

清 华 大 学

综 合 论 文 训 练

题目：车联网分布式训练仿真平台搭
建与优化算法设计

系 别：电子工程系

专 业：电子信息科学与技术

姓 名：谢博文

指导教师：牛志升 教授

2022 年 6 月 3 日

中文摘要

未来智能网联汽车具备强大的数据采集、存储、计算与通信能力，同时其自动驾驶技术需要大量应用机器学习算法。在车联网中部署面向机器学习的分布式训练任务，既可以充分利用智能网联车辆的数据与计算资源，又可以有效提高其自动驾驶性能，具有实际应用价值。

已有研究大多关注于如何在给定资源约束下提高分布式训练的收敛速度。然而，这些工作主要关注准静态场景，即可调度的终端设备集合在分布式训练过程中保持不变，这与车联网场景的动态特性不符。由于车辆的移动性以及路侧单元信号覆盖范围的有限性，车联网分布式训练过程中可调度的车辆集合随时间变化。此外，交通环境与车流量对于每回合可调度车辆数也有直接影响。基于车联网场景的特殊性，本文针对未来车联网中的“车路协同”分布式训练场景进行研究，旨在提出有效的优化算法，在总训练时长约束下提高分布式训练的收敛速度。

具体来说，本文工作分为以下三个部分。首先，本文针对车联网分布式训练的典型场景进行建模，该场景中一个路侧单元统筹分布式训练过程，其覆盖范围下参与训练的车辆到达服从泊松过程，且匀速行驶。而后，基于该场景模型，本文对每回合成功完成模型上传的车辆数进行理论分析，发现其服从泊松分布。利用该结论，本文针对分布式训练回合时长 T 与本地训练迭代次数 H 建立优化问题，以最大化训练收敛速率，并结合循环迭代法与基于偏微分的二分搜索法，设计了一个移动性感知的 $H-T$ 联合优化算法。最后，本文设计并实现了一个车联网分布式训练仿真平台，基于毫米波波束选择和车辆轨迹预测任务，对所提算法进行了仿真验证。

关键词：车联网；分布式训练；机器学习；边缘计算；仿真平台

ABSTRACT

Intelligent connected vehicles (ICV) have promising data sampling, storage, computing and communication capabilities, while their autonomous driving needs machine learning (ML) algorithms. The deployment of ML-oriented distributed training tasks in the Internet of vehicles (IoV) can not only make full use of the data and computing resources of ICV, but also effectively improve their autonomous driving performance, which is of great practical application value.

Most of the existing researches focus on how to improve the convergence speed in distributed training given specific resource constraints. However, these works mainly study quasi-static scenarios, where the set of schedulable devices remains unchanged in the distributed training process, which is inconsistent with the dynamic characteristics of the IoV scenarios. Due to the mobility of vehicles and the limitation of signal coverage of road side units (RSU), the set of schedulable devices in the IoV will be changing all the time in the distributed training process. In addition, traffic environment and traffic flow also have a direct impact on the number of schedulable vehicles in each round. Based on the characteristics of the IoV scenarios, we study the distributed training scenarios of "vehicle-road collaboration" in the future IoV, aiming to propose effective optimization algorithms and improve the convergence speed of the distributed training.

The contributions of this paper are as follows. Firstly, we model a typical IoV distributed training scenario. In this scenario, an RSU coordinates the distributed training process, and the vehicles participating in the training arrive at the road following a Poisson process and drive at a uniform speed. Based on the scenario model, we analyze the probability distribution of the number of vehicles that successfully complete model uploading in each round, and find that it follows a Poisson distribution. Based on this conclusion, we formulate an optimization problem about the round duration T and the local iteration number H to maximize the convergence speed of distributed training. To solving this problem, we propose a mobility-aware H - T joint optimization algorithm by combining the iterative algorithm and the partial-differential-based binary search method. Finally, we design and implement an IoV distributed training simulation platform. Based on the millimeter wave beam selection and the driving trajectory

prediction tasks, simulation experiments verify the rationality and effectiveness of the proposed optimization algorithm.

Keywords: Internet of vehicles; distributed training; machine learning; edge computing; simulation platform

目 录

插图和附表索引	VII
第 1 章 引言	1
1.1 研究背景与动机	1
1.2 研究现状	2
1.3 论文研究思路与结构安排	3
1.3.1 研究思路	3
1.3.2 结构安排	4
第 2 章 车联网分布式训练场景建模	5
2.1 导言	5
2.2 交通道路及车流过程建模	5
2.3 分布式训练相关要素	6
2.3.1 目标任务	6
2.3.2 神经网络模型	7
2.3.3 数据集	8
2.3.4 损失函数	9
2.4 车联网中的分布式训练框架	10
2.4.1 模型下发	12
2.4.2 本地训练	12
2.4.3 模型上传	13
2.4.4 模型聚合	14
2.5 本章小结	14
第 3 章 车联网分布式训练优化算法设计	16
3.1 导言	16
3.2 单回合成功车数的概率分布	17
3.3 优化问题建模	23
3.4 优化算法设计	24
3.5 本章小结	27

第 4 章 仿真平台搭建及算法验证	29
4.1 导言	29
4.2 仿真平台设计	29
4.2.1 交通道路及车流仿真模块	30
4.2.2 分布式训练回合表生成模块	32
4.2.3 目标任务调用模块	34
4.2.4 本地数据集配给模块	37
4.2.5 分布式训练模块	37
4.2.6 模型性能测试模块	38
4.3 仿真平台总体架构与程序运行流程	39
4.4 仿真实验结果	41
4.4.1 典型参数下的分布式训练过程图	41
4.4.2 分布式训练回合时长 T 的影响	42
4.4.3 本地训练迭代次数 H 的影响	43
4.4.4 优化问题建模有效性与普适性检验	44
4.4.5 算法性能验证	47
4.5 本章小结	48
第 5 章 总结与展望	49
5.1 研究总结	49
5.2 未来展望	50
参考文献	51

插图和附表索引

图 2.1	交通道路及车流过程	5
图 2.2	深度学习推断算法流程示意图	7
图 2.3	车联网中的分布式训练流程图	11
图 3.1	$T \geq T'$ 时单回合成功车数理论分析	18
图 3.2	$T < T'$ 时单回合成功车数理论分析	20
图 3.3	通过仿真验证定理 3.1.....	22
图 3.4	两组典型参数设置下, $g(H, T)$ 在 $H-T$ 平面上的三维曲面图	25
图 4.1	仿真平台各模块间关系图	30
图 4.2	SUMO 图形用户界面	32
图 4.3	仿真平台总体架构	40
图 4.4	针对波束选择任务的分布式训练过程图	41
图 4.5	针对轨迹预测任务的分布式训练过程图	42
图 4.6	波束选择任务下的最优模型性能指标与优化问题 $P4-1$ 的目标函数 $g(H, T)$ 随回合时长 T 的变化关系	45
图 4.7	波束选择任务下的最优模型性能指标与优化问题 $P4-1$ 的目标函数 $g(H, T)$ 随本地训练迭代次数 H 的变化关系	46
图 4.8	轨迹预测任务下的最优模型验证损失函数与优化问题 $P4-1$ 的目标函数 $g(H, T)$ 间的变化关系.....	46
图 4.9	波束选择任务下优化算法效果对比	47
图 4.10	轨迹预测任务下优化算法效果对比	48
表 2.1	车联网中分布式训练相关符号	11
表 4.1	分布式训练典型参数设置	43
表 4.2	波束选择任务分布式训练受 T 影响.....	43
表 4.3	轨迹预测任务分布式训练受 T 影响.....	44
表 4.4	波束选择任务分布式训练受 H 影响.....	44
表 4.5	轨迹预测任务分布式训练受 H 影响.....	45

主要符号对照表

ADE	平均偏移误差 (Average Displacement Error)
C-V2X	基于蜂窝网的车与万物通信 (Cellular Vehicle-to-Everything)
DNN	深度神经网络 (Deep Neural Network)
DSRC	专用短程通信 (Dedicated Short Range Communication)
FDE	终点偏移误差 (Final Displacement Error)
ICV	智能网联车辆 (Intelligent and Connected Vehicle)
ML	机器学习 (Machine Learning)
OFDM	正交频分多址 (Orthogonal Frequency Division Multiplexing)
RSU	路侧单元 (Road Side Unit)
SGD	随机梯度下降 (Stochastic Gradient Descent)
V2X	车与万物通信 (Vehicle-to-Everything)
IoV	车联网 (Internet of Vehicles)
i.i.d.	独立同分布 (Independent and Identically Distributed)
$\mathbb{E}[\cdot]$	数学期望
$\mathbb{P}[\cdot]$	事件概率
\mathbb{N}	非负整数集合
\mathbb{R}	实数集合
\mathbb{R}^+	正实数集合
\mathcal{D}	数据集
\mathcal{B}	批数据
\mathcal{M}	车辆集合
$\mathcal{T}_{\text{traf}}$	车流信息表
\mathcal{T}_{rnd}	分布式训练回合表
L	道路长度
λ	车辆到达率
v	车速
w	神经网络模型参数
T_{sim}	分布式训练总时长
T	分布式训练回合时长
H	本地训练迭代次数

τ^{cm}

通信时延

τ^{cp}

计算时延

第 1 章 引言

1.1 研究背景与动机

车联网是 5G、人工智能等新一代信息通信技术在汽车、交通行业的重要应用场景^[1]。车联网技术以专用短程通信 (Dedicated Short Range Communication, DSRC)^[2]和基于蜂窝网的车与万物通信 (Cellular Vehicle-to-Everything, C-V2X)^[3]为技术基础,可以形成“人-车-路-云”协同连接的通信网络体系,实现交通运输场景下的万物智联。而随着我国进入“十四五”时期,车联网技术将在我国得到大规模部署应用^[4]。

在未来车联网应用场景中,智能网联车辆 (Intelligent and Connected Vehicle, ICV) 将得到广泛普及,成为车联网发展的基石^[5]。它们配备多种类型的传感器,能对周围环境进行探测与感知,实时产生相关数据,同时其车载处理器具备良好的计算与存储能力,能完成目标检测、轨迹预测、导航决策、运动控制等一系列自动驾驶任务。此外,智能网联车辆还能通过 V2X (Vehicle to Everything) 技术和路侧单元 (Road Side Unit, RSU) 进行相互通信,实现“车路协同”下的智能交通。

智能网联车辆同时具备良好的数据生成、存储、计算与通信能力,这与分布式机器学习技术的要求不谋而合,因此智能网联车辆十分适合作为分布式机器学习技术的实现载体与应用对象。另一方面,机器学习技术作为目前学界与业界的研究热点之一,在自动驾驶相关领域上得到了广泛应用^[6],且基于机器学习技术的相关算法在性能上往往优于传统算法,因此在未来车联网中,智能网联车辆的自动驾驶同样离不开机器学习。可以说,智能网联车辆与分布式机器学习是一种相辅相成的关系。

根据上述原因,基于智能网联车辆实现分布式机器学习技术在车联网中的部署具有很高的研究价值与广阔的应用前景。而相比于分布式机器学习中的分布式推断,本文更关注对分布式训练的研究。其中,分布式训练指的是将一个神经网络模型借由若干终端设备的本地数据和运算资源进行训练,而分布式推断指的是将训练完的神经网络模型部署到若干终端设备上,由这些终端设备联合执行该模型的推断任务。

事实上,车联网中的分布式训练与一般意义上的分布式训练存在一定差异。一方面,一般意义上的分布式训练的调度中心所能调用的终端设备集合是固定

的，而车联网中的分布式训练的调度中心为路侧单元，它所能调用的终端设备集合由其信号覆盖范围内的所有智能网联车辆所组成，由于车辆的移动性，该终端设备集合将随时间动态变化。另一方面，一般意义上的分布式训练的调度中心可以访问其终端设备的数据，甚至终端设备的数据就是由调度中心从总数据集中选取一部分分配的，该总数据集是固定不变的；而车联网中智能网联车辆携带的数据由自身传感器产生，路侧单元一般不能直接访问车辆的原始数据，且由于车辆的移动性，不同时刻的总数据集将发生变化。

此外，为了反映车联网分布式训练的实际应用价值，本文的仿真平台分别针对毫米波通信中的波束选择任务和自动驾驶中的轨迹预测任务，实现了对它们相应的神经网络模型的分布式训练仿真。之所以选择这两个目标任务，一方面是因为它们分别是未来车联网中 V2X 通信与车辆自动驾驶的关键问题，另一方面则是因为它们所需的训练数据都可以由车辆通过车载传感器实时采集。随着环境的动态变化，车辆采集数据的特征分布也在发生变化，而利用这些实时数据进行在线分布式训练，就可以让神经网络模型适应环境变化，从而在不同的环境下都能保持良好的性能表现。这也体现了车联网分布式训练相对于准静态场景分布式训练的优势。

总而言之，车辆的移动性导致车联网分布式训练中的可调度终端设备集合以及训练数据的特征分布随时间动态变化，而如何利用好这一特性在车联网中高效地进行分布式训练，就是我们课题研究的目标。

1.2 研究现状

针对与车联网分布式训练类似的边缘计算中的联邦学习问题，已有较多研究工作。根据是否在每一回合调度所有终端参与训练，可以将它们分为两类：全体调度式训练与选择调度式训练。

对于全体调度式训练，文献[7] 从分布式梯度下降收敛界的理论分析出发，提出了针对本地训练更新次数与模型参数聚合频率的联合优化算法，以在给定分布式训练资源开销下最小化全局模型性能损失。文献[8] 同时考虑设备计算时延与通信开销之间的矛盾以及分布式训练收敛速度与设备能耗之间的矛盾，通过优化本地更新精度、本地计算频率以及设备通信速率，以在设备总能耗、分布式训练总时长以及最终模型精度三者间进行平衡折衷。文献[9] 在端-边-云三级联邦学习体系下提出 HierFAVG 算法，进一步提高分布式训练中通信计算平衡折衷的优

化性能。

对于选择调度式训练，文献[10] 在移动边缘计算框架下提出了 FedCS 协议，根据移动终端的计算能力与信道条件在有限资源约束的情况下基于贪心算法调度各终端参与分布式训练，以此最大化回合延时约束下的终端调度数，从而提高分布式训练的模型收敛速度。文献[11] 为了在保证训练速度的前提下减少设备能耗，提出基于终端设备计算能力与信道状态的高能效的终端调度与带宽分配策略，平衡能效与分布式训练性能的折衷关系。文献[12] 在总训练时长受约束的车联网联邦学习场景下通过联合优化终端调度与频谱分配策略来最大化模型训练精度。

上述工作在边缘计算的联邦学习场景下，对于终端本地计算与端边通信产生的时延与能耗代价进行了相关分析讨论，并基于神经网络模型特性、设备算力资源、无线信道条件等因素进行了相关优化算法的设计，从而提高分布式训练中的模型收敛速度。然而，上述工作的研究场景与我们的车联网分布式训练场景存在一定差异。上述工作中可被边缘服务器调度的终端设备集合不发生改变，即使在选择调度式训练中一个终端设备在某一回合未被调度，其在之后的回合中仍有可能被调度。而在我们的车联网分布式训练场景下，由于车辆的移动性以及路侧单元有限的信号覆盖范围，能被路侧单元调度的车辆集合随时间推移将发生改变。在一次分布式训练过程中，一辆车一旦驶出训练区域，那么之后便不再参与分布式训练。此外，交通环境和车流量大小也会显著影响各个时刻可调度的车辆数量。因此，车联网场景下的分布式训练仍有许多可待研究的问题。

1.3 论文研究思路与结构安排

1.3.1 研究思路

本文针对车联网中的“车路协同”分布式训练场景，研究如何提高分布式训练中神经网络模型的收敛速度，从而在车联网中高效地执行分布式训练任务。为此，本文采取如下研究思路：

1. 首先，对车联网分布式训练场景进行数学建模，通过合理的抽象与简化使得该场景模型在准确刻画车联网分布式训练过程的同时便于理论分析与仿真验证。
2. 基于上述场景，对车联网分布式训练的相关要素进行理论分析，研究影响分布式训练中神经网络模型收敛速度的主要因素，并对如何优化车联网分布式训练的问题进行建模。

3. 针对优化问题，设计相应的优化算法，确保算法能在一定误差允许范围内得到优化问题的近似最优解。
4. 进行仿真实验，验证本文理论分析的正确性、优化问题建模的有效性与普适性，以及优化算法设计的合理性与有效性。

1.3.2 结构安排

基于上述研究思路，本文内容结构安排如下：

第2章对车联网中的“车路协同”分布式训练场景进行数学建模。该场景模型包含交通道路及车流过程、分布式训练相关要素以及车联网分布式训练框架等三个部分，第2章将对它们进行分节介绍。此外，分布式训练相关要素又包含了目标任务、神经网络模型、数据集和损失函数等要素，车联网分布式训练框架又包含模型下发、本地训练、模型上传、模型聚合等环节，因此在对应章节中还会针对这些要素/环节进行分小节详细介绍。该场景模型为第3章和第4章中的工作起到了铺垫作用。

第3章从对影响分布式训练模型收敛速度的主要因素的理论分析出发，对于车联网分布式训练相关的优化问题进行建模，并为该优化问题设计了一个优化算法。该算法针对车联网分布式训练的关键参数进行优化求解，最大化优化问题中的目标函数，从而提高分布式训练模型收敛速度。

第4章针对第2章中的车联网分布式训练场景以模块化的方式设计并实现了一个仿真平台。利用该仿真平台，通过在不同目标任务和不同仿真参数设置下的多组仿真实验，验证了第3章中的优化问题建模的合理性与有效性，并验证了相应优化算法确实可以有效提高分布式训练模型收敛速度。

第5章总结本文研究成果，并讨论未来研究方向。

第 2 章 车联网分布式训练场景建模

2.1 引言

为了对车联网中的“车路协同”分布式训练进行研究，我们首先需要对该场景进行建模。在本章中，我们将按顺序对交通道路及车流过程、分布式训练相关要素以及车联网分布式训练框架进行建模。其中，分布式训练相关要素包含了目标任务、神经网络模型、数据集和损失函数等要素，车联网分布式训练框架包含模型下发、本地训练、模型上传、模型聚合等环节。

基于本章构建的车联网分布式训练场景模型，本文后续的章节中将对车联网分布式训练相关的优化问题进行建模，设计相关优化算法，搭建仿真平台，进行仿真实验，并通过实验结果验证优化问题建模与优化算法设计的合理性与有效性。因此，本章的场景建模对于后文的研究而言至关重要。

2.2 交通道路及车流过程建模

我们首先对于车联网中的交通道路进行建模。考虑到车辆在诸如十字路口、三岔路口、弯道等较为复杂的交通道路上需要高负荷执行自动驾驶相关算法以保障行驶安全，难以给优先级相对较低的分布式训练任务提供充足的计算资源，因此我们仅考虑简单的直路交通环境，即将我们的交通道路模型建模为一条笔直道路。在现实生活中，直路既可以是单向行驶的也可以是双向行驶的，同时对于直路的一个方向可能存在若干条车道。但对于分布式训练而言，无论是双向多车道的直路，还是单向单车道的直路，它们对训练过程的影响实际上是等效的。因此，为了便于后续的理论分析，我们采用简单的单向单车道直路模型作为我们的交通道路模型。该模型仅需用道路长度 L 这一个参数即可完全确定。

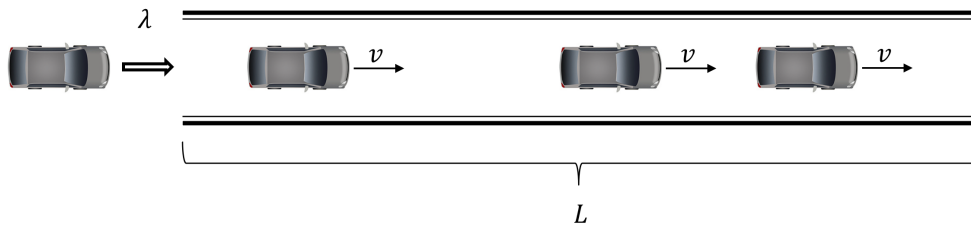


图 2.1 交通道路及车流过程

基于上述交通道路模型，我们对车流过程进行建模。车流过程由车辆到达过程与行驶过程所确定。为了能在充分反映现实情况的同时便于理论分析，我们将车辆到达过程建模为一个泊松过程，其到达率为 λ 。由于相同的原因，我们将车辆行驶过程建模为定速行驶，即所有车辆在直路上以相同的速度 v 匀速行驶。对于这样的车流过程模型，仅需车辆到达率 λ 和车速 v 这两个参数即可完全确定。

