中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION

边缘智能与协同计算：前沿与进展

侯祥鹏, 兰兰, 陶长乐, 寇小勇, 丛佩金, 邓庆绪, 周俊龙

引用本文:
侯祥鹏,兰兰,陶长乐,寇小勇,丛佩金,邓庆绪,周俊龙. 边缘智能与协同计算: 前沿与进展[J]. 控制与决策, 2024, 39(7): 2385−2404.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2024.0206

边缘智能与协同计算: 前沿与进展

侯祥鹏[1], 兰 兰[1], 陶长乐[1], 寇小勇[1], 丛佩金[1], 邓庆绪[2], 周俊龙[1,3†]

(1. 南京理工大学 计算机科学与工程学院，南京 210094；2. 东北大学 计算机科学与工程学院，沈阳 110819；
3. 东南大学 移动通信全国重点实验室，南京 211111)

摘 要: 随着万物互联时代的到来,边缘设备规模急剧增加,海量数据在网络边缘产生,人工智能技术的飞速发展为分析和处理这些数据提供了强大的支撑. 然而,传统云计算的集中处理模式难以满足用户对任务低时延和设备低功耗的需求,并带来数据隐私泄露的潜在隐患. 与此同时,嵌入式高性能芯片的发展显著提升了边缘设备的计算能力,使其能够在边缘侧实时处理部分计算密集型任务. 在此背景下,边缘计算和人工智能有机融合,孕育了一种新的计算范式: 边缘智能. 鉴于此,聚焦边缘智能与协同计算的前沿与进展,首先概述边缘计算、人工智能和边缘智能的相关背景、基本原理与发展趋势;然后从训练、推理和缓存 3 个方面回顾面向单个设备的边缘智能方法;接着从架构、技术和功能 3 个维度介绍多个设备合作实现边缘智能协同的相关工作;最后总结边缘智能在工业物联网、智慧城市和虚拟现实等领域的广泛应用.

关键词: 边缘智能；边缘设备；边缘训练；边缘推理；边缘缓存；协同计算

中图分类号: TP391 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2024.0206

引用格式: 侯祥鹏,兰兰,陶长乐,等. 边缘智能与协同计算: 前沿与进展[J]. 控制与决策, 2024, 39(7): 2385-2404.

Edge intelligence and collaborative computing: Frontiers and advances

HOU Xiang-peng[1], *LAN Lan*[1], *TAO Chang-le*[1], *KOU Xiao-yong*[1], *CONG Pei-jin*[1], *DENG Qing-xu*[2], *ZHOU Jun-long*[1,3†]

(1. School of Computer Science and Engineering，Nanjing University of Science and Technology，Nanjing 210094，China；2. School of Computer Science and Engineering，Northeastern University，Shenyang 110819，China；3. National Mobile Communications Research Laboratory，Southeast University，Nanjing 211111，China)

Abstract: With the advent of the Internet of Everything era, there has been a dramatic increase in the number of edge devices, leading to the generation of massive amounts of data at the network edge. The development of artificial intelligence (AI) technology provides powerful support for analyzing and processing these data. However, the traditional centralized processing model of cloud computing fails to meet users' demands for low latency of tasks and low power consumption of devices. In addition, it poses potential threats to data privacy and security. At the same time, the development of embedded high-performance chips has greatly enhanced the computing capabilities of edge devices, enabling them to process computation-intensive tasks in real-time at the edge. In light of this, edge computing (EC) and AI are organically integrated, giving rise to a new computing paradigm known as edge intelligence (EI). This paper focuses on the frontiers and advances in EI and collaborative computing. Firstly, we introduce the relevant background, basic principles, and development trends of EC, AI, and EI. Secondly, we review EI methods for individual devices, covering edge training, edge inference, and edge caching. Thirdly, we present the collaborative EI works on multiple devices from the perspectives of architecture, technology, and functionality. Finally, we summarize the wide-ranging applications of EI in various fields, such as the industrial Internet of Things, smart cities, and virtual reality.

Keywords: edge intelligence；edge device；edge training；edge inference；edge caching；collaborative computing

0 引 言

近年来,随着信息技术的不断进步,物联网节点(即边缘设备)在工业制造、城市交通、社交生活等各个领域得到了广泛的应用和快速发展. IDC (international data corporation) 数据表明,全球物联网产业市场规模不断扩大,总支出在 2022 年达到 7 300

收稿日期: 2024-02-29；**录用日期:** 2024-05-09.

基金项目: 国家自然科学基金项目 (62172224, 62302221, U23B2006)；江苏省自然科学基金项目 (BK20220138, BK20230913)；中央高校基本科研业务费专项资金项目 (30922010318, 30922010406)；教育部产学研创新基金项目 (2021ITA01004)；东南大学移动通信全国重点实验室开放研究基金项目 (2024D07).

†通讯作者. E-mail: jlzhou@njust.edu.cn.

亿美元,预计2027年接近1.2万亿美元[1]. 与此同时,GSMA (global system for mobile communications association)也指出,截至2022年底,全球移动互联网用户数为44亿,预计到2030年将达到55亿[2]. 在这个万物互联的时代,每天都在产生海量数据,数据规模急剧增加[3]. 然而,传统云计算集中处理模式面对上述边缘设备数量规模的不断扩大和数据量的爆炸式增长表现出诸多不足之处,例如:

1) 带宽资源有限. 边缘设备规模庞大,每天产生大量的实时数据,如果将海量数据都上传至云服务器进行计算,将占用大量的网络带宽资源.

2) 网络延迟较高. 云计算模式下,终端设备将数据卸载至云服务器进行处理,云服务器再将处理后的结果传回终端设备,增加了网络延迟[4].

3) 隐私泄露风险. 大数据时代,终端数据经常会涉及个人隐私,直接上传至云端处理将带来很高的隐私泄露风险[5].

4) 能量消耗增大. 云数据中心因海量计算任务而产生较高的能量消耗,并且高排放也对实现碳中和目标构成了巨大挑战.

为应对上述挑战,边缘计算应运而生. 与传统的集中式云计算不同,边缘计算是一种将计算基础设施部署到网络边缘的计算新范式,通过在终端设备或者靠近网络边缘的服务器上处理数据和用户请求,能够有效减少数据传输带宽、降低服务延迟、保护数据隐私、缓解云数据中心压力[6]. 凭借这些优势,边缘计算近年来得到了迅速发展. 根据IDC调查,2023年上半年,中国边缘云市场规模同比增长46.3%,并且还在保持快速增长[7].

嵌入式高性能芯片的发展让边缘设备的算力获得了显著提升,使边缘设备拥有实时处理部分计算密集型任务的能力,越来越多的人工智能应用迁移到边缘侧执行. 边缘计算与人工智能的有效融合催生了一个全新的研究领域:边缘智能(edge intelligence, EI)[8]. 它赋予了边缘设备执行智能算法的能力,也能使端侧计算变得更加智能高效. 近年来,边缘智能已成为学术界和工业界的研究热点.

本文对边缘智能的前沿与进展进行梳理和介绍,如图1所示. 首先介绍边缘计算、人工智能和边缘智能的相关背景、基本原理与发展趋势;然后从边缘训练、边缘推理和边缘缓存3个方面回顾面向单个设备的边缘智能研究工作;接着从架构、技术和功能3个维度回顾面向多个设备的边缘智能协同研究工作;最后总结边缘智能在工业物联网、智慧城市、虚拟现实等诸多场景中的应用.

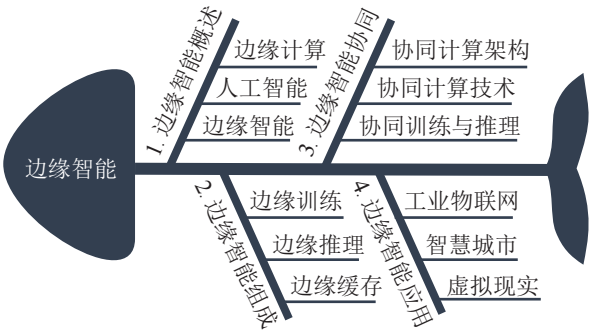

图1 本文综述内容整体脉络

1 边缘智能概述

1.1 边缘计算

边缘计算是一种将数据处理和计算能力推向接近数据源和终端设备的网络边缘的新型计算模式,其中网络边缘涵盖从数据源到云服务中心之间的任意节点[9]. 边缘计算主要是将计算和存储功能移至网络边缘,在边缘处进行计算和资源管理,为任务处理提供更低的延迟和更高的带宽. 它的快速发展与物联网、5G通信、人工智能[10]等技术密切相关. 万物互联的时代背景下,边缘设备和传感器产生大量数据[11],需要本地实时处理和分析.

以无人驾驶汽车为例,相关研究显示,无人车的传感器每秒产生约1 GB数据[9],但车与云服务器之间的带宽不足以支持实时传输. 于是在边缘计算场景下,计算和数据处理将被推向车辆附近的边缘节点,利用数据在网络边缘处理相较云服务器更近的优势,缓解云端带宽占用,降低数据传输延迟,提供实时响应和决策. 此外,车辆会产生大量敏感信息,包括定位信息、路况信息、车载信息等. 边缘计算可以将数据处理和存储功能保留在本地或者信任的边缘节点,降低数据泄露风险和减少传输带宽占用.

1.2 人工智能

随着深度学习的蓬勃发展,人工智能(artificial intelligence, AI)近些年再次走进大众视野并受到广泛关注. 它的提出最早可以追溯到20世纪50年代,在达特茅斯会议上,各学者探讨、规划AI未来发展[12]. 简单地说,人工智能是一种使计算机系统具备模拟、理解和执行人类智能任务的能力[13],它的发展经历了符号主义、连接主义再到后来的统计学习阶段. 机器学习是人工智能的一个核心分支,它使用算法来解析数据,从中学习,并做出决策或预测,包括监督学习、无监督学习等多种分类. 深度学习,作为机器学习的一个子集,依赖于使用多层神经网络处理大量数据,借助较强的计算能力,能够在图像和语音识别等复杂任务中达到很高的准确度[14-15]. 大数据时代,AI

凭借出色的海量数据处理和泛化能力,在数据挖掘、计算机视觉和自然语言处理[16]等应用领域效果显著,带来智能化的解决方案. 然而,实际部署面临着诸多挑战. 例如边缘端设备资源有限,部署AI模型需要借助云端强大的算力,但这会导致延迟和通信压力增大,同时数据上传至云端也会带来隐私安全问题[17]. 因此,需要综合考虑计算能力、存储容量、网络带宽、隐私保护等因素,在边缘设备上高效地部署和运行AI模型.

1.3　边缘智能

边缘计算与AI的融合给未来带来无限可能,两者相辅相成. 一方面,AI近些年高速发展,但需要依靠云端算力支撑,边缘计算的出现可以让数据处理更靠近数据源,为AI带来更多的应用场景. 另一方面,边缘计算的发展也依靠AI为其提供更多的解决方案,帮助其释放更大的发展潜力[18].

边缘智能是边缘计算框架与机器学习和深度学习等AI模型的结合[12],旨在增强数据处理和用户数据隐私保护等. 近年来,边缘智能发展迅速,边缘训练、边缘推理和边缘缓存都是其聚焦点. 在谷歌学术上搜索以“Edge Training”“Edge Inference”和“Edge Caching”为关键词的文章,可得到近10年边缘智能相关文章发表数量随时间变化的趋势,如图2所示. 2016年之前,边缘智能研究处于起步阶段,但在过去7年中,由于物联网的繁荣和人工智能等技术的推动,边缘智能的发展呈现爆炸式增长. 由此可见,边缘智能的前景极为广阔. 例如,提高虚拟现实视频的服务质量,为用户提供低延迟、高渲染的沉浸式服务体验[19],结合区块链[20]技术实现在传输过程中的隐私保护. 目前5G网络技术已经得到很大发展,但依据国际电信联盟统计数据,预计2030年现有的5G网络技术将达到容量极限,因此结合6G通信技术有望更进一步缩短延迟和降低成本等[21].

