[20]	2.29	5.22
LSTM-map ^[20]	2.25	4.67
VectorNet ^[13]	1.66	3.67
SGANAM ^[21]	1.55	3.57
GA-MTP ^[22]	1.55	3.55
本文方法	1.45	3.21

通过表 3 的对比可以看出所提轨迹预测算法预测精度相较于恒速模型有明显的提升, 两项指标分别提升 53.3% 与 54.2%, 其余深度学习模型的预测精度由于恒速模型, 因此使用深度学习的方法显著提升了预测的准确性。对比 LSTM-social 与 LSTM-map, 可以看相较于交互信息, 加入车道信息对于预测准确性有更大的提升。所提方法对比 LSTM-map, 两项指标分别提升了 35.5% 与 31.26%, 证明了所提方法中交互模块对周围环境信息有更强的提取能力。VectorNet 使用了较为简单的注意力模块提取交互信息, 而 SGANAM 与 GA-MTP 均使用了车道卷积操作提取环境信息, 因此有相近的预测精度且精度优于 VectorNet。所提轨迹预测算法在指标上均优于其他使用注意力机制的方法, 本文方法与 GA-MTP 相比在指标上分别提升了 6.5%、9.57%, 在 minFDE 指标上有更大的提升, 证明本文方法所提的交互模块充分考虑了目标车辆与周围车辆以及车道线间存在的交互, 并且目标点预测模块进一步提升了模型的预测精度, 且对最终点预测有更大的提升。

除了历史信息, 预测时长也是影响预测精度的因素之一, 此处选择恒速模型、SGANAM 以及本文方法在不同预测时长下进行对比, 结果如图 6 所示, 图中展示了 3 种方法的平均距离误差以及最终距离误差。

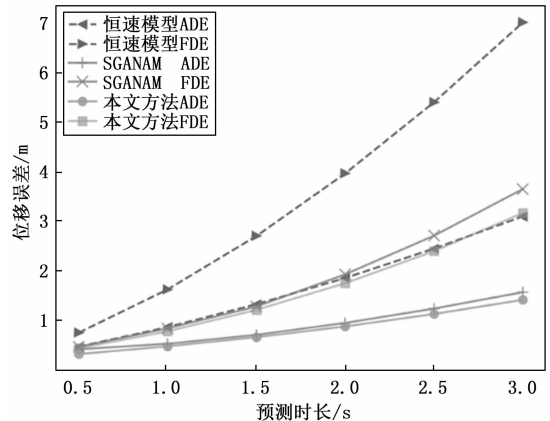


图 6 不同预测时长下的预测误差

在 0.5 s 预测时长下, 3 种方法的预测误差相近。随着预测时间的增加, 3 种方法的预测误差都随之增加, 预测准确性下降。恒速模型由于不能识别驾驶员的意图与周围环境, 因此预测误差随预测时长将大幅增加, 两种深度学习模型的误差增长明显小于恒速模型。SGANAM 与本文方法使用了不同深度学习模型学习驾驶员与周围的交互及驾驶员的驾驶意图, 因此在误差增长趋势上呈现一致, 且所提方法在所有预测时长的误差均小于 SGANAM, 进一步证明所提方法的有效性。

3.6 消融实验

为了验证各个模块在预测中的作用, 消融实验将逐步添加不同的模块, 从而定量分析每个模块在网络结构中对性能提升的占比。

表 4 为消融实验的结果, 其中 A2A 是交互模块中用于提取车辆间交互信息的模块; MAP 表示交互模块中的 L2L、L2L、A2L 三个模块, 这 3 个模块帮助网络提取道路信息和交通流信息; Target 表示加入目标点预测。

表 4 消融实验

A2A	Map	Target	minADE/m	minFDE/m
✓			1.55	3.47
	✓		1.49	3.33
✓	✓		1.47	3.29
✓	✓	✓	1.45	3.21