至此，我们完成了对车联网分布式训练场景下交通道路及车流过程的建模，模型示意图如图2.1所示。虽然该模型十分简单，但它能仅以三个参数来描述交通道路与车流过程的信息，且能很好地近似真实场景中最为常见的直路情况，同时简化了理论分析，因此我们最终采用了该模型。

2.3 分布式训练相关要素

在本节中，我们对车联网分布式训练中需要确定的相关要素进行刻画。它们包括目标任务、神经网络模型、数据集以及损失函数等。

2.3.1 目标任务

在机器学习技术的实际应用中，首先需要确定目标任务。目标任务决定了一个机器学习算法要解决的具体问题。更确切地说，目标任务规定了输入数据和输出结果的取值集合，要求配备至少一种推断算法以将合法的输入数据映射为对应的输出结果，并配有一个或多个评价指标以对推断算法的性能做出评价。以手写数字识别这一目标任务为例，它要做的就是根据一张手写数字图片，判断图片上的数字是 $\{0, \dots, 9\}$ 中的哪一个数字。换句话说，将目标任务选定为手写数字识别，就约束了输入数据必须是一张图片，而输出结果必须是 $\{0, \dots, 9\}$ 中的一个整数。该目标任务要求人们设计推断算法将作为输入数据的图片映射为对应的数字，并以映射的准确率作为任务的评价指标。

为后续讨论方便，我们先对目标任务进行规范化表示。一个目标任务由以下几个部分组成：

1. 输入数据的取值集合 \mathcal{X} ：对于该目标任务下的任意输入数据 \mathbf{x} ，都有 $\mathbf{x} \in \mathcal{X}$ 。一般情况下，输入数据可以以张量形式来表示，且 \mathcal{X} 中所有元素的张量形状均应相同。
2. 输出结果的取值集合 \mathcal{Y} ：对于该目标任务下的任意输出结果 \mathbf{y} ，都有 $\mathbf{y} \in \mathcal{Y}$ ，一般情况下，输出结果同样可以以张量形式来表示，且 \mathcal{Y} 中所有元素的张量形状均应相同。 $\forall \mathbf{x} \in \mathcal{X}$ ，有且仅有一个 $\hat{\mathbf{y}} \in \mathcal{Y}$ 与之对应，我们将这样的

\hat{y} 称为 \mathbf{x} 的标签，并将从 \mathbf{x} 到 \hat{y} 的映射关系记为 $\hat{f}(\cdot)$ 。

3. 推断算法 $f(\cdot)$: 一个目标任务可以采用若干种合法的推断算法，我们将所有合法的推断算法所构成的集合记为 \mathcal{F} 。 $\forall f(\cdot) \in \mathcal{F}$, $f(\cdot)$ 满足对于 \mathcal{X} 中的任意元素 \mathbf{x} , $f(\mathbf{x})$ 均有定义，且 $f(\mathbf{x}) \in \mathcal{Y}$ 。在分类问题中， $f(\cdot)$ 的选取应尽可能地满足 $f(\mathbf{x}) = \hat{f}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$ 。而在回归问题中， $f(\cdot)$ 的选取应尽可能地最小化 $|f(\mathbf{x}) - \hat{f}(\mathbf{x})|, \forall \mathbf{x} \in \mathcal{X}$ 。可以说， $\hat{f}(\cdot)$ 就是最优的推断算法。
4. 评价函数 $Q(\cdot; \cdot)$: 目标任务需要对其选用的推断算法的性能做出评价，得到评价指标，而评价函数 $Q(\cdot; \cdot)$ 即为将一个推断算法 $f(\cdot)$ 和一个评价数据集 \mathcal{D}_{eval} 映射为对应评价指标 $Q(f; \mathcal{D}_{eval})$ 的函数。可以说，评价指标 $Q(f; \mathcal{D}_{eval})$ 指的就是以评价数据集 \mathcal{D}_{eval} 为标准，衡量推断算法 $f(\cdot)$ 性能表现的指标。评价指标既可以是一个实数，也可以是一个向量，这取决于目标任务的具体要求。

对于车联网中的分布式训练场景，只有给定了目标任务，我们才能选取合适的神经网络模型、数据集、分布式训练算法等其他要素，最终将分布式训练应用于车联网的环境中。在第4章中，我们将针对毫米波通信中的波束选择任务以及车辆自动驾驶中的轨迹预测任务搭建仿真平台并进行仿真实验。

2.3.2 神经网络模型

确定了目标任务后，我们需要为其设计推断算法，而所谓的深度学习，就是在推断算法中应用了深度神经网络模型（简称为神经网络）这一强大工具。神经网络相当于一个结构化的函数 $f_N(\cdot)$ ，它能将一个输入张量 \mathbf{X} 通过若干层神经网络层的处理，映射为一个输出张量 \mathbf{Y} 。如图2.2所示，在一个应用了神经网络模型的推断算法 f_I 的执行过程中，原始输入数据 $\mathbf{x} \in \mathcal{X}$ 经过预处理操作 f_P 得到神经网络的输入张量 $\mathbf{X} = f_P(\mathbf{x})$ ，神经网络 f_N 将该输入张量通过若干层神经网络层的处理映射为输出张量 $\mathbf{Y} = f_N(\mathbf{X})$ ，最后通过一个后处理操作 f_B 得到推断算法的输出结果 $\mathbf{y} = f_B(\mathbf{Y})$ 。这一推断过程也可直接表示为 $\mathbf{y} = f_I(\mathbf{x})$ 。

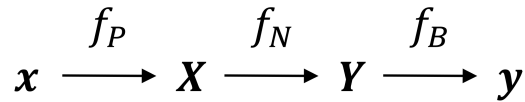


图 2.2 深度学习推断算法流程示意图

对于一个给定的神经网络模型 f_N ，它具有特定的网络结构 \mathbf{A}_N 和相应的网络参数 \mathbf{w}_N 。其中， \mathbf{w}_N 也被称为网络权重，一般以张量的形式表示，其结构由

A_N 所决定。设计一个神经网络模型，指的是设计一个网络结构 A_N ；而对一个神经网络模型进行训练，指的是通过机器学习算法让具有特定网络结构 A_N 的神经网络模型 f_N 学到合适的网络参数 w_N ，以实现准确合理的推断算法。

由于 w_N 的张量结构可以体现神经网络模型 f_N 的网络结构 A_N ，因此我们也可以直接用 w_N 表示一个神经网络。为了表示方便，在后面的章节中，对于一个以 w 为模型参数的神经网络 f_N ，我们直接以 w 表示之，而只有在以该神经网络对输入张量进行处理时，才以 f_N 表示之。

2.3.3 数据集

为了对神经网络模型进行训练，我们需要基于目标任务选取合适的数据集。一个数据集由许多数据样本所构成。我们将数据集记为 D ，其大小记为 $D = |D|$ 。对于数据集中的任意一个数据样本 $d_i \in D, i = 0, \dots, D - 1$ ，其包含一个输入-输出对，即 $d_i = \{x_i, y_i\}$ ，其中 x_i 和 y_i 应满足 $x_i \in \mathcal{X}, y_i \in \mathcal{Y}$ ， \mathcal{X} 和 \mathcal{Y} 分别指代目标任务下的输入数据取值集合和输出结果取值集合。我们将数据集的样本输入构成的集合记为 $\mathcal{X}_D = \{x_i\}_{i=0}^{D-1}$ ，样本输出构成的集合记为 $\mathcal{Y}_D = \{y_i\}_{i=0}^{D-1}$ ，显然有 $\mathcal{X}_D \subseteq \mathcal{X}, \mathcal{Y}_D \subseteq \mathcal{Y}$ 。

对于一个给定的目标任务，可能有许多种可供选择的数据集，如图像分类任务下常用的 ImageNet^[13]、CIFAR-100^[14]、MNIST^[15]，轨迹预测任务下常用的 Argoverse^[16]、nuScenes^[17]、ApolloScape^[18] 等。数据集的选取在很大程度上影响了经过训练得到的神经网络模型的性能表现。一般而言，数据集大小 D 越大，数据样本类型越多样，数据样本分布越接近实际应用场景中的数据分布，那么使用该数据集训练出来的神经网络模型的性能就越好。

在对一个神经网络模型 f_N 进行训练时，数据集 D 一般会被划分为三个不相交的子集：训练集 D_{train} 、验证集 D_{val} 和测试集 D_{test} 。在模型训练的过程中，我们在训练集 D_{train} 上对神经网络模型 f_N 应用以梯度下降法为典型代表的训练算法更新其网络参数 w_N ，在验证集 D_{val} 上以评价指标 $Q(f; D_{val})$ 评估模型训练后的性能表现，并以此为参照来调整模型训练的超参数，让确定网络架构 A_N 下的神经网络模型 f_N 在验证集 D_{val} 上的性能表现最优。最后在测试集 D_{test} 上以评价指标 $Q(f; D_{test})$ 作为其最终的性能指标。如果要以一个通俗的比喻来解释这三个子数据集之间的关系，那么训练集就像是课堂上老师传授的知识，用来让学生学会某一领域的能力；验证集像是课后作业，用以评估学生的学习效果并指导其学习过程；而测试集则像是考试，用以评价学生学到的能力水平的高低。

在一般的分布式训练场景中，所有参与分布式训练的终端设备都配有一个本地数据集，该数据集通常由调度中心分配给各终端设备。我们将所有终端设备构成的集合记为 \mathcal{M} ，设备数量记为 $M = |\mathcal{M}|$ 。对于 \mathcal{M} 中的任意一个终端设备 m ，其本地数据集记为 D_m ，其大小记为 $D_m = |D_m|$ 。分布式训练中的总数据集 D 是所有终端设备的本地数据集的并集，即 $D = \bigcup_{m=0}^{M-1} D_m$ 。值得注意的是，各终端设备的本地数据集间可以有重复的数据样本，也可以没有重复，这取决于调度中心对终端设备的数据分配方式。

而对于车联网的分布式训练场景，每辆智能网联车辆（以后简称为车辆）作为终端设备携带有一个本地数据集。与一般的分布式训练场景不同，车联网场景中各车辆的本地数据集由它们自身所携带，且数据来源于车辆各类传感器所采集的数据，作为调度中心的路侧单元一般不会直接访问车辆的本地数据集。此外，由于车辆的移动性，路侧单元信号覆盖范围下的分布式训练区域（以后简称为训练区域）中的车辆集合将随时间发生变化，这也意味着车联网分布式训练中的数据集将随时间发生变化。记 t 时刻训练区域内的车辆集合为 $\mathcal{M}(t)$ ，车辆数量为 $M(t) = |\mathcal{M}(t)|$ ，总数据集为 $D(t) = \bigcup_{m \in \mathcal{M}(t)} D_m$ 。在该场景下，虽然在 t 时刻属于 $\mathcal{M}(t)$ 的车辆在未来将会离开训练区域，其携带的本地数据集也将不再被分布式训练所利用，但是未来也将有新的车辆携带着新的本地数据集到达训练区域，使得分布式训练得以继续下去。

本文的研究中主要用到了 Raymobtime 数据集^[19]和 Argoverse Motion Forecasting 数据集^[16]。Raymobtime 数据集是 2020 国际电联人工智能/机器学习 5G 挑战赛的官方数据集，包括了激光雷达、相机图片、GPS 以及各波束方向的信道信息等数据，用以训练用于毫米波波束选择任务的神经网络模型。Argoverse Motion Forecasting 数据集是一个配备了高清地图数据的包含 323,557 个交通场景片段的数据集，每个场景片段时长 5 秒，采样频率为 10Hz，包含每个跟踪对象（包括车、行人等）的二维质心运动轨迹，用以训练用于车辆轨迹预测任务的神经网络模型。

2.3.4 损失函数

为了对神经网络模型进行训练，我们除了需要选取数据集，还需要确定模型训练算法中的损失函数 $\mathcal{L}(\cdot)$ 。损失函数是机器学习领域的重点研究对象之一，它可以度量一个神经网络模型对一个数据样本的预测值与真实值之间的差异程度。基于给定的损失函数 $\mathcal{L}(\cdot)$ ，我们可以将神经网络 w 在数据集 D 上的平均损失函

数记为 $F(\mathbf{w}, \mathcal{D})$ ，其表达式为：

$$F(\mathbf{w}, \mathcal{D}) = \frac{1}{D} \sum_{\{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{D}} \mathcal{L}(f_I(\mathbf{x}_i); \mathbf{y}_i), \quad (2.1)$$

其中 f_I 表示应用了神经网络 \mathbf{w} 的推断算法， D 表示数据集 \mathcal{D} 的大小。

机器学习中一般假设， $F(\mathbf{w}, \mathcal{D})$ 越小，模型 \mathbf{w} 在数据集 \mathcal{D} 上的性能表现越好。在训练集 \mathcal{D} 上对一个神经网络进行训练，其本质为求取最优模型参数 \mathbf{w}^* ， \mathbf{w}^* 满足如下关系：

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} F(\mathbf{w}, \mathcal{D}). \quad (2.2)$$

在实际应用中，由于神经网络模型过于复杂，我们无法求得 \mathbf{w}^* 的闭式解。为了解决这一问题，人们通常采用以梯度下降法为代表的更新策略对模型参数进行多轮迭代更新，让更新后的模型参数尽可能收敛于 \mathbf{w}^* 。

对于一个神经网络模型的训练算法而言，损失函数是其优化目标，对损失函数的设计是训练算法中的重要部分。在分类问题中，损失函数常常以交叉熵的形式出现；而在回归问题中，损失函数往往采用均方误差函数或其相关变体。可以看出，损失函数的设计取决于目标任务、神经网络模型以及数据集等影响因素。

2.4 车联网中的分布式训练框架

基于2.3节中介绍的分布式训练相关要素，我们可以对车联网中的分布式训练框架进行建模。在该框架中，有且仅有一个路侧单元作为分布式训练调度中心，训练区域由其信号覆盖范围所决定，所有在训练过程中经过训练区域的车辆都可视为参与分布式训练的终端设备。该框架以分布式训练回合（以后简称为回合）为循环单元，通过迭代的方式对神经网络模型进行分布式训练。

如图2.3所示，一个回合又分为模型下发、本地训练、模型上传以及模型聚合等四个环节。其中，模型下发是指作为分布式训练调度中心的路侧单元向训练区域中的车辆下发全局模型参数；本地训练是指车辆在收到全局模型参数后，将其作为初始本地模型参数，然后基于本地数据集对其进行训练，得到更新后的本地模型参数；模型上传是指车辆在完成本地训练后，将本地模型参数传回给路侧单元；模型聚合是指路侧单元在回合结束时对该回合内收到的所有模型参数进行加权聚合，从而得到下一回合的全局模型参数。

为了便于对以上分布式训练流程进行详细说明，我们在表2.1中引入了相关符号，并对它们的含义进行了具体描述。值得注意的是，对于 $k \in \{0, \dots, K-1\}$,

分布式训练回合

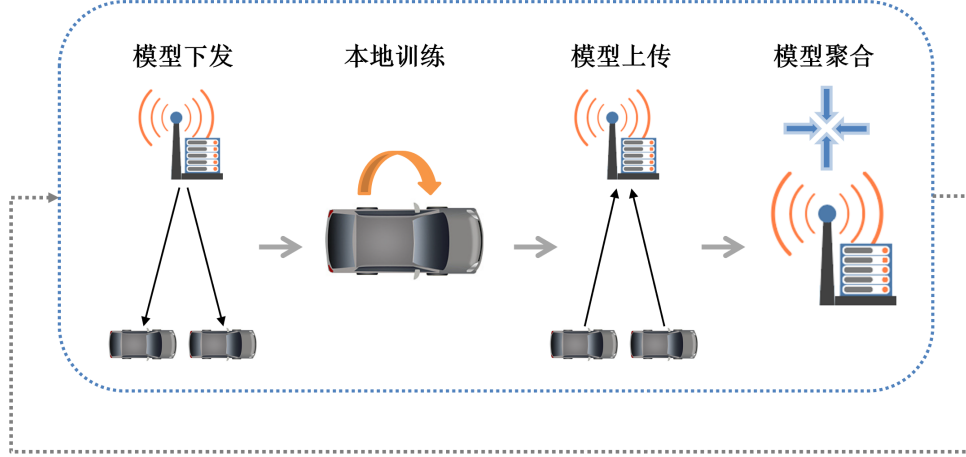


图 2.3 车联网中的分布式训练流程图

表 2.1 车联网中分布式训练相关符号

符号	描述
K	在分布式训练过程中完成的回合总数
\mathcal{M}	在分布式训练过程中曾处于训练区域内的车辆构成的集合
T_k	第 k 个回合的时长, $k \in \{0, \dots, K-1\}$
H_k	第 k 个回合车辆进行本地训练时的模型迭代次数, $k \in \{0, \dots, K-1\}$
\mathcal{M}_k	第 k 个回合曾处于训练区域内的车辆构成的集合, $k \in \{0, \dots, K-1\}$
M_k	第 k 个回合曾处于训练区域内的车辆的总数, $k \in \{0, \dots, K-1\}$
w_k	第 k 个回合的全局模型参数, $k \in \{0, \dots, K\}$
$w_{k,m}^{[h]}$	第 k 个回合中编号为 m 的车辆经过 h 次模型迭代后的本地模型参数, $k \in \{0, \dots, K-1\}$, $m \in \mathcal{M}_k$, $h \in \{0, \dots, H_k-1\}$
D_m	编号为 m 的车辆携带的本地数据集, $m \in \mathcal{M}$
D_m	编号为 m 的车辆携带的本地数据集的大小, $m \in \mathcal{M}$

w_k 指的是从第 k 回合的起始时刻 t_k 到结束时刻 t_{k+1} 这段时间 $[t_k, t_{k+1})$ 内的全局模型参数, 而 w_K 指的是经过这 K 个回合的分布式训练后最终得到的全局模型参数。接下来, 我们将基于表2.1中引入的符号, 对一个回合中的四个环节进行详细建模。

2.4.1 模型下发

我们记 $t_0 = 0, t_k = \sum_{i=0}^{k-1} T_i, k = 1, \dots, K$, 则第 k 回合的起始时刻和结束时刻分别为 t_k 和 t_{k+1} 。在 t_k 时刻, 路侧单元向训练区域内的所有车辆下发全局模型参数 \mathbf{w}_k 。而对于在 $[t_k, t_{k+1})$ 时段内到达训练区域的车辆, 路侧单元同样将全局模型参数 \mathbf{w}_k 下发给它们。为表示方便, 我们将 t_k 时刻处于训练区域内的车辆和在 $[t_k, t_{k+1})$ 时段内到达训练区域的车辆统称为第 k 回合曾处于训练区域内的车辆, 它们构成的集合记为 \mathcal{M}_k 。在第 k 回合, $\forall m \in \mathcal{M}_k$, 模型下发完成对车辆 m 的本地模型参数的初始化, 即:

$$\mathbf{w}_{k,m}^{[0]} = \mathbf{w}_k. \quad (2.3)$$

对于路侧单元向车辆下发全局模型参数这一过程, 我们采用点对点通信的方式进行建模。由于通信带宽有限, 模型传输将会带来时延。我们将第 k 回路侧单元向编号为 m 的车辆下发全局模型参数的通信时延记为 $\tau_{k,m}^{\text{down}}$ 。由于发射功率、带宽分配的差异, 以及信道衰落、信号干扰和噪声等影响, $\tau_{k,m}^{\text{down}}$ 是一个随机变量, 服从一定的概率分布, 且不同的 $\tau_{k,m}^{\text{down}}$ 间可能具有一定相关关系。不过在理论分析中, 为了简化分析, 一般会将 $\tau_{k,m}^{\text{down}}$ 的概率分布建模为一个偏移负指数分布, 不同的 $\tau_{k,m}^{\text{down}}$ 独立同分布, 或者直接将 $\tau_{k,m}^{\text{down}}$ 取定值。对于通信时延的建模, 我们将在之后的章节中做进一步讨论。

需要注意的是, 在模型下发过程中, 车辆可能会行驶出训练区域, 从而导致路侧单元对该车辆模型下发的失败, 此时的 $\tau_{k,m}^{\text{down}}$ 不再具有意义。而如何降低模型下发的失败率, 则是后续我们做优化算法设计时需要考虑的问题之一。