图2　边缘智能相关文献发表数量变化趋势

2　边缘智能组成

本章聚焦单个设备上的边缘智能技术研究,即在边缘设备本地进行数据处理和分析. 具体由3个方面的研究组成: 边缘训练、边缘推理、边缘缓存.

2.1　边缘训练

标准的深度神经网络训练任务依赖于大量数据集的集中式处理和高性能计算资源,然而将用户数据汇聚至云端会面临着数据隐私安全问题,边缘训练是解决这一问题的有效途径. 边缘训练是指在边缘侧利用数据对模型进行迭代训练,仅在边缘设备上完成训练任务,而不依赖于云服务器. 虽然边缘训练可以实现更加安全的数据处理,但是由于边缘设备通常计算和能量资源受限,可能导致训练效率低下. 因此,大量研究人员致力于优化边缘设备上神经网络模型训练的速度、能耗和内存效率.

这类工作主要采用软硬件协同设计模式,借助数据压缩、稀疏、流水线技术等方法减少片外内存访问和计算量以优化训练[22-24]. 例如,Li等[22]提出了一种近似计算的批归一化(batch normalization, BN)算法,通过使用每个特征图在小批量中的最大神经元和最小神经元的平均值近似小批量均值,同时使用尺度调整因子,通过将其与每个特征图范围的平均值相乘来近似小批量标准差,以达到训练过程中减少BN层的浮点运算次数和外部内存访问次数的目的,提高训练效率. 其设计了一个集成前向和后向计算的硬件模块,进一步提高深度神经网络(deep neural network, DNN)的训练效率. 模型训练通常需要进行反向传播和权重更新过程. 在这两个过程中,梯度和权重都需要与外部内存进行昂贵的通信. Jung等[23]观察到,权重和梯度包含负值,并且集中在零附近. 当这些数据进行二值化时,小的负幅度会产生大量1值位,对稀疏性产生不利影响,从而降低稀疏性压缩的效率. 其提出了一种附加的数据转换方法,通过反转以增加零位和将增加的零位推向最高有效位来保证数据的稀疏性,提高端设备的个性化训练效率. Hong等[24]提出了一种变体反向传播算法,将符号对称固定反馈作为误差梯度的调制器信号,并采用随机方法对误差梯度剪枝,通过自身设定的调制器信号与反向传播之间达到小角度来保持学习能力;然后设计了一种混合数据流的数据重用架构,通过消除转置权矩阵的获取/存储和反向相位中涉及的小梯度,显著提高了能量效率,可以适应能量受限的边缘设备.

2.2　边缘推理

边缘推理是将训练后的模型部署在边缘设备上,通过前向传递计算输出,实现对端侧海量数据的实时处理. 例如,手机端可以部署超分辨率算法模型,在视频播放时对图像实时处理,从而在网络环境较

差或流量受限情况下达到画质增强的效果,提高用户体验. 虽然端侧执行可提高服务的响应速度,减轻网络带宽压力的同时保护数据隐私安全,但是性能优异的算法模型往往是为在具有强大GPU (graphics processing unit, GPU) 的设备上运行而设计的,需要极高的算力和电力支持,在资源受限的边缘环境中并不适用. 因此,实现边缘高效推理存在两方面的挑战: 1) 如何使参数量庞大的算法模型可以部署在存储、算力、电力等资源受限的边缘设备上; 2) 在算法模型确定后,如何加速推理,以提供对数据的高效处理和实时响应. 相关工作主要集中于通过模型压缩和推理优化来分别解决上述挑战,具体介绍如下.

2.2.1 模型压缩

现有研究表明,神经网络中存在许多冗余的组件和参数,参与推理过程但对最终结果没有影响或影响甚微[25-27]. 基于这一观察,研究人员发现通过删除冗余的计算参数,可有效降低模型对存储和计算资源的需求,提高边缘推理效率. 本文将这些工作按所采用技术归类为低秩分解、参数量化、网络剪枝和知识蒸馏.

1) 低秩分解. 低秩分解的核心思想是将参数矩阵分解为多个较小的矩阵,以达到节省参数的目的. 这类工作源于Denil等在2013年的研究[28],其证明了几个深度学习模型的参数存在显著冗余和低秩的特点,并且根据每个特征的少量权重值便可以准确预测其余的值. 一般而言,卷积层计算更耗时,全连接层则存储占用更多. 早期工作大多数集中在加速卷积层计算[29]或压缩全连接层[30],没有联合优化两者. Lin等[31]设计了一种整体的压缩框架,提出一种基于低秩分解的层间压缩方法的同时压缩卷积层和全连接层,并利用闭式求解器来弥补迭代优化求解器[32]效率低下的问题,高效消除了卷积核和全连接层的参数冗余.

2) 参数量化. 神经网络中的参数通常以32位浮点数存储和计算. 因此,一个很显然的思路是采用更低位宽 (如1-bit, 2-bit, 8-bit等) 表示这些参数,从而降低模型的存储和计算需求,即参数量化. 相关工作按照模型训练和量化的顺序可以分为两类: 量化感知训练 (quantization-aware training, QAT)[33-34]和训练后量化 (post-training quantization, PTQ)[35-36]. QAT方法在模型训练过程中进行量化,使模型能够适应低位表示并保持精度. 例如, Liu等[33]针对大语言模型的量化感知训练问题,对权重和键-值缓存进行量化,能够将大型LLaMA模型[37]蒸馏为仅使用4位的模型. 而PTQ则在训练完成后再进行量化,可实现更高效的模型压缩,但是可能会引入一定程度的精度损失. 为了应对这种挑战, Frantar等[36]提出了一种基于近似二阶信息的新型一次性权重量化技术,可以将每个权重的位宽降低到3或4位,与未压缩基准相比几乎无精度损失.

3) 网络剪枝. 网络剪枝的主要思想是通过对训练完成的模型删除冗余参数和组件来提高存储和计算效率,达到压缩模型的目的. 按照修剪对象和产生的网络结构,近期研究可以分为非结构化剪枝[38-40]和结构化剪枝[41-42]两类. 非结构化剪枝的修剪对象是网络中冗余的单个权重连接. 例如, SparseGPT[38]采用单次修剪策略在最大的GPT模型 (如BLOOM-176B) 上实现了高达60%的显著稀疏. Zhang等[39]针对循环神经网络设计了一种基于循环雅可比矩阵谱的一次性剪枝算法. 这类方法虽然可以实现高稀疏度,但可能会导致低计算并行性. 结构化剪枝则通过删除网络中的整个结构组件 (例如注意力头、通道或层) 来保留整体网络结构,实现更快加速. 例如, Michel等[41]通过动态移除Transformer中的部分多头注意力获得推理速度、内存效率的提升. Ma等[42]针对大语言模型,提出了基于梯度信息的结构剪枝方法LLM-Pruner,有选择性地移除非关键的耦合结构.

4) 知识蒸馏. 知识蒸馏的核心思想是通过从较大的模型 (教师模型) 中提取精华知识,训练参数量较小的模型 (学生模型),使其拥有接近教师模型推理能力的同时兼具模型紧凑的优点. 按照参与知识蒸馏的模型数量和类别,可以将此类工作归纳为多教师学习、教师助理、相互蒸馏以及自蒸馏4个方面. 多教师学习[43-44]是将多个教师模型的知识迁移到单个学生模型中,可以避免单个教师模型知识不足的情况. Li等[43]提出了一种多教师知识蒸馏模型IsMt-KD,蒸馏出轻量级的驾驶员分心姿势检测模型. 其设计了一个特定实例教师评分模块,根据每个实例动态分配教师模型的权重,以此避免教师模型对不相关类别过度自信的情况. 然而,当教师模型和学生模型规模差异较大时,学生模型的学习效果可能较差. 为解决这类问题,研究工作[45-46]使用一个或多个规模位于教师模型与学生模型之间的网络作为知识传递的中间模型,即教师助理. 与上述两种方法不同,相互蒸馏方法[47-48]中知识是双向传递的,即让一组未经训练的网络模型互相训练,可进一步提高网络推理性能. 自蒸馏方法[49-51]的对象是单个网络,利用其自身

的知识进行自我蒸馏. 比如,将模型网络较深层的知识蒸馏到网络较浅部分[49-50],或者将网络早期阶段的知识蒸馏到后期阶段[51],而无需采用复杂的教师模型. 模型压缩方法对比如表1所示.

表 1　模型压缩方法对比

方法	文献	细分方法	主要优化目标				性能	
			推理速度	模型尺寸	稀疏度	准确度	提升	基准
低秩分解	[29]	CP分解	√	–	–	–	↑ 4.5× 速度	AlexNet
	[30]	tensor-train分解	–	√	–	–	↓ 7× 尺寸	VGG-19
	[31]	rank-1分解	√	–	–	√	↑ 2.12× 速度	AlexNet
	[32]	低秩近似	√	–	–	–	↑ 5.48× 速度	AlexNet
参数量化	[33]	量化感知训练	–	–	–	√	69.7% (↑ 19%) 零样本准确率	SmoothQuant[26]
	[34]	量化感知训练	√	–	–	√	94.5%准确度	REDDIT-BINARY
	[35]	训练后量化	–	–	–	√	68.48% (↑ 27.07%) top-1准确度	APQ-ViT[27]
	[36]	训练后量化	–	√	–	–	3.72× smaller	OPT-175B
网络剪枝	[38]	非结构化剪枝	–	–	√	–	60%稀疏度	OPT-175B
	[39]	非结构化剪枝	–	–	√	–	95%稀疏度	标准LSTM
	[40]	非结构化剪枝	–	–	√	–	99%稀疏度	LeNet-5-Caffe
	[41]	结构化剪枝	–	√	–	–	修剪40%注意力头	BERT
	[42]	结构化剪枝	–	√	–	–	20% smaller	LLaMA-7B
知识蒸馏	[43]	多教师单学生蒸馏	–	–	–	√	92.32%准确度	StateFarm数据集
	[44]	多教师单学生蒸馏	√	√	–	–	↓ 36× 尺寸,↑ 13.17× 速度	CIFAR-10数据集
	[45]	教师助理	–	√	–	√	89.02% (↑ 1.01%) 准确度	TAKD
	[46]	教师助理	–	√	–	√	88.98% (↑ 0.46%) 准确度	NOKD
	[47]	双向蒸馏	–	√	–	√	↓ 2.34% top-1 错误率	ResNet-56
	[48]	双向蒸馏	–	√	–	√	↑ 1.12%准确度	ResNet-164
	[49]	自蒸馏	–	–	–	√	↑ 4.07%准确度	VGG-19
	[50]	自蒸馏	–	–	–	√	↑ 3.49% 准确度	CIFAR-100数据集
	[51]	自蒸馏	–	–	–	√	↓1.23% top-1错误率	ResNet-152

综上所述,低秩分解将大权重矩阵分解为多个小矩阵,以近似原有权重矩阵,减少参数数量和计算开销,但需要特定的优化算法来保持模型性能,并且对于小卷积核可能不适用. 参数量化将模型中的浮点数参数转换为低精度的表示,以加速推理过程,但通常会导致精度损失,影响模型性能,需要额外的训练或调整来恢复部分精度. 网络剪枝通过移除模型中不重要的权重或神经元来简化模型结构,可以提高模型的稀疏性,从而减少存储和计算需求,但选取移除的权重或神经元通常是有挑战性的,而且可能会损害模型的泛化能力. 知识蒸馏将一个大型“教师”模型的知识转移到一个小型“学生”模型中,使学生模型在保持相似性能的同时拥有更少的参数,但需要着重选取高质量的教师模型. 简而言之,低秩分解和参数量化主要关注于减少模型的存储和计算需求,而网络剪枝和知识蒸馏则侧重于减少模型的复杂性和提高效率. 在选择模型压缩技术时,需要综合考虑具体的应用场景、资源限制以及性能要求.