从实验结果可以看出相较于只使用车辆间交互信息的 A2A 模块, MAP 模块加入了道路信息, 将帮助预测网络更好地学习驾驶的未来意图, 两项指标下降了 3.8% 与 4.0%, 预测网络性能在 minFDE 上有明显的下降。在单独使用 L2L、A2L、L2L 这 3 个模块时由于该模块通过各个道路点传播了交通流信息, 该信息隐含了车辆间可能的交互信息, 所以在预测精度上与同时使用地图信息和车辆交互信息相近, 但是直接提取交互信息的 A2A 提升了网络预测最终目标点的能力, 使得两项指标分别下降了 1.3% 与 1.2%。进一步加入目标点预测模块提升了预测网络的整体预测能力, 预测轨迹的平均误差和最终误差都有所下降, 分别下降了 1.3% 与 2.4%, 证明加入目标点预测对于最终点的预测有更大的提升。

4 结束语

本文提出了基于交互性的轨迹预测网络, 使用轨迹信息以及车道线信息作为输入, 预测网络主要包括了 3 个模块: 编码模块、交互模块、预测模块。历史轨迹信息以及车道线信息经过编码后将送入到基于注意力机制的交互模块, 该模块用于提取车辆和道路之间的交互信息。最后经预测模块输出预测轨迹, 在预测模块中使用目标点预测的方式增加了预测的准确性。通过在 Argoverse 公开数据集上的实验证明了算法在预测准确性上的提高, minADE 为 1.45 m, minFDE 为 3.21 m。目前所提算法聚焦于单条未来轨迹的预测, 无法预测驾驶员的多种驾驶意图, 后续的工作将以多模态预测为主。

参考文献:

- [1] 王珂, 王艳阳, 黄秋实, 等. 基于高级运动模型轨迹预测的不确定性分析 [J/OL]. 武汉大学学报 (工学版): 1-14 [2023-04-17]. <http://kns.cnki.net/kcms/detail/42.1675.T.20220913.1509.002.html>.
- [2] 谢枫, 李永乐, 苏致远, 等. 一种城市交叉路口转弯车辆轨迹预测方法 [J]. 军事交通学院学报, 2019, 21 (11): 78-83.
- [3] XIE G, GAO H, QIAN L, et al. Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using in-

- teractive multiple models [J]. IEEE Transactions on Industrial Electronics, 2017, 65 (7): 5999-6008.
- [4] KIM B D, KANG C M, KIM J, et al. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network [C] //2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017: 399-404.
- [5] 温惠英, 张伟罡, 赵胜. 基于生成对抗网络的车辆换道轨迹预测模型 [J]. 华南理工大学学报 (自然科学版), 2020, 48 (5): 32-40.
- [6] 季学武, 费聪, 何祥坤, 等. 基于 LSTM 网络的驾驶意图识别及车辆轨迹预测 [J]. 中国公路学报, 2019, 32 (6): 34-42.
- [7] DAI S, LI L, LI Z. Modeling vehicle interactions via modified LSTM models for trajectory prediction [J]. IEEE Access, 2019, 7: 38287-38296.
- [8] 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.
- [9] 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.
- [10] LIN L, LI W, BI H, et al. Vehicle Trajectory Prediction Using LSTMs With Spatial-Temporal Attention Mechanisms [J]. IEEE Intelligent Transportation Systems Magazine, 2021, 14 (2): 197-208.
- [11] KHANDELWAL S, QI W, SINGH J, et al. What-if motion prediction for autonomous driving [J/OL]. arXiv, 2020: 1-16. 2020.08.24. <https://arxiv.org/abs/2008.10587>.
- [12] LIANG M, YANG B, HU R, et al. Learning lane graph representations for motion forecasting [C] //European Conference on Computer Vision. Springer, Cham, 2020: 541-556.
- [13] GAO J, SUN C, ZHAO H, et al. Vctornet: 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.
- [14] HUANG Z, MO X, LV C. Multi-modal motion prediction with transformer-based neural network for autonomous driving [C] //2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022: 2605-2611.
- [15] GILLES T, SABATINI S, TSISHKOU D, et al. Home: Heatmap output for future motion estimation [C] //2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, 2021: 500-507.
- [16] FANG L, JIANG Q, SHI J, et al. TpNet: Trajectory proposal network for motion prediction [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 6797-6806.

(下转第 118 页)