2.4.2 本地训练

在车辆收到当前回合的全局模型参数后, 就会进入本地训练环节。车辆的本地训练与非分布式的神经网络模型训练相似, 都是基于数据集、损失函数、更新策略等要素对神经网络模型参数进行迭代式的更新。因此, 要对车辆 m 的本地模型进行训练, 除了需要指定训练用的本地数据集 \mathcal{D}_m , 还需要指定损失函数 $F(\cdot)$ 和更新策略。其中, 本地数据集由目标任务决定, 损失函数是训练算法的优化目标, 对损失函数的设计是训练算法中的重要部分, 而更新策略则一般采用梯度下降法或其变体形式。

在我们的训练框架中, 车辆进行本地训练的更新策略采用小批量梯度下降法 (Mini-Batch Gradient Descent, MBGD), 其相关参数包括本地训练迭代次数 H_k 、

学习率 η 、批大小 B 等。该更新策略表示如下：

$$\mathbf{w}_{k,m}^{[h]} = \mathbf{w}_{k,m}^{[h-1]} - \eta \cdot \nabla F(\mathbf{w}_{k,m}^{[h-1]}, \mathcal{B}_{k,m}^{[h]}), \quad (2.4)$$

其中 $h \in \{1, \dots, H_k\}$, $\mathcal{B}_{k,m}^{[h]}$ 为批数据集，它代表第 k 回合车辆 m 在第 h 次本地迭代时从本地数据集 \mathcal{D}_m 中随机抽取的 B 个数据样本构成的集合。由于我们固定了更新策略的形式，因此后续对于更新策略的设计仅需要给定 H_k 、 η 以及 B 这三个参数。

必须注意到的是，车辆进行本地训练势必会产生计算时延。我们将第 k 回合车辆 m 进行本地训练的计算时延记为 $\tau_{k,m}^{\text{cp}}$ ，它由车辆的计算能力、模型的复杂度以及本地迭代次数 H_k 等因素决定。由于车辆间的硬件设备存在差异，它们的计算能力存在区别。此外，由于车辆在参与分布式训练的同时，还需要对其他任务（如自动驾驶等）进行处理，其计算资源占用率也不尽相同。因此，我们将计算时延 $\tau_{k,m}^{\text{cp}}$ 建模为一个随机变量，它服从偏移负指数分布：

$$\mathbb{P}[\tau_{k,m}^{\text{cp}} \leq t] = \begin{cases} 1 - e^{-\frac{1}{\beta_{k,m}H_k}(t - \mu_{k,m}H_k)}, & t \geq \mu_{k,m}H_k, \\ 0, & \text{otherwise,} \end{cases} \quad (2.5)$$

其中 $\mu_{k,m}$ 表示执行一次本地迭代的最小时延， $\beta_{k,m}$ 表征该分布的随机性。由式(2.5)可知， $\tau_{k,m}^{\text{cp}}$ 的均值为 $\mathbb{E}[\tau_{k,m}^{\text{cp}}] = (\mu_{k,m} + \beta_{k,m})H_k$ ，方差为 $\mathbb{C}(\tau_{k,m}^{\text{cp}}) = \beta_{k,m}H_k$ 。

在车辆进行本地训练的过程中，有可能驶出训练区域，导致该车辆的本地训练结果无法被路侧单元所利用。而如何尽可能地减少该事件的发生，是我们后续做优化算法设计时需要考虑的问题之一。

2.4.3 模型上传

以第 k 回合编号为 m 的车辆为例，在其完成本地训练后，若其仍处于训练区域内，且本回合仍未结束（即本地训练完成时刻早于 t_{k+1} ），则该车辆将进入模型上传环节，将更新后的本地模型参数 $\mathbf{w}_{k,m}^{[H_k]}$ 上传给路侧单元。与模型下发环节类似，模型上传的过程同样可以建模为点对点通信，其通信时延记为 $\tau_{k,m}^{\text{up}}$ 。 $\tau_{k,m}^{\text{up}}$ 是一个随机变量，对其概率分布的建模与模型下发环节的通信时延 $\tau_{k,m}^{\text{down}}$ 类似，二者应属于同类型的概率分布函数。

同样需要注意的是，在模型上传过程中，车辆有可能会驶出训练区域，从而导致该车辆上传模型失败，此时的 $\tau_{k,m}^{\text{up}}$ 不再具有意义。而如何降低模型上传的失败率，同样是后续我们做优化算法设计时需要考虑的问题之一。

2.4.4 模型聚合

在一个回合的结束时刻，路侧单元将对全局模型参数进行更新。由于此更新通常以加权聚合的形式执行，这一行为也被称为模型聚合。以第 k 回合为例，在该回合时段 $[t_k, t_{k+1})$ 内，共有 M_k 辆车曾处于训练区域，这些车辆构成集合 \mathcal{M}_k 。其中，共有 M_k^{suc} 辆车成功完成模型上传，这些车构成集合 $\mathcal{M}_k^{\text{suc}}$ ， $\mathcal{M}_k^{\text{suc}} \subseteq \mathcal{M}_k$ 。为得到第 $k+1$ 回合的全局模型参数 w_{k+1} ，路侧单元在 t_{k+1} 时刻以如下策略更新全局模型参数：

$$w_{k+1} = \sum_{m \in \mathcal{M}_k^{\text{suc}}} \alpha_{k,m} w_{k,m}^{[H_k]}, \quad (2.6)$$

式(2.6)中的 $\alpha_{k,m}$ 为在第 k 回合编号为 m 的车辆的模型参数加权系数，其应满足如下约束：

$$\sum_{m \in \mathcal{M}_k^{\text{suc}}} \alpha_{k,m} = 1, \forall k \in \{0, \dots, K-1\}, \quad (2.7)$$

$$0 \leq \alpha_{k,m} \leq 1, \forall k \in \{0, \dots, K-1\}, \forall m \in \mathcal{M}_k^{\text{suc}}. \quad (2.8)$$

在分布式训练中，加权系数 $\alpha_{k,m}$ 一般由终端设备的本地数据集大小与总数据集大小之比决定。而在我们的车联网分布式训练框架中， $\alpha_{k,m}$ 由车辆携带的本地数据集大小与本回合的总数据集大小之比决定。其表示形式如下：

$$\alpha_{k,m} = \frac{D_m}{\sum_{m' \in \mathcal{M}_k^{\text{suc}}} D_{m'}}, \forall k \in \{0, \dots, K-1\}, \forall m \in \mathcal{M}_k^{\text{suc}}. \quad (2.9)$$

值得一提的是，模型聚合的计算任务由路侧单元完成，存在一个计算时延 τ_k^{sum} 。然而由于模型聚合算法十分简单，计算消耗的时间极短，我们一般会忽略该计算时延对分布式训练的影响。在之后的理论分析与仿真实验中，我们也不会考虑 τ_k^{sum} 的影响。

2.5 本章小结

针对车联网中的分布式训练任务，本章给出了一个典型场景。在该场景中，交通道路以单向单车道直路模型建模，车流过程以泊松到达过程与定速行驶过程建模。基于分布式训练相关要素，我们对车联网中的分布式训练框架进行建模。该框架以分布式训练回合为循环单元，每个回合包括模型下发、本地训练、模型上传、模型聚合等四个环节。通过迭代执行分布式训练回合，神经网络模型的全

局模型参数将逐渐趋于收敛，同时模型性能不断提升，最终达到应用预期。

第 3 章 车联网分布式训练优化算法设计

3.1 引言

在本章中，我们考虑在车联网分布式训练的各回合分布式训练回合时长 T_k 和本地训练迭代次数 H_k 分别取固定参数 T 和 H 的情况下，如何对 T 和 H 进行优化，以提高车联网分布式训练下全局神经网络模型的收敛速度。

车联网分布式训练的收敛速度受到许多因素的影响。为了分析方便，我们仅在其他参数固定不变的情况下，讨论 T 和 H 对分布式训练的影响。在其他参数不变的情况下，由于总训练回合数 K 由总训练时长 T_{sim} 和回合时长 T 决定：

$$K = \left\lfloor \frac{T_{\text{sim}}}{T} \right\rfloor, \quad (3.1)$$

因此当 T 增大时，总训练回合数 K 减小，导致全局模型参数的更新次数减小，从而使得分布式训练变慢；而当 T 减小时，每个回合内路侧单元收到的模型数量的期望减小，也会影响分布式训练的收敛速度。同样在其他参数不变的情况下对 H 进行分析，当 H 增大时，每辆车进行本地训练的计算时延变长，更有可能上传模型失败，因此每个回合内路侧单元收到的模型数量的期望变小，这对分布式训练的收敛速度不利；而当 H 减小时，每回合车辆的本地模型更新次数变少，本地模型参数变化更慢，从而导致每回合全局模型参数的变化更慢，影响分布式训练的收敛速度。可见 T 和 H 是一对权衡参数，在其他参数不变的情况下， T 和 H 应该存在一个或多个最优值，使得一次分布式训练过程的模型收敛速度最快。

从路侧单元的角度出发， T 的选取既影响了全局模型参数更新的总次数 K ，又影响了每回合路侧单元能从车辆端收到的本地更新模型的数量 M_k^{suc} ；而 H 的选取既影响了每回合全局模型参数更新的变化程度 $\|\Delta \mathbf{w}_k\|$ ，同时也影响了 M_k^{suc} 。每回合模型更新的变化程度 $\|\Delta \mathbf{w}_k\|$ 虽然难以定量分析，但无论是从直觉上还是从实验上来看， $\|\Delta \mathbf{w}_k\|$ 都与 H 呈正相关关系：

$$\|\Delta \mathbf{w}_k\| \propto H, \quad (3.2)$$

在理想情况下 $\|\Delta \mathbf{w}_k\|$ 甚至可以直接由 H 近似表示。而 M_k^{suc} 与 T 正相关，与 H 负相关，且在给定的 T 和 H 下应具有特定的概率分布。

在本章的后续部分，我们会先对 M_k^{suc} 的概率分布进行理论分析。之后，通过结合 K 、 $\|\Delta \mathbf{w}_k\|$ 以及 M_k^{suc} 对分布式训练的影响，我们将构建一个具体的优化

问题，并针对该问题设计一个对 T 和 H 进行联合优化的优化算法。该算法设计的合理性与有效性将在4.4节中通过仿真实验进行验证，并对其局限性进行说明。

3.2 单回合成功车数的概率分布

我们称在一个分布式训练回合内成功完成模型上传的车辆为该回合的成功车。对应于第2章的符号，第 k 回合的所有成功车构成的集合为 $\mathcal{M}_k^{\text{suc}}$ ，成功车数为 M_k^{suc} 。由于 M_k^{suc} 也等于第 k 回合路侧单元收到的本地更新模型的数量，而其对于分布式训练的收敛性而言有一定影响（一般 M_k^{suc} 越大意味着分布式训练在第 k 回合的收敛性越好），因此我们对于 M_k^{suc} 的概率分布很感兴趣。下面，我们将在第2章描述的车联网分布式训练场景下对 M_k^{suc} 的概率分布进行分析。

定理 3.1： 在车联网分布式训练场景中，若满足以下假设条件：

1. 交通道路为单向单车道直路，道路长度为 L ；
2. 车辆依泊松到达，到达率为 λ ，车辆定速行驶，车速为 v ；
3. 车辆与路侧单元间进行模型下发与模型上传的通信时延为固定值，分别为 τ^{down} 和 τ^{up} ，二者之和为 τ^{cm} ；
4. 车辆每次进行本地训练的计算时延 τ^{cp} 独立同分布，其概率分布以公式(3.3)描述，其中 β 和 μ 为该偏移负指数分布的固定参数， H 为本地训练迭代次数；
5. 回合时长 T 与本地训练迭代次数 H 在整个分布式训练过程中保持不变；
6. $T \geq \mu H + \tau^{\text{cm}}$, $\frac{L}{v} \geq \mu H + \tau^{\text{cm}}$,

则第 k 回合成功车数 M_k^{suc} 服从参数为 Λ 的泊松分布， Λ 由公式(3.4)决定，其中 $T' = \frac{L}{v}$ ，表示车辆在道路上的行驶时间， $T_{\min} = \min\{T, T'\}$ ，表示 T 与 T' 间的最小值。

$$\mathbb{P}[\tau^{\text{cp}} \leq t] = \begin{cases} 1 - e^{-\frac{1}{\beta H}(t - \mu H)}, & t \geq \mu H, \\ 0, & \text{otherwise,} \end{cases} \quad (3.3)$$

$$\Lambda = 2\lambda (T_{\min} - \mu H - \tau^{\text{cm}}) + \lambda (|T - T'| - 2\beta H) \left(1 - e^{-\frac{1}{\beta H}(T_{\min} - \mu H - \tau^{\text{cm}})} \right). \quad (3.4)$$

证明： 以第 m 回合为例，若一辆车可以参与第 m 回合的分布式训练，根据2.4节的分布式训练规则，该车应在 mT 时刻前到达道路入口且在 mT 时刻仍在道路上，或在 $[mT, (m+1)T)$ 时间区间内到达道路入口。由于车辆在定长的道路上定

速行驶，车辆在道路上的行驶时间 T' 固定，因此可以参与第 m 回合分布式训练的
车辆到达道路入口的时刻应处于 $[mT - T', (m + 1)T)$ 内。此外，若该车能在第
 m 回合成功上传模型，那么其完成模型上传的最短时间必须同时小于 T 和 T' ，即
 $\mu H + \tau^{\text{cm}} < T_{\min}$ 。下面我们根据 T 与 T' 间的大小关系对 $[mT - T', (m + 1)T)$ 区
间进行分类讨论：

(1) 当 $T \geq T'$ 时：

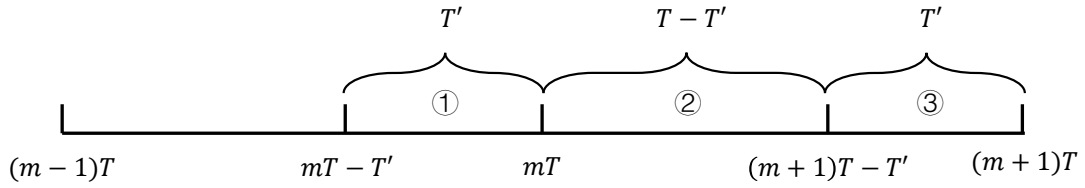


图 3.1 $T \geq T'$ 时单回合成功车数理论分析

如图3.1所示，对时间区间 $[mT - T', (m + 1)T)$ 进行划分，得到三个子时间
区间，其中区间①为 $[mT - T', mT)$ ，区间②为 $[mT, (m + 1)T - T')$ ，区间③为
 $[(m + 1)T - T', (m + 1)T)$ 。我们记 N_1 为在区间①到达道路入口的车辆数目，记 p_1
为在区间①到达道路入口的车辆在第 m 回合完成模型上传的概率，记 M_1 为在区
间①到达道路入口的车辆中最终在第 m 回合完成模型上传的车辆数目。类似地可
以定义 N_2 、 N_3 、 p_2 、 p_3 、 M_2 、 M_3 。我们所关心的成功车数为 $M_m^{\text{suc}} = M_1 + M_2 + M_3$ 。

由于车辆的到达过程是一个参数为 λ 的泊松过程，可知 N_1 、 N_2 、 N_3 服从
泊松分布，其泊松分布的参数分别为 $\lambda T'$ 、 $\lambda(T - T')$ 、 $\lambda T'$ ，且 N_1 、 N_2 、 N_3 相
互独立。由泊松分布的特性，以及各车成功上传模型事件的独立性，可以推知
 M_1 、 M_2 、 M_3 同样服从泊松分布，其泊松分布的参数分别为 $\lambda p_1 T'$ 、 $\lambda p_2 (T - T')$ 、
 $\lambda p_3 T'$ ，且 M_1 、 M_2 、 M_3 相互独立。由于 M_1 、 M_2 、 M_3 为相互独立的服从泊松
分布的随机变量， $M_m^{\text{suc}} = M_1 + M_2 + M_3$ ，因此 M_m^{suc} 同样服从泊松分布，其参数
 Λ 为 M_1 、 M_2 、 M_3 三者的参数之和，即：

$$\Lambda = \lambda p_1 T' + \lambda p_2 (T - T') + \lambda p_3 T'. \quad (3.5)$$

接下来对 p_1 、 p_2 、 p_3 进行求解。

对于在区间①内到达道路入口的车辆，它们在第 m 回合结束前必定会离开道
路，因此导致其上传模型失败的原因只可能是未能在驶出道路前完成模型上传。
此外，这些车辆到达道路入口的时刻 t 会影响它们参与第 m 回合分布式训练的时
间上限，即它们只能在 $T' - (mT - t)$ 时间内完成模型下载、本地训练以及模型上

传，否则便会上传模型失败。基于上述分析，我们可以算出 p_1 :

$$p_1 = \int_{mT-T'}^{mT} \mathbb{P} [\tau^{\text{cp}} + \tau^{\text{cm}} \leq T' - (mT - t)] \cdot \frac{1}{T'} \cdot dt \quad (3.6)$$

$$= \int_0^{T'} \mathbb{P} [\tau^{\text{cp}} \leq t - \tau^{\text{cm}}] \cdot \frac{1}{T'} \cdot dt \quad (3.7)$$

$$= \int_{\tau^{\text{cm}}+\mu H}^{T'} \left(1 - e^{-\frac{1}{\beta H}(t-\tau^{\text{cm}}-\mu H)} \right) \cdot \frac{1}{T'} \cdot dt \quad (3.8)$$

$$= \int_0^{T'-\tau^{\text{cm}}-\mu H} \left(1 - e^{-\frac{t}{\beta H}} \right) \cdot \frac{1}{T'} \cdot dt \quad (3.9)$$

$$= \frac{1}{T'} \left[T' - \tau^{\text{cm}} - \mu H - \beta H \left(1 - e^{-\frac{1}{\beta H}(T'-\tau^{\text{cm}}-\mu H)} \right) \right]. \quad (3.10)$$

对于在区间②内到达道路入口的车辆，它们在第 m 回合结束前必定会离开道路，因此导致其上传模型失败的原因只可能是未能在驶出道路前完成模型上传。此外，这些车辆到达道路入口的时刻 t 不会影响它们参与第 m 回合分布式训练的时间上限，该上限固定为 T' 。基于上述分析，我们可以算出 p_2 :

$$p_2 = \mathbb{P} [\tau^{\text{cp}} + \tau^{\text{cm}} \leq T'] \quad (3.11)$$

$$= \mathbb{P} [\tau^{\text{cp}} \leq T' - \tau^{\text{cm}}] \quad (3.12)$$

$$= 1 - e^{-\frac{1}{\beta H}(T'-\tau^{\text{cm}}-\mu H)}. \quad (3.13)$$

对于在区间③内到达道路入口的车辆，它们在第 m 回合结束前必定不会离开道路，因此导致其上传模型失败的原因只可能是未能在第 m 回合结束前完成模型上传。此外，这些车辆到达道路入口的时刻 t 会影响它们参与第 m 回合分布式训练的时间上限，即它们只能在 $(m+1)T - t$ 时间内完成模型下载、本地训练以及模型上传，否则便会上传模型失败。基于上述分析，我们可以算出 p_3 :

$$p_3 = \int_{(m+1)T-T'}^{(m+1)T} \mathbb{P} [\tau^{\text{cp}} + \tau^{\text{cm}} \leq (m+1)T - t] \cdot \frac{1}{T'} \cdot dt \quad (3.14)$$

$$= \int_0^{T'} \mathbb{P} [\tau^{\text{cp}} + \tau^{\text{cm}} \leq t] \cdot \frac{1}{T'} \cdot dt \quad (3.15)$$

$$= \int_0^{T'} \mathbb{P} [\tau^{\text{cp}} \leq t - \tau^{\text{cm}}] \cdot \frac{1}{T'} \cdot dt \quad (3.16)$$

$$= \frac{1}{T'} \left[T' - \tau^{\text{cm}} - \mu H - \beta H \left(1 - e^{-\frac{1}{\beta H}(T' - \tau^{\text{cm}} - \mu H)} \right) \right]. \quad (3.17)$$

