

面向多目标状态感知的自适应云边协同调度研究

周文晖, 彭清桦, 谢磊

引用本文

周文晖, 彭清桦, 谢磊. 面向多目标状态感知的自适应云边协同调度研究[J]. 计算机科学, 2024, 51(9): 319-330.

ZHOU Wenhui, PENG Qinghua, XIE Lei. [Study on Adaptive Cloud-Edge Collaborative Scheduling Methods for Multi-object State Perception](#) [J]. Computer Science, 2024, 51(9): 319-330.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

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

[基于不确定性权重的保守Q学习离线强化学习算法](#)

Offline Reinforcement Learning Algorithm for Conservative Q-learning Based on Uncertainty Weight
计算机科学, 2024, 51(9): 265-272. <https://doi.org/10.11896/jsjcx.230700151>

[无人机辅助边缘计算安全通信能力最大化方案](#)

Scheme for Maximizing Secure Communication Capacity in UAV-assisted Edge Computing Networks
计算机科学, 2024, 51(6A): 230800032-7. <https://doi.org/10.11896/jsjcx.230800032>

[基于联邦学习的智能电网AMI入侵检测方法研究](#)

Study on Smart Grid AMI Intrusion Detection Method Based on Federated Learning
计算机科学, 2024, 51(6A): 230700077-8. <https://doi.org/10.11896/jsjcx.230700077>

[边缘计算下差分隐私的应用研究综述](#)

Survey of Application of Differential Privacy in Edge Computing
计算机科学, 2024, 51(6A): 230700089-9. <https://doi.org/10.11896/jsjcx.230700089>

[基于深度强化学习的二进制代码模糊测试方法](#)

Fuzz Testing Method of Binary Code Based on Deep Reinforcement Learning
计算机科学, 2024, 51(6A): 230800078-7. <https://doi.org/10.11896/jsjcx.230800078>

面向多目标状态感知的自适应云边协同调度研究

周文晖 彭清桦 谢磊

南京大学软件新技术全国重点实验室 南京 210000

(whzhou@smail.nju.edu.cn)

摘要 随着智能城市和工业智能制造的蓬勃发展,从监控摄像头获取详尽信息以进行多目标视觉分析的需求日益突出。现有研究主要关注在服务器上进行资源调度以及改良视觉模型,往往不能很好地应对设备资源状态和任务状态的动态变化。随着边缘端硬件资源的升级和任务处理模型的改进,设计一个自适应的云边协同调度模型来满足任务的实时用户需求成为优化多目标状态感知任务的重要方式。因此,在深入分析云边场景下多目标状态感知任务特性的基础上,提出了一种基于深度强化学习的自适应云边协同调度模型 ATS-SAC。ATS-SAC 通过实时解析多目标状态感知任务的运行时状态,动态给出任务执行的视频流配置、模型部署配置等调度决策,从而显著优化环境不稳定的云边场景下多目标状态感知任务的精度、时延的综合性质量。同时,还引入了一种基于用户体验极限阈值的动作筛选方法,有助于去除冗余的决策动作,进一步优化模型的决策空间。针对用户对多目标状态感知任务性能结果的不同需求,ATS-SAC 模型能提供包括极速模式、均衡模式和精度模式在内的多种灵活的调度策略。实验结果表明,相比其他的任务执行方式,在 ATS-SAC 模型的调度策略下,多目标状态感知任务在精度质量和处理时延上更能满足用户的体验需求。同时,当实时运行状态变化时,ATS-SAC 模型能够动态调整其调度策略,以保持稳定的任务处理效果。

关键词: 边缘计算;云边协同;调度策略;多目标状态感知;深度强化学习

中图分类号 TP393

Study on Adaptive Cloud-Edge Collaborative Scheduling Methods for Multi-object State Perception

ZHOU Wenhui, PENG Qinghua and XIE Lei

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210000, China

Abstract With the development of smart cities and intelligent industrial manufacturing, the demand for comprehensive information from surveillance cameras for multi-objective visual analysis has become increasingly prominent. Existing research mainly focuses on resource scheduling on servers and improvements of visual model, which often struggle to adequately handle dynamic changes in system resource and task state. With the advancement of edge hardware resources and task processing models, designing an adaptive cloud-edge collaborative scheduling model to meet the real-time user requirements of tasks has become an essential approach to optimize multi-objective state perception tasks. Thus, based on a profound analysis of characteristics of multi-objective state perception tasks in cloud-edge scenarios, this paper proposes a model of adaptive task scheduler based on soft actor-critic(ATS-SAC). ATS-SAC intelligently decides key factors of tasks such as video stream configuration and model deployment configuration according to real-time analysis of runtime state, thereby significantly optimizing the accuracy and delay of multi-objective state perception tasks in cloud-edge scenarios. Furthermore, we introduce an action filtering method based on user experience threshold that helps to eliminate redundant decision-making actions, so as to reduce the decision-making space of the model. Depending on user's varied demands for performance outcomes of the multi-objective state perception tasks, ATS-SAC model can provide three flexible scheduling strategies, namely speed mode, balance mode, and precision mode. Experimental results show that, comparing to other executing methods, the scheduling strategies of ATS-SAC model make multi-objective state perception tasks more satisfactory in terms of accuracy and delay. Moreover, when the real-time operating state changes, the ATS-SAC model can dynamically adjust its scheduling strategies to maintain stable task processing results.

Keywords Edge computing, Cloud-Edge collaboration, Scheduling policy, Multi-object state perception, Deep reinforcement learning

到稿日期:2024-02-05 返修日期:2024-06-07

基金项目:国家重点研发计划(2022YFB3303900);国家自然科学基金(62272216)

This work was supported by the National Key R & D Program of China(2022YFB3303900) and National Natural Science Foundation of China(62272216).

通信作者:谢磊(lxie@nju.edu.cn)

1 引言

在当前时代,物联网设备的普及、智慧城市和智能制造的兴起使得监控摄像头在众多智能化场景中得到广泛应用^[1-2]。与过去仅仅获取并保存监控视频流的方式不同,现代的应用场景需要实时且有效地智能分析视频内容以应对社会数字化和智能化需求,如在交通摄像头的视频流中实时分析当前路口车流量、判断是否发生事故等。边缘端摄像头的视频画面往往同时存在多个待分析目标,其目标个数、大小等状态也实时动态变化,这些场景下的多目标状态感知任务需要较高的实时响应性^[3-4]。

许多研究集中于以云服务器为中心的处理方式^[5-6],即将所有分布式边缘节点获取的视频数据上传至云端进行集中处理。然而,随着监控摄像头的视频质量提升以及边缘端节点数量的指数级增长,云端中心常常难以负荷如此大的工作量;此外,这种处理方式很大程度上受网络质量限制,导致网络传输时延存在不稳定性^[7];原始视频数据信息通过网络传输还会产生安全性问题^[8]。这些都是云中心计算需要处理的难题。在一些极端情况下(如网络信号中断),视频处理分析任务可能会被迫中断,这对许多实时场景来说是无法接受的。

因此,完全依赖云服务器的处理方式对于部分场景是不适合的。加入边缘端节点的视频流处理架构如图1所示。边缘端靠近数据源,可以快速获取来自数据源摄像头的监控视频流;同时,边缘端通过网络连接云处理中心,向云中心传输待处理数据信息。随着边缘端硬件算力的提升,将任务部分或全部加载到靠近数据源的边缘端节点上完成处理成为一种可行的解决方案^[9]。这种云边协同处理的方案可以充分利用边缘端算力资源,从而减少源数据在网络传输中带来的不稳定时延开销以及隐私数据泄露问题。



图1 云端和边缘端之间动态任务负载

Fig. 1 Dynamic task offloading between cloud and edge

当前的云边协同调度算法的研究很多集中于基于测量配置的优化方法^[9-10],即预先测量不同调度参数下的任务表现后,在任务执行时通过组合优化的方式选择最佳组合配置进行调度。然而,在实际的多目标状态感知任务执行过程中,硬件的系统状态和任务的复杂度都会随时间而变化,运行时状态环境的组合情况往往具有爆炸式的组合空间;同时,调度动作执行本身会对任务运行环境状态产生进一步的影响,从而导致运行时状态、调度决策、任务执行表现之间具有复杂的数学关系。对于基于配置优化方案来说,很难考虑和覆盖到这种复杂非线性关系。因此,针对不稳定环境下的多目标状态感知任务,其动态自适应能力具有一定的局限性。