2.2.2　推理优化

在不改变网络结构的前提下,优化模型在边缘设备上运行的推理能耗和时延也尤为重要. 根据优化方式,研究工作可分为两类:硬件优化和软件优化.

1) 硬件优化方法侧重于将模型推理任务并行化到可用的异构处理器资源上,如中央处理器(central processing unit, CPU)、图形处理器GPU、数据处理器(data processing unit, DPU)和神经处理器(neural processing unit, NPU)等. 研究人员[52-53]评估了常见的DNN模型在这些处理器上推理性能的差异性. 例如,根据Tan等的评估[52],与HUAWEI Mate10Pro手机上搭载的CPU相比,在NPU上运行VGG、 VocNet、

AlexNet 等网络模型可降低 95 % 的处理时间, 但同时也会因为 NPU 只支持 FP16 精度计算而产生最高 30 % 的精度损失. 因此, 在模型部署时应充分利用设备硬件的异构特性, 进一步提高模型的推理性能, 优化能耗和时延. Du 等[54] 提出一种部署在 FPGA (field programmable gate array) 平台异构多 DPU 引擎上的流水线卷积神经网络 (convolutional neural network, CNN) 推理方案, 自适应判断 CNN 推理任务的最佳执行模式, 包括任务级并行和流水线推理模式. CNN 模型被划分为多个由连续层组成的片段, 每个片段可以映射到不同的 DPU 引擎上, 以加速推理. 文献 [55] 同时考虑推理速度和功耗的协同优化, 建立 DNN 需求-资源匹配模型, 利用多目标强化学习方法自适应配置 CPU-GPU 的资源分配方案.

2) 软件优化方法包含资源管理、流水线设计和编译器优化等方式[56-59]. Georgiev 等[56] 研究了能量受限设备上语音识别模型的性能与能耗之间的权衡问题. 其利用内存访问并行模式, 建立更多线程独立地处理音频输入流来增加数据并行性, 并且将线程所需的数据策略性地放置在 GPU 内存缓存中, 以降低访问延迟和能耗. Yang 等[57] 提出了一种自适应深度神经网络加速器 NetAdapt, 根据预算 (如时延、能量等) 自动且逐步简化预训练网络, 直到满足资源预算同时最大化准确性. Zhang 等[58] 和 Qi 等[59] 针对视频实时处理任务, 均提出了一种在数据源处过滤掉时间冗余数据的方法, 过滤掉没有用户定义事件的帧或者与相邻帧相似度高的帧, 以减轻单位时间内推理任务数量, 提高模型的实时性.

2.3 边缘缓存

数据和计算的冗余会增加核心网络的负载, 这通常由用户对高流行度数据和相同服务重复请求引起. 边缘缓存通过将数据或信息存储在网络边缘的设备上, 可以更快速地提供给终端用户, 降低延迟, 提高访问速度, 并缓解核心网络压力. 文献 [60] 通过在线预测和在线学习的方法预测内容文件 (如视频、音乐等) 的受欢迎程度, 以确定缓存的内容. 优先缓存流行度高的内容有助于降低延迟, 但新流入互联网的内容可能难以预测或预先建模[61]. Sun 等[62] 设计了一种基于异构信息网络的未知域文件流行度预测算法来预测终端用户对新内容文件的偏好. 此外, 考虑了多个接入点 (access point, AP) 中的流量使用模式, 使用季节性自回归移动平均模型预测每个 AP 未来的流量使用情况, 并基于按需分配规则得到适当的匹配配额. 当新内容交付并且需要缓存, 但是所有缓存单元都被占用时, 需要替换一些缓存的旧内容. Li 等[63] 针对边缘数据的完整性问题, 采用分布式共识机制形成自管理的边缘缓存系统. 在该系统中, 边缘服务器协作地确保缓存副本的完整性并修复受损的副本.

3 边缘智能协同

前文介绍的模型压缩等技术虽然有助于降低模型部署对内存和算力的需求, 但仍不适用于资源极端受限的设备, 且无法满足参数量庞大的高精度模型部署时对时延、能耗等的要求. 协同计算作为一种创新的解决方案, 通过网络连接多个节点 (端设备、边缘服务器或云数据中心) 协同工作, 可以实现计算能力的扩展与并行化, 以及数据资源的共享[64], 从而满足边缘智能中计算密集型任务的实时执行需求. 本节将介绍边缘智能协同相关研究, 首先对广泛采用的协同计算架构进行归类总结, 然后介绍以计算卸载和资源分配为主的协同计算关键技术, 最后介绍边缘协同训练与推理的相关工作.

3.1 协同计算架构

3.1.1 端-边架构

端-边架构是端侧设备与边缘服务器之间的协同计算架构. 如图 3 所示, 端侧设备包括各种终端设备, 例如智能手机、智能汽车等, 具有一定的计算和存储能力; 边缘服务器则通常位于靠近端侧用户设备的网络边缘, 具有更强大的计算和存储能力[65]. 通过将移动应用程序的部分或全部数据处理任务从资源有限的终端设备迁移到边缘服务器执行, 可有效减轻终端设备的计算负载, 提高其用户服务响应速度[66].

图 3 端-边架构

研究者对不同应用场景下的端-边协同计算展开研究. 文献 [67] 研究了一种无人机辅助的移动边缘计算网络, 在用户根据时变概率生成任务的动态环境中, 最大限度地降低系统的计算成本. 文献 [68] 在支持 5G 的车载网络场景下, 考虑车辆用户激励兼容性和个体合理性, 提出了一种高效的局部计算卸载和自适应任务调度算法. 同样是研究车载网络, 文献 [69] 提出了一种基于边缘计算的车辆网络节能协同卸载

方案,该方案将任务分解为多个子任务,并将其卸载到位于车辆路线前方的不同路边单元.

端-边架构将数据处理和存储任务分布在网络的边缘和终端设备上,将其推向离数据源更近的地方执行,满足了任务对实时性、带宽和隐私保护的需求,尤其适用于需要快速响应和数据处理的场景. 此外,敏感数据可以在本地处理,无需传输到云端,从而降低了数据泄露的风险. 例如,端边架构可以为物联网中大量传感器提供海量数据的实时处理和反馈,为工业自动化生产线上的设备提供快速响应和决策.

3.1.2　端-边-云架构

尽管端-边协同展现出不错的能力,但是也不能忽视云中心所拥有的巨大资源. 随着智能移动设备数量的不断增加及其对资源需求的日益增长,仅依赖边缘层资源来满足智能设备的业务需求将变得越来越具有挑战性[70]. 因此,充分结合边缘计算和云计算,使两者互补,端-边-云协同计算应运而生. 如图4所示,端-边-云架构可以在纵向和横向进行协同.

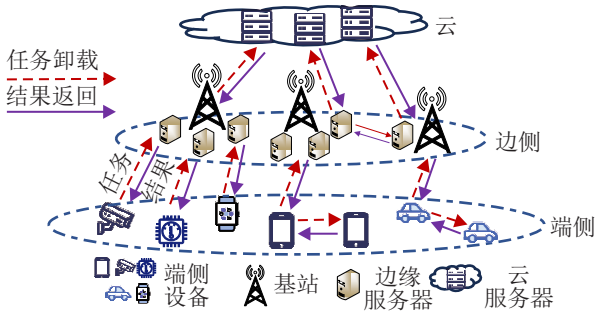

图 4　端-边-云架构

纵向协同是指端边云不同层次间的协同,旨在充分利用端边云不同层次间设备的资源和特征来满足不同的应用需求. 文献[71]通过在相邻边缘服务器或远程云之间划分任务,全面研究了边缘服务器与集中式云之间的协作. 文献[72]研究了端-边-云系统中的多用户任务卸载问题,用户任务既可以在用户设备上本地处理,也可以通过无线通道卸载到边缘服务器或卸载到云端处理. 文献[73]提出基于蜂群劳动分工“激发-抑制”模型的边云协同任务调度算法,实现了高效的边云协同任务调度. 与上述研究场景不同,郑莹莹等[74]针对端-边-云架构的车路协同系统,设计了一种基于多智能体强化学习的资源调度算法,通过任务卸载和计算资源分配,实现可靠性约束下的系统时延和能耗优化.

横向协同是指在端边云架构中同一层次的多方之间进行数据交互. 为满足多样的应用需求,需要多方之间进行数据共享,在端边云架构下,数据主要在边缘层进行汇集,而边缘设备是地理上分布式部署的,因此需要边缘节点之间进行数据共享,满足多样的应用需求[75]. 文献[76]提出了一种新的任务卸载架构Flex-MEC,实现了边缘服务器间高效的任务分配和调度. 与物理邻居数量有限的单跳情况相比,文献[77]关注的是多跳的情况,它可以将任务卸载到位于1跳邻居之外的更强大的边缘云上,更好地探索协同计算能力. He等[78]研究了协作边缘服务器之间的任务调度问题,提出了一种在最差延迟要求和长期能耗约束下系统效用最大化的在线算法.

端-边-云架构作为端-边架构的演进,不仅继承了端边架构的低延迟和隐私保护特性,还融入了云计算的计算和存储优势. 在这一架构中,终端设备主要承担数据收集和执行简单计算的任务,而边缘设备则充当中间层,负责处理任务或将预处理数据转发至云端进行更复杂地分析和长期存储. 端边云架构在车联网、智慧城市等多个应用场景中显示出其灵活性和适应性. 然而,这种多样化的处理策略也使得任务卸载调度和资源分配变得更加复杂. 设计者必须综合考虑数据的产生位置、处理需求、网络环境以及隐私安全需求,在终端、边缘和云端之间作出合理的任务卸载调度以及计算、存储和带宽资源分配,以实现成本效益和系统性能的最优化.

3.2　协同计算技术

3.2.1　计算卸载

计算卸载是指将部分或全部计算任务从端侧设备传输到其他可用计算节点(如其他端侧设备、边缘服务器、云服务器)进行计算的过程. 计算卸载通常有两种分类方式. 一种方式是按照卸载目的地进行分类,可以分为设备到云服务器(device to cloud, D2C)卸载、设备到边缘服务器(device to edge, D2E)卸载、设备到设备(device to device, D2D)卸载以及混合卸载(hybrid)四种模式.

1) D2C卸载. D2C卸载是指将计算密集型任务从端侧设备转移到云服务器执行,从而减轻端设备的计算负载和突破其存储限制. 文献[79]研究了D2C卸载中任务最大完工时间和云服务器卸载成本的加权和的最小化问题. Mahmoodi等[80]联合考虑了移动应用组件的无线感知调度与卸载策略,用线性规划解决了带有任意依赖关系图的移动应用的云卸载问题. 文献[81]提出了一种基于Lyapunov优化的节能卸载决策算法,用于确定哪些应用程序组件在端设备执行,哪些组件在云服务器处理.

2) D2E卸载. 相比于D2C卸载,将计算任务传输

到边缘服务器上执行可以减少网络带宽消耗,提升服务的响应速度. 例如: Zhan等[82]研究车辆边缘计算场景中的任务卸载调度问题,设计了一种基于深度强化学习的卸载调度方法以优化任务时延和能耗成本. Luo等[83]建立一个具有通信和计算能力的车辆边缘计算卸载框架,提出了一种基于粒子群优化的计算卸载算法,以实现延迟和成本的双目标优化. 文献[84]提出了一种基于无模型深度强化学习的分布式算法,使得每个设备可以在不知道其他设备的卸载决策的情况下确定自己的卸载方案.