将公式(3.10)(3.13)(3.17)代入公式(3.5)，可得：

$$\begin{aligned} \Lambda &= \lambda (T - T') \left(1 - e^{-\frac{1}{\beta H}(T' - \tau^{\text{cm}} - \mu H)} \right) \\ &\quad + 2\lambda \left[T' - \tau^{\text{cm}} - \mu H - \beta H \left(1 - e^{-\frac{1}{\beta H}(T' - \tau^{\text{cm}} - \mu H)} \right) \right] \end{aligned} \quad (3.18)$$

$$= 2\lambda (T' - \mu H - \tau^{\text{cm}}) + \lambda (T - T' - 2\beta H) \left(1 - e^{-\frac{1}{\beta H}(T' - \mu H - \tau^{\text{cm}})} \right). \quad (3.19)$$

(2) 当 $T < T'$ 时：

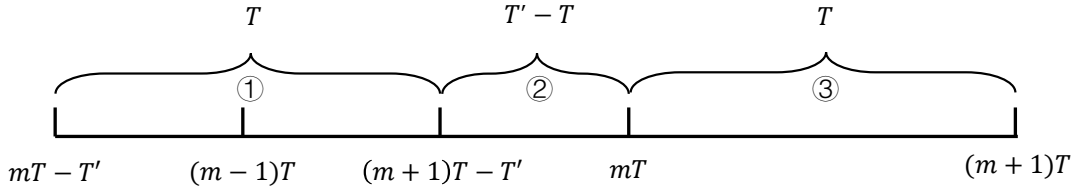


图 3.2 $T < T'$ 时单回合成功车数理论分析

与 (1) 中类似地，我们对时间区间 $[mT - T', (m+1)T)$ 进行划分，划分方法如图3.2所示。此外，与 (1) 中的分析一致，我们同样可以定义 $\{N_i\}_{i=1}^3$ 、 $\{p_i\}_{i=1}^3$ 、 $\{M_i\}_{i=1}^3$ 。 M_1 、 M_2 、 M_3 服从泊松分布，其泊松分布的参数分别为 $\lambda p_1 T$ 、 $\lambda p_2 (T' - T)$ 、 $\lambda p_3 T$ ，且 M_1 、 M_2 、 M_3 相互独立。 M_m^{suc} 服从泊松分布，其参数 Λ 为 M_1 、 M_2 、 M_3 三者的参数之和，即：

$$\Lambda = \lambda p_1 T + \lambda p_2 (T' - T) + \lambda p_3 T. \quad (3.20)$$

对于在区间①内到达道路入口的车辆，它们在第 m 回合结束前必定会离开道路，因此导致其上传模型失败的原因只可能是未能在驶出道路前完成模型上传。此外，这些车辆到达道路入口的时刻 t 会影响它们参与第 m 回合分布式训练的时间上限，即它们只能在 $T' - (mT - t)$ 时间内完成模型下载、本地训练以及模型上传，否则便会上传模型失败。基于上述分析，我们可以算出 p_1 ：

$$p_1 = \int_{mT - T'}^{(m+1)T - T'} \mathbb{P} [\tau^{\text{cp}} + \tau^{\text{cm}} \leq T' - (mT - t)] \cdot \frac{1}{T} \cdot dt \quad (3.21)$$

$$= \frac{1}{T} \left[T - \tau^{\text{cm}} - \mu H - \beta H \left(1 - e^{-\frac{1}{\beta H}(T - \tau^{\text{cm}} - \mu H)} \right) \right]. \quad (3.22)$$

对于在区间②内到达道路入口的车辆，它们在第 m 回合结束前必定不会离开道路，因此导致其上传模型失败的原因只可能是未能在第 m 回合结束前完成模型上传。此外，这些车辆到达道路入口的时刻 t 不会影响它们参与第 m 回合分布式训练的时间上限，该上限固定为 T 。基于上述分析，我们可以算出 p_2 ：

$$p_2 = \mathbb{P} [\tau^{\text{cp}} + \tau^{\text{cm}} \leq T] \quad (3.23)$$

$$= 1 - e^{-\frac{1}{\beta H}(T - \tau^{\text{cm}} - \mu H)}. \quad (3.24)$$

对于在区间③内到达道路入口的车辆，它们在第 m 回合结束前必定不会离开道路，因此导致其上传模型失败的原因只可能是未能在第 m 回合结束前完成模型上传。此外，这些车辆到达道路入口的时刻 t 会影响它们参与第 m 回合分布式训练的时间上限，即它们只能在 $(m+1)T - t$ 时间内完成模型下载、本地训练以及模型上传，否则便会上传模型失败。基于上述分析，我们可以算出 p_3 ：

$$p_3 = \int_{mT}^{(m+1)T} \mathbb{P} [\tau^{\text{cp}} + \tau^{\text{cm}} \leq (m+1)T - t] \cdot \frac{1}{T} \cdot dt \quad (3.25)$$

$$= \frac{1}{T} \left[T - \tau^{\text{cm}} - \mu H - \beta H (1 - e^{-\frac{1}{\beta H}(T - \tau^{\text{cm}} - \mu H)}) \right]. \quad (3.26)$$

将公式(3.22)(3.24)(3.26)代入公式(3.20)，可得：

$$\begin{aligned} \Lambda &= \lambda(T' - T) \left(1 - e^{-\frac{1}{\beta H}(T - \tau^{\text{cm}} - \mu H)} \right) \\ &\quad + 2\lambda \left[T - \tau^{\text{cm}} - \mu H - \beta H \left(1 - e^{-\frac{1}{\beta H}(T - \tau^{\text{cm}} - \mu H)} \right) \right] \end{aligned} \quad (3.27)$$

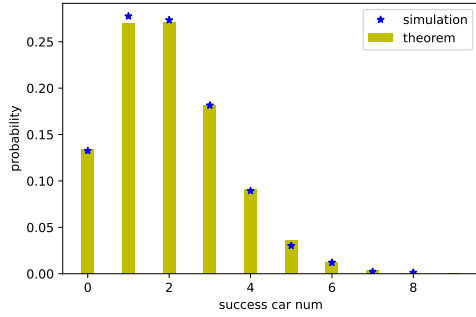
$$= 2\lambda(T - \mu H - \tau^{\text{cm}}) + \lambda(T' - T - 2\beta H) \left(1 - e^{-\frac{1}{\beta H}(T - \mu H - \tau^{\text{cm}})} \right). \quad (3.28)$$

综上(1)(2)所述， M_m^{suc} 服从泊松分布，其参数 Λ 在 $T \geq T'$ 时由公式(3.19)表示，在 $T < T'$ 时由公式(3.28)表示。为了统一表示 Λ ，我们可以设 $T_{\min} = \min\{T, T'\}$ ，则以下公式在(1)(2)两种情况下都成立：

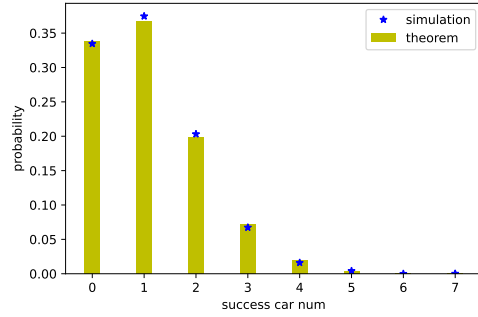
$$\Lambda = 2\lambda(T_{\min} - \mu H - \tau^{\text{cm}}) + \lambda(|T - T'| - 2\beta H) \left(1 - e^{-\frac{1}{\beta H}(T_{\min} - \mu H - \tau^{\text{cm}})} \right). \quad (3.29)$$

■

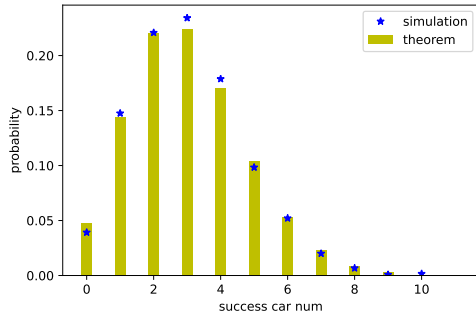
需要注意的是，根据定理3.1，虽然每个回合的成功车数 M_k^{suc} 服从相同的泊松分布，但它们之间并不独立，其原因在于相邻回合的成功车集合之间可能存在



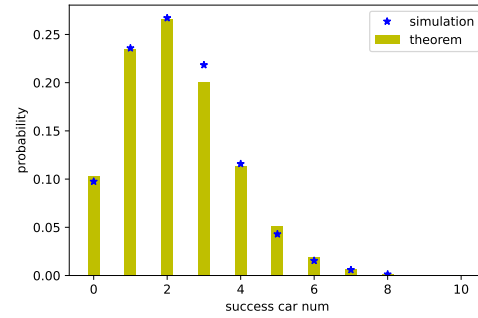
(a) $T = 10, H = 10$



(b) $T = 10, H = 20$



(c) $T = 20, H = 10$



(d) $T = 20, H = 20$

图 3.3 通过仿真验证定理3.1

重叠，即当前回合的成功车可能也是下一回合的成功车，由此带来了相关性。对于该相关性的研究有待后续工作继续探索，在此不做进一步分析。

除了上述的理论性证明，我们还通过仿真实验验证了定理3.1的正确性。如图3.3所示，在其他参数设置相同的情况下，通过改变分布式训练回合时长 T 和本地训练迭代次数 H 进行四次仿真实验，比较仿真结果和理论分析中单回合成功车数的频率/概率分布。图3.3中每张子图的横轴为单回合成功车数，纵轴为频率/概率，蓝色星形点为仿真得到的结果，代表一次仿真实验中相应单回合成功车数出现的频率，黄色柱状图为理论分析得到的结果，代表定理3.1给出的单回合成功车数的概率分布。从图3.3中可以看到，我们的理论分析与仿真结果十分接近，而其中的误差主要来源于：1. 仿真中车辆具有一定车长与最小间距，而理论分析中则忽略了车辆长度与最小间距的影响；2. 仿真中车辆速度有一定波动，而理论分析中则假设车辆定速行驶；3. 仿真实验具有一定随机性的扰动。可以说，在误差允许范围内，仿真实验很好地验证了定理3.1的正确性。

3.3 优化问题建模

由于分布式训练的模型收敛过程十分复杂，我们难以精准刻画其收敛性能，因此常常需要在一些理想假设下进行问题转换，建模出一个新的优化问题进行求解，这便是本节中优化问题的建模思路。

我们假设一个理想情况，在该理想情况下，每辆车本地数据集内的数据样本足够多，其包含的数据特征足够充分，本地训练的批大小足够大，神经网络模型足够鲁棒的理想情况。我们简称该理想情况为情况 1。在情况 1 下，第 k 回合内路侧单元从每辆成功车端接收的本地模型近似相同，因此只要该回合有成功车，即 $M_k^{\text{suc}} > 0$ ，那么不论成功车数 M_k^{suc} 是多少，全局模型都能得到更新，且更新结果与 M_k^{suc} 无关。我们将有成功车的回合称为成功回合，记整个分布式训练过程中的成功回合总数为 K^{suc} 。在情况 1 下，经过整个分布式训练过程，全局模型参数的等效总迭代次数为 HK^{suc} ，其与最终模型 w_K 的性能间呈正相关关系。根据上述分析，我们可以得到情况 1 下的优化目标为最大化 $\mathbb{E}[HK^{\text{suc}}]$ 。由于 H 为固定值， $\mathbb{E}[HK^{\text{suc}}] = H\mathbb{E}[K^{\text{suc}}] = H \sum_{k=0}^{K-1} p_k^{\text{suc}}$ ，其中 $p_k^{\text{suc}} = \mathbb{P}[M_k^{\text{suc}} > 0] = 1 - \mathbb{P}[M_k^{\text{suc}} = 0]$ 表示第 k 回合是成功回合的概率。根据定理 3.1，可知 $\mathbb{P}[M_k^{\text{suc}} = 0] = e^{-\Lambda(H,T)}$ ，其中 $\Lambda(H,T)$ 由公式(3.4)决定。因此，我们有：

$$\mathbb{E}[HK^{\text{suc}}] = HK(1 - e^{-\Lambda(H,T)}). \quad (3.30)$$

将公式(3.1)代入公式(3.30)，可得：

$$\mathbb{E}[HK^{\text{suc}}] = H \left\lfloor \frac{T_{\text{sim}}}{T} \right\rfloor (1 - e^{-\Lambda(H,T)}). \quad (3.31)$$

当 $T_{\text{sim}} \gg T$ 时， $\left\lfloor \frac{T_{\text{sim}}}{T} \right\rfloor$ 可近似为 $\frac{T_{\text{sim}}}{T}$ ，同时由于 T_{sim} 为固定参数，我们的优化目标可以等价于最大化 $\frac{H}{T} (1 - e^{-\Lambda(H,T)})$ 。记 $\tilde{\tau}(T) = \min\{T, T'\}$ ，优化目标函数为 $g(H, T)$ ，则有：

$$g(H, T) = \frac{H}{T} (1 - e^{-\Lambda(H,T)}), \quad (3.32)$$

其中：

$$\begin{aligned} \Lambda(H, T) = & 2\lambda (\tilde{\tau}(T) - \mu H - \tau^{\text{cm}}) \\ & + \lambda (|T - T'| - 2\beta H) \left(1 - e^{-\frac{1}{\beta H}(\tilde{\tau}(T) - \mu H - \tau^{\text{cm}})} \right) \end{aligned} \quad (3.33)$$

综上所述，我们最终可将情况 1 下的，满足定理 3.1 假设条件的优化问题建模

为:

$$\mathcal{P4-1}: \max_{H,T} g(H,T) \quad (3.34)$$

$$\text{s.t. } 0 < T \leq T_{\text{sim}}, T \in \mathbb{R}, \quad (3.35)$$

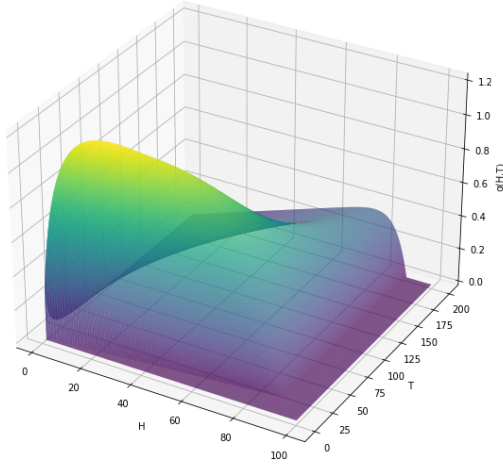
$$1 \leq H < \frac{\tilde{\tau}(T) - \tau^{\text{cm}}}{\mu}, H \in \mathbb{N}, \quad (3.36)$$

$$T', \tau^{\text{cm}}, \mu, \beta, \lambda \in \mathbb{R}^+. \quad (3.37)$$

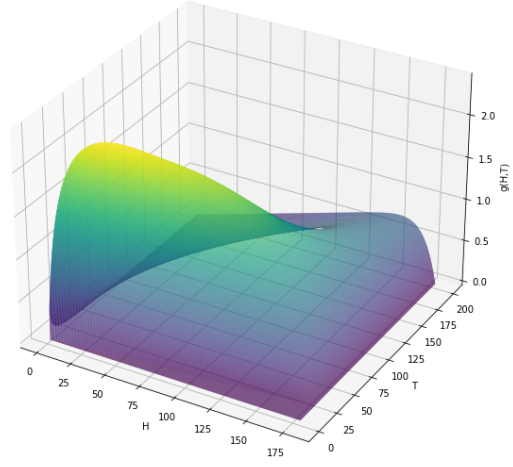
由于优化问题 $\mathcal{P4-1}$ 建模的前提假设是需要满足情况 1，而情况 1 中对数据集和神经网络模型的假设条件较强，在实际应用中不一定能得到满足，因此优化问题 $\mathcal{P4-1}$ 建模的有效性与普适性仍待检验。而为了验证优化问题 $\mathcal{P4-1}$ 的目标函数 $g(H,T)$ 是否能有效地反映实际应用中车联网分布式训练的收敛性能，我们基于第4章的车联网分布式训练仿真平台，以控制变量法进行了若干组仿真实验，并将各组实验的模型性能指标与 $g(H,T)$ 进行了比较，发现 $g(H,T)$ 确实可以有效反映车联网分布式训练的收敛性能。相关实验结果将在4.4.4节中进行展示与讨论。

3.4 优化算法设计

本节对于优化问题 $\mathcal{P4-1}$ 进行优化算法设计。在该问题中，优化目标函数 $g(H,T)$ 的形式十分复杂，我们难以通过一般的数学方法得到一组最优参数 (H^*, T^*) 的闭式解。为了启发我们的优化算法设计思路，我们分别在波束选择任务和轨迹预测任务的典型参数设置的基础上，通过改变 H 和 T 的取值，利用计算机作图工具画出了 $g(H,T)$ 在 H - T 平面上的三维曲面图，结果如图3.4所示。从图中我们可以看出， $g(H,T)$ 在两组典型参数设置下均存在唯一极大值点 (H^*, T^*) ，且在该极大值点附近为局部下凸函数。此外，对于给定的 T ， $g(H,T)$ 关于 H 是一个单峰函数，具有唯一极大值点；而对于给定的 H ， $g(H,T)$ 关于 T 同样是一个单峰函数，也具有唯一极大值点。由此我们可以很容易想到一个可行的优化算法设计思路：从初始点 (H_0, T_0) 出发，先固定 T 而对 H 进行优化，再固定 H 而对 T 进行优化，如此循环迭代，直至 $g(H,T)$ 收敛。而要根据该思路设计优化算法，我们还需要确定 H 和 T 的搜索范围，初始点 (H_0, T_0) 的选取方法，在固定 T （或 H ）时对 H （或 T ）的优化方案，以及该迭代优化算法的终止条件。下面我们将对这些问题依次进行讨论。



(a) 波束选择任务典型参数



(b) 轨迹预测任务典型参数

图 3.4 两组典型参数设置下, $g(H, T)$ 在 H - T 平面上的三维曲面图

根据优化问题 $\mathcal{P}4$ -1 中公式(3.36)的约束, 可知 H 的搜索范围为 $\left[1, \frac{T' - \tau^{\text{cm}}}{\mu}\right)$ 内的整数。而对于 T 的搜索范围, 虽然公式(3.35)对 T 的取值范围做出限制, 但 T_{sim} 一般远大于 T^* , 直接在 $(0, T_{\text{sim}}]$ 上对 T 进行搜索开销过大, 因此需要找到一个更紧的搜索范围。更确切地说, 我们需要找到一个 $T^{\text{sup}}(H)$, 使得当 $T > T^{\text{sup}}(H)$ 时, $\frac{\partial g(H, T)}{\partial T} < 0$ 恒成立。为此, 我们证明了定理3.2, 该定理解决了如何选取这样一个 $T^{\text{sup}}(H)$ 的问题。需要指出的是, 该定理给出的 $T^{\text{sup}}(H)$ 并不是唯一满足需求的选择, 也不是最紧的选择, 但在实际应用中能将 T 的搜索区间控制在一个合理范围内, 从而有效减少每次对 T 进行优化时的搜索开销。

定理 3.2: 对于优化问题 $\mathcal{P}4$ -1, 当 $T > T^{\text{sup}}(H)$ 时, $\frac{\partial g(H, T)}{\partial T} < 0$ 恒成立。其中 $T^{\text{sup}}(H)$ 由公式(3.38)-(3.40)给出。

$$C_0(H) = 1 - e^{-\frac{1}{\beta H}(T' - \mu H - \tau^{\text{cm}})}, \quad (3.38)$$

$$C_1(H) = 2(T' - \mu H - \tau^{\text{cm}}) - (T' + 2\beta H)C_0(H), \quad (3.39)$$

$$T^{\text{sup}}(H) = \begin{cases} T', & \text{if } C_1(H) \geq 0, \\ T' + \frac{1-12\lambda C_1(H)}{4\lambda C_0(H)}, & \text{if } C_1(H) < 0. \end{cases} \quad (3.40)$$

证明: 显然有 $0 < C_0(H) < 1$ 。

当 $T > T'$ 时, 由公式(3.33)可得:

$$\Lambda(H, T) = 2\lambda(T' - \mu H - \tau^{\text{cm}}) + \lambda(T - T' - 2\beta H) \left(1 - e^{-\frac{1}{\beta H}(T' - \mu H - \tau^{\text{cm}})} \right), \quad (3.41)$$

两边对 T 求偏微分可得:

$$\frac{\partial \Lambda(H, T)}{\partial T} = \lambda \left(1 - e^{-\frac{1}{\beta H}(T' - \mu H - \tau^{\text{cm}})} \right), \quad (3.42)$$