鉴于此,本文提出一种基于深度强化学习算法的自适应云边协同调度模型 ATS-SAC(Adaptive Task Scheduler based on Soft Actor-Critic)。在边缘端设备资源有限、边缘端与

云端交互受网络波动影响的场景下,ATS-SAC模型采用强化学习的架构,根据系统的资源状况以及当前的任务状态,动态地完成视频流配置和云边之间的任务负载。在充分利用云边资源的同时,这一模型可以更好地满足应用场景的多目标状态感知的精度、时延需求。本文的主要贡献可以归纳为以下3个方面:

1)分析了云边场景下多目标状态感知任务的特性,并探索运行时状态和配置参数对任务精度和时延等性能结果的影响。在此基础上,建立了基于多目标状态感知任务精度和时延多目标优化的云边协同调度任务数学模型。

2)提出了基于深度强化学习的自适应云边协同调度模型 ATS-SAC。ATS-SAC模型分为调度决策部分和调度执行部分。调度决策部分根据实时的多目标状态感知任务运行时状态,给出调度的配置决策;调度执行部分将调度决策在云边协同的场景下具体实施。同时,通过对精度和时延做不同权重的加权和,从而将多目标优化问题转变为单目标优化问题进行求解,并根据不同的精度与时延权重制定对应的 ATS-SAC 调度策略模式。

3)提出了基于用户体验极限阈值的动作筛选方法,通过设定时延和精度的用户体验极限阈值,剔除超过用户体验极限的调度决策动作,从而大大减小决策空间的维度,使算法更容易收敛。

本文第2章介绍了云边任务优化的相关工作;第3章分析了多目标状态感知任务的特性并对本文研究的云边协同调度任务进行系统化数学建模,从而阐述本文提出的云边协同调度模型;第4章通过对比实验论证本文提出的 ATS-SAC 模型在满足用户体验需求、自适应调整等方面的能力;最后总结全文并展望未来。

2 相关工作

监控摄像头被广泛部署在智慧工业、智能城市等场景中^[11-12],基于这些边缘摄像头获取的视频画面进行多目标状态感知任务是当前计算机视觉研究的热点之一,其中包括多目标视频检测(识别)^[13-15]、多目标视频分割^[16-17]、多目标跟踪^[18-19]、多目标分类^[20]等细分技术领域。

然而,很多针对这些多目标状态感知任务提出的处理模型并不能很好地应用在云边的实际场景中,因为这些模型通常受到系统状态波动、硬件资源有限等限制,难以在精度和速度之间取得良好的平衡^[21]。因此,在云边场景下,需要通过调度优化技术来改进这些模型的任务执行效果。目前,国内外的研究主要有3个方向:1)基于系统资源配置的优化;2)基于模型配置的优化;3)基于视频流配置的任务优化。

2.1 基于系统资源配置的任务优化

对于那些以云服务器为处理中心的调度框架,如何在有大量边缘节点的情况下通过调度算法在单个云服务器上实现云上资源的合理分配非常关键。系统资源往往包括 CPU 的执行线程分配、内存分配、GPU 算力分配等。

Zhang 等^[11]提出了一种视频分析系统 VideoStorm,该系统将边缘摄像头传输的待处理监控视频视为查询服务,并

使用贪心算法将服务器现有资源分配到每个查询服务。Han等^[10]使用优化编译器和运行时调度器组成的MCDNN模型来权衡CPU的线程分配,并尽可能满足每个边缘端服务的需求。Shen等^[22]针对云服务器上GPU集群的高效服务问题提出了Nexus系统,从而持续高效地将GPU分配给深度神经网络(DNNs)。Jiang等^[23]构建了基于视频分析管道的Chameleon调度系统,与前几种资源调度模型不同,Chameleon会根据系统的持续动态状态搜索最佳配置,并将系统切换配置的开销平摊到多个时间片上。

2.2 基于模型配置的任务优化

对多目标状态感知任务的处理模型的优化也能够提升任务性能,在国内外的研究中主要有两个方向:针对模型本身的改进和针对模型卸载的改进。

对于模型本身,许多研究尝试通过改进结构、模型压缩蒸馏等手段来减小模型的规模、加快模型的推理速度,从而使其可以在资源有限的边缘节点上执行任务。例如,Liu等^[24]提出的单深度神经网络检测方法SSD通过预先分割不同尺度下的预测框,在保证精度质量的同时显著提升了推理速度;Zhao等^[25]提出的ICNet通过多分辨率分支和级联特征融合的方式在语义分割任务中显著提升了速度;He等^[26]通过在特征图采样阶段减少瓶颈架构,压缩了MobileNet,得到了更轻量级的模型;Jiang等^[27]提出的MainStream分析系统通过在云端的所有任务模型之间共享参数来提升资源利用率。

对于模型卸载策略,研究主要集中在模型在云端或边缘端的动态切割、动态卸载上,以便根据系统状态和任务状态及时调整云、边的任务负载。例如,Matsubara等^[28]将任务处理的DNN模型分割为头模型和尾模型两部分,分别部署在边缘端和云端,从而避免直接传输视频帧,减少传输带宽;Ra等^[29]采用贪心策略,根据网络状态来决定模型卸载在云端或边缘端;Ran等^[9]提出了模型卸载调度框架DeepDecision,该框架

可以通过对模型准确率、视频质量、网络状态等参数的评估求解组合优化问题得到模型的卸载策略。

2.3 基于视频流配置的任务优化

多目标状态感知任务的处理对象是视频流或视频帧,因此,针对视频流的配置优化也能提升整体任务性能。视频流的配置优化通常会关注视频的分辨率、帧率、码率等。

对于视频分辨率,虽然较高的分辨率有助于模型更好地得到推理结果,但也会导致视频帧变得很大,从而产生较大的传输延迟。Wang等^[30]提出了边缘端到云端的传输框架CloudSeg,它采用较低分辨率向云端传输视频流,并在云端通过对人物质量进行分析,实现视频流的超分辨率处理,这样既保证了较小的传输时延,也能得到较好的云端推理结果。

对于视频帧率,常见的优化方法是采用帧过滤的方式过滤掉部分重复帧,从而减少不必要的任务负载。例如,Li等^[31]基于视频帧内容的特征分析来过滤掉特征重复度较高的连续帧;Chen等^[32]通过维护一个活动缓存来存储视频帧,并根据缓存的剩余容量来决定是否丢弃视频帧。

对于视频码率,较小的码率可以在相同条件下带来更小的传输时延。Huang等^[33]通过学习模仿已有的专家经验来给出码率的生成建议;Zhang等^[7]提出了基于深度强化学习的CASVA模型,CASVA通过对实时视频状态的分析,学习自适应地调整边缘端向云端传输视频的码率。

3 基于深度强化学习的自适应云边协同调度算法

3.1 多目标视频流处理任务分析

多目标视频处理任务涉及对视频流中目标进行满足各种场景需求的视觉处理任务,如目标检测、目标分类、目标跟踪等。在云边场景下的多目标视频流处理任务主要分为两种情形:单任务场景和多任务场景。图2描述了本文研究中两种任务的具体执行过程。

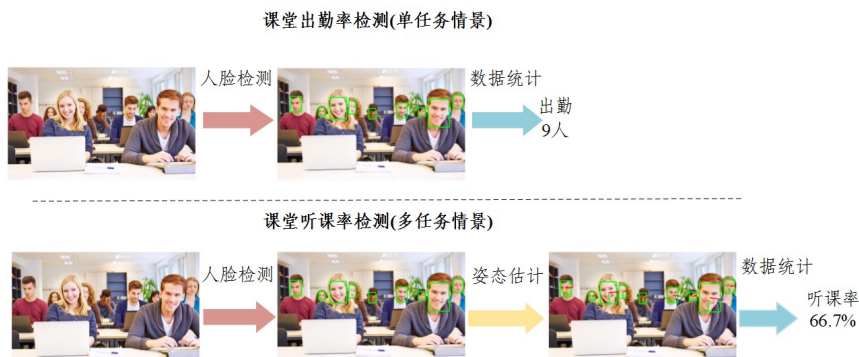


图2 课堂出勤率检测和课堂听课率检测过程

Fig. 2 Detection process of classroom attendance rate and classroom listening rate