3) D2D卸载. D2D卸载可以充分利用边缘网络中闲置的端侧设备资源,将工作负载从一端设备转移到其他端设备,减少计算延迟和能量成本. Peng等[85]利用移动终端之间的协作通信能力,提出一种通用的D2D底层框架,实现了应用部分卸载、协同调度和计算分配的动态结合. 文献[86]开发了一种基于深度强化学习的计算卸载方案,其中车辆可以将部分计算密集型任务卸载给相邻车辆. Ko等[87]设计了一种分布式D2D卸载系统,确保已卸载的任务能在预先指定的截止日期内完成.

4) Hybrid卸载. Hybrid卸载结合了上述3种不同的卸载模式,可以根据任务的特性和需求,灵活选择合适的卸载目的地. 该方法可以充分利用服务器和众多端设备的资源,缓解高负载设备的压力,提高任务执行效率和用户体验. 例如: Yu等[88]构建了一种新的考虑社会关系的混合计算卸载系统模型,将D2E卸载与D2D卸载相结合. Liu等[89]提出了一个针对多个移动应用程序的依赖任务卸载框架,其中移动设备可以将具有依赖约束的计算密集型任务自适应地分配给边缘服务器和云服务器,以改善用户体验.

在实际应用中,卸载目的地的选择取决于具体的需求,包括任务的计算复杂性、实时性要求、网络带宽、能耗限制等因素. D2C卸载凭借强大的云端计算能力可以处理大规模和更复杂的任务,然而它对网络带宽的需求较大,尤其是对于图像、视频处理等数据传输量较大的任务,远距离传输产生的高延迟不可忽视. 因此,D2C卸载通常适合对计算能力要求极高的同时对时延有较高容忍度的任务. D2E卸载中的边缘服务器位于网络的边缘,更靠近数据源,可以避免数据远距离传输而产生的高延迟,适合对实时性要求高的任务,如自动驾驶、工业自动化控制等. 但是边缘服务器通常计算和存储资源有限,且服务的连续性易受端侧设备位置移动的影响,因此D2E模式中需要着重考虑边缘服务器放置、资源分配、服务迁移等问题. D2D卸载允许设备之间直接通信,可以不通过中心化的服务器或云端进行数据交换和计算卸载. 这种模式能够充分利用邻近设备的计算资源,但设备之间的通信可能受到距离和兼容性的限制,同时安全性和数据隐私保护也变得更加复杂. 因此,D2D模式通常适合在高密度设备环境中的局部数据处理,如紧急情况下的快速信息传播,不适用于大规模数据处理和移动性高的设备. Hybrid卸载结合了D2C、D2E和D2D卸载的特点,灵活性高,可以根据任务需求和网络条件动态选择最合适的卸载策略. 但环境的高度动态变化,使计算卸载和资源分配策略的制定更具挑战,其需要更复杂的网络管理和协调机制、精确的上下文感知和预测能力等.

另一种是按照任务的卸载模式进行分类,可以分为二进制卸载和部分卸载两种模式.

1) 二进制卸载. 即“0-1卸载”,是指整个计算任务或者在本地处理,或者卸载至其他计算节点处理,“0”和“1”表示任务是否被卸载. 例如: Huang等[90]考虑一个采用二进制卸载策略的无线供电移动边缘计算网络,提出了一种基于深度强化学习的在线卸载框架,使得无线设备的每个计算任务或者在本地执行,或者整体卸载至边缘服务器执行. 文献[91]在车辆边缘计算网络中构建数据调度模型,其中每个计算任务可以在本地处理,或者迁移到协作车辆,或者卸载到沿路部署支持边缘计算的路边单元中.

2) 部分卸载. 是指将计算任务划分为多个子任务,并将其中部分子任务卸载至远程处理. 在拆分任务时,需注意任务数据是否相互独立. 如果相互独立,则可以任意拆分; 如果任务数据之间存在依赖关系,则拆分后子任务的执行顺序必须满足顺序要求. 例如,Hu等[92]考虑到作业中任务之间的依赖关系,提出了一种基于贪心策略的作业卸载调度算法,以最小化作业的总延迟,同时最大化队列的资源利用率. 为了使移动车辆的任务总延迟和等待时间最小,文献[93]建立将多移动车辆的任务划分为连续子任务的动态卸载模型,提出了一种基于双深度Q网络的动态分帧卸载算法. 文献[94]提出了一种面向DNN的层级计算分区策略,将每个移动设备的任务划分为在本地计算或卸载到服务器执行的子任务.

两种卸载模式的选取通常取决于任务的属性,二进制卸载方式适用于任务不可分割或者任务分割带来的开销大于其收益的情况. 这种卸载模式不涉及任务的分割和部分执行的问题,因此在建模和实际部署中决策过程相对简单. 例如,对于基于DNN的图像

识别任务,远程执行时,只需将需要输入的原始图像数据全部传输到远程服务器上,然后将识别结果传回本地. 但这种卸载模式缺乏灵活性,可能无法实现计算资源的最优化利用,而且传输图像、视频等原始数据时通常会因数据量较大导致高延迟. 部分卸载模式通常更加灵活,可以根据任务和系统状态选取分割方案,将任务传输到不同计算节点上并行处理,优化资源的利用效率. 部分卸载需要考虑任务之间的依赖关系,包括顺序依赖、并行依赖或更复杂的通用依赖. 在能量受限的移动设备上,部分卸载可能更有利于节省能耗,因为它允许设备只执行那些对性能要求不高的部分任务. 同时,部分卸载模式下需要精确的计算性能和网络状况预测机制、任务分割和调度策略,这也会增加计算和实现的复杂性. 设计卸载算法时,还需要考虑卸载的总体性能,包括能耗、时延、任务完成率以及系统稳定性等指标.

3.2.2 资源分配

边缘网络中的资源可以分为3类: 通信资源、计算资源和存储资源[95]. 通信资源一般指用于数据传输的网络带宽、发射功率等. 计算资源通常指端侧设备、边缘服务器和云服务器的处理器资源. 存储资源指具有缓存功能的硬件设备,通过将计算任务和热门内容缓存到网络边缘提高用户体验并且减轻网络负担. 资源分配是指根据计算任务特性和系统状态,将通信、计算、存储资源进行合理分配的过程[96]. 通过阅读大量相关文献[86,97-107],发现只研究单一资源分配的很少,大多数研究考虑资源的联合分配. 下面将详细进行分类说明.

1) 单一资源分配. 现有的单一资源分配研究主要集中在通信和计算资源上. 对于通信资源分配,文献[97]提出了基于深度强化学习的网络资源分配框架以最大化通信链路的传输成功率. Chen等[98]考虑到物联网节点发射功率的有限性、可再生能量的不确定性和无线信道的动态性,提出了一种基于李雅普诺夫优化理论的在线随机时间和发射功率分配算法,以最大化物联网节点的长期总吞吐量. 对于计算资源分配,文献[86]提出了一种在线卸载框架使移动应用程序能够同时将其相关任务卸载到边缘云系统,并通过动态探索边缘服务器与远程云之间的任务负载,在线作出高效的分配决策. Yang等[99]将单个边缘服务器计算资源分配问题表述为一个回归问题,设计并训练了基于多任务学习的前馈神经网络模型,同时获取卸载决策和计算资源分配方案. 对于存储资源分配,文献[100]考虑内容传递和缓存替换,设计了基于分布式多智能体强化学习的缓存策略,以最小化系统长期内容访问成本.

上述工作只考虑一种类型的资源分配,如仅考虑计算资源或存储资源,易于实现,适用于资源需求相对单一或者某种资源特别紧张的场景. 例如,边缘服务器被用于处理来自工业传感器的实时数据分析任务时,可以采用单一资源分配,优化特定计算资源的使用. 边缘计算场景中通常涉及到计算、存储、网络等多种资源,因此更多研究专注于联合资源分配.

2) 联合资源分配. 联合资源分配即在更复杂的应用场景中同时考虑多种类型的资源分配. 文献[101]提供了一种用于边缘云协同计算的在线任务卸载和计算通信资源分配方法,旨在最小化任务的平均延迟. 文献[102]针对边缘计算系统中的计算卸载问题,构建了设备与运营商之间互动的博弈理论模型,以实现成本最小化或时间公平的通信计算资源分配策略. 文献[103]将任务分配和功率分配共同表述为一个混合整数非线性规划问题,并提出了一种启发式的动态感知任务调度方法,以获得低复杂度的有效资源分配. 为了找到使任务延迟和能耗最小的内容缓存和并行任务卸载策略,Xiao等[105]设计了一种改进的二元粒子群优化算法来解决存储资源分配问题. 文献[91]在车辆边缘计算场景下,创新出一种增强型深度Q网络算法,用以解决联合通信、计算和缓存资源的数据调度问题. 针对边缘云协同的多任务多服务场景,Fan等[107]提出了一种包含服务放置、任务调度、计算资源和传输速率分配的方案,保证长期任务队列稳定性的同时最小化总任务处理延迟.

联合资源分配考虑了不同资源之间的相互作用,更可能找到全局最优解,但也使得问题规模、解空间更大,算法设计和实现难度增加. 例如,在车联网应用中,需要同时考虑计算、网络和存储资源,以确保车辆产生的大量数据能够及时处理和分析,提高数据处理速度和准确性,从而提升整体系统性能. 但联合资源分配需要更复杂的动态优化算法,如深度强化学习来实现多种资源的动态分配和调度. 而目前的工作对资源分配的数学建模较为抽象,在实际部署时,可能无法准确估计在特定数量资源下(如CPU、带宽等)任务的时延、能耗等指标,导致数学求解出的最优解在实际中无法取得最好的效果. 因此,在联合资源分配的复杂系统中,还需探究如何对资源分配过程进行更加真实地建模.

部分文献在卸载模式和资源分配方面的对比如表2所示.

表2 计算卸载与资源分配

分类	文献	卸载模式	资源分配		
			通信资源	计算资源	存储资源
D2C	[79]	0-1卸载	–	√	–
	[81]	部分卸载	–	√	–
D2E	[82, 84]	0-1卸载	–	√	–
	[83]	0-1卸载	√	√	–
	[90]	0-1卸载	√	–	–
	[92, 94]	部分卸载	√	√	–
	[93]	部分卸载	–	√	–
D2D	[85]	部分卸载	√	√	–
	[86]	部分卸载	–	√	–
	[87]	0-1卸载	–	√	–
	[91]	0-1卸载	√	√	√
hybrid	[105]	0-1卸载	–	√	√
	[88]	部分卸载	–	√	–
	[89]	0-1卸载	–	√	–

3.3 协同训练与推理

训练与推理是边缘智能协同的两个重要组成部分. 边缘协同训练是指在多个端侧设备、边缘服务器或云服务器的共同协作下,进行机器学习或深度学习模型的训练. 边缘协同推理是将端侧产生的部分推理任务通过网络传输分配到其他可用计算节点进行处理,从而满足端侧资源受限设备对推理时延、精度、能耗等方面的要求.

3.3.1 边缘协同训练

与在单个端侧设备上进行模型训练相比,边缘协同训练可以充分利用多个设备的算力资源,实现更高效的训练. 与传统的云中心集中式模型训练相比,边缘协同训练在网络边缘处进行,无需将所有原始底层数据传输到云端,降低了隐私泄露风险. 本节主要介绍边缘协同训练中的联邦学习和持续学习相关工作.

1) 联邦学习. 联邦学习(federated learning, FL)利用分布式数据和计算资源,在不集中端侧设备私有原始数据的情况下训练全局模型,确保数据安全和隐私,其中边缘服务器负责聚合本地模型、更新并传播全局模型[108-109]. 本节主要关注FL中优化设备通信效率和能耗以及解决数据异构问题方面的工作.