代入公式(3.38)可得:

$$\frac{\partial \Lambda(H, T)}{\partial T} = \lambda C_0(H). \quad (3.43)$$

优化目标函数 $g(H, T) = \frac{H}{T} (1 - e^{-\Lambda(H, T)})$ 对 T 求偏微分可得:

$$\frac{\partial g(H, T)}{\partial T} = \frac{e^{-\Lambda(H, T)}}{T} \cdot \frac{\partial \Lambda(H, T)}{\partial T} - \frac{1 - e^{-\Lambda(H, T)}}{T^2}, \quad (3.44)$$

代入公式(3.41)和(3.43)可得:

$$\frac{\partial g(H, T)}{\partial T} = \frac{(\lambda T C_0(H) + 1)e^{-\lambda T C_0(H) - \lambda C_1(H)} - 1}{T^2}, \quad (3.45)$$

要使得 $\frac{\partial g(H, T)}{\partial T} \leq 0$, 需满足:

$$\lambda T C_0(H) + 1 \leq e^{\lambda T C_0(H) + \lambda C_1(H)}. \quad (3.46)$$

当 $C_1(H) \geq 0$ 时: 由不等式 $x + 1 \leq e^x$ 可得:

$$\lambda T C_0(H) + 1 \leq e^{\lambda T C_0(H)} \leq e^{\lambda T C_0(H) + \lambda C_1(H)}, \quad (3.47)$$

因此该情况下只需取 $T^{\text{sup}}(H) = T'$, 即可满足不等式(3.46)对于任意 $T > T^{\text{sup}}(H)$ 均成立。

当 $C_1(H) < 0$ 时: 由不等式 $1 + x + \frac{x^2}{2} \leq e^x, \forall x \geq 0$ 可得:

$$1 + \lambda T C_0(H) + \lambda C_1(H) + \frac{(\lambda T C_0(H) + \lambda C_1(H))^2}{2} \leq e^{\lambda T C_0(H) + \lambda C_1(H)}, \quad (3.48)$$

为满足不等式(3.46), 只需有:

$$\lambda T C_0(H) + 1 \leq 1 + \lambda T C_0(H) + \lambda C_1(H) + \frac{(\lambda T C_0(H) + \lambda C_1(H))^2}{2}, \quad (3.49)$$

化简可得:

$$-2\lambda C_1(H) \leq (\lambda T C_0(H) + \lambda C_1(H))^2, \quad (3.50)$$

为满足不等式(3.50), 可取 $T^{\text{sup}}(H) = T' + \frac{1-12\lambda C_1(H)}{4\lambda C_0(H)}$, 此时对于任意 $T > T^{\text{sup}}(H)$,

均有：

$$T > T', \quad T > \frac{1 - 12\lambda C_1(H)}{4\lambda C_0(H)}, \quad (3.51)$$

对其中第二个不等式进行变换可得：

$$\lambda T C_0(H) + \lambda C_1(H) > -2\lambda C_1(H) + \frac{1}{4}, \quad (3.52)$$

根据不等式 $x + \frac{1}{4} \geq \sqrt{x}, \forall x \geq 0$ 可得：

$$-2\lambda C_1(H) + \frac{1}{4} \geq \sqrt{-2\lambda C_1(H)}, \quad (3.53)$$

结合不等式(3.52)和(3.53)立即可得不等式(3.50)成立，进而满足不等式(3.46)。

综上所述，定理3.2成立。 ■

在确定了 H 和 T 的搜索范围后，我们便可从中选取初始点 (H_0, T_0) ，这里我们可直接取 $H_0 = 1, T_0 = T'$ ，该点一定处于之前确定的搜索范围内。

关于固定 T 时对 H 的优化方案，由于 H 的搜索范围为 $\left[1, \frac{T' - \tau^{cm}}{\mu}\right)$ 内的整数， H 取值的个数有限，因此可以直接遍历所有可能的取值，从中选取令 $g(H, T)$ 最大的 H 。

关于固定 H 时对 T 的优化方案，我们可采用基于偏微分的二分搜索法。由于 $g(H, T)$ 是一个连续可微的二元函数，且当 H 固定时，在 T 的搜索范围内有且仅有一个极大值点，在该极大值点左侧 $\partial_T g(H, T) > 0$ ，在该极大值点右侧 $\partial_T g(H, T) < 0$ ，因此基于偏微分的二分搜索法可以给出固定 H 时 $g(H, T)$ 关于 T 的最大值点。

最后关于迭代优化算法的终止条件，我们可以设定一个判别阈值 ϵ ，当经过一次循环迭代后 $g(H, T)$ 的相对增量小于 ϵ 时，搜索终止。

综上所述，我们设计出了算法3.1，其通过迭代优化的策略对 H - T 进行联合优化，最终能在误差允许范围内逼近最优解 (H^*, T^*) 。为了验证该算法设计的合理性与有效性，我们在4.4.5节中进行了相关仿真实验，其实验结果表明算法3.1可以有效提高车联网分布式训练模型收敛速度。

3.5 本章小结

在本章中，我们先对单回合成功车数的概率分布进行了理论分析，发现其服从一个泊松分布，其参数由公式(3.4)给出。接下来，我们刻画了一个理想情况，在该理想情况下每一回合每辆车进行本地训练得到的模型参数近似相同，由此我

算法 3.1 $H - T$ 联合优化算法

Require: $T', \tau^{\text{cm}}, \mu, \beta, \lambda, \epsilon, \gamma$ **Ensure:** H^*, T^* 初始化: $H_0 = 1, T_0 = T', k = 0$ **repeat** $k \leftarrow k + 1$ $\mathcal{H}_k \leftarrow \left\{ h \in \mathbb{N} : 1 \leq h \leq \left\lfloor \frac{\min\{T_{k-1}, T'\} - \tau^{\text{cm}}}{\mu} \right\rfloor \right\}$ $H_k \leftarrow \arg \max_{h \in \mathcal{H}_k} g(h, T_{k-1})$ $T_k^{\min} \leftarrow \mu H_k + \tau^{\text{cm}}$ $T_k^{\max} \leftarrow T^{\text{sup}}(H_k)$, 其中 $T^{\text{sup}}(\cdot)$ 由公式(3.40)给出 $T_k = \frac{1}{2} (T_k^{\min} + T_k^{\max})$ **while** $T_k^{\max} - T_k^{\min} > \gamma$ **do** **if** $\partial_T g(H_k, T_k) > 0$ **then** $T_k^{\min} \leftarrow T_k$ **else** $T_k^{\max} \leftarrow T_k$ **end if** $T_k = \frac{1}{2} (T_k^{\min} + T_k^{\max})$ **end while****until** $g(H_k, T_k) < (1 + \epsilon)g(H_{k-1}, T_{k-1})$ $H^* \leftarrow H_k$ $T^* \leftarrow T_k$

们可以对该理想情况下的车联网分布式训练优化问题进行建模，得到优化问题 $\mathcal{P}4-1$ 。之后，我们针对该优化问题设计了一个联合优化算法3.1。虽然我们的优化问题和优化算法建立在理想情景的假设下，但4.4节的相关仿真实验结果将表明，在实际应用中我们的优化算法确实行之有效。

第4章 仿真平台搭建及算法验证

4.1 引言

在第2章中，我们对车联网中分布式训练的相关问题进行介绍，刻画了一个典型的车联网分布式训练场景。在此基础上，第3章对车联网分布式训练优化问题进行建模，并针对该优化问题设计了一个联合优化算法。为了验证第3章中的理论分析与算法设计的合理性与有效性，我们需要搭建一个车联网分布式训练仿真平台，利用仿真平台进行仿真实验来模拟车联网分布式训练过程，并对相关实验结果进行分析讨论。

本章基于第2章中的车联网分布式训练场景，以模块化的方式设计了一个车联网分布式训练仿真平台（以后简称为仿真平台）。之所以采用模块化设计的方式，是因为其便于我们对该仿真平台进行拓展，如选用更为复杂的交通道路，更换目标任务、神经网络模型与数据集，采用不同的分布式训练策略等。我们已通过 Python 编程的方式完成了对该仿真平台的搭建。在本章的后续部分，将对仿真平台中调用的各个模块的设计方案进行说明，并给出其实现上的细节。设计完各个模块后，我们将通过对仿真程序的运行流程进行描述，来说明这些模块是如何构成这样一个仿真平台的。最后，我们将利用该仿真平台，针对毫米波通信中的波束选择任务与车辆自动驾驶中的轨迹预测任务，进行若干组仿真实验。基于各组仿真实验的结果，我们成功验证了第3章中的理论分析与算法设计的合理性与有效性。

4.2 仿真平台设计

仿真平台以模块化的方式进行设计，其中的各个模块分别为：交通道路及车流仿真模块，分布式训练回合表生成模块，目标任务调用模块，本地数据集配给模块，分布式训练模块以及模型性能测试模块。如图4.1所示，各模块以不同颜色的圆角矩形表示，模块间的依赖关系以有向箭头表示，箭头起点模块为箭头终点模块提供其所需的输入数据。本节将对上述模块的设计方案进行详细说明，从而为4.3节中的仿真平台总体架构设计做好铺垫。

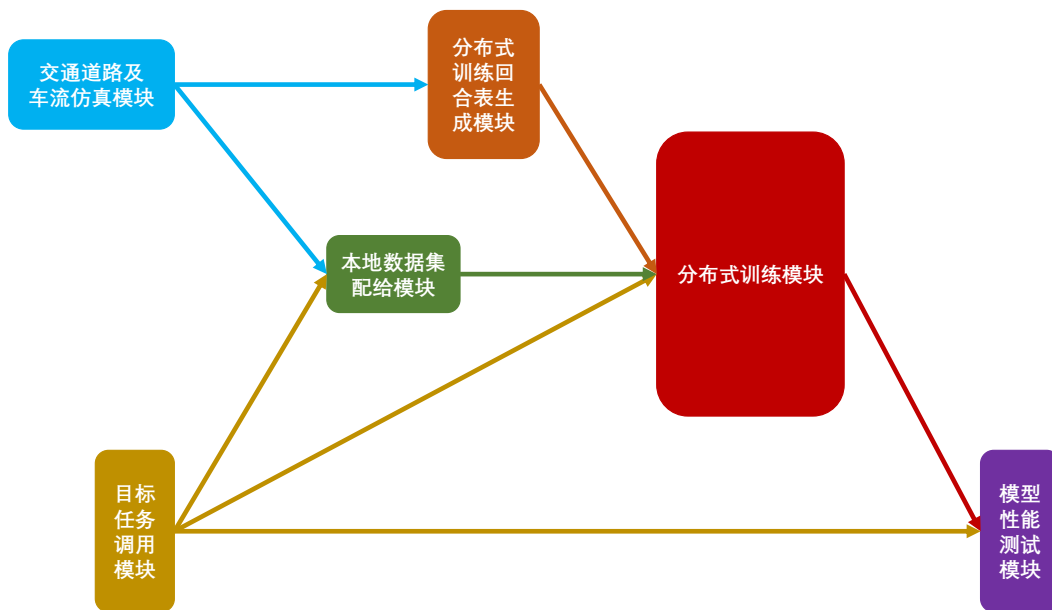


图 4.1 仿真平台各模块间关系图

4.2.1 交通道路及车流仿真模块

本模块对应于2.2节的交通道路及车流过程模型。与简单的数学建模不同的是，真实场景下，即便是在单向单车道直路上，车辆由于具有一定车长，相互之间存在最小相邻距离，这在一定程度上将导致车辆的泊松到达模型产生偏移。此外，真实场景下车辆一条直路上的行驶速度会因种种原因产生波动，最典型情况的便是在一条连接了两个十字路口的直路上，车辆会在第一个十字路口处加速，而在第二个十字路口处减速。注意到到包含上述情况在内的真实场景中交通道路及车流过程的复杂性，考虑到在分布式训练中可能需要利用车辆在各个时刻的状态信息，同时兼顾到未来可能会在其他不同的交通环境下研究车联网的分布式训练，本模块应具备精确性、可测性以及可拓展性。其中，精确性指的是能精确模拟真实场景中的交通情况，可测性指的是仿真中各车辆的状态信息能被实时观测，而可拓展性指的是能对交通道路和车流过程进行编辑与拓展。为此，我们选用 SUMO 交通系统仿真工具包对本模块进行设计与实现。

SUMO (Simulation of Urban MObility) 是一款开源、免费、能提供交通系统微观级控制的仿真软件，它由德国航空航天中心 (German Aerospace Center) 的交通系统研究所 (Institute of Transportation Systems) 开发，初次发布于 2001 年，其相应参考文献为 Lopez et al.^[20]。基于 SUMO，我们可以自定义交通道路网络，设计网络的拓扑结构，每条道路的尺寸、车道数、最高限速以及行驶方向等属性；

除此之外，我们还可以自定义各种类型的车辆，设置它们的行驶路径、最高配速、加速度、车长、最小间距等属性。更方便的是，SUMO 为 Python 提供了一个在线交互接口 TraCI (Traffic Control Interface 的缩写)，让我们能在 SUMO 的仿真过程中实时观测各个时刻的交通情况，获取交通系统中每辆车的编号、位置、速度等信息，而这些信息则是车联网分布式训练仿真过程中必不可少的部分。由于 SUMO 同时满足精确性、可测性以及可拓展性的要求，我们的仿真平台最终选用 SUMO 来实现交通道路及车流仿真模块。

在仿真平台中应用本模块时，需要给定三个 xml 格式的配置文件，我们分别以 net、rou、cfg 表示它们。其中 net 文件对交通道路网络进行配置，如设置各交通结点的位置与连接关系，道路的长度、宽度、最高限速，以及其他细节信息。而 rou 文件中存储了所有的车辆类型与路由路线，其中车辆类型指定了该类车的加速度、车长、最小车距与最高配速等属性，路由路线指定了一辆车在交通道路网络中的行驶路线。除此之外，rou 文件中还包含仿真中出现的所有车辆的基本信息，包括车辆编号、类型、路由路线、出发时刻等。最后 cfg 文件则是指定了 SUMO 仿真中用到的 net 文件和 rou 文件的文件位置，并设置了仿真的起始时刻。对于 net、rou、cfg 这三个文件，其生成方式如下：

1. net: 我们采用 SUMO 特定的图形编辑器对交通道路网络进行编辑，得到初始的 net 文件。然后我们可以基于该文件的源代码，通过 Python 脚本的文件读写操作修改其中的关键参数改变道路长度、车道限速等属性，得到我们最终需要的 net 文件。
2. rou: 根据 SUMO 规定的格式与仿真设置中的关键参数，设置一个车辆类型与一个路由路线，仿真中出现的所有车辆共用该车辆类型与路由路线，因此对于这些车辆，剩余需要指定的基本属性只有车辆编号与出发时刻。由于我们将车流到达过程建模为泊松过程，其到达率为 λ ，因此可以设置一个尽可能小的仿真步长 Δt ，对于仿真总时长以 Δt 为间隔划分为若干时隙，在每个时隙的起始时刻做成功率为 $\lambda \Delta t$ ($\lambda \Delta t \ll 1$) 的伯努利试验，若试验成功，则在 rou 文件中新增一辆车，其出发时刻为该时隙的起始时刻，车辆编号为前一辆车的编号加一（首辆车编号为 0）。由此我们可以得到所需的 rou 文件。
3. cfg: 根据 SUMO 规定的格式，设置 net 文件和 rou 文件的文件位置，并将仿真起始时刻置为 0，即可得到所需的 cfg 文件。

在给定 net、rou、cfg 文件后，便可通过 TraCI 接口运行 SUMO 仿真软件，其

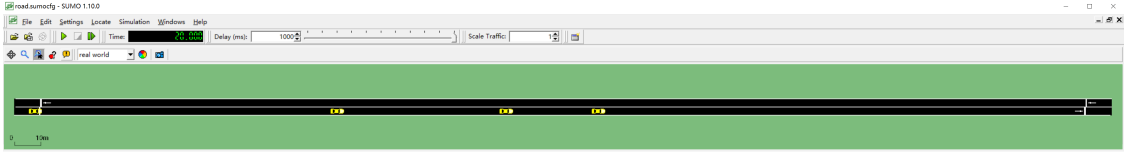


图 4.2 SUMO 图形用户界面

图形用户界面如图4.2所示。经过 SUMO 仿真后，我们可以得到仿真车流信息，其同样以 xml 文件格式存储，我们记其为 `tripinfo`。通过读取 `tripinfo` 文件信息，我们可以获取每辆车在仿真过程中的出发时刻与到达时刻，这些信息在后续模块中将得到利用。此外，对于 SUMO 仿真中的每个时隙，我们还可以记录当前道路中所有车辆的位置与速度，为车联网分布式训练的仿真提供更为精确的信息。

至此，我们完成了对交通道路及车流仿真模块的设计与实现。在我们的车联网分布式训练场景中，如果将该模块视为一个广义函数，那么其输入就是道路长度 L 、车辆到达率 λ 、车辆速度 v 和仿真总时长 T_{sim} ，输出结果则是车流信息表 $\mathcal{T}_{\text{traf}}$ 。对于 SUMO 仿真过程中出现的任意车辆 $m \in \mathcal{M}$ ，在 $\mathcal{T}_{\text{traf}}$ 中均可找到其对应的车流信息 q_m ， q_m 中又包含车辆 m 从直路入口出发的时刻 t_m^{in} 和到达直路出口的时刻 t_m^{out} 。我们可以将这一查表操作记为 $q_m = \mathcal{T}_{\text{traf}}(m)$ 。而当我们记录了每个时隙所有车辆的位置与速度信息时， q_m 中还包含 t_m 、 v_m 和 l_m ，其中 t_m 为时隙向量，它由所有车辆 m 处于这条直路上的时隙所构成，而 v_m 和 l_m 分别是与 t_m 形状相同的速度向量和位置向量，它们的向量元素分别代表相应时隙车辆 m 的速度与位置。

4.2.2 分布式训练回合表生成模块

在设计本模块之前，我们先引入分布式训练回合表这一概念。对于车联网分布式训练过程而言，每一回合参与训练的车辆都不尽相同。要对车联网分布式训练过程进行仿真，必须要确定每一回合中哪些车辆参与训练并成功完成了模型上传，而这便是分布式训练回合表所包含的信息。我们记分布式训练回合表为 \mathcal{T}_{rnd} ，对于分布式训练中的第 k 回合， $k \in \{1, \dots, K\}$ ，在 \mathcal{T}_{rnd} 中均可找到其对应的 $\mathcal{M}_k^{\text{suc}}$ ， $\mathcal{M}_k^{\text{suc}}$ 表示第 k 回合中成功完成模型上传的车辆构成的集合。我们可以将这一查表操作记为 $\mathcal{M}_k^{\text{suc}} = \mathcal{T}_{\text{rnd}}(k)$ 。

基于上述概念，我们对分布式训练回合表生成模块进行设计。该模块同样可以视为一个广义函数，其输入为车流信息表 $\mathcal{T}_{\text{traf}}$ 、分布式训练相关参数以及通信时延与计算时延的概率分布，输出结果为分布式训练回合表 \mathcal{T}_{rnd} 。下面我们以算法的形式描述该模块的设计思路。

算法 4.1 生成分布式训练回合表 \mathcal{T}_{rnd}

输入: $\mathcal{T}_{\text{traf}}, \{T_k\}_{k=1}^K, \{H_k\}_{k=1}^K, \mathcal{F}^{\text{up}}(\tau), \mathcal{F}^{\text{down}}(\tau), \mathcal{F}^{\text{cp}}(\tau; H)$

输出: \mathcal{T}_{rnd}

初始化: $t_0 = 0, t_k = \sum_{i=1}^k T_i, k = 1, \dots, K$, 由 $\mathcal{T}_{\text{traf}}$ 得到 \mathcal{M}

for $k = 1, \dots, K$ **do**

$\mathcal{M}_k^{\text{suc}} \leftarrow \emptyset$

for $m \in \mathcal{M}$ **do**

$q_m \leftarrow \mathcal{T}_{\text{traf}}(m)$

 从 q_m 中获取车辆 m 从直路入口出发的时刻 t_m^{in} 与到达直路出口的时刻 t_m^{out}

$\tau_{k,m}^{\text{th}} \leftarrow \min(t_k, t_m^{\text{out}}) - \max(t_{k-1}, t_m^{\text{in}})$