如图2所示,单任务场景仅包含一个多目标视频处理任务,例如在课堂出勤率检测中,执行人脸检测任务就可以得出出勤人数。在这种情况下,任务的调度负载只需要考虑在云端或边缘端执行。而多任务场景则包含多个任务,例如在课堂听课率检测任务中,需要先进行人脸检测,然后对每个人脸框进行姿态估计,才能得出抬头人数。在这种情况下,任务的调度负载需要考虑将多个串行的任务

分配在云端和边缘端执行的问题。

针对多目标视频处理任务,我们选用精度和时延作为评价指标,精度和时延的综合表现通常决定了最终的用户体验。

为了使调度策略能够优化多目标状态感知任务的执行结果,本文通过一系列实验探究影响多目标状态感知任务的执行结果。实验结果如表1所列。通过研究发现,影响多目标状态感知任务的执行结果的因素主要分为两类:一类是主观可改变的因素,如

输入视频流的配置和模型的配置;另一类是客观不可改变的因素,包括硬件系统状态和任务状态。

表 1 多目标状态感知任务影响因素实验结论

Table 1 Experiments results of factors in multi-object state perception tasks

因素分类	因素属性	具体影响因素名称	实验结论
运行时状态	系统状态	CPU 占用	其他进程 CPU 占用增加,任务处理进程 CPU 占用会减少 任务处理进程 CPU 占用越小,任务处理时延越大
运行时状态	系统状态	内存占用	其他任务内存占用增加会导致任务处理进程可占用内存减小 任务处理进程内存占用小于需要内存大小会导致任务执行失败
运行时状态	系统状态	网络带宽	网络带宽越大,单位信息的云边传输时延越小
运行时状态	任务状态	目标数量	目标数量越多,任务处理时延越大
运行时状态	任务状态	目标大小	目标平均大小越大,任务处理时延越大
配置参数	视频流配置	分辨率	分辨率越高的视频帧,任务处理时延越大,任务处理精度越高
配置参数	视频流配置	帧率	视频处理帧率较高时,更可能导致任务未处理帧数增加
配置参数	视频流配置	编码方式	不同的编码方式会有不同的解析速度和视频流大小
配置参数	模型配置	模型选择	不同模型、同一模型不同预训练参数会影响精度与时延结果
配置参数	模型配置	模型部署位置	不同方式的云边部署会使得云边系统负载、传输内容不同 不同的系统处理会导致处理时延和精度不同 不同的云边传输内容会导致传输时延不同

3.2 云边协同调度任务数学建模

在多目标状态感知任务中,配置因素、运行时状态都会对任务的执行结果产生影响。而对于调度算法而言,配置因素是可主动改变的主观量,运行时状态是不可改变、不可预知的客观量。因此,想要通过调度使多目标状态感知任务达到精度和时延的相对最优解,需要根据实际的运行时状态给出对多目标状态感知任务的配置。

对于本文研究的面向多目标状态感知的自适应云边协同调度任务,我们给出如下数学建模过程:

多目标状态感知任务执行的运行时状态包括系统状态、任务状态。式(1)展示了运行时状态与系统状态、任务状态的函数映射关系,其中 S_{system} 表示系统状态, S_{task} 表示任务状态, S 表示运行时状态, F_s 表示系统状态和任务状态到运行时状态的映射关系。

$$S = F_s(S_{system}, S_{task}) \quad (1)$$

多目标状态感知任务中,视频流配置包括视频的分辨率、帧率、编码方式。式(2)展示了视频流配置情况与视频的分辨率、帧率、编码方式的映射关系,其中 C_{video} 表示视频流配置, r 表示视频的分辨率, f 表示视频的处理帧率, e 表示视频流的编码方式, F_v 表示视频的分辨率、帧率、编码方式到视频流配置的映射关系。

$$C_{video} = F_v(r, f, e) \quad (2)$$

多目标状态感知任务中,模型配置包括模型的选择、模型的云边部署方式。式(3)展示了模型配置情况与模型选择、云边部署方式的映射关系,其中 C_{model}^i 表示第 i 个任务的模型配置($i=0,1,2,\dots,n$), m^i 表示第 i 阶段任务的模型选择, p^i 表示第 i 阶段任务的云边部署情况, F_m 表示任务模型选择、任务云边部署情况到任务模型配置情况的映射关系。

$$C_{model}^i = F_m(m^i, p^i) \quad (3)$$

式(4)展示了多目标状态感知处理任务的精度求解过程,式(5)展示了多目标状态感知处理任务的时延求解过程。

$$accuracy = \sum_i A(F_v(r, v, f), F_m(m^i, p^i), F_s(S_{system}, S_{task})) \quad (4)$$

$$latency = \sum_i L(F_v(r, v, f), F_m(m^i, p^i), F_s(S_{system}, S_{task})) \quad (5)$$

那么,面向多目标状态感知任务的云边协同调度算法的数学模型可以表达为式(6)所示的内容。简单来说,即对于某一时间点,在给定运行时状态下(包括系统状态和任务状态),求解能使得多目标状态感知任务精度最高、时延最小的配置(包括视频流配置、模型配置)。由此可见,云边协同调度问题本质上是最大化精度和最小化时延的多目标优化求解问题。

$$\begin{aligned} & \min_{r, v, f, m, p} \sum_i L(F_v(r, v, f), F_m(m^i, p^i), F_s(S_{system}, S_{task})) \text{ and} \\ & \max_{r, v, f, m, p} \sum_i A(F_v(r, v, f), F_m(m^i, p^i), F_s(S_{system}, S_{task})), \\ & \text{given } S_{system}, S_{task} \end{aligned} \quad (6)$$

3.3 自适应云边协同调度模型 ATS-SAC

本文根据实际云边场景下的多目标状态感知任务优化需求提出了基于 Soft Actor-Critic 的自适应任务调度模型(ATS-SAC),模型的框架结构如图 3 所示。

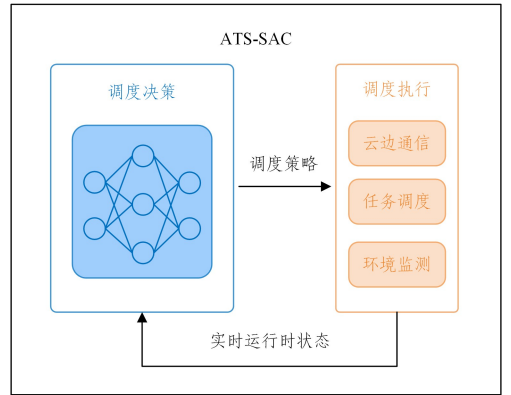


图 3 ATS-SAC 模型框架

Fig. 3 Framework of ATS-SAC

从图中可以看出,ATS-SAC 模型主要由两部分组成:调度决策部分和调度执行部分。调度决策部分根据实时的运行时状态来给出调度决策,调度执行部分根据调度决策具体地在云边场景下实现多目标状态感知任务的调度,并将监测到的实时运行时状态提供给调度决策部分。

3.4 ATS-SAC 模型调度决策部分

根据任务的数学建模分析,我们使用深度强化学习算法构建云边协同调度模型的决策部分。决策部分的

结构如图 4 所示。

从模型框架图中可以看出,ATS-SAC 调度模型决策部分主要分为状态输入部分、深度强化学习网络结构部分和动作决策输出部分。状态输入部分通过解析运行时状态为模型提供状态空间输入;深度强化学习网络结构部分通过学习迭代给出调度动作;动作决策输出部分将调度动作解析成一系列的系统配置决策。

ATS-SAC 模型的决策部分主要基于深度强化学习的演员-评论家(Actor-Critic)架构^[34],在整个模型中,由策略网络(Actor)做出动作决策,由价值网络(Critic)根据动作决策的