在优化设备通信效率方面,早期的经典工作FedAvg[110]采用周期性模型平均方式. 由于每次迭代中同步数据的大小与模型的大小几乎相同,引入了大量的通信开销,导致通信效率低下. 针对该挑战,Zhou等[111]设计了Overlap-FedAvg框架,将模型训练阶段与模型上传和下载阶段并行操作,从而使后者完全被前者覆盖,提高通信效率. Zhang等[112]提出了一种新型的合作移动边缘网络FL框架CE-FedAvg,其中边缘服务器协调区域内部设备训练,并且边缘服务器之间通过去中心化学习方式共享模型参数,从而实现延迟和准确度的平衡. Wu等[113]提出一种FL通信框架FedComp,通过利用张量级的索引共享机制、细粒度的参数打包策略和残差压缩器,在降低内存成本的同时实现了显著的通信压缩比.

在优化设备能耗方面,Xu等[114]针对电池供电的端侧设备提出了一种FL算法,该算法通过拍卖策略选择参与训练的设备并允许设备在低功耗模式下进行训练. Li等[115]考虑到参与FL的设备之间的异构环境,提出了一种压缩学习算法,通过调整梯度压缩参数来平衡设备在本地计算和通信上的能量消耗. Chen等[116]将FL训练中的能量最小化问题形式化为一个混合整数非线性问题,通过共同确定每个移动设备的带宽分配和权重量化级别来优化训练中的计算和通信能耗.

在解决数据异构方面,数据的非独立同分布问题可能导致模型收敛不稳定和预测准确性下降[117]. Tan等[118]将抽象的样本类别作为原型,在客户端与服务器之间传输原型而非梯度,但这种方法面临着隐私安全和性能的权衡问题. Yang等[119]提出了一种联邦特征蒸馏方法,将数据分成对模型性能贡献大的性能敏感特征数据和对模型性能贡献有限的性能稳健特征数据,在全局范围内共享性能敏感特征以缓解数据异质性,本地则保留性能稳健特征.

联邦学习作为一种分布式机器学习框架,允许在不泄露数据的前提下,利用分散的数据进行模型训练,因此受到了广泛关注与研究. 尽管FL在数据隐私和分布式计算方面展现出巨大潜力,但面向未来大规模异构边缘设备及多样化智能应用时仍存在一些问题. 例如,异构设备、异构数据分布等特点可能导致模型聚合后的性能下降,而目前常用的聚合算法普遍缺乏理论保障. 客户端的网络连接可能存在带宽限制、连接不稳定等问题,严重时会中断模型的上传与下载过程,需要高可靠的网络通信和容错机制来进一步保证模型聚合的稳定性. 此外,目前大多数研究侧重于训练优化,忽略了FL参与者的参与度和公平性问题或建立的模型较为粗糙. 如何鼓励数据持有者积极参与FL,并确保其贡献与所获奖励之间的公平分配,是一个亟待解决的现实问题.

2) 持续学习. 不同于联邦学习,持续学习旨在使模型能够连续地从新任务或数据流中学习,并保留先前学到的知识,避免灾难性遗忘,以适应环境动态变化. 这是因为真实环境中的数据与训练数据在特征上不完全一致,导致特定数据集上预训练的模型在

现实世界应用时往往无法达到预期性能,从而需要实施持续学习,即利用新采集数据对模型进行迭代训练. 本节主要关注多节点协同进行持续学习的相关工作. Kong 等[120] 针对边缘辅助实时视频分析系统设计了一种改善端侧摄像头推理精度的方法,该方法中,摄像头选取最能代表当前环境的帧,将其传输至边缘服务器上作为模型更新的重新训练数据,一旦控制器检测到所选帧的准确性显著下降,边缘服务器将会重新训练模型,并将更新后的参数下发到端侧摄像头,然而这种直接传输模型参数的方式会增加带宽消耗. Wang 等[121] 解耦了数据样本的标注和模型训练过程,将标注过程卸载到云服务器,而在边缘设备上进行模型训练,以协同完成推理模型的更新任务. AdaEvo 框架[122] 通过平衡边缘服务器上的计算资源、内存资源以及不同移动用户发起的异步任务之间的竞争关系,使得资源有限的边缘服务器可以处理来自多个移动端的DNN更新任务.

部署在端侧的人工智能应用模型通常需要灵活高效的更新,因此持续学习展现出了巨大的潜力和广阔的研究前景. 除视觉领域外,持续学习的应用扩展到自然语言处理和强化学习等其他领域,人工智能模型的快速发展拓宽了持续学习的理解和应用的同时,也带来了新的挑战. 例如,多模态模型的广泛应用促使研究人员需要考虑持续学习的可塑性和跨任务的普适性. 随着模型参数量的急剧增加,持续学习算法需要在资源消耗(如计算和存储)和学习效率之间找到平衡,其中用于衡量持续学习算法性能的评估指标和基准尤为重要. 此外,GPT[123] 等的基础模型,通过在海量数据上的广泛预训练,在多种下游任务上展现了卓越的性能. 然而,这些预训练数据不仅体量庞大,而且通常是逐步累积的,这就要求模型必须具备高效的更新机制. 另一方面,预训练模型规模的持续增长对于知识的传递具有积极作用,有助于在后续的持续学习过程中减少因学习新任务而导致的对先前知识的灾难性遗忘问题.

边缘协同训练相关工作对比如表3所示.

表 3 边缘协同训练

研究范畴	文献	主要优化目标	方法	基准	性能
联邦学习	[111]	优化通信效率	模型本地训练与通信阶段并行	Transformer, Wikitext-2	↓34%每轮时间
	[112]	优化训练时间	多个端设备与多个边缘服务器构成双层协作模型训练架构	FedAvg, FEMNIST	↓62.5%训练时间
	[113]	优化通信效率	压缩模型训练参数	DeepReduce[124]	28.5× 通信压缩比
	[114]	降低能耗	选择参与训练设备、低功耗处理模式	LeNet	↓20%能耗
	[115]	降低能耗	模型梯度压缩	ResNet20, CIFAR-10	↓1.5×(−100)× 能耗
	[116]	降低能耗	为每个设备选择不同的权重量化级别	ResNet34,FlexibleSpar[115]	↓28% 能耗
	[118]	解决数据异构	客户端和服务器之间只传输样本类别	MNIST, 异构标准差为2	97.13%精度
	[119]	解决数据异构	设备间部分数据共享	CIFAR-100, MNIST	加速收敛
持续学习	[120]	优化准确度和时间	更新模型,选取最优epoch数、帧数和教师模型的配置	无再训练策略	↑24%精度,↓50%再训练时间
	[121]	优化准确度	在线知识蒸馏,标注过程卸载到云端	仅边缘处理	↑15%～20% mAP
	[122]	优化移动端平均QoE	视频帧采样来控制重新训练数据的大小	默认GPU调度	↑32% QoE

3.3.2 边缘协同推理

在单个设备上实现加速推理的方法(例如模型压缩)常伴随精度损失问题,与此不同,边缘协同推理方法得益于边缘服务器的算力支持,能够在不牺牲推理精度的前提下进行模型运算. 根据协同计算模式,将相关工作主要分为两类:模型分割协同推理[125-129]和任务卸载协同推理[130-135].

1) 模型分割协同推理. 模型分割协同推理是将DNN进行分割,并部署到不同设备和计算节点上,仅传输少量中间结果. 得益于神经网络易于分割的特点,每个推理任务可以划分为两个甚至多个子任务执行. 此外某些中间层的数据明显小于原始输入数据的大小,传输时可以减少带宽消耗,降低传输时延. 对DNN不同层推理性能的准确评估是实现模型高效分割的前提. Zhang 等[125] 将模型按照推理时的独立执行单元进行分割,形成一组核心,并分别预测这些核心的推理延迟. 但是这种需要剖析模型内部结构的侵入式方法需要昂贵的计算资源和开源代码,不适用于无法容忍额外开销的云边协同推理场景. Liu 等[126] 开发了一种非侵入式性能表征网络,基于神经网络的相似度预测DNN的推理时间. Wu 等[127] 考虑了边缘服务器上有服务于多个设备的多种DNN模型共享资源的情况,通过构建DNN性能预测模型,计算最优的DNN分割点和计算资源分配,最大化系

统吞吐量. 这种针对链式结构网络模型的划分方法并不适用于非链式结构的DNN模型. Yang等[128]将非链式结构的网络模型建模成有向无环图(directed acyclic graphs, DAG), 根据每个设备的计算能力动态划分网络层并映射到多个设备上. 输入数据被送入第1个设备, 推理结果在最后一个设备产生, 从而组成了一个推理管道, 吞吐量可以提高1.8～6.8倍. Wu等[129]针对服务器处理大量多样的不对齐DNN片段会导致资源效率差的问题, 通过计算服务器上所有片段的最优重新分割点、批处理大小、每个实例的GPU分配以及实例数量来最小化服务器上GPU资源消耗.

2) 任务卸载协同推理. 与模型分割协同推理中多个计算节点共同完成一个推理任务不同, 任务卸载协同推理方法通常将推理任务完整卸载到其他计算节点或分享数据信息, 这些节点的计算过程是互相解耦的. Hanyao等[130]针对同时具备目标跟踪和边缘辅助分析功能的系统, 通过决定边缘辅助推理的频率来最大化目标检测的整体准确性. Ran等[131]提出了一个分布式基础设施DeepDecision, 将计算能力较弱的前端设备与更强大的后端辅助设备相结合协同进行任务推理. 通过动态选择帧率、卸载决策、视频压缩、分辨率及采用的DNN模型规模来优化前端设备上增强现实应用的准确性、能耗和时延. Galanopoulos等[132]实验发现在边缘服务器辅助移动设备进行视频分析时, 图像中物体差异、无线信道的变化都会影响到系统的准确性和延迟, 因此通过确定图像编码率、服务时间分配和神经网络输入层大小来最大化系统的识别准确性. Luo等[133]在边缘辅助多车辆感知系统中, 让边缘服务器与车辆共享原始传感器数据, 形成一个更高分辨率的整体视图, 从而增强车辆上传感器感知的稳健性并扩大感知范围. Zhang等[134]为了适应动态的视频内容和网络条件, 引入一种轻量级算法来为视频处理任务分配最合适的本地或远程模型, 以实现低延迟和高准确性.

4 边缘智能应用

随着物联网技术的高速发展, 万物互联成为现实, 并产生海量数据. 除了大规模的云数据中心外, 更多的数据是由地理分布广泛的移动设备产生, 为边缘计算提供了丰富的应用场景和平台[8], 促使AI算法从云端迁移到边缘[108]. 与此同时, 越来越多的企业加大力度研发具有计算加速能力的AI芯片(如FPGA、GPU、TPU、NPU等), 使其与移动边缘设备逐渐融合, 为边缘智能的发展注入了生机和活力[136]. 从上述讨论可以看出, 边缘智能在缓解云数据中心压力、将智能服务下沉到边缘侧有着独特优势. 下面将简要介绍边缘智能在工业物联网、智慧城市和虚拟现实3个领域的应用.

4.1 工业物联网

工业物联网(industrial internet of things, IIoT)是指将各类传感器或控制器, 借助通信技术融入到工业生产、应用等各个环节领域, 实现对作业的实时监控. 边缘智能的融入则进一步提高了工业环境的智能化水平, 通过在传感器和控制器附近进行数据处理、控制指令下发, 实现对关键操作的迅速响应, 确保工业过程的实时性和同步性. 边缘智能融入工业物联网的优势如下:

1) 提升决策效率. 边缘推理使工业数据的智能处理和分析可以在数据产生点即时完成, 如车间、流水线等, 并根据分析结果实时作出反馈控制决策.