 依概率分布 $\mathcal{F}^{\text{down}}(\tau)$ 随机产生模型下发通信时延 $\tau_{k,m}^{\text{down}}$

 依概率分布 $\mathcal{F}^{\text{cp}}(\tau; H_k)$ 随机产生本地训练计算时延 $\tau_{k,m}^{\text{cp}}$

 依概率分布 $\mathcal{F}^{\text{up}}(\tau)$ 随机产生模型上传通信时延 $\tau_{k,m}^{\text{up}}$

if $\tau_{k,m}^{\text{down}} + \tau_{k,m}^{\text{cp}} + \tau_{k,m}^{\text{up}} < \tau_{k,m}^{\text{th}}$ **then**

$\mathcal{M}_k^{\text{suc}} \leftarrow \mathcal{M}_k^{\text{suc}} \cup \{m\}$

end if

end for

 将 $\mathcal{M}_k^{\text{suc}}$ 加入到 \mathcal{T}_{rnd}

end for

在算法4.1中, $\mathcal{T}_{\text{traf}}$ 为车流信息表, \mathcal{M} 为所有车辆构成的集合, K 为回合总数, T_k 为第 k 回合时长, H_k 为第 k 回合本地训练迭代次数, $\mathcal{F}^{\text{up}}(\tau)$ 为模型上传通信时延的概率分布, $\mathcal{F}^{\text{down}}(\tau)$ 为模型下发通信时延的概率分布, $\mathcal{F}^{\text{cp}}(\tau; H)$ 为本地训练迭代 H 次的计算时延的概率分布, 其形式可参照式(2.5)。该算法的目的是对于每一个回合, 找出其中所有能够参与该回合分布式训练并成功完成模型上传的车辆。在该算法中, $\tau_{k,m}^{\text{th}}$ 表示车辆 m 在第 k 回合处于训练区域 (即4.2.1节中的直路) 的时长, $\tau_{k,m}^{\text{th}}$ 既受到车辆 m 进入直路时刻 t_m^{in} 与离开直路时刻 t_m^{out} 的影响, 也受到第 k 回合的开始时刻 t_{k-1} 与结束时刻 t_k 的影响, 其具体表达式为:

$$\tau_{k,m}^{\text{th}} = \min \{t_k, t_m^{\text{out}}\} - \max \{t_{k-1}, t_m^{\text{in}}\}. \quad (4.1)$$

而车辆 m 要想在第 k 回合成功上传模型, 其通信与计算的总时延 $\tau_{k,m}^{\text{down}} + \tau_{k,m}^{\text{cp}} + \tau_{k,m}^{\text{up}}$ 必须小于 $\tau_{k,m}^{\text{th}}$, 这便是算法4.1的判定条件。需要注意的是, 式(4.1)中 $\tau_{k,m}^{\text{th}}$ 可能为负数, 此时车辆 m 在第 k 回合处于训练区域的时长等于 0 而非 $\tau_{k,m}^{\text{th}}$, 但通过合理推断我们可以发现, 在算法4.1中应用式(4.1)并不影响其正确性。

最后对算法4.1的复杂度进行简单分析。该算法的复杂度为 $\mathcal{O}(KM)$, 其中 $M = |\mathcal{M}|$ 。由于 K 与 M 均与分布式训练总时长 T_{sim} 成正相关, 因此算法4.1的复杂度也可表示为 $\mathcal{O}(T_{\text{sim}}^2)$, 可见算法4.1在复杂度方面还有进一步优化的空间。由于在实际运行仿真程序时, 该算法的执行时间远小于后续对分布式训练做仿真的运行时间, 因此本文不再对该算法做进一步优化。

基于算法4.1, 我们设计并实现了分布式训练回合表生成模块。由该模块生成的分布式训练回合表 \mathcal{T}_{rnd} 将在4.2.5节中介绍的分布式训练模块中得到利用。

4.2.3 目标任务调用模块

要对车联网分布式训练过程进行仿真, 就必须要有确定的推断算法 $f_I(\cdot)$ 、评价函数 $Q(\cdot; \cdot)$ 、神经网络模型 \mathbf{w} 、数据集 \mathcal{D} 以及损失函数 $\mathcal{L}(\cdot)$ 等分布式训练相关要素。而根据2.3节中的解释, 以上要素均取决于目标任务的选取。因此, 仿真平台需要一个目标任务调用模块, 该模块负责根据给定的目标任务返回上述相关要素, 为后续的分布式训练做准备。可以说, 对目标任务调用模块的设计与实现就是针对我们所关心的目标任务定义一系列分布式训练相关要素。

由于我们的仿真场景为车联网中的分布式训练, 因此我们选择两个与车联网背景有关的目标任务供本模块进行调用, 它们分别是毫米波通信中的波束选择任务, 以及自动驾驶中的轨迹预测任务。对于这两个目标任务, 已存在诸多可用的数据集以及评价指标。此外, 前人在非分布式训练场景下对这两个目标任务已经进行了许多研究, 设计出了诸多性能良好的基于神经网络模型的推断算法。而在我们的车联网分布式训练仿真平台中, 可以将他们的推断算法、神经网络模型、损失函数等研究成果迁移至车联网分布式训练场景, 用以设计与实现我们的目标任务调用模块。下面我们将分别介绍这两个目标任务以及调用它们所需的分布式训练相关要素。

4.2.3.1 波束选择任务

在下行链路 OFDM 毫米波通信系统中, 记发端 (基站) 波束码本集合为 \mathcal{C}_t , 收端 (车辆) 波束码本集合为 \mathcal{C}_r , 波束对信道增益为 $y_{i,j}, i \in \mathcal{C}_t, j \in \mathcal{C}_r$ 。在传统方案中, 为了选择信道条件最优的波束对, 需要进行遍历式波束搜索, 而这将导致较大的时延与信令开销。为降低遍历式波束搜索过程的开销, 车辆可以基于神经网络推断算法, 利用激光雷达、GPS 等传感器提供的数据, 在所有的 $\mathcal{C}_t \times \mathcal{C}_r$ 个波束对 $\{(i,j)\}_{i \in \mathcal{C}_t, j \in \mathcal{C}_r}$ 中, 选取 K 个波束对, 使得这 K 个波束对中包含信道条件最好的波束对 $\mathbf{b}^* = (i^*, j^*) = \arg \max_{\{(i,j)\}_{i \in \mathcal{C}_t, j \in \mathcal{C}_r}} y_{i,j}$, 然后再在这 K 个波束对中通过遍历式搜索得到信道条件最优的波束。通过该方法, 遍历式波束搜索过程的开销变为原来的 $\frac{K}{C_t \times C_r}$ 倍, 当 $K \ll C_t \times C_r$ 时, 此开销大大减少。而从这 $\mathcal{C}_t \times \mathcal{C}_r$ 个波束对中选取包含最优波束对 \mathbf{b}^* 的 K 个波束对的操作, 就是我们的波束选择任务。

根据该任务的描述, 不难看出它本质上就是一个分类问题, 因此我们可以采用分类精确度 $Q_K(\cdot; \cdot)$ 作为评价函数, $Q_K(f_I; \mathcal{D})$ 指的是推断算法 f_I 在数据集 \mathcal{D}

上执行波束选择任务时，所选 K 个波束对包含最优波束对的数据样本比例。此外，我们还可以采用分类问题中最经典的交叉熵函数作为损失函数 $\mathcal{L}(\cdot)$ 。

在数据集方面，我们选用 Raymobtime 数据集^[19]。它是 2020 国际电联人工智能/机器学习 5G 挑战赛的官方数据集，包括了激光雷达、相机图片、GPS 以及各波束方向的信道信息等数据，用以训练用于毫米波波束选择任务的神经网络模型。

最后，我们采用文献[21]中的推断算法与神经网络模型。该算法的思路是将激光雷达的点云数据和 GPS 的坐标数据经过预处理操作 f_P 转为一个二维特征图 X ，然后将其输入到由多个二维卷积层、全连接层序贯排列构成的神经网络模型 f_N 中， f_N 最终通过 Softmax 激活函数输出一个特征向量 Y ， Y 的每个分量对应一个候选波束对的置信度，而后处理操作 f_B 则根据 Y 选择置信度最高的 K 个波束对作为推断算法的输出结果。

综上我们确定了波束选择任务的分布式训练相关要素，并在仿真平台的目标任务调用模块中实现了这部分内容。

4.2.3.2 轨迹预测任务

人们在开车时，经常需要通过预判其他车或行人在未来几秒的行动路线来调整自己的驾驶行为，而自动驾驶车辆也有类似的需求。在自动驾驶场景中，车辆 c_m 根据周围物体（包括车辆、自行车、行人等）在过去一段时间的轨迹信息，结合自身当前所处环境的地图信息，推断未来一段时间内某个目标车辆 c_n 的行驶轨迹，这便是轨迹预测。虽然轨迹本身是连续的，但在实际应用中，通常采用离散化处理，即将轨迹以固定时间间隔 Δt 采样表示。在离散化处理下，我们常常将车辆 c_n 的历史轨迹表示为 $\left[\left(x_{-T_p}^n, y_{-T_p}^n \right), \dots, \left(x_{-1}^n, y_{-1}^n \right) \right]$ ，其中 (x_{-i}^n, y_{-i}^n) 表示车辆 c_n 在过去 $i\Delta t$ 时刻的坐标。类似的，车辆 c_n 的未来轨迹可以表示为 $\left[\left(x_1^n, y_1^n \right), \dots, \left(x_{T_f}^n, y_{T_f}^n \right) \right]$ ，其中 (x_i^n, y_i^n) 表示车辆 c_n 在未来 $i\Delta t$ 时刻的坐标。为表示方便，我们将目标车辆的历史轨迹记为 \mathbf{x} ，真实的未来轨迹记为 \mathbf{y} ，预测的未来轨迹记为 $\hat{\mathbf{y}}$ 。

轨迹预测任务可以有多个评价指标，常见的有平均偏移误差（Average Displacement Error, ADE）、终点偏移误差（Final Displacement Error, FDE）以及未命中率（Missing Rate, MR）等。其中，对于一条真实未来轨迹 \mathbf{y} 和预测未来轨迹 $\hat{\mathbf{y}}$,

平均偏移误差可表示为:

$$Q_{\text{ADE}}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{T_f} \cdot \sum_{i=1}^{T_f} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]}, \quad (4.2)$$

终点偏移误差可表示为:

$$Q_{\text{FDE}}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{(x_{T_f} - \hat{x}_{T_f})^2 + (y_{T_f} - \hat{y}_{T_f})^2}. \quad (4.3)$$

而未命中率则可以表示为 $Q_{\text{MR}}(f_I; \mathcal{D}, d)$, 它指的是推断算法 f_I 在数据集 \mathcal{D} 上执行轨迹预测任务时, 终点偏移误差小于 d 的数据样本比例。以上是单模轨迹预测的评价指标, 而针对多模轨迹预测, 即对目标车辆预测 K 条轨迹 $\{\hat{\mathbf{y}}^k\}_{k=0}^{K-1}$, 还可以定义多模评价指标, 如 ADE_K 、 FDE_K 以及 MR_K 。其中, FDE_K 可表示为:

$$Q_{\text{FDE}_K}(\mathbf{y}, \{\hat{\mathbf{y}}^k\}_{k=0}^{K-1}) = \min_k Q_{\text{FDE}}(\mathbf{y}, \hat{\mathbf{y}}^k), \quad (4.4)$$

ADE_K 可表示为:

$$Q_{\text{ADE}_K}(\mathbf{y}, \{\hat{\mathbf{y}}^k\}_{k=0}^{K-1}) = Q_{\text{ADE}}(\mathbf{y}, \hat{\mathbf{y}}^{k^*}), k^* = \arg \min_k Q_{\text{FDE}}(\mathbf{y}, \hat{\mathbf{y}}^k). \quad (4.5)$$

而 MR_K 则可以表示为 $Q_{\text{MR}_K}(f_I; \mathcal{D}, d)$, 它指的是推断算法 f_I 在数据集 \mathcal{D} 上执行多模轨迹预测任务时, FDE_K 小于 d 的数据样本比例。

在数据集方面, 我们选用 Argoverse Motion Forecasting 数据集^[16]。它是一个包含 323,557 个场景片段的数据集, 每个场景片段时长 5 秒, 采样频率为 10Hz, 包含每个跟踪对象的二维质心运动轨迹。该数据集还配备了对应于各个场景的高清地图, 可从中提取丰富的几何信息和语义信息, 便于模型更好地理解当前的自动驾驶环境。

最后, 我们采用文献[22]中的多模轨迹预测推断算法、神经网络模型与损失函数。在一个场景片段中, 该算法将车辆的历史轨迹通过 ActorNet (由一维卷积神经网络与特征金字塔网络构成) 转为车辆特征输入 X_{act} , 将车辆周围的地图数据通过 MapNet (由文献[22]设计的图卷积神经网络 LaneGCN 构成) 转为地图特征输入 X_{map} , 接着将 X_{act} 和 X_{map} 输入到 FusionNet (由空间注意力模块与 LaneGCN 构成) 进行特征融合处理得到融合后的车辆特征 \tilde{X}_{act} , 最后将 \tilde{X}_{act} 输入到 Header (包含一个回归分支与一个分类分支, 二者都为全连接网络) 得到预测结果 Y , Y 中包含对目标车辆的 K 条预测轨迹以及其相应的置信度。由于多模轨迹预测同时包含回归与分类, 损失函数 $\mathcal{L}(\cdot)$ 也应同时包含回归损失与分类损失, 即 $\mathcal{L}(\cdot) = \mathcal{L}_{\text{reg}}(\cdot) + \alpha \mathcal{L}_{\text{cls}}(\cdot)$, 其中 α 为权衡系数, 仿真实验中将其置为 1。

综上我们确定了轨迹预测任务的分布式训练相关要素，并在仿真平台的目标任务调用模块中实现了这部分内容。

4.2.4 本地数据集配给模块

在我们设想的车联网分布式训练场景中，车辆的本地数据由每辆车自行实时生成。然而，在做仿真平台的实现时，我们无法做到让仿真程序中的车像在现实环境下那样通过传感器实时采集与生成本地数据。作为替代，我们只能基于现有的数据集，按照特定的采样规则从中采取部分数据作为车辆的本地数据集。因此，我们需要设计一个本地数据集配给模块，它负责基于给定的总训练数据集为每辆车分配一个本地数据集。

我们记总训练数据集为 D_{train} ，车辆集合为 \mathcal{M} ，车辆 m ($m \in \mathcal{M}$) 的本地数据集为 D_m 。其中，总训练数据集 D_{train} 由目标任务调用模块给出，车辆集合 \mathcal{M} 由交通道路及车流仿真模块给出，而 D_m 则由本模块给出。对车辆 m 分配本地数据集 D_m ，除了需要一个总训练数据集 D_{train} ，还需要给定如下参数：

1. 本地数据集大小 D_m ：从 D_{train} 中抽取 D_m 个数据样本，作为车辆 m 的本地数据集 D_m 。在仿真平台中，默认每辆车的本地数据集大小相等，因此只需给定一个统一的本地数据集大小 D_{local} 即可。通过修改本模块，还可以做到为每辆车分配大小不等的本地数据集。
2. 采样概率分布函数 $P_s(\cdot)$ ：本模块基于概率分布函数 $P_s(\cdot)$ 从 D_{train} 中独立同分布地抽取数据样本， $P_s(d)$ 表示数据样本 d ($d \in D_{\text{train}}$) 被抽中的概率。在仿真平台中，默认以随机均匀采样的策略抽取数据样本。通过修改本模块，还可以使用其他概率分布对数据样本进行抽取，这对于研究数据非独立同分布 (Non-IID) 的分布式训练过程而言是十分必要的。

需要注意的是，对于 \mathcal{M} 中的两辆车 m_1 和 m_2 ，本模块分配给它们的本地数据集可能有重叠部分，即 $D_{m_1} \cap D_{m_2} \neq \emptyset$ ，而这在实际的车联网场景下几乎不可能发生。

综上我们设计并实现了本地数据集配给模块。由本模块生成的车辆本地数据集 $\{D_m\}_{m \in \mathcal{M}}$ 将在4.2.5节中介绍的分布式训练模块中得到利用。

4.2.5 分布式训练模块

基于2.4节中介绍的车联网分布式训练框架，我们可以为仿真平台设计一个分布式训练模块，对车联网场景下的分布式训练过程进行仿真。该模块是仿真平台中的核心模块，其他模块都直接或间接地为其提供服务，且仿真程序在该模块上

的运行时间远高于其他模块。

算法 4.2 分布式训练模块算法

输入: $\mathcal{M}, \mathcal{T}_{\text{rnd}}, \{D_m\}_{m \in \mathcal{M}}, f_I(\cdot), w_0, \mathcal{L}(\cdot), \eta, B, H$

输出: $\{w_k\}_{k=1}^K$

初始化: 由 \mathcal{T}_{rnd} 得到总回合数 K

for $k = 0, \dots, K - 1$ **do**

 由 \mathcal{T}_{rnd} 得到第 k 回合完成模型上传的车辆集合 \mathcal{M}_k 以及该集合大小 M_k

$\mathcal{W}_k \leftarrow \emptyset$

for $m \in \mathcal{M}_k$ **do**

$w_{k,m}^{[0]} \leftarrow w_k$

for $h = 1, \dots, H$ **do**

 从 D_m 中随机均匀抽取 B 个数据样本作为批训练数据 $B_{k,m}^{[h]}$

$w_{k,m}^{[h]} \leftarrow w_{k,m}^{[h-1]} - \eta \cdot \nabla F(w_{k,m}^{[h-1]}, B_{k,m}^{[h]})$, $F(\cdot, \cdot)$ 由公式(2.1)给出

end for

$\mathcal{W}_k \leftarrow \mathcal{W}_k \cup \{w_{k,m}^{[H]}\}$

end for

$w_{k+1} \leftarrow \frac{1}{M_k} \sum_{w \in \mathcal{W}_k} w$

end for

我们通过算法4.2对分布式训练模块的设计思路进行描述。对分布式训练模块进行调用时, 需要给出以下输入, 它们和算法4.2的输入一一对应:

1. 车辆集合 \mathcal{M} : 由交通道路及车流仿真模块给出。
2. 分布式训练回合表 \mathcal{T}_{rnd} : 由分布式训练回合表生成模块给出。
3. 所有车辆的本地数据集 $\{D_m\}_{m \in \mathcal{M}}$: 由本地数据集配给模块给出。
4. 推断函数 $f_I(\cdot)$: 由目标任务调用模块给出。
5. 神经网络模型 w : 由目标任务调用模块给出, 对其参数进行初始化后得到初始全局神经网络模型 w_0 。
6. 损失函数 $\mathcal{L}(\cdot)$: 由目标任务调用模块给出。
7. 学习率 η : 仿真参数之一, 在执行仿真程序前人为给定。
8. 批大小 B : 仿真参数之一, 在执行仿真程序前人为给定。
9. 本地训练迭代次数 H : 仿真参数之一, 在执行仿真程序前人为给定。

在完成如算法4.2所示的步骤后, 分布式训练模块最终会返回各个回合的全局神经网络模型 $\{w_k\}_{k=1}^K$ 。而对于这些模型性能的评价, 将由之后介绍的模型性能测试模块完成。

4.2.6 模型性能测试模块

在分布式训练过程中, 我们需要对全局神经网络模型的性能做出评价, 以此评估分布式训练的训练速度、收敛性以及可达的模型最优性能, 这便是模型性能

测试模块负责的任务。如果将本模块视为一个广义函数，那么其输入为推断算法 $f_I(\cdot)$ 、评价函数 $Q(\cdot; \cdot)$ 、损失函数 $\mathcal{L}(\cdot)$ 、验证数据集 \mathcal{D}_{val} 以及神经网络模型参数 \mathbf{w} ，输出为以 \mathbf{w} 为神经网络模型参数的推断算法 $f_I(\cdot)$ 在验证数据集 \mathcal{D}_{val} 上的评价指标 $Q(f_I; \mathcal{D}_{\text{val}})$ 。其中， $f_I(\cdot)$ 和 $Q(\cdot; \cdot)$ 由目标任务调用模块给定， \mathcal{D}_{val} 取为目标任务调用模块给定的验证数据集， \mathbf{w} 则取为第 k 回合的全局网络模型参数 \mathbf{w}_k 。