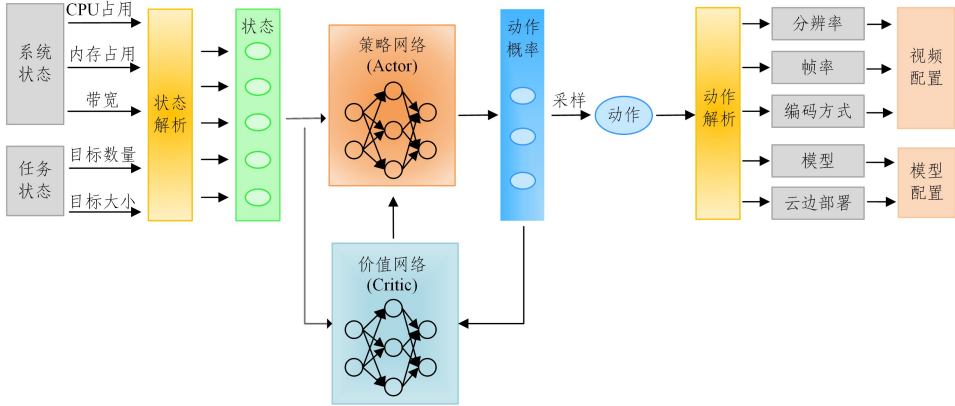


图 4 ATS-SAC 模型调度决策部分结构

Fig. 4 Structure of scheduling decision in ATS-SAC

调度器的实际结构基于 Soft Actor-Critic^[35]模型,主要由一个策略网络、两个价值网络、两个目标价值网络构成。

式(7)展示了模型中目标值的计算,其中, y 是目标值, r 是奖励, γ 是折扣因子, d 是结束标志, $Q_{\theta_i}(s', \pi_{\phi}(s'))$ 是使用策略 $\pi_{\phi}(s')$ 采样的两个目标 Q 函数的最小值。

$$y = r + \gamma(1-d) \min_{i \in \{1,2\}} Q_{\theta_i}(s', \pi_{\phi}(s')) \quad (7)$$

式(8)展示了模型中 Q 函数的更新方式,其中, θ 是 Q 函数的参数, α 是学习率, B 是采样的批次, (s, a, r, s', d) 是经验数据。

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} (Q_{\theta}(s,a) - y)^2 \quad (8)$$

式(9)展示了模型中策略网络的更新,其中, ϕ 是策略网络的参数, $\mathbb{E}_{a \sim \pi_{\phi}(s)} [Q_{\theta}(s,a)]$ 是 Q 函数关于策略的期望, $\mathbb{H}(\pi_{\phi}(s))$ 是策略的熵。在策略网络参数的更新过程中,ATS-SAC 同时兼顾了期望回报的最大化和熵的最大化。期望回报的最大化即价值函数 Q 函数的期望最高,表示希望模型能够产生平均意义上最优价值的策略;策略熵的最大化即模型给出的策略在各个可能的动作上概率更接近,表示希望模型搜索策略的不确定性和随机性更高,能够更全面地探索策略,这样更可能找到更好的解。

$$\phi \leftarrow \phi + \alpha \nabla_{\phi} \frac{1}{|B|} \sum_{s \in B} (\mathbb{E}_{a \sim \pi_{\phi}(s)} [Q_{\theta}(s,a)] - \alpha \mathbb{H}(\pi_{\phi}(s))) \quad (9)$$

式(10)展示了模型中目标 Q 函数的软更新,其中, τ 是软更新的系数。在训练的过程中,由于一个很小的软更新系数的存在,目标价值网络的更新将比价值网络的更新慢,从而更使得模型稳定更新,避免其过拟合。

结果给出价值评估。

在 ATS-SAC 进行云边协同调度的过程中,调度模型先采集当前多目标状态感知任务的运行时状态并解析为状态空间,运行时状态包括系统状态(CPU 占用、内存占用、带宽)和任务状态(目标数量、目标平均大小),可以反映当前任务的执行上下文情境。策略网络根据状态空间输入给出动作空间的概率分布决策,调度模型基于所有动作的概率分布进行采样得到具体的调度决策动作选择,并通过对动作选择的解析得到具体的配置决策信息,从而对整个云边场景下的多目标状态感知任务进行调度。

$$J(\pi) = \mathbb{E}_{s \sim D, a \sim \pi} [Q(s,a) - \tau \log \pi(a|s)] \quad (10)$$

在 ATS-SAC 模型决策部分,策略的输出是离散的,将其所有可能的动作输出编码为一维向量,在每次决策中针对一维向量的所有动作输出一个选择的概率,最终模型输出的策略结果即为选择概率最大的动作。

3.5 ATS-SAC 模型调度执行部分

ATS-SAC 调度模型的执行部分负责执行调度策略,在云边协同的场景下优化多目标感知任务,调度执行部分的主要结构如图 5 所示。

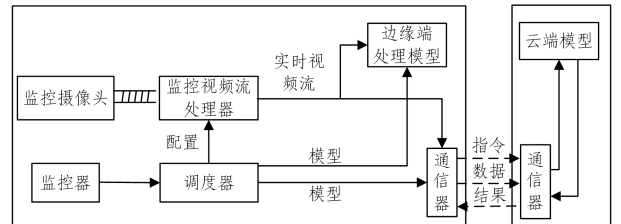


图 5 ATS-SAC 模型调度执行部分结构

Fig. 5 Structure of scheduling execution in ATS-SAC

调度执行部分主要包括 3 方面的工作,即环境监测、云边通信、任务调度。

1) 环境监测

ATS-SAC 模型通过对边缘端结点硬件系统状况的实时监测得到实时的系统状态,通过对视频帧的快速预处理得到任务状态,由此可以对多目标状态感知任务的实时运行时状态进行监测,并将其提供给模型的决策部分用于生成调度策略。

2) 云边通信

ATS-SAC 模型部署在云边协同的场景下,其中 ATS-

SAC的调度决策部分部署在边缘端以便进行实施决策,因此云端与边缘端之间需要进行多种信息的传递,如边缘端向云端传送待处理的数据、边缘端向云端发送模型部署调度指令信息、云端向边缘端发送处理结果。图6展示了本文定义的云边之间交流的数据帧格式,其中,模式字段指定了当前数据帧的信息内容模式,识别码字段指定了当前传输数据的视频帧。通过构建云边之间的通信机制,ATS-SAC模型的执行部分可以实现对云和边的分布式调度控制。

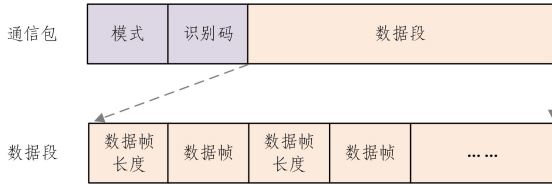


图6 云边通信协议

Fig. 6 Communication protocol between cloud and edge

3) 任务调度

ATS-SAC模型的调度执行部分接收调度决策部分生成的调度策略,并对系统进行一系列的配置。在图5中可以看到,调度器生成的调度策略包括视频流配置和模型配置,调度执行部分通过监控视频流处理器组件来控制边缘端本地的摄像头输入,并调整输入视频流的配置。对于调度策略中的模型配置,调度执行部分根据具体的部署策略将相应的模型选择指令和任务执行指令分别发送给边缘端、云端,并调整多目标状态感知任务的执行流程。

3.6 调度模型动作空间映射

对于ATS-SAC模型而言,动作的输出是离散的一维数字,然而,云边协同调度任务下需要的调度决策包括了多维配置,这就要求在一维动作与多维配置之间建立映射关系。本文通过排列组合的方式将所有配置构建为一张动作空间表,将动作空间表的每一组动作视为一种可行的调度策略并对其编码,最后可以得到一维的动作编码向量,其对应了整个调度决策的动作空间。表2列出了当需要调度控制的配置参数为A、B、C且每个配置参数分别有两个可选择的结果(A₁, A₂, B₁, B₂, C₁, C₂)时,通过配置参数排列组合得到的部分调度决策动作。

表2 配置的排列组合方式

Table 2 Permutations and combinations of configurations

调度策略序号	配置A	配置B	配置C
1	A ₁	B ₁	C ₁
2	A ₁	B ₁	C ₂
3	A ₁	B ₂	C ₁
4	A ₁	B ₂	C ₂
5	A ₂	B ₁	C ₁
—	—	—	—