2) 增加系统鲁棒性. 分布式的边缘智能计算可以避免单点故障, 并且在网络连接不稳定或中断的情况下, 边缘节点仍可以继续进行局部作和处理.

3) 加强数据安全和隐私保护. 数据在本地处理可以减少敏感数据的传输, 实施数据加密和访问控制, 从而降低数据泄露或被截获的风险.

在智能制造应用方面, 边缘智能可实现生产线的实时监控和数据分析, 及时发现异常情况并发出预警信号, 避免生产事故的发生[137].

边缘智能还可实现设备的预测性维护和自动化控制, 降低维修成本和停机时间. 将智能网关用于执行本地数据的采集与处理, 包括数据过滤和清洗, 实现实时操作[138]. 此外, 边缘智能还支持跨层协议转换, 实现工业网络的统一接入和管理. 为了利用边缘计算在这方面的优势, 一些工厂开始采用软件定义的方法来实现机械和控制的分离, 包括在边缘节点上运行虚拟化的工业控制器, 便于集中控制生产线上的机械臂, 展现了边缘智能在工业自动化中的广泛应用前景[139]. 在智能物流应用方面, 边缘智能可实现货物的实时追踪和优化调度, 提高物流效率. 例如, 通过部署在货车上的传感器, 边缘智能设备可以监测货物的位置、温度、湿度等信息, 为货物提供安全保障和最优路线建议. 边缘智能还可实现仓库的智能管理, 通过机器视觉和机器人技术, 实现货物的自动识别、分类和搬运[140]. 在智能电网应用方面, 边缘智能可实现能量的实时监测和优化分配, 提高能量效率. 例如, 通过部署在电网上的传感器, 边缘智能设备可以监测电网的负荷、电压、频率等信息, 为电网提供稳定保障和

需求响应建议[141]. 边缘智能还可实现分布式能源的智能管理,通过协同计算和机器学习技术,实现可再生能量的预测、调度和优化. 边缘智能在保护用户隐私方面有独特优势,例如,Hudson等[142]基于FL设计了一个边缘智能配电网框架,既降低了通信成本又保障了用户隐私.

4.2 智慧城市

智慧城市是基于信息和通信技术的集成和应用,旨在优化城市管理、提升公共服务效率、增强居民生活质量和推动可持续发展. 智慧城市的建设依靠单一的集中式云计算模型已无法有效应对所有计算需求,亟需多种计算模式融合. 通过将计算、数据和服务从集中式的云数据中心转移到分布式的边缘服务器[143],边缘智能可满足用户对于移动网络高带宽、低时延的要求,同时使得低成本、小型的传感器海量连接成为可能,为大规模的城市数据决策和治理提供基础保障[139]. 边缘智能在智慧城市领域的应用主要如下:

1) 在智慧交通方面,包括智能调度[144]、自动驾驶系统[145]、交通设施改造升级[146]、交通数据采集与分析[147]等. 例如,通过在交通灯、监控摄像头等关键位置安装边缘计算节点,城市管理者可实现对路况、车辆、行人等的实时监测和控制,优化交通信号灯控制策略,减轻拥堵,提高交通安全和效率[147].

2) 在智慧医疗方面,边缘智能可应用于医疗数据的采集、分析和共享,医疗设备的智能控制和优化,医疗服务的智能辅助和推荐,医疗知识的智能获取和推理等方面[148]. 例如,边缘智能可以实现对医疗设备和患者的数据进行实时采集和分析,实现远程诊断、治疗、监护等,提高医疗服务的效率和质量[148]. 边缘智能也可以对医疗数据进行智能加密和共享,保护数据的隐私安全,促进数据的价值发挥[149]. 边缘智能还可以对医疗知识进行智能获取和推理,提供医疗决策的智能辅助和推荐,提升医疗水平和效果[150].

3) 智慧城市离不开智能家居,边缘智能使得设备能够在本地处理和响应语音指令,提供更快速的服务,同时保护住户隐私[151]. 例如,Nour等[152]提出一种利用联邦边缘学习的方法保护住户的数据隐私,并通过节点筛选和优化来提高训练效果并降低家居网络的通信开销. Li等[151]在智能家居系统中引入容器化技术,用于部署和管理智能家居中各种设备上的深度学习模型. 根据IDC预测,预计到2024年,近2%的智能家居设备将配备基站功能;到2025年,边缘智能技术在智能家居领域的渗透率有望接近50%[153].

边缘智能的技术促进智慧城市建设的过程中,将面临着数据隐私保护、网络稳定性保障等多种挑战. 例如,智慧城市时刻产生大量隐私数据并将其传输到联网设备进行处理,如何确保数据在边缘设备上的安全,防止数据泄露、同时符合隐私保护法规是一大难题. 此外,终端设备与边缘设备、边缘设备与边缘设备之间需频繁地通信,势必占用大量的网络带宽资源. 自动驾驶等服务通常具有强实时性约束,对网络要求极高,因此如何实现不同计算服务之间网络资源的动态合理分配,同时保障网络的稳定性是一大挑战.

4.3 虚拟现实

虚拟现实(virtual reality, VR)和增强现实(augment reality, AR)技术的发展为用户与虚拟世界的交互提供了新模式. 与传统媒体应用不同,移动AR/VR应用需要持续传输高分辨率的视频,因此对数据计算能力、网络稳定性和响应速度有很高的要求[154]. 按照当前的数据增长速度,云中心的计算能力越来越难以满足这些要求[155]. 然而,将所有数据上传到云端会导致严重的网络拥塞. 同时由于网络带宽有限,大量物联网设备产生的数据会给网络带宽带来很大的压力,导致云计算无法满足这些场景中的低延迟和高响应速度要求[6,140]. 边缘智能有助于解决上述问题,通过在离用户更近的边缘服务器上处理数据,缩短了数据在网络中的传输距离,从而显著减少了延迟. 这对于需要快速反应的VR应用(如虚拟手术模拟、高速运动游戏等)尤其重要. 例如,Pan等[154]利用混合整数非线性规划方法解决了移动设备视频流的卸载问题,在识别精度的约束下能够显著降低VR系统的延迟和设备的能耗. He等[156]研究了移动增强现实系统中时延和能量感知的通信和计算资源联合优化问题,以支持AR应用. Wang等[157]提出了一种高效在线调度算法,可以在考虑能耗、延迟、分析准确性的同时,根据网络状况和视频内容为多个视频流选择合适的带宽配置. 利用边缘智能技术,谷歌眼镜能够预测用户的注视区域从而做到实时选择360°视频或图像的关键部分,以减少帧率的下降,同时降低延迟,提高用户体验,因此得以广泛应用[158].

VR和AR技术的发展革新了人类与虚拟世界的交互方式,而边缘智能的加入进一步加速了这些技术的落地应用. 然而,边缘智能在虚拟现实领域的应用仍面临一些挑战. 例如,如何准确建模用户对VR视频不同帧率、分辨率的感知水平,以便在计算或网络资源受限时,确定最优的资源分配和视频配置方案,

最优化用户的感知质量. 在多个边缘节点处理数据时,确保数据的同步和一致性也是一个挑战. 不一致的数据可能导致用户体验不连贯或产生错误的决策.

5 总结与展望

边缘智能融合了边缘计算和人工智能的优势,有望将计算密集型的智能算法部署在网络边缘,以实现在边缘设备上对海量数据的安全高效处理. 作为一种新兴计算范式,边缘智能引起了学术界和工业界研究人员的广泛关注. 本文系统性地回顾并探讨了近几年边缘智能的前沿技术与最新进展. 首先概述了边缘计算、人工智能和边缘智能的相关背景、基本原理与发展趋势;然后梳理了在单个设备上实现边缘智能的相关工作,并根据研究内容分为边缘训练、边缘推理和边缘缓存3个方面;接着介绍了协同计算的典型架构和关键技术,并梳理了多个设备合作实现边缘智能协同的相关工作,将其分为边缘协同训练和边缘协同推理;最后,总结了边缘智能在工业物联网、智慧城市和虚拟现实等领域的广泛应用及意义. 边缘智能通过在数据产生源附近进行数据处理和智能决策,可以减少延迟、保护隐私、降低带宽使用,并提升服务的可靠性. 近期,AI领域的大模型展现出了令人印象深刻的卓越推理能力,但其对能源和算力的巨大需求给数据中心带来了重大挑战,难以保障用户体验. 因此,对大模型端侧部署推理的需求正日益高涨,特别是在需要快速响应和数据隐私性高的应用场景中,边缘智能技术有望满足这一需求. 因此,随着AI算法、网络的发展和边缘算力的不断提升,以及与绿色能量采集等新型技术的融合,可以预见边缘智能正迅速成为实现可持续、高效和普及化AI应用的关键途径,为工业、医疗、教育和交通等多个领域创造更高的价值.

参考文献(References)

[1] IDC. IDC: 中国物联网市场支出逐步攀升,预计2027年将位列全球第一[EB/OL]. (2023-07-17) [2024-01-20]. https://www.idc.com/getdoc.jsp?containerId=prCHC51041723.

[2] GSMA. The mobile economy 2023[EB/OL]. (2023-03-27) [2024-01-20]. https://www.gsma.com/mobileeconomy/wp-content/uploads/2023/03/270223-The-Mobile-Economy-2023.

[3] Dai M H, Su Z, Li J L, et al. An energy-efficient edge offloading scheme for UAV-assisted internet of things[C]. IEEE 40th International Conference on Distributed Computing Systems. Singapore, 2020: 1293-1297.

[4] Tian S J, Chang C, Long S Q, et al. User preference-based hierarchical offloading for collaborative cloud-edge computing[J]. IEEE Transactions on Services Computing, 2023, 16(1): 684-697.

[5] Li Y C, Liang W F, Li J, et al. Energy-aware, device-to-device assisted federated learning in edge computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(7): 2138-2154.

[6] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J]. 计算机研究与发展, 2019, 56(1): 69-89.
(Shi W S, Zhang X Z, Wang Y F, et al. Edge computing: State-of-the-art and future directions[J]. Journal of Computer Research and Development, 2019, 56(1): 69-89.)

[7] IDC. IDC: 2023上半年中国边缘云市场逆势增长, 走出独立行情[EB/OL]. (2023-12-18) [2024-02-06]. https://www.idc.com/getdoc.jsp?containerId=prCHC51569923.

[8] Deng S G, Zhao H L, Fang W J, et al. Edge intelligence: The confluence of edge computing and artificial intelligence[J]. IEEE Internet of Things Journal, 2020, 7(8): 7457-7469.

[9] 施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型[J]. 计算机研究与发展, 2017, 54(5): 907-924.
(Shi W S, Sun H, Cao J, et al. Edge computing — An emerging computing model for the internet of everything era[J]. Journal of Computer Research and Development, 2017, 54(5): 907-924.)

[10] Qin X D, Xie Q M, Li B. Distributed threshold-based offloading for heterogeneous mobile edge computing[C]. IEEE 43rd International Conference on Distributed Computing Systems. Hong Kong, 2023: 202-213.

[11] Sarwar Murshed M G, Murphy C, Hou D Q, et al. Machine learning at the network edge: A survey[J]. ACM Computing Surveys, 2022, 54(8): 1-37.

[12] 张晓东, 张朝昆, 赵继军. 边缘智能研究进展[J]. 计算机研究与发展, 2023, 60(12): 2749-2764.
(Zhang X D, Zhang C K, Zhao J J. State-of-the-art survey on edge intelligence[J]. Journal of Computer Research and Development, 2023, 60(12): 2749-2764.)

[13] Zhou Z, Chen X, Li E, et al. Edge intelligence: Paving the last mile of artificial intelligence with edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1738-1762.