在仿真过程中，分布式训练模块每更新一次全局网络模型参数，便调用一次模型性能测试模块，得到当前回合的模型性能评价指标以及验证损失。经过 K 个回合的分布式训练，共得到 K 组评价指标 $\{Q_k\}_{k=1}^K$ 与验证损失 $\{F_k\}_{k=1}^K$ 。基于 $\{Q_k\}_{k=1}^K$ 与 $\{F_k\}_{k=1}^K$ ，仿真平台可以自动生成分布式训练过程图，让我们直观地看到模型性能在每一回合的变化过程。

4.3 仿真平台总体架构与程序运行流程

我们在4.2节中针对仿真平台所需的各个模块进行了设计与实现。在本节中，我们将基于图4.3，从总体架构的角度详细介绍各个模块之间的相互关系，并基于仿真程序的运行流程描述如何调用这些模块以搭建我们的仿真平台。

在图4.3中，各个模块以不同颜色的圆角矩形表示，仿真程序的输入参数统一放在图片最左侧的虚线框中，所有的有向连线表示仿真数据的流动方向，其中部分仿真数据被多个模块所利用，我们使用分叉连线的形式表示这一关系。对于一次仿真实验所求的结果，即各回合的全局模型参数 \mathbf{w}_k 和对应的评价指标 Q_k ，我们统一放在图片最右侧。

虽然图4.3中各模块间的相互依赖关系看似错综复杂，但是在代码实现中，我们的仿真程序是逐个调用这些模块的，而其调用顺序与4.2节中的介绍顺序相同。下面我们将描述在一次仿真实验中，仿真程序的运行流程：

1. 仿真程序读取所有仿真参数后，首先调用交通道路及车流仿真模块，基于道路长度 L 、车辆到达率 λ 、车辆速度 v 和仿真总时长 T_{sim} ，生成车流信息表 $\mathcal{T}_{\text{traf}}$ 与车辆集合 \mathcal{M} 。
2. 调用分布式训练回合表生成模块，基于车流信息表 $\mathcal{T}_{\text{traf}}$ 、车辆集合 \mathcal{M} 、分布式训练回合时长 T 、本地训练迭代次数 H ，以及通信时延与计算时延的概率分布 \mathcal{F}^{up} 、 $\mathcal{F}^{\text{down}}$ 、 \mathcal{F}^{cp} ，生成分布式训练回合表 \mathcal{T}_{rnd} 。
3. 调用目标任务调用模块，根据指定的目标任务 $task$ ，调出数据集 \mathcal{D} 、推断函数 $f_I(\cdot)$ 及其使用的神经网络模型 \mathbf{w} 、损失函数 $\mathcal{L}(\cdot)$ 以及评价函数 $Q(\cdot; \cdot)$ 。

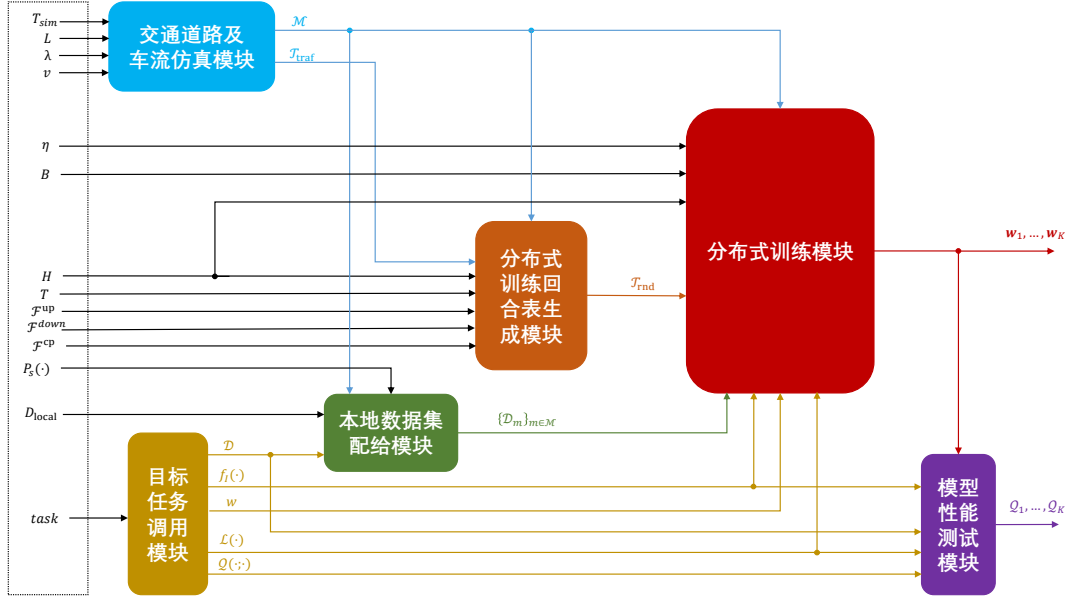


图 4.3 仿真平台总体架构

4. 调用本地数据集配给模块，基于训练数据集 $\mathcal{D}_{\text{train}}$ （由 \mathcal{D} 划分出的大部分数据样本构成）、本地数据集大小 D_{local} 、车辆集合 \mathcal{M} 以及采样概率分布函数 $P_s(\cdot)$ ，生成每辆车的本地数据集 $\{\mathcal{D}_m\}_{m \in \mathcal{M}}$ 。
5. 调用分布式训练模块，基于车辆集合 \mathcal{M} 、分布式训练回合表 \mathcal{T}_{rnd} 、每辆车的本地数据集 $\{\mathcal{D}_m\}_{m \in \mathcal{M}}$ 、推断函数 $f_I(\cdot)$ 、神经网络模型 w 、损失函数 $\mathcal{L}(\cdot)$ 、学习率 η 、批大小 B 以及本地训练迭代次数 H ，以算法4.2对分布式训练进行仿真，得到各个回合的全局模型参数 $\{w_k\}_{k=1}^K$ 。
6. 调用模型性能测试模块，基于推断函数 $f_I(\cdot)$ 、损失函数 $\mathcal{L}(\cdot)$ 、评价函数 $Q(\cdot; \cdot)$ ，在验证数据集 \mathcal{D}_{val} （由 \mathcal{D} 中划分出的小部分数据样本构成，与 $\mathcal{D}_{\text{train}}$ 正交）上对各回合的全局模型参数 $\{w_k\}_{k=1}^K$ 进行性能评价，得到它们对应的性能评价指标 $\{Q_k\}_{k=1}^K$ 和验证损失 $\{F_k\}_{k=1}^K$ 。
7. 仿真程序对 $\{Q_k\}_{k=1}^K$ 和 $\{F_k\}_{k=1}^K$ 进行作图，生成分布式训练过程图，并对相关数据进行保存。

基于上述流程以及4.2节中对各模块的设计，我们可以从代码层面搭建车联网分布式训练仿真平台。利用该仿真平台，我们可以在不同参数设置下进行仿真实验，研究车联网分布式训练的特性与影响因素。

4.4 仿真实验结果

本节基于前文设计并实现的车联网分布式训练仿真平台，在不同仿真参数设置下进行了各组仿真实验，根据实验结果研究车联网分布式训练收敛速度的影响因素。在本节中，我们先展示了两个目标任务在典型参数设置下的分布式训练过程图，验证车联网分布式训练仿真平台设计与实现的正确性。接下来，我们针对分布式训练回合时长 T 以及本地训练迭代次数 H 这两个参数，应用控制变量法进行了多组仿真实验，基于实验结果解释 T 和 H 如何对分布式训练的模型收敛速度造成影响。最后，我们基于第3章中提出的优化问题及优化算法，分别进行了相关实验，验证了优化问题建模与优化算法设计的合理性与有效性。

需要说明的是，我们对一次分布式训练过程的模型收敛速度进行观测时，其观测指标为该分布式训练过程中评价指标在所有回合中的最优值。对于波束选择任务，我们观测最小验证损失和最高 Top-K 准确度；对于轨迹预测任务，我们观测最小验证损失、最低 minADE(K=6) 以及最低 minFDE(K=6)。其中 Top-K 准确度对应于4.2.3.1节中的 Q_K 评价函数，该指标越高模型性能越好；minADE(K=6) 和 minFDE(K=6) 分别对应于4.2.3.2节中的 Q_{ADE_K} 和 Q_{FDE_K} 评价函数，该指标越低模型性能越好。

4.4.1 典型参数下的分布式训练过程图

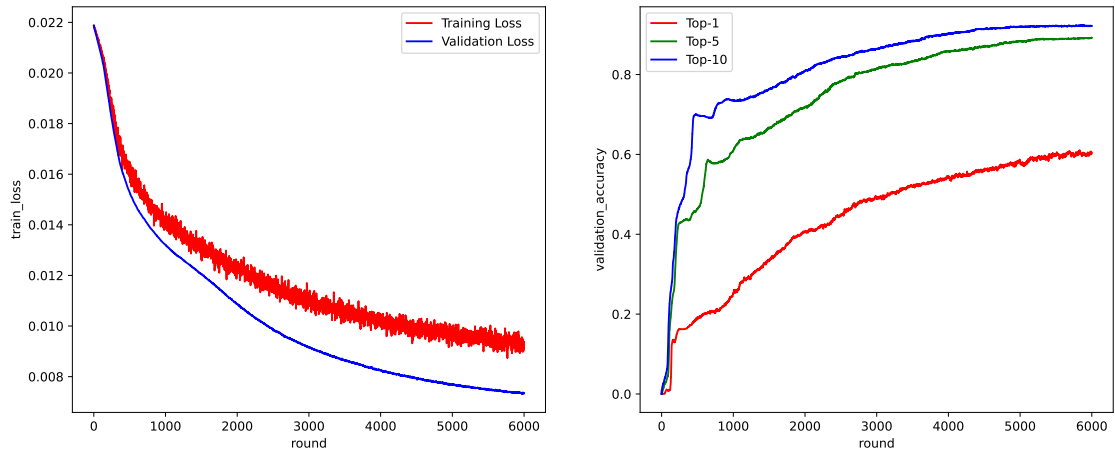


图 4.4 针对波束选择任务的分布式训练过程图

图4.4和图4.5分别是针对波束选择任务和轨迹预测任务在典型参数设置下的分布式训练过程图。由于两种任务的差异性，它们的典型参数选取并不相同，其具体取值详见表4.1。在图4.4中，左子图为波束选择任务下训练损失与验证损失

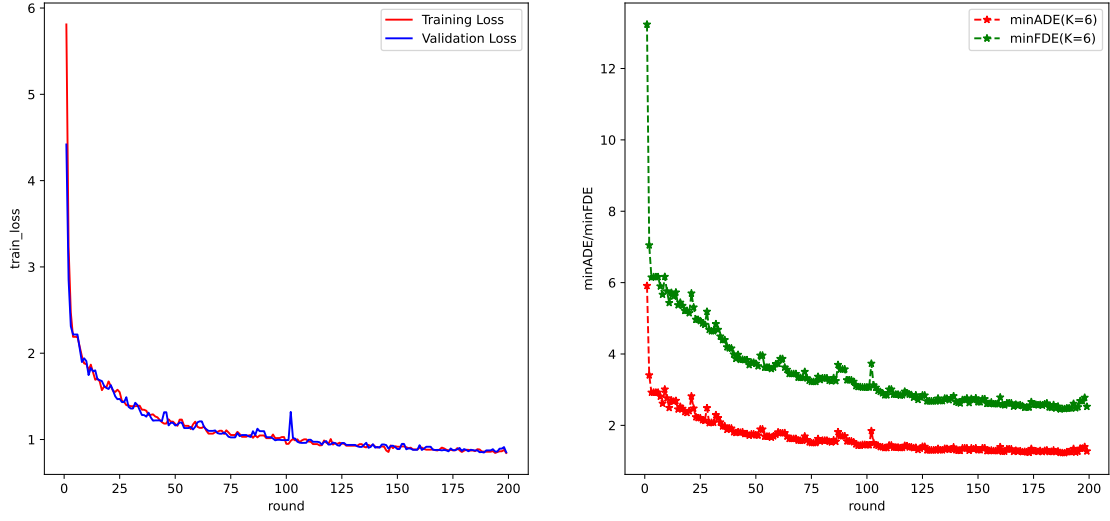


图 4.5 针对轨迹预测任务的分布式训练过程图

每回合的变化过程，右子图为 Top-1/Top-5/Top-10 准确度每回合的变化过程。在图4.5中，左子图为轨迹预测任务下训练损失与验证损失每回合的变化过程，右子图为 minADE(K=6) 与 minFDE(K=6) 指标每回合的变化过程。

可以看到，随着分布式训练的进行，训练损失/验证损失逐步降低，评价指标逐步变优，经过充分多个回合后，全局模型趋于收敛，可见我们的车联网分布式训练框架适用于这两个目标任务及其对应的神经网络模型。

4.4.2 分布式训练回合时长 T 的影响

在表4.1的典型参数设置的基础上，通过改变分布式训练回合时长 T ，我们针对波束选择任务与轨迹预测任务分别进行了两组实验，实验结果如表4.2和表4.3所示。可以看到，无论是在波束选择任务还是轨迹预测任务下， T 的变化都会影响分布式训练过程的模型收敛速度，且 T 过大或过小均会对其造成不利影响。对于该现象，结合我们的车联网分布式训练框架，可以给出一个较为直观的解释。在其他参数不变的情况下，当 T 增大时，总训练回合数 $\left\lfloor \frac{T_{\text{sim}}}{T} \right\rfloor$ 减小，导致全局模型参数的更新次数减小，从而使得分布式训练变慢；而当 T 减小时，每个回合内路侧单元收到的模型数量的期望 $\mathbb{E}[M^{\text{suc}}]$ 减小，也会影响分布式训练的收敛速度。可见 T 相当于一个权衡参数，在 $\left\lfloor \frac{T_{\text{sim}}}{T} \right\rfloor$ 与 $\mathbb{E}[M^{\text{suc}}]$ 间做权衡折中，并且在其他参数不变的情况下， T 应该存在一个或多个最优值，使得一次分布式训练过程的模型收敛速度最快。该现象及其解释与第3章中的理论分析相吻合。

表 4.1 分布式训练典型参数设置

目标任务	波束选择	轨迹预测
总仿真时长 T_{sim}	60000	2000
道路长度 L	420	420
车辆到达率 λ	0.1	0.1
车速 v	20	20
本地数据集大小 D_{local}	1024	1024
回合时长 T	10	10
本地训练迭代次数 H	10	30
批大小 B	64	64
学习率 η	0.05	0.01
模型下发通信时延 τ^{down}	1	1
模型上传通信时延 τ^{up}	1	1
偏移负指数分布参数 μ	0.2	0.1
偏移负指数分布参数 β	0.2	0.1

表 4.2 波束选择任务分布式训练受 T 影响

T	最小验证损失	最高 Top-1 准确度	最高 Top-5 准确度	最高 Top-10 准确度
4.5	0.00756	59.54%	88.93%	92.09%
5	0.00704	62.45%	89.64%	92.81%
10	0.00731	60.97%	89.18%	92.35%
20	0.00883	51.22%	82.65%	87.70%

4.4.3 本地训练迭代次数 H 的影响

在表4.1的典型参数设置的基础上,通过改变本地训练迭代次数 H ,我们针对波束选择任务与轨迹预测任务分别进行了两组实验,实验结果如表4.4和表4.5所示。与4.4.2节中的 T 类似, H 的变化会影响分布式训练过程的模型收敛速度,且 H 过大或过小均会对其造成不利影响。我们同样可以对该现象给出直观解释。在其他参数不变的情况下,当 H 增大时,每辆车进行本地训练的计算时延变长,更有可能上传模型失败,因此每个回合内路侧单元收到的模型数量的期望 $\mathbb{E}[M^{\text{suc}}]$ 变小,这对分布式训练的收敛速度不利;而当 H 减小时,每回合车辆的本地模

表 4.3 轨迹预测任务分布式训练受 T 影响

T	最小验证损失	最低 minADE(K=6)	最低 minFDE(K=6)
6	0.8693	1.267	2.538
10	0.8488	1.226	2.452
20	0.9151	1.305	2.709

表 4.4 波束选择任务分布式训练受 H 影响

H	最小验证损失	最高 Top-1 准确度	最高 Top-5 准确度	最高 Top-10 准确度
5	0.00938	47.81%	81.07%	85.82%
10	0.00731	60.97%	89.18%	92.35%
20	0.00673	63.78%	90.10%	93.06%
30	0.00699	62.55%	89.80%	92.76%

型更新次数变少，本地模型参数变化更慢，从而导致每回合全局模型参数的变化 $\Delta \mathbf{w}$ 更慢，影响分布式训练的收敛速度。可见 H 同样是一个权衡参数，在 $\mathbb{E}[M^{\text{suc}}]$ 与 $\Delta \mathbf{w}$ 间做权衡折中，并且在其他参数不变的情况下， H 应该存在一个或多个最优值，使得一次分布式训练过程的模型收敛速度最快。该现象及其解释同样与第3章中的理论分析相吻合。

4.4.4 优化问题建模有效性与普适性检验

我们先以波束选择任务为例，在表4.1中的典型参数设置的基础上，通过改变回合时长 T ，进行多次仿真实验。如图4.6所示，我们比较在回合时长 T 取不同值时，经过相同总时长 T_{sim} 的分布式训练得到的最优模型性能指标与优化问题 P4-1 目标函数 $g(H, T)$ 的变化关系。图4.6(a)中每个蓝色星形点代表一次仿真实验中最优全局神经网络模型在验证集上的平均损失函数，其横坐标为回合时长 T ，纵坐标为损失函数值。图4.6(b)中每个蓝色/绿色/红色星形点代表一次仿真实验中最优全局神经网络模型在验证集上的 Top-10/Top-5/Top-1 准确度，其横坐标为回合时长 T ，纵坐标为分类准确度。而这两张子图中的紫色曲线代表了优化问题 P4-1 的目标函数 $g(H, T)$ 在其他参数不变的情况下随回合时长 T 的变化情况。

一般情况下，波束选择任务的神经网络模型性能越好，其验证损失函数越小，Top-K 准确度越高。而通过观察图4.6(a)和图4.6(b)可以发现，经过分布式训练得到的最优模型的验证损失函数与 $g(H, T)$ 间存在显著的负相关关系，Top-K 准确

表 4.5 轨迹预测任务分布式训练受 H 影响

H	最小验证损失	最低 minADE(K=6)	最低 minFDE(K=6)
15	0.9396	1.356	2.812
30	0.8511	1.237	2.474
45	0.8444	1.238	2.455
60	0.8839	1.257	2.516

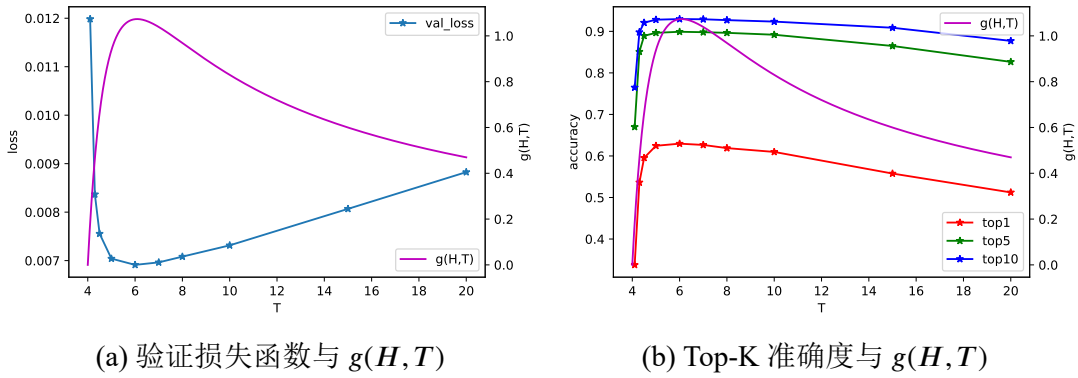
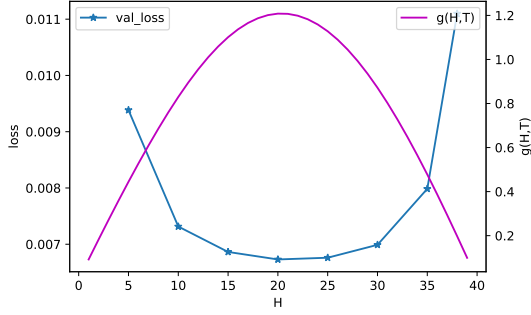