在本研究中,配置数量的增加会使动作维度呈指数级增长。以课堂抬头率检测任务为例,当视频流配置中分辨率有3种配置、帧率有3种配置、编码方式有2种配置时,云边的模型部署有8种配置,这种情况下动作空间的维度达到了 $3 \times 3 \times 2 \times 8 = 144$,这往往会使得调度算法难以搜索到最佳的调度决策。然而,在这些配置排列组合得到的调度动作中,有

很大一部分的动作决策在现实决策中并不能很好地发挥作用,实际意义不大。

因此,本文提出了一种基于用户体验极限阈值的动作筛选方法来降低动作空间的维度。用户体验极限指严重影响用户对于任务处理体验的精度或时延方面表现,我们将那些会严重影响用户体验的动作决策剔除。图7展示了不同动作的精度指标(负相关)-时延指标测试情况,其性能表现越接近坐标原点越能给用户带来较好的体验。我们分别在时延和精度维度选定阈值,超出用户体验极限阈值部分的动作决策将会被剔除,被剔除的动作表示其调度下的任务执行往往会很大程度上降低用户体验,由此进一步降低动作空间的复杂度。

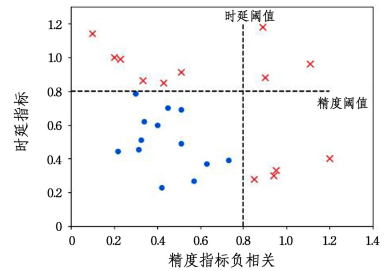


图7 不同动作的精度-时延测试图

Fig. 7 Accuracy-Latency relations of different actions

3.7 调度模型结果评估方式

对于多目标状态感知任务而言,这是一个精度最大化和时延最小化的多目标优化任务,这里通过将精度和时延进行加权得到综合的评价指标,来更新调度模型的网络参数,如式(11)所示,其中 $accuracy$ 和 $latency$ 为归一化后的精度和时延结果, α 和 β 为可调节的权重参数。

$$reward = \alpha \cdot accuracy + \beta \cdot latency \quad (11)$$

归一化精度的方式为将实际多目标感知任务运行得到的任务结果与数据集中各任务的ground truth进行对比,得到任务的完成度(无量纲数)。例如,对于课堂听课率检测场景,人脸检测任务的完成度为正确检测到的人脸数占实际总人脸数的比例,姿态估计任务的完成度为是否抬头估计正确的个数占全部检测出的人脸框数的比例;归一化时延的方式为将任务执行时延与云边传输时延的综合与数据集的标准任务执行时延做比值,得到无量纲的归一化时延结果。

在奖励机制中可以通过调节模式选择来为调度器选择不同的训练方向,权重参数与模式的关系如表3所列。

表3 不同模式下调度结果奖励计算的参数

Table 3 Reward parameters in different scheduling modes

调度模式	精度系数 α	时延系数 β
均衡模式	0.5	-0.5
极速模式	0.4	-0.6
精度模式	0.7	-0.3

3.8 调度模型训练过程

ATS-SAC调度模型的决策部分是基于深度强化学习算法的,其训练过程包括两个部分:经验的收集和经验的训练。我们在云边之间构建模拟的任务执行环境,调度模型通过与模拟环境的交互收集经验,再通过经验回放的方式更新网络参数。

本文构建了用于调度模型训练的模拟环境,如图 8 所示。在模拟环境中,我们仿真了云边协同场景下的多目标状态感知任务,通过环境监视器、时间检测器、任务执行器等

封装的功能组件,可以在云端与边缘端之间模拟调度决策下多目标状态感知任务的执行,并及时测量评估精度和时延结果。

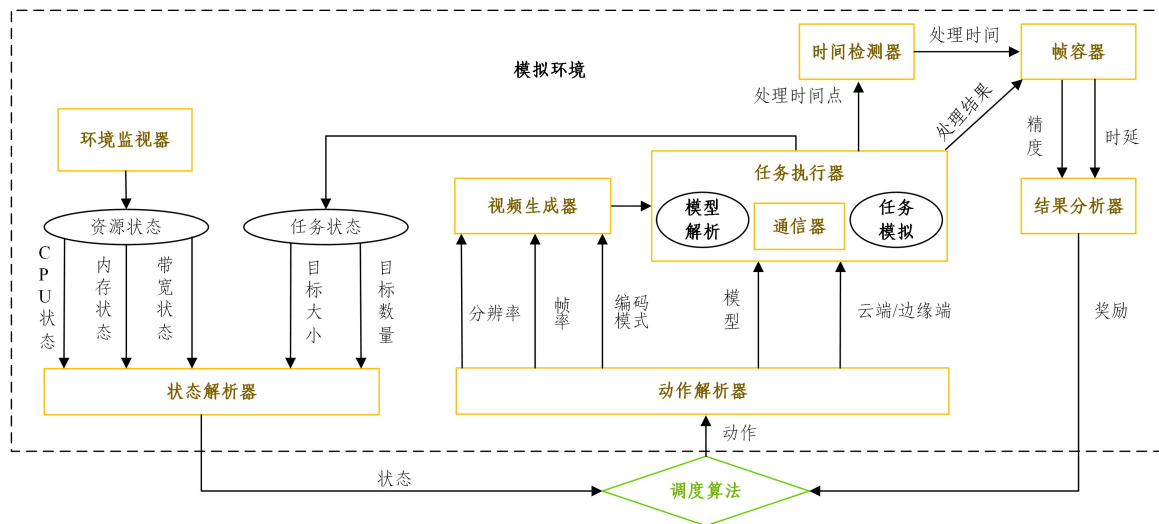


图 8 用于调度模型训练的模拟环境

Fig. 8 Simulation environment for training of scheduling model

调度模型与模拟环境的交互训练过程如下:

1) 多目标状态感知场景给出当前运行时状态 s_t , s_t 包括系统状态(CPU 占用率、内存占用率、带宽)和任务状态(目标数量、目标平均大小)。

2) ATS-SAC 算法根据状态 s_t 做出调度决策 a_t , a_t 包含模型配置决策(模型选择、模型卸载策略)和视频流配置决策(处理视频的分辨率、频率、编码方式)。

3) 系统根据调度算法给出的调度决策实施调度,调整模型配置和视频流配置,并模拟推演 3s 多目标视频处理任务,得到精度和时延结果;通过奖励机制计算出奖励值 r_t ,同时计算系统模拟任务推演后新的运行时状态 s_{t+1} ,将 r_t 和 s_{t+1} 返回给调度算法。

4) 调度算法将经验元组 $(s_t, a_t, r_t, s_{t+1}, d)$ 存储到经验回放缓冲区中,其中 d 表示本轮环境推演是否已结束。若结束则重置环境,并准备用经验回放训练调度算法;若未结束则回到 2) 继续下一次交互与模拟推演。

在经验回放缓冲区 Replay Buffer 中收集到部分经验时,可以通过经验回放的方式不断训练并更新模型中的网络参数。

经验的收集与经验的回放训练并不是完全的先后关系,它们可以交替进行,即收集一部分经验再回放训练,之后再重复上述过程。训练完成之后的策略网络可以用来在测试或推理的过程中生成调度策略。

4 实验与评估

4.1 实验配置

本文实验使用 ATS-SAC 模型作为调度算法,同时对其均衡模式、极速模式、精度模式进行测试实验。

调度算法可决策选择的模型配置如表 4 所列。

表 4 实验模型配置

Table 4 Model configurations in experiment

任务类型	模型部署位置	模型名称	预训练模型
人脸检测	边缘端	SSD	Tiny_Model
人脸检测	边缘端	CenterFace	CenterFace_Small
人脸检测	云端	SSD	RFB_Model
姿态估计	边缘端	PnP	—
姿态估计	云端	HopeNet	HopeNet_Lite
姿态估计	云端	HopeNet	HopeNet_Simple