[14] Li T, Sun J Y, Liu Y L, et al. ESMO: Joint frame scheduling and model caching for edge video analytics[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(8): 2295-2310.

[15] Park J, Kwon H, Kim S, et al. QuiltNet: Efficient deep learning inference on multi-chip accelerators using model partitioning[C]. Proceedings of the 59th ACM/IEEE Design Automation Conference. San Francisco, 2022: 1159-1164.

[16] Kashikar P, Sentieys O, Sinha S. Lossless neural network model compression through exponent sharing[J]. IEEE Transactions on Very Large Scale Integration: VLSI Systems, 2023, 31(11): 1816-1825.

[17] Cheng L S, Wang J L, Li Y H. ViTrack: Efficient tracking

on the edge for commodity video surveillance systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(3): 723-735.

[18] Hua H C, Li Y T, Wang T H, et al. Edge computing with artificial intelligence: A machine learning perspective[J]. ACM Computing Surveys, 2023, 55(9): 1-35.

[19] Xia Q F, Ren W H, Xu Z C, et al. When edge caching meets a budget: Near optimal service delivery in multi-tiered edge clouds[J]. IEEE Transactions on Services Computing, 2022, 15(6): 3634-3648.

[20] Wang Z L, Hu Q, Li R N, et al. Incentive mechanism design for joint resource allocation in blockchain-based federated learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(5): 1536-1547.

[21] Hu N, Tian Z H, Du X J, et al. An energy-efficient in-network computing paradigm for 6G[J]. IEEE Transactions on Green Communications and Networking, 2021, 5(4): 1722-1733.

[22] Li B T, Wang H, Luo F J, et al. ACBN: Approximate calculated batch normalization for efficient DNN on-device training processor[J]. IEEE Transactions on Very Large Scale Integration: VLSI Systems, 2023, 31(6): 738-748.

[23] Jung Y, Kim H, Choi S, et al. Energy-efficient CNN personalized training by adaptive data reformation[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023, 42(1): 332-336.

[24] Hong Z Y, Yue C P. Efficient-grad: Efficient training deep convolutional neural networks on edge devices with gradient optimizations[J]. ACM Transactions on Embedded Computing Systems, 2022, 21(2): 1-24.

[25] Ren A, Zhang T Y, Ye S K, et al. ADMM-NN: An algorithm-hardware co-design framework of DNNs using alternating direction methods of multipliers[C]. Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems. New York, 2019: 925-938.

[26] Xiao G X, Lin J, Seznec M, et al. Smoothquant: Accurate and efficient post-training quantization for large language models[C]. Proceedings of the 40th International Conference on Machine Learning. Hawaii, 2023: 38087-38099.

[27] Ding Y F, Qin H T, Yan Q H, et al. Towards accurate post-training quantization for vision transformer[C]. Proceedings of the 30th ACM International Conference on Multimedia. New York, 2022: 5380-5388.

[28] Denil M, Shakibi B, Dinh L, et al. Predicting parameters in deep learning[C]. Proceedings of the Advances in Neural Information Processing Systems. Nevada, 2013: 2148-2156.

[29] Lebedev V, Ganin Y, Rakhuba M, et al. Speeding-up convolutional neural networks using fine-tuned CP-decomposition[C]. Proceedings of the 3rd International Conference on Learning Representations. San Diego, 2015: 1412.6553.

[30] Novikov A, Podoprikhin D, Osokin A, et al. Tensorizing neural networks[J]. Advances in Neural Information Processing Systems, 2015(1): 442-450.

[31] Lin S H, Ji R R, Chen C, et al. Holistic CNN compression via low-rank decomposition with knowledge transfer[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(12): 2889-2905.

[32] Lin S H, Ji R R, Chen C, et al. ESPACE: Accelerating convolutional neural networks via eliminating spatial and channel redundancy[C]. Proceedings of the AAAI Conference on Artificial Intelligence. San Francisco, 2017: 1424-1430.

[33] Liu Z C, Oguz B, Zhao C S. LLM-QAT: Data-free quantization aware training for large language models[J/OL]. 2023, arXiv: 2305.17888.

[34] Novkin R, Klemme F, Amrouch H. Approximation- and quantization-aware training for graph neural networks[J]. IEEE Transactions on Computers, 2024, 73(2): 599-612.

[35] Li Z K, Xiao J R, Yang L W, et al. RepQ-ViT: Scale reparameterization for post-training quantization of vision transformers[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. Paris, 2023: 17227-17236.

[36] Frantar E, Ashkboos S, Hoefler T, et al. GPTQ: Accurate post-training quantization for generative pre-trained transformers[C]. Proceedings of the 11th International Conference on Learning Representations. Kigali Rwanda, 2023.

[37] Touvron H, Lavril T, Izacard G, et al. Llama: Open and efficient foundation language models[J/OL]. 2023, arXiv: 2302.13971.

[38] Frantar E, Alistarh D. Sparsegpt: Massive language models can be accurately pruned in one-shot[C]. Proceedings of the 40th International Conference on Machine Learning. Hawaii, 2023: 10323-10337.

[39] Zhang S S, Stadie B. One-shot pruning of recurrent neural networks by jacobian spectrum evaluation[J/OL]. 2019, arXiv: 1912.00120.

[40] Lee N, Ajanthan T, Torr P. Snip: Single-shot network pruning based on connection sensitivity[J/OL]. 2019, arXiv: 1810.02340.

[41] Michel P, Levy O, Neubig G. Are sixteen heads really better than one?[C]. Proceedings of the Advances in Neural Information Processing Systems. Vancouver, 2019: 14014-14024.

[42] Ma X Y, Fang G F, Wang X C. LLM-Pruner: On the structural pruning of large language models[C]. Proceedings of the Advances in Neural Information Processing Systems. New Orleans, 2021: 2533-2552.

[43] Li W J, Wang J, Ren T T, et al. Learning accurate, speedy, lightweight CNNs via instance-specific multi-teacher knowledge distillation for distracted driver posture identification[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(10): 17922-17935.

[44] You S, Xu C, Xu C, et al. Learning from multiple teacher

networks[C]. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, 2017: 1285-1294.

[45] Son W, Na J, Choi J, et al. Densely guided knowledge distillation using multiple teacher assistants[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, 2021: 9395-9404.

[46] Mirzadeh S I, Farajtabar M, Li A, et al. Improved knowledge distillation via teacher assistant[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(4): 5191-5198.

[47] Kim J, Hyun M, Chung I, et al. Feature fusion for online mutual knowledge distillation[C]. The 25th International Conference on Pattern Recognition. Milan, 2021: 4619-4625.

[48] Yao A B, Sun D W. Knowledge transfer via dense cross-layer mutual-distillation[C]. European Conference on Computer Vision. Cham: Springer, 2020: 294-311.

[49] Zhang L F, Song J B, Gao A N, et al. Be your own teacher: Improve the performance of convolutional neural networks via self distillation[C]. IEEE/CVF International Conference on Computer Vision. Seoul, 2019: 3712-3721.

[50] Zhang L F, Bao C L, Ma K S. Self-distillation: Towards efficient and compact neural networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(8): 4388-4403.

[51] Yang C L, Xie L X, Su C, et al. Snapshot distillation: Teacher-student optimization in one generation[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, 2019: 2854-2863.

[52] Tan T X, Cao G H. Deep learning on mobile devices through neural processing units and edge computing[C]. IEEE Conference on Computer Communications. London, 2022: 1209-218.

[53] Wang T, Cao K, Zhou J L, et al. Power-efficient layer mapping for CNNs on integrated CPU and GPU platforms: A case study[C]. Proceedings of the 26th Asia and South Pacific Design Automation Conference. Tokyo, 2021: 627-632.

[54] Du Z L, Zhang W, Zhou Z M, et al. Accelerating DNN inference with heterogeneous multi-DPU engines[C]. The 60th ACM/IEEE Design Automation Conference. San Francisco, 2023: 1-6.

[55] Wu Y S, Gong Y F, Zhan Z, et al. MOC: Multi-objective mobile CPU-GPU co-optimization for power-efficient DNN inference[C]. IEEE/ACM International Conference on Computer Aided Design. San Francisco, 2023: 1-10.

[56] Georgiev P, Lane N D, Mascolo C, et al. Accelerating mobile audio sensing algorithms through on-chip GPU offloading[C]. Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services. New York, 2017: 306-318.

[57] Yang T J, Howard A, Chen B, et al. NetAdapt: Platform-aware neural network adaptation for mobile applications[C]. Proceedings of the European Conference on Computer Vision. Munich, 2018: 285-300.

[58] Zhang C, Cao Q, Jiang H, et al. A fast filtering mechanism to improve efficiency of large-scale video analytics[J]. IEEE Transactions on Computers, 2020, 69(6): 914-928.

[59] Qi C Y, Li Z L, Song Z R, et al. ViTframe: Vision transformer acceleration via informative frame selection for video recognition[C]. IEEE 41st International Conference on Computer Design. Washington, 2023: 383-390.

[60] Garg N, Sellathurai M, Bhatia V, et al. Online content popularity prediction and learning in wireless edge caching[J]. IEEE Transactions on Communications, 2020, 68(2): 1087-1100.

[61] Zhao T, Hou I H, Wang S Q, et al. Red/LeD: An asymptotically optimal and scalable online algorithm for service caching at the edge[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(8): 1857-1870.

[62] Sun H, Chen Y R, Sha K W, et al. A proactive on-demand content placement strategy in edge intelligent gateways[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(7): 2072-2090.

[63] Li B, He Q, Chen F F, et al. Cooperative assurance of cache data integrity for mobile edge computing[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 4648-4662.

[64] Chen R T, Wang X B. Maximization of value of service for mobile collaborative computing through situation-aware task offloading[J]. IEEE Transactions on Mobile Computing, 2023, 22(2): 1049-1065.

[65] Nguyen D C, Ding M, Pathirana P N, et al. Cooperative task offloading and block mining in blockchain-based edge computing with multi-agent deep reinforcement learning[J]. IEEE Transactions on Mobile Computing, 2023, 22(4): 2021-2037.

[66] Zhang Y Q, Kishk M A, Alouini M S. Computation offloading and service caching in heterogeneous MEC wireless networks[J]. IEEE Transactions on Mobile Computing, 2023, 22(6): 3241-3256.

[67] Ning Z L, Yang Y X, Wang X J, et al. Dynamic computation offloading and server deployment for UAV-enabled multi-access edge computing[J]. IEEE Transactions on Mobile Computing, 2023, 22(5): 2628-2644.

[68] Ning Z L, Dong P R, Wang X J, et al. Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks[J]. IEEE Transactions on Mobile Computing, 2022, 21(4): 1319-1333.

[69] Cho H, Cui Y, Lee J. Energy-efficient cooperative offloading for edge computing-enabled vehicular networks[J]. IEEE Transactions on Wireless Communications, 2022, 21(12): 10709-10723.

[70] Ebrahimzadeh A, Maier M. Cooperative computation offloading in WiFi enhanced 4G HetNets using self-organizing MEC[J]. IEEE Transactions on Wireless

Communications, 2020, 19(7): 4480-4493.

[71] Ho T M, Nguyen K K. Joint server selection, cooperative offloading and handover in multi-access edge computing wireless network: A deep reinforcement learning approach[J]. IEEE Transactions on Mobile Computing, 2022, 21(7): 2421-2435.

[72] Chen Y, Zhao J, Wu Y, et al. QoE-aware decentralized task offloading and resource allocation for end-edge-cloud systems: A game-theoretical approach[J]. IEEE Transactions on Mobile Computing, 2024, 23(1): 769-784.