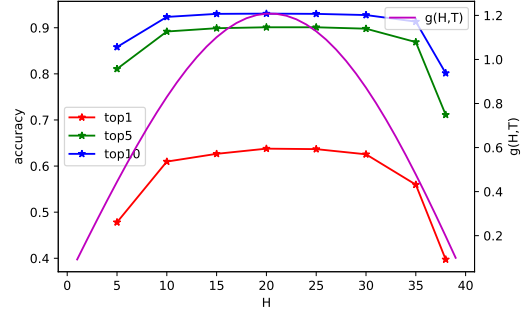
图 4.6 波束选择任务下的最优模型性能指标与优化问题 $\mathcal{P}4-1$ 的目标函数 $g(H, T)$ 随回合时长 T 的变化关系

度与 $g(H, T)$ 间存在显著的正相关关系，二者都表明该组实验中最优模型的性能与 $g(H, T)$ 正相关。由于该组实验中各次实验的总时长 T_{sim} 相同，我们可以推知该组实验中的模型收敛速度与 $g(H, T)$ 正相关。此外，我们还在表4.1中的典型参数设置的基础上，通过改变本地训练迭代次数 H ，进行了另一组仿真实验。如图4.7所示，该组实验中的模型收敛速度同样与 $g(H, T)$ 正相关。

为了验证优化问题 $\mathcal{P}4-1$ 的普适性，我们还以轨迹预测任务为例，同样在表4.1中的典型参数设置的基础上，分别进行了两组仿真实验，一组仅改变回合时长 T ，另一组仅改变本地训练迭代次数 H ，观察最优模型验证损失函数与优化问题 $\mathcal{P}4-1$ 的目标函数 $g(H, T)$ 间的相关关系，实验结果如图4.8所示。两张子图中每个蓝色星形点代表一次仿真实验中最优全局神经网络模型在验证集上的平均损失函数，其横坐标在图4.8(a)中为回合时长 T ，在图4.8(b)中为本地训练迭代次数 H ，纵坐标为损失函数值。紫色曲线在两张子图中分别代表 $g(H, T)$ 在其他参数不变的情况下随回合时长 T 和本地训练迭代次数 H 的变化过程。可以看到，在轨迹预测任务中，模型收敛速度同样与 $g(H, T)$ 有较为明显的正相关关系。

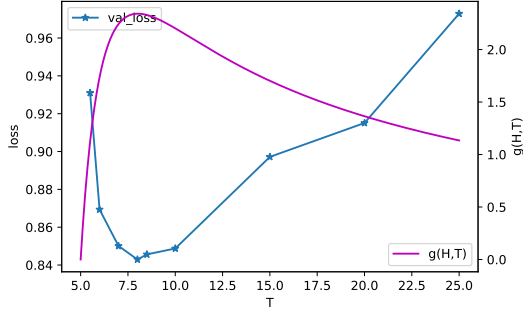


(a) 验证损失函数与 $g(H, T)$

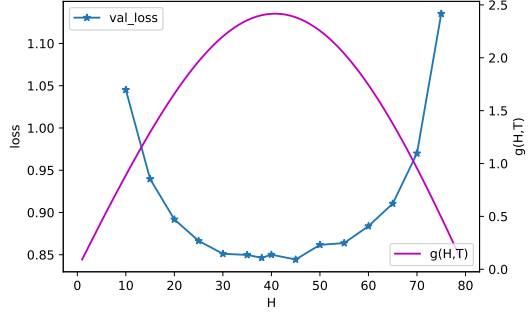


(b) Top-K 准确度与 $g(H, T)$

图 4.7 波束选择任务下的最优模型性能指标与优化问题 $\mathcal{P4-1}$ 的目标函数 $g(H, T)$ 随本地训练迭代次数 H 的变化关系



(a) 改变回合时长 T



(b) 改变本地训练迭代次数 H

图 4.8 轨迹预测任务下的最优模型验证损失函数与优化问题 $\mathcal{P4-1}$ 的目标函数 $g(H, T)$ 间的变化关系

根据上述在波束选择任务和轨迹预测任务下进行的若干组实验，以及两种目标任务间的差异性，我们可以合理推知优化问题 $\mathcal{P4-1}$ 具有一定的普适性，它不仅适用于波束选择任务和轨迹预测任务的神经网络模型的分布式训练，还能适用于其他目标任务下的神经网络模型的分布式训练。

最后需要指出的是，在两个目标任务的典型参数设置中，车辆本地数据集大小 $D_{\text{local}} = 1024$ ，每次本地迭代的批大小 $B = 64$ ，这两个参数都足够大，可以近似认为每辆车的本地数据集足够丰富，车辆进行本地训练时模型参数的更新较为稳定，因而近似满足情况 1 的假设条件。而如果 D_{local} 和 B 较小，或者不同车辆间的本地数据分布差异较大，那么就无法近似满足情况 1 的假设条件。在该情况下，优化问题 $\mathcal{P4-1}$ 的建模不再适用，如果进行与上述实验类似的实验，其结果将会有所不同。

4.4.5 算法性能验证

为了验证3.4节中优化算法3.1设计的合理性与有效性，我们应用该算法，先在波束选择任务的典型参数设置下得到了 (H, T) 的一个近似最优解 $(H_{\text{beam}}^*, T_{\text{beam}}^*)$ ，其中 $H_{\text{beam}}^* = 24$, $T_{\text{beam}}^* = 11.8$ 。接着，我们在 $(H_{\text{beam}}^*, T_{\text{beam}}^*)$ 下执行一次仿真实验，并将其结果与之前的一组仿真实验结果（对应于图4.6）进行比较。如图4.9(a)所示，其中每个蓝色星形点代表在 $H = 10$ 时，在不同回合时长 T 下经过分布式训练得到的最优模型的验证损失函数值，我们称该组实验为对照组，而红色虚线则代表在 $H = H_{\text{beam}}^*, T = T_{\text{beam}}^*$ 时，经过分布式训练得到的最优模型的验证损失函数值，我们称该次实验为优化算法实验，其中红色虚线的横坐标无意义。类似的，图4.9(b)中还比较了优化算法实验与对照组实验的Top-K准确度，其中紫色/绿色/红色星形点代表在 $H = 10$ 时，在不同回合时长 T 下经过分布式训练得到的最优模型的Top-10/Top-5/Top-1准确度，紫色/绿色/红色虚线代表在 $H = H_{\text{beam}}^*, T = T_{\text{beam}}^*$ 时，经过分布式训练得到的最优模型的Top-10/Top-5/Top-1准确度。以上各次实验的总训练时长 T_{sim} 相同。可以看到，得益于算法3.1的优化，经过分布式训练得到的最优模型的性能有了一定提升。

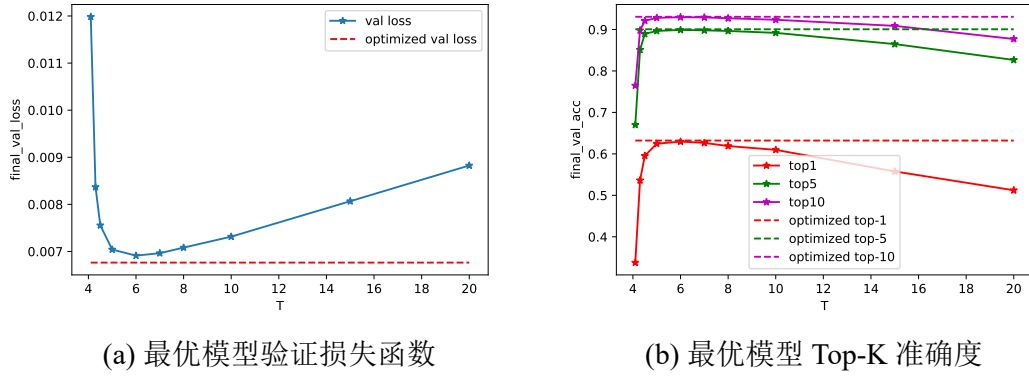
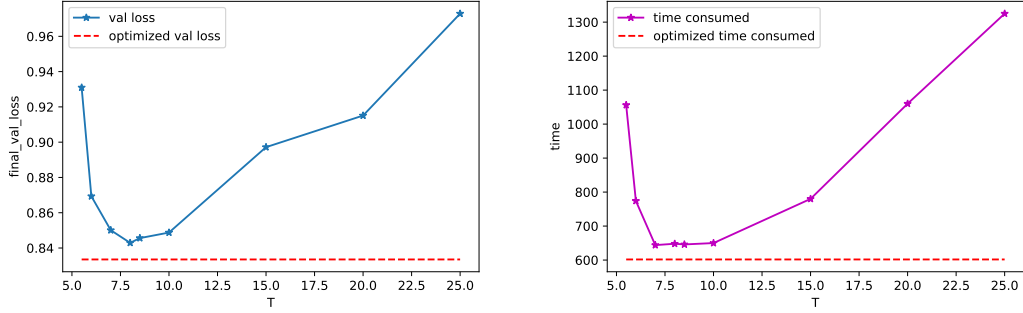


图 4.9 波束选择任务下优化算法效果对比

此外，我们还在轨迹预测任务的典型参数设置下得到了 (H, T) 的一个近似最优解 $(H_{\text{traj}}^*, T_{\text{traj}}^*)$ ，其中 $H_{\text{traj}}^* = 48$, $T_{\text{traj}}^* = 11.8$ 。我们同样基于 $(H_{\text{traj}}^*, T_{\text{traj}}^*)$ 进行了一次优化算法实验，并将其结果与之前的一组仿真实验结果（对应于图4.8(a)）进行比较。如图4.10(a)所示，其中蓝色星形点代表在 $H = 30$ 时，在不同回合时长 T 下经过分布式训练得到的最优模型的验证损失函数值，而红色虚线则代表在 $H = H_{\text{traj}}^*, T = T_{\text{traj}}^*$ 时，经过分布式训练得到的最优模型的验证损失函数值。此外，图4.10(b)中还比较了优化算法实验与对照组实验中，全局模型验证损失函数



(a) 最优模型验证损失函数

(b) 模型验证损失首次低于阈值所耗时间

图 4.10 轨迹预测任务下优化算法效果对比

值首次小于给定阈值（这里设为 1.1）时所消耗的时间。其中紫色星形点代表在 $H = 30$ 时，在不同回合时长 T 下全局模型验证损失函数值首次小于给定阈值时所消耗的时间，红色虚线代表在 $H = H_{\text{traj}}^*$, $T = T_{\text{traj}}^*$ 时，全局模型验证损失函数值首次小于给定阈值时所消耗的时间。可以看到优化算法实验的全局模型验证损失函数值能更快地达到给定阈值，这意味着其分布式训练的收敛速度更快。由此可知，经过算法3.1的优化，轨迹预测任务的分布式训练收敛速度也得到了一定增益。

当然，根据上述实验结果只能验证算法3.1的合理性与有效性，还无法说明该算法的最优性。而要想验证算法3.1的最优性，需要遍历 H 和 T 搜索范围内的所有取值点进行仿真实验，这显然并不现实。因此，本小节的仿真实验仅对算法设计的合理性与有效性进行验证。

4.5 本章小结

在本章中，我们对车联网分布式训练仿真平台进行了模块化设计与代码级实现。基于该仿真平台，我们进行了多组仿真实验。根据各组实验的结果，我们依次验证了车联网分布式训练仿真平台设计与实现的正确性；观察到分布式训练回合时长 T 与本地训练迭代次数 H 对于分布式训练收敛速度的影响，并对其影响机理进行解释；验证了优化问题 $P4-1$ 建模的有效性与普适性；验证了优化算法3.1设计的合理性与有效性。

第 5 章 总结与展望

5.1 研究总结

本文面向未来车联网中具有广阔应用前景的“车路协同”分布式训练场景，对于如何提高分布式训练中神经网络模型的收敛速度的问题进行了相关研究。

我们首先对于一个基本的车联网分布式训练场景进行建模，该场景模型在简化实际交通环境与车流过程的同时，准确刻画了车联网分布式训练的基本框架。

基于该场景模型，我们通过分析可知，分布式训练回合时长 T 以及本地训练迭代次数 H 的选取对于模型收敛速度有一定影响。为了设计一个针对 T 和 H 的优化算法，我们先对单回合成功车数的概率分布进行理论分析，发现当通信时延为常量，计算时延服从偏移负指数分布时，单回合成功车数服从泊松分布，该分布的参数由分布式训练的各项参数共同决定。基于该结论，我们在车辆本地数据集数据特征足够丰富、神经网络模型足够鲁棒、本地训练批大小足够大的理想情景下，对车联网分布式训练中模型收敛速度的优化问题进行建模。针对该优化问题，我们设计了一个 H - T 联合优化算法，该算法通过循环迭代的方式，交替更新 H 和 T ，最终得到优化问题的近似最优解。

为了验证优化算法设计的合理性与有效性，需要进行仿真实验验证。基于本文的车联网分布式训练场景模型，我们以模块化的方式设计了一个车联网分布式训练仿真平台，并通过 Python 编程完成了对该仿真平台的代码级实现。利用该仿真平台，我们针对毫米波通信中的波束选择任务以及自动驾驶中的轨迹预测任务，进行了若干组仿真实验，发现 T 和 H 的选取确实对模型收敛速度有明显影响，并且我们所设计的优化算法能够最大限度地提高分布式训练的模型收敛速度。

上述工作对于未来车联网中如何高效地进行“车路协同”分布式训练提供了一定指导。其创新之处体现在对车联网分布式训练场景的建模、对模型收敛速度的理论分析，以及对优化算法的设计。然而，本文的工作仍有许多局限性，如未考虑通信时延受信道条件、通信带宽、信干噪比等因素的影响而直接将其视为固定值，未考虑车辆本地数据集间非独立同分布的影响，未考虑车辆到达率随时间动态变化的情况等。这些局限之处都是未来可以继续研究的问题。

5.2 未来展望

在本文工作的基础上，未来可继续深入研究的方向包括：

1. 更加贴合真实情景的车辆本地数据生成方式。在本文设计的仿真平台中，每辆车的本地数据集都是从一个总数据集中随机抽取部分数据样本构成的。由于我们所选用的现有数据集的样本数量有限，当仿真时长足够长时必然存在不同车辆间拥有相同数据样本的情况，而这和真实情景中各车辆间不存在相同数据样本的情况不符。为了更加贴合真实情景，需要采用其他的车辆本地数据生成方式，让每辆车的本地数据独一无二。
2. 通信资源约束下的车辆调度与带宽分配策略。在本文的优化问题建模中，为了简化理论分析，将模型下发与模型上传的通信延时建模为常量，而这与真实情景不符。在车联网环境中，车辆与路侧单元间的通信速率受信道条件、通信带宽、信干噪比等因素的影响。当多辆车同时需要与路侧单元进行通信时，由于通信资源有限，车辆间会出现竞争通信资源的情况。因此，车联网分布式训练还需要考虑车辆调度与带宽分配策略，即在每个回合内，路侧单元调度哪些车辆参与分布式训练，车路通信时每条通信链路分配多少带宽。
3. 针对数据非独立同分布的模型聚合策略。在本文的优化问题建模中，假设车辆的本地数据集间独立同分布，因此在模型聚合环节路侧单元对收到的所有模型参数做平均聚合。而在真实情景下，不同车辆间本地数据集的大小，数据样本特征分布，数据价值均有所差异。在模型聚合环节，如果采用加权聚合策略，对本地数据集更优的车辆上传的模型参数赋以更高的加权系数，就能提高全局模型向着更优方向更新的概率，从而提高分布式训练的模型收敛速度。
4. 针对车辆到达率时变的动态优化算法设计。在本文的优化问题建模中，假设车辆泊松到达且到达率不变。但在真实情景中，一条道路的车辆到达率在不同时段往往有明显变化，而车辆到达率对于本文设计的优化算法具有影响。因此，针对车辆到达率时变的真实情景，还需要设计动态优化算法，实时地调整车联网分布式训练的相关参数，从而提高分布式训练的模型收敛速度。

参考文献

- [1] 中国信息通信研究院. 车联网白皮书（网联自动驾驶分册）[EB/OL]. 2020. <http://www.caict.ac.cn/kxyj/qwfb/bps/202012/P020201215382968589778.pdf>.
- [2] Kenney J B. Dedicated short-range communications (dsrc) standards in the united states[J/OL]. Proceedings of the IEEE, 2011, 99(7): 1162-1182. DOI: 10.1109/JPROC.2011.2132790.
- [3] Naik G, Choudhury B, Park J M. Ieee 802.11bd amp; 5g nr v2x: Evolution of radio access technologies for v2x communications[J/OL]. IEEE Access, 2019, 7: 70169-70184. DOI: 10.1109/ACCESS.2019.2919489.
- [4] 中国信息通信研究院. 车联网白皮书[EB/OL]. 2021. <http://www.caict.ac.cn/kxyj/qwfb/bps/202112/P020211224634954242855.pdf>.
- [5] Lu N, Cheng N, Zhang N, et al. Connected vehicles: Solutions and challenges[J/OL]. IEEE Internet of Things Journal, 2014, 1(4): 289-299. DOI: 10.1109/JIOT.2014.2327587.
- [6] Grigorescu S, Trasnea B, Cocias T, et al. A survey of deep learning techniques for autonomous driving[J/OL]. Journal of Field Robotics, 2020, 37(3): 362-386. <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21918>. DOI: <https://doi.org/10.1002/rob.21918>.
- [7] Wang S, Tuor T, Salonidis T, et al. Adaptive federated learning in resource constrained edge computing systems[J/OL]. IEEE Journal on Selected Areas in Communications, 2019, 37(6): 1205-1221. DOI: 10.1109/JSAC.2019.2904348.
- [8] Tran N H, Bao W, Zomaya A, et al. Federated learning over wireless networks: Optimization model design and analysis[C/OL]//IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. 2019: 1387-1395. DOI: 10.1109/INFOCOM.2019.8737464.
- [9] Liu L, Zhang J, Song S, et al. Client-edge-cloud hierarchical federated learning[C/OL]//ICC 2020 - 2020 IEEE International Conference on Communications (ICC). 2020: 1-6. DOI: 10.1109/ICC40277.2020.9148862.
- [10] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge[C/OL]//ICC 2019 - 2019 IEEE International Conference on Communications (ICC). 2019: 1-7. DOI: 10.1109/ICC.2019.8761315.
- [11] Zeng Q, Du Y, Huang K, et al. Energy-efficient radio resource allocation for federated edge learning[C/OL]//2020 IEEE International Conference on Communications Workshops (ICC Workshops). 2020: 1-6. DOI: 10.1109/ICCWorkshops49005.2020.9145118.

- [12] Shi W, Zhou S, Niu Z, et al. Joint device scheduling and resource allocation for latency constrained wireless federated learning[J/OL]. IEEE Transactions on Wireless Communications, 2021, 20(1): 453-467. DOI: 10.1109/TWC.2020.3025446.
- [13] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database[C]// 2009 IEEE conference on computer vision and pattern recognition. IEEE, 2009: 248-255.
- [14] Krizhevsky A. Learning multiple layers of features from tiny images[C]//2009.
- [15] Deng L. The mnist database of handwritten digit images for machine learning research[J]. IEEE Signal Processing Magazine, 2012, 29(6): 141-142.
- [16] Chang M F, Lambert J, Sangkloy P, et al. Argoverse: 3d tracking and forecasting with rich maps[C/OL]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019: 8740-8749. DOI: 10.1109/CVPR.2019.00895.
- [17] Caesar H, Bankiti V, Lang A H, et al. nuscenes: A multimodal dataset for autonomous driving [J]. arXiv preprint arXiv:1903.11027, 2019.
- [18] Ma Y, Zhu X, Zhang S, et al. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 33. 2019: 6120-6127.
- [19] Klautau A, Batista P, González-Prelcic N, et al. 5g mimo data for machine learning: Application to beam-selection using deep learning[C/OL]//2018 Information Theory and Applications Workshop (ITA). 2018: 1-9. DOI: 10.1109/ITA.2018.8503086.
- [20] Lopez P A, Behrisch M, Bieker-Walz L, et al. Microscopic traffic simulation using sumo [C/OL]//The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE, 2018. <https://elib.dlr.de/124092/>.
- [21] Mashhadi M B, Jankowski M, Tung T Y, et al. Federated mmwave beam selection utilizing lidar data[J/OL]. IEEE Wireless Communications Letters, 2021, 10(10): 2269-2273. DOI: 10.1109/LWC.2021.3099136.
- [22] Liang M, Yang B, Hu R, et al. Learning lane graph representations for motion forecasting[Z]. 2020.