调度算法可决策选择的视频流配置如表 5 所列。

表 5 实验视频流配置

Table 5 Video configurations in experiment

视频属性	可选配置
分辨率	1080 p, 720 p, 360 p
处理帧率	10 fps, 6 fps, 3 fps
编码方式	H.264, MJPEG

实验过程中,考虑的运行时状态包括系统状态和任务状态,具体采用的运行时参数如表 6 所列。

表 6 运行时状态情况

Table 6 Runtime state situation

状态名称	考查方面
系统状态	处理器占用率,内存占用率,网络带宽
任务状态	目标数量,目标平均大小

4.2 实验数据及处理

本文实验采用的数据集来源于课堂真实录像视频,数据集的基本情况如表 7 所列。

表 7 实验数据情况

Table 7 Experiments datasets

数据类别	总时长/s	视频个数	原始分辨率/p	原始帧率/fps	编码方式	数据来源
训练	611	13	1080	30	H.264	人工收集
测试	247	8	1080	30	H.264	人工收集

由于本研究所使用的数据集为人工采集,因此并没有

用于训练的 ground truth,这就需要人工对这些数据集进行标注。本文借鉴 CASVA 模型^[7]的数据集处理方法,采用“最佳配置标注法”对数据集标注 ground truth,具体流程如下:

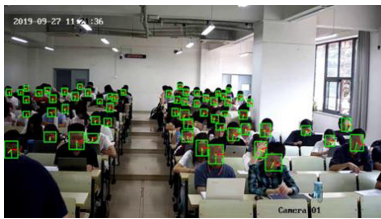
1) 最佳视频配置:将视频数据提取成一系列视频帧,视频帧分辨率采用最佳分辨率(本文中为 1080p)。

2) 最佳模型配置:在云边调度所使用的云服务器同等配置的硬件平台,针对每个任务分别部署精度最优的重量型模型和时延最少的轻量型模型。

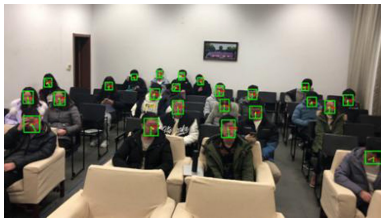
3) 数据标注:用 2) 中部署的任务模型以及 1) 中获取的视频帧执行任务,得到精度标注结果(在本文中为 人脸框 bounding box 以及姿态角度)和时延标注结果(在本文中为 人脸检测时间、姿态估计时间)。

4) 其他视频配置标注:用 3) 中获得的最佳视频帧标注结果生成其他视频配置标注(例如人脸检测的结果 bounding box 需要根据分辨率的比例进行等比放缩)。

“最佳配置标注法”得到的 ground truth 如图 9 所示(涉及隐私部分已做模糊处理),其代表了理想情况下本研究中多目标视频任务处理的性能上限,在调度算法的执行结果评估中会将实际测量的精度、时延结果与 ground truth 中的相应标准求比值,得到去量纲的表现结果指数,从而可以进一步计算用于调度算法更新的奖励值。



(a) 训练数据示例



(b) 测试数据示例

图 9 实验数据的 ground truth 示例

Fig. 9 Examples of ground truth in experiment data

4.3 实验环境

实验使用 Nvidia Jetson Nano 作为边缘端节点,该硬件节点的处理器的 Maxwell 架构的 Quad-Core ARM Cortex-A57,同时具有 128 个 NVIDIA CUDA 核心的 GPU,如图 10 所示。

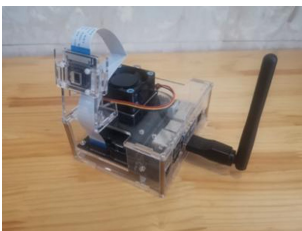


图 10 实验所用的 Jetson Nano 边缘节点

Fig. 10 Jetson Nano used as edge node in experiments

同时,服务器采用 Intel Core I7-8750H 处理器以及 Nvidia GeForce GTX 1060 显卡,具体实验硬件配置如表 8 所列。

表 8 实验硬件环境配置

配置	边缘端	云端
处理器	Quad-Core ARM Cortex-A57	Intel Core I7-8750H
内存容量/GB	4	16
显卡	128 个 NVIDIA CUDA 核心	Nvidia GeForce GTX 1060
显存容量/GB	4	6

4.4 实验结果与分析

本文实验中采用课堂出勤率检测任务和课堂听课率检测任务作为实验任务。其中,课堂出勤率检测任务包含了人脸检测,可以代表单阶段任务的特性;课堂听课率检测任务包含人脸检测、姿态估计,可以代表多阶段任务的特性。

实验中用于对比的基线策略如下:

1) 完全云端部署策略。任务完全在云端处理,从边缘摄像头获取视频帧画面后,立即将视频数据发往云端处理并得到结果。其中,人脸检测的模型为 SSD RFB,姿态估计模型为 FSA Net。

2) 完全边缘端部署策略。任务完全在边缘端处理,从本地边缘摄像头获取视频帧后直接在本地处理得到结果。其中,人脸检测模型为 Center Face,姿态估计模型为 PnP。

3) 组合配置优化策略。预先测量不同配置组合下的调度方案的时延性能结果,在任务执行时,根据当前任务的实时执行时延匹配最接近的配置组合方案作为调度决策,调度决策中的模型配置和视频流配置同 ATS-SAC 模型(如表 4 和表 5 所列)。

我们统计了不同调度策略在多目标状态感知任务中的总体调度效果,选取 120s 的连续测试视频进行测试,每 3s 为一个测试单元。在一个测试单元内,调度策略完成一次调度计划输出,系统完成调度计划的执行,并统计任务执行结果。

表 9 列出了课堂出勤率检测任务中 ATS-SAC 的均衡模式、极速模式、精度模式与云端策略、边缘端策略调度下多目标状态感知任务的精度质量和时延情况,表 10 列出了课堂听课率检测任务中这 5 种策略调度下的精度和时延情况。其中,对于课堂出勤率检测任务,人脸检测精度为检测出的正确人脸框数量占总人脸框数量的比例,综合任务质量与人脸检测精度相同,人脸检测时延、云边传输时延为任务执行和消息传递的真实记录时延(单位为 s),综合任务时延为任务执行整个过程的总计时间开销(单位为 s);对于课堂听课率检测任务,人脸检测精度为检测出的正确人脸框数量占总人脸框数量的比例,姿态估计误差为检测出的人脸框中估计姿态角与实际姿态角的平均绝对误差(MAE),综合任务质量为最终是否抬头检测判断正确的数量占总人数的比例,人脸检测时延、姿态估计时延、云边传输时延均为任务执行和消息传递的真实记录时延(单位为 s),综合任务时延为任务执行整个过程的总计时间开销(单位为 s)。

表9 不同调度策略下课堂出勤率检测任务执行情况

Table 9 Performance of different scheduling policies in classroom attendance rate detection task

策略	精度		时延		
	人脸检测精度	综合任务质量	人脸检测时延	云边传输时延	综合任务时延
ATS-SAC 均衡模式	0.8127	0.8127	0.2101	0.2013	0.4114
ATS-SAC 极速模式	0.7945	0.7945	0.1988	0.0699	0.2687
ATS-SAC 精度模式	0.8687	0.8687	0.3008	0.2574	0.5582
完全云端部署策略	0.8627	0.8627	0.3694	0.3908	0.7602
完全边缘端部署策略	0.7839	0.7839	0.1108	0.0000	0.1108
组合配置优化策略	0.8275	0.8275	0.2610	0.2584	0.5194

表10 不同调度策略下课堂听课率检测任务执行情况

Table 10 Performance of different scheduling policies in classroom listening rate detection task

策略	精度			时延			
	人脸检测精度	姿态估计误差	综合任务质量	人脸检测时延	姿态估计时延	云边传输时延	综合任务时延
ATS-SAC 均衡模式	0.8595	10.2041	0.7081	0.2899	0.1705	0.0714	0.5318
ATS-SAC 极速模式	0.7744	10.3208	0.6629	0.2976	0.1566	0.0375	0.4907
ATS-SAC 精度模式	0.8715	9.8702	0.7097	0.3528	0.2733	0.1748	0.8009
完全云端部署策略	0.8969	8.2213	0.8501	0.3514	0.3316	0.3978	1.0808
完全边缘端部署策略	0.7764	12.9765	0.6013	0.2084	0.1213	0.0000	0.3297
组合配置优化策略	0.8243	10.8731	0.6939	0.3255	0.1656	0.1073	0.5984