[73] 赵璞, 肖人彬. 基于自组织劳动分工的边云协同任务调度与资源缓存算法[J]. 控制与决策, 2023, 38(5): 1352-1362.
(Zhao P, Xiao R B. Edge-cloud collaborative task scheduling and resource cache algorithm based on self-organizing division of labor[J]. Control and Decision, 2023, 38(5): 1352-1362.)

[74] 郑莹莹, 周俊龙, 申钰凡, 等. 时间和能量敏感的端-边-云车路协同系统资源调度优化方法[J]. 计算机研究与发展, 2023, 60(5): 1037-1052.
(Zheng Y Y, Zhou J L, Shen Y F, et al. Time and energy-sensitive end-edge-cloud resource provisioning optimization method for collaborative vehicle-road systems[J]. Journal of Computer Research and Development, 2023, 60(5): 1037-1052.)

[75] 佟兴, 张召, 金澈清, 等. 面向端边云协同架构的区块链技术综述[J]. 计算机学报, 2021, 44(12): 2345-2366.
(Tong X, Zhang Z, Jin C Q, et al. Blockchain for end-edge-cloud architecture: A survey[J]. Chinese Journal of Computers, 2021, 44(12): 2345-2366.)

[76] Teng H J, Li Z T, Cao K, et al. Game theoretical task offloading for profit maximization in mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2023, 22(9): 5313-5329.

[77] Li Y Q, Wang X, Gan X Y, et al. Learning-aided computation offloading for trusted collaborative mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2020, 19(12): 2833-2849.

[78] He X Q, Wang S, Wang X. Providing worst-case latency guarantees with collaborative edge servers[J]. IEEE Transactions on Mobile Computing, 2023, 22(5): 2955-2971.

[79] Champati J P, Liang B. Delay and cost optimization in computational offloading systems with unknown task processing times[J]. IEEE Transactions on Cloud Computing, 2021, 9(4): 1422-1438.

[80] Mahmoodi S E, Uma R N, Subbalakshmi K P. Optimal joint scheduling and cloud offloading for mobile applications[J]. IEEE Transactions on Cloud Computing, 2019, 7(2): 301-313.

[81] Wu H M, Sun Y, Wolter K. Energy-efficient decision making for mobile cloud offloading[J]. IEEE Transactions on Cloud Computing, 2020, 8(2): 570-584.

[82] Zhan W H, Luo C B, Wang J, et al. Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing[J]. IEEE Internet of Things Journal, 2020, 7(6): 5449-5465.

[83] Luo Q Y, Li C L, Luan T H, et al. Minimizing the delay and cost of computation offloading for vehicular edge computing[J]. IEEE Transactions on Services Computing, 2022, 15(5): 2897-2909.

[84] Tang M, Wong V W S. Deep reinforcement learning for task offloading in mobile edge computing systems[J]. IEEE Transactions on Mobile Computing, 2022, 21(6): 1985-1997.

[85] Peng J, Qiu H B, Cai J, et al. D2D-assisted multi-user cooperative partial offloading, transmission scheduling and computation allocating for MEC[J]. IEEE Transactions on Wireless Communications, 2021, 20(8): 4858-4873.

[86] Shi J M, Du J, Shen Y, et al. DRL-based V2V computation offloading for blockchain-enabled vehicular networks[J]. IEEE Transactions on Mobile Computing, 2023, 22(7): 3882-3897.

[87] Ko H, Pack S. Distributed device-to-device offloading system: Design and performance optimization[J]. IEEE Transactions on Mobile Computing, 2021, 20(10): 2949-2960.

[88] Yu S, Dab B, Movahedi Z, et al. A socially-aware hybrid computation offloading framework for multi-access edge computing[J]. IEEE Transactions on Mobile Computing, 2020, 19(6): 1247-1259.

[89] Liu J G, Ren J, Zhang Y M, et al. Efficient dependent task offloading for multiple applications in MEC-cloud system[J]. IEEE Transactions on Mobile Computing, 2023, 22(4): 2147-2162.

[90] Huang L, Bi S Z, Zhang Y J A. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks[J]. IEEE Transactions on Mobile Computing, 2020, 19(11): 2581-2593.

[91] Luo Q Y, Li C L, Luan T H, et al. Collaborative data scheduling for vehicular edge computing via deep reinforcement learning[J]. IEEE Internet of Things Journal, 2020, 7(10): 9637-9650.

[92] Hu S H, Li G H, Shi W S. LARS: A latency-aware and real-time scheduling framework for edge-enabled Internet of vehicles[J]. IEEE Transactions on Services Computing, 2023, 26(1): 398-411.

[93] Tang H J, Wu H M, Qu G J, et al. Double deep Q-network based dynamic framing offloading in vehicular edge computing[J]. IEEE Transactions on Network Science and Engineering, 2023, 10(3): 1297-1310.

[94] Gao M J, Shen R J, Shi L, et al. Task partitioning and offloading in DNN-task enabled mobile edge computing networks[J]. IEEE Transactions on Mobile Computing, 2023, 22(4): 2435-2445.

[95] Wang P F, Yao C, Zheng Z J, et al. Joint task assignment,

transmission, and computing resource allocation in multilayer mobile edge computing systems[J]. IEEE Internet of Things Journal, 2019, 6(2): 2872-2884.

[96] Zhang J, Hu X P, Ning Z L, et al. Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching[J]. IEEE Internet of Things Journal, 2019, 6(3): 4283-4294.

[97] Xu Y Y, Zhu K, Xu H, et al. Deep reinforcement learning for multi-objective resource allocation in multi-platoon cooperative vehicular networks[J]. IEEE Transactions on Wireless Communications, 2023, 22(9): 6185-6198.

[98] Chen X H, Liu Y, Cai L X, et al. Resource allocation for wireless cooperative IoT network with energy harvesting[J]. IEEE Transactions on Wireless Communications, 2020, 19(7): 4879-4893.

[99] Yang B, Cao X L, Bassey J, et al. Computation offloading in multi-access edge computing: A multi-task learning approach[J]. IEEE Transactions on Mobile Computing, 2021, 20(9): 2745-2762.

[100] Zhou H, Jiang K, He S B, et al. Distributed deep multi-agent reinforcement learning for cooperative edge caching in internet-of-vehicles[J]. IEEE Transactions on Wireless Communications, 2023, 22(12): 9595-9609.

[101] Liu T, Fang L, Zhu Y M, et al. A near-optimal approach for online task offloading and resource allocation in edge-cloud orchestrated computing[J]. IEEE Transactions on Mobile Computing, 2022, 21(8): 2687-2700.

[102] Josilo S, Dan G. Joint management of wireless and computing resources for computation offloading in mobile edge clouds[J]. IEEE Transactions on Cloud Computing, 2021, 9(4): 1507-1520.

[103] Saleem U, Liu Y, Jangsher S, et al. Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing[J]. IEEE Transactions on Wireless Communications, 2021, 20(1): 360-374.

[104] 李燕君, 蒋华同, 高美惠. 基于强化学习的边缘计算网络资源在线分配方法[J]. 控制与决策, 2022, 37(11): 2880-2886.
(Li Y J, Jiang H T, Gao M H. Reinforcement learning-based online resource allocation for edge computing network[J]. Control and Decision, 2022, 37(11): 2880-2886.)

[105] Xiao Z, Shu J M, Jiang H B, et al. Multi-objective parallel task offloading and content caching in D2D-aided MEC networks[J]. IEEE Transactions on Mobile Computing, 2023, 22(11): 6599-6615.

[106] 付主木, 王俊朋, 司鹏举, 等. 基于李雅普诺夫随机优化的车辆边缘计算资源管理[J]. 控制与决策, 2022, 37(3): 721-728.
(Fu Z M, Wang J P, Si P J, et al. Resource management of vehicle edge computing based on Lyapunov stochastic optimization[J]. Control and Decision, 2022, 37(3): 721-728.)

[107] Fan W H, Zhao L, Liu X, et al. Collaborative service placement, task scheduling, and resource allocation for task offloading with edge-cloud cooperation[J]. IEEE Transactions on Mobile Computing, 2024, 23(1): 238-256.

[108] Letaief K B, Shi Y M, Lu J M, et al. Edge artificial intelligence for 6G: Vision, enabling technologies, and applications[J]. IEEE Journal on Selected Areas in Communications, 2022, 40(1): 5-36.

[109] Cui Y G, Cao K, Zhou J L, et al. Optimizing training efficiency and cost of hierarchical federated learning in heterogeneous mobile-edge cloud computing[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023, 42(5): 1518-1531.

[110] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Ft Lauderdale, 2017: 1273-1282.

[111] Zhou Y H, Ye Q, Lv J C. Communication-efficient federated learning with compensated overlap-fedavg[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(1): 192-205.

[112] Zhang Z X, Gao Z D, Guo Y X, et al. Scalable and low-latency federated learning with cooperative mobile edge networking[J]. IEEE Transactions on Mobile Computing, 2024, 23(1): 812-822.

[113] Wu D L, Yang W H, Jin H Y, et al. FedComp: A federated learning compression framework for resource-constrained edge computing devices[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024, 43(1): 230-243.

[114] Xu Z C, Li L, Zou W T. Exploring federated learning on battery-powered devices[C]. Proceedings of the ACM Turing Celebration Conference. Chengdu, 2019: 1-6.

[115] Li L, Shi D, Hou R H, et al. To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices[C]. IEEE Conference on Computer Communications. Vancouver, 2021: 1-10.

[116] Chen R, Li L, Xue K P, et al. Energy efficient federated learning over heterogeneous mobile devices via joint design of weight quantization and wireless transmission[J]. IEEE Transactions on Mobile Computing, 2023, 22(12): 7451-7465.

[117] Li Q B, Diao Y Q, Chen Q, et al. Federated learning on non-IID data silos: An experimental study[C]. Proceedings of the 38th International Conference on Data Engineering. Kuala Lumpur, 2022: 965-978.

[118] Tan Y, Long G D, Liu L, et al. FedProto: Federated prototype learning across heterogeneous clients[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(8): 8432-8440.

[119] Yang Z Q, Zhang Y G, Zheng Y, et al. Fed: Feature distillation against data heterogeneity in federated learning[C]. Proceedings of the Conference on Neural

Information Processing Systems. New Orleans, 2023, DOI: 10.1007/s12035-015-9112-7.

[120] Kong Y X, Yang P, Cheng Y. Edge-assisted on-device model update for video analytics in adverse environments[C]. Proceedings of the 31st ACM International Conference on Multimedia. Ottawa, 2023: 9051-9060.

[121] Wang L, Lu K, Zhang N, et al. Shoggoth: Towards efficient edge-cloud collaborative real-time video inference via adaptive online learning[C]. Proceedings of the 60th ACM/IEEE Design Automation Conference. San Francisco, 2023: 1-6.

[122] Wang L H, Yu Z W, Yu H Y, et al. AdaEvo: Edge-assisted continuous and timely DNN model evolution for mobile devices[J]. IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2023.3316388.

[123] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[C]. Proceedings of the Advances in Neural Information Processing Systems. Vancouver, 2020: 1877-1901.

[124] Xu H, Kostopoulou K, Dutta A, et al. DeepReduce: A sparse-tensor communication framework for federated deep learning[C]. Proceedings of the Advances in Neural Information Processing Systems. Piscataway: IEEE, 2021: 21150-21163.

[125] Zhang L L, Han S H, Wei J Y, et al. Nn-Meter: Towards accurate latency prediction of deep-learning model inference on diverse edge devices[C]. Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services. New York: ACM, 2021: 81-93.

[126] Liu W H, Geng J W, Zhu Z W, et al. Sniper: Cloud-edge collaborative inference scheduling with neural network similarity modeling[C]. Proceedings of the 59th ACM/IEEE Design Automation Conference. San Francisco California, 2022: 505-510.

[127] Wu J