从表9和表10的对比结果中可以观察到,完全云端策略下的任务都在云端完成,这使得它能充分利用云端丰富的硬件资源以及发挥模型的最佳推理能力,因此在精度性能上往往能达到最优,然而,这种策略的整体任务时延却不可避免地大幅增加。完全边缘端策略下的任务都在边缘端完成,任务推理模型直接接收本地的视频流输入,没有网络传输的时间开销,受限硬件资源平台上轻量型的部署模型也带来了更小的时延,因此其在整体时延性能上均处于最小值,但精度上却有较大的损失。相较于完全云端部署策略和完全边缘端部署策略这两种策略,ATS-SAC的调度策略更能平衡精度与时延的关系,能够达到精度-时延的帕累托最优解(Pareto Optimality),从而为用户提供更优质的整体体验。同时,ATS-SAC模型相较于组合配置优化策略在精度和时延质量上都有所提升,例如在课堂听课率检测任务中,ATS-SAC均衡模式比组合配置优化策略在精度上提升2.05%,在时延上降低了11.13%。这主要是由于组合配置优化策略在面对复杂状态变化的环境下难以很好地匹配出最优配置组合,而ATS-SAC调度模型采用深度强化学习算法,可以较好地学习环境改变对调度效果的影响,从而调整调度策略适应环境。此外,ATS-SAC的3种不同调度模式也会产生不同的调度结果。例如,在精度模式下,调度策略会进行调整以使整体时延相对较大,而处理质量显著提升;在极速模式下,调度策略会牺牲部分精度性能以换取更小的处理时延。这表明,ATS-SAC能够满足不同的用户体验需求倾向。

从表9和表10的对比中也可以看出,相比单任务场景,ATS-SAC调度算法在多任务场景下对时延的优化更显著。例如,在课堂出勤率检测任务中,ATS-SAC均衡模式的综合任务时延相较于完全云端部署策略降低了45.88%,而在课堂听课率检测任务中,ATS-SAC均衡模式的综合任务时延相较于完全云端部署策略降低了50.8%。主要原因是,在多任务场景下,ATS-SAC调度算法有更灵活的模型云边部署安排策略,能够将不同任务均衡负载在云端和边缘端,更有利于进一步优化时延性能表现。

为了探究ATS-SAC模型对不同运行时状态的自适应调整能力,我们从任务状态和系统状态两方面进行实验测试。

先测试任务状态变化下ATS-SAC的调度表现。我们采用人群稀疏到人群密集的视频场景模拟刚上课时学生逐渐进入教室导致的目标数量、目标大小等任务状态变化的情况,并测试调度策略在课堂听课率检测任务中的表现。测试视频中目标数量的变化如图11所示,目标大小的变化如图12所示。测试中,调度算法输出并执行调度决策的时间间隔为3s。

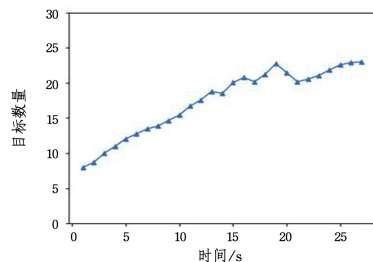


图11 目标数量变化情况

Fig. 11 Changes in the number of targets

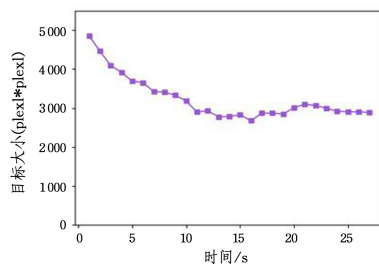


图12 平均目标大小变化情况

Fig. 12 Changes in the size of targets

图13展示了不同调度策略随任务状态变化的时延变化情况,图14展示了不同调度策略随视频的进展的精度变化情况。其中背景颜色为ATS-SAC调度算法的调度决策动作,不同的颜色代表了不同配置决策组合的编码。第1个时间段(0-3s)中调度策略为Action3:(分辨率:1080P,帧率:20FPS,编码方式:H.264,[人脸检测:边缘端/SSD_Tiny,姿态估计:

云端/HopeNet]};第3个时间段(7-9 s)中调度策略为 Action20;{分辨率:720 P,帧率:10 FPS,编码方式:H. 264,[人脸检测:边缘端/SSD_Tiny,姿态估计:云端/HopeNetLite]};第5个时间段(13-15 s)中调度策略为 Action6;{分辨率:1 080 P,帧率:10 FPS,编码方式: MJPEG,[人脸检测:边缘端/Center_Face,姿态估计:边缘端/PnP]}。

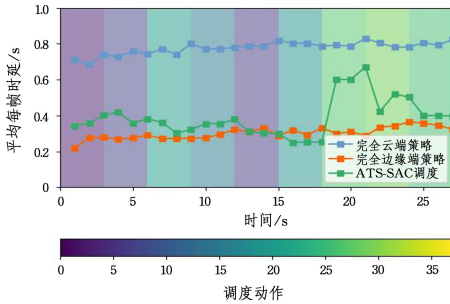


图 13 任务状态变化下不同调度策略的时延变化情况

Fig. 13 Latency of different policies changes with task states

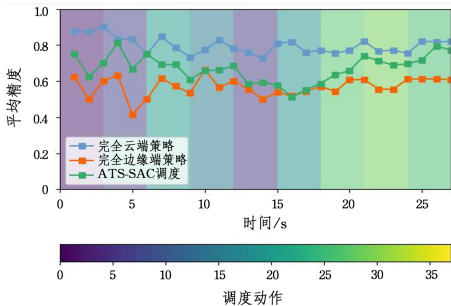


图 14 任务状态变化下不同调度策略的精度变化情况

Fig. 14 Accuracy of different policies changes with task states

从图中可以看出,相比完全云端策略和完全边缘策略,ATS-SAC算法可以很好地根据任务状态的复杂情况动态改变任务策略,从而达到更稳定的用户体验效果。例如,在第12 s时,随着进入教室的同学大量增加,任务执行时延上升,ATS-SAC算法输出的调度策略中将原来的“人脸检测在边缘端、姿态估计在云端”模型部署方式更改为“人脸检测、姿态估计都在边缘端”的方式,从而及时降低了相对任务时延,稳定了用户对任务执行结果的体验情况。

类似地,我们对ATS-SAC模型在系统状态变化情况下的调度效果进行测试。以系统状态中对云边协同调度任务影响最大的网络情况为例进行实验。如图15所示,我们手动改变云端与边缘端之间的网络情况,使云边之间通信的网络带宽由大变,并测试课堂听课率检测任务的执行效果。

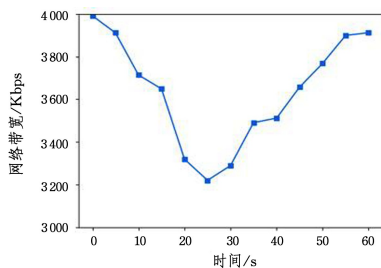


图 15 带宽变化情况

Fig. 15 Changes in network bandwidth

在网络质量变化的情况下,调度策略时延变化情况如图16所示。

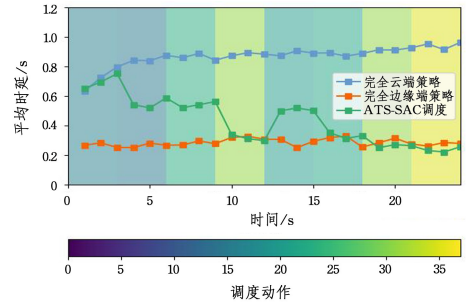


图 16 系统状态变化下不同调度策略的时延变化情况

Fig. 16 Latency of policies changes with system states

可以看出,随着网络质量的降低,云边之间的通信时延增加,ATS-SAC调度算法可以及时调整策略,通过将任务负载更多地放在边缘端、降低视频流质量等方式,减小云边之间通信数据的规模,从而保证多目标状态感知任务在网络质量较差情况下顺利、稳定地执行。

相比没有调度的多目标感知任务处理,ATS-SAC模型调度会带来额外的时间开销。图3所示的ATS-SAC结构图中显示,ATS-SAC模型的调度决策部分和调度执行部分互相独立,因此调度决策的生成和任务的执行是并行的,调度决策的时间开销并不会给多目标状态感知任务的执行带来影响;在调度执行部分,相较于没有调度的多目标感知任务,框架增加了云边通信的时间开销。从表9和表10所列的结果中可以看到,在考虑云边通信时间开销后的综合时延上,ATS-SAC仍可以达到时延和精度的相对较优。同时,如图16所示,当网络状态变差时,云边通信时延增加,ATS-SAC调度模型可以自适应调整策略使得整体时延符合用户需求。因此,ATS-SAC调度模型所带来的额外开销不会降低其对于多目标状态感知任务的优化效果。

结束语 针对云边场景下的多目标状态感知任务的优化需求,本研究提出了一种基于深度强化学习的自适应云边协同调度模型ATS-SAC。该调度模型由调度决策部分和调度执行部分组成,通过获取当前系统的运行时状态(包括系统状态和任务状态),然后产生调度决策配置(包括模型配置和视频配置),从而优化多目标状态感知任务的精度和时延。我们采用了基于用户体验极限阈值的动作筛选方法对调度决策动作进行筛选,以减少动作向量的维度。同时,利用不同的精度和时延加权,设定了调度模型的不同调度策略。

在未来,随着基础设施的持续完善和硬件性能的进一步提升,多边与云的协同以及更复杂的云-边-端协同网络可能会成为提升实时场景智能处理任务性能的途径。基于边缘计算的云边协同将为智慧监控网络的构建提供一种可行的解决方案。

参考文献

- [1] SHAO Z, CAI J J, WANG Z Y, et al. Big data analysis and processing of surveillance video for intelligent surveillance cameras [J]. Journal of Electronics and Information Technology, 2017,

- 39(5):1116-1122.
- [2] WANG N. Surveillance camera visualization management system in smart grid [J]. *Journal of North China Institute of Water Resources and Hydropower*, 2011, 32(1):111-113.
- [3] SMOLYANSKIY N, KAMENEV A, SMITH J, et al. Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness [C] // 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017:4241-4247.
- [4] KOURIS A, BOUGANIS C S. Learning to fly by myself: A self-supervised cnn-based approach for autonomous navigation [C] // 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018:1-9.
- [5] GEORGANAS E, AVANCHA S, BANERJEE K, et al. Anatomy of high-performance deep learning convolutions on simd architectures [C] // International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2018:830-841.
- [6] JOUPPI N P, YOUNG C, PATIL N, et al. In-datacenter performance analysis of a tensor processing unit [C] // Proceedings of the 44th Annual International Symposium on Computer Architecture. 2017:1-12.
- [7] ZHANG M, WANG F, LIU J. Casva: Configuration-adaptive streaming for live video analytics [C] // IEEE Conference on Computer Communications. IEEE, 2022:2168-2177.
- [8] SINGH A, CHATTERJEE K. Cloud security issues and challenges: A survey [J]. *Journal of Network and Computer Applications*, 2017, 79:88-115.
- [9] RAN X, CHEN H, ZHU X, et al. Deepdecision: A mobile deep learning framework for edge video analytics [C] // IEEE Conference on Computer Communications. IEEE, 2018:1421-1429.
- [10] HAN S, SHEN H, PHILIPOSE M, et al. Mednn: An approximation-based execution framework for deep stream processing under resource constraints [C] // Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. 2016:123-136.
- [11] ZHANG H, ANANTHANARAYANAN G, BODIK P, et al. Live video analytics at scale with approximation and {Delay-Tolerance} [C] // 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17). 2017:377-392.
- [12] PAKHA C, CHOWDHERY A, JIANG J. Reinventing video streaming for distributed vision analytics [C] // 10th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 18). 2018.
- [13] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. Scaled-yolov4: Scaling cross stage partial network [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021:13029-13038.
- [14] CHEN Q, WANG Y, YANG T, et al. You only look one-level feature [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021:13039-13048.
- [15] LI X, WANG W, HU X, et al. Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021:11632-11641.
- [16] FAN M, LAI S, HUANG J, et al. Rethinking bisenet for real-time semantic segmentation [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021:9716-9725.
- [17] YU C, ZHOU Q, LI J, et al. Foundation Model Drives Weakly Incremental Learning for Semantic Segmentation [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023:23685-23694.
- [18] CAI J, XU M, LI W, et al. Memot: Multi-object tracking with memory [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022:8090-8100.
- [19] BEWLEY A, GE Z, OTT L, et al. Simple online and realtime tracking [C] // 2016 IEEE International Conference on Image Processing (ICIP). IEEE, 2016:3464-3468.
- [20] WANDT B, RUDOLPH M, ZELL P, et al. Canonpose: Self-supervised monocular 3d human pose estimation in the wild [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021:13294-13304.
- [21] ALMEIDA M, LASKARIDIS S, LEONTIADIS I, et al. Embench: Quantifying performance variations of deep neural networks across modern commodity devices [C] // The 3rd International Workshop on Deep Learning for Mobile Systems and Applications. 2019:1-6.
- [22] SHEN H, CHEN L, JIN Y, et al. Nexus: A GPU cluster engine for accelerating DNN-based video analysis [C] // Proceedings of the 27th ACM Symposium on Operating Systems Principles. 2019:322-337.
- [23] JIANG J, ANANTHANARAYANAN G, BODIK P, et al. Chameleon: Scalable adaptation of video analytics [C] // Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. 2018:253-266.
- [24] LIU W, ANGUELOV D, ERHAN D, et al. Ssd: Single shot multibox detector [C] // 14th European Conference. Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016:21-37.
- [25] ZHAO H, QI X, SHEN X, et al. Icnets for real-time semantic segmentation on high-resolution images [C] // Proceedings of the European Conference on Computer Vision (ECCV). 2018:405-420.
- [26] HE Y, PAN Z, LI L, et al. Real-time vehicle detection from short-range aerial image with compressed mobilenet [C] // 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019:8339-8345.
- [27] JIANG A H, WONG D L K, CANEL C, et al. Mainstream: Dynamic {Stem-Sharing} for {Multi-Tenant} Video Processing [C] // 2018 USENIX Annual Technical Conference (USENIX ATC 18). 2018:29-42.
- [28] MATSUBARA Y, BAIDYA S, CALLEGARO D, et al. Distilled

split deep neural networks for edge-assisted real-time systems [C]//Proceedings of the 2019 Workshop on Hot Topics in Video Analytics and Intelligent Edges. 2019;21-26.

- [29] RA M R, SHETH A, MUMMERT L, et al. Odessa: enabling interactive perception applications on mobile devices [C]// Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. 2011;43-56.
- [30] WANG Y, WANG W, ZHANG J, et al. Bridging the {Edge-Cloud} Barrier for Real-time Advanced Vision Analytics [C]// 11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19). 2019.
- [31] LI Y, PADMANABHAN A, ZHAO P, et al. Reducto: On-camera filtering for resource-efficient real-time video analytics [C]// Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. 2020;359-376.
- [32] CHEN T Y H, RAVINDRANATH L, DENG S, et al. Glimpse: Continuous, real-time object recognition on mobile devices [C]// Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. 2015;155-168.
- [33] HUANG T, ZHOU C, ZHANG R X, et al. Comyco: Quality-aware adaptive video streaming via imitation learning [C]// Pro-

ceedings of the 27th ACM International Conference on Multimedia. 2019;429-437.

- [34] WANG Z, BAPST V, HEES N, et al. Sample efficient actor-critic with experience replay [J]. arXiv:1611.01224, 2016.
- [35] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement [C]// Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden. 2018;1861-1870.



ZHOU Wenhui, born in 2000, postgraduate, is a student member of CCF (No. E0259G). His main research interests include edge computing and edge intelligence.



XIE Lei, born in 1982, Ph.D, professor, Ph.D supervisor, Young Chang Jiang Scholar, is a distinguished member of CCF (No. 17652D). His main research interests include wireless sensing, wearable computing and edge computing.

(责任编辑:李亚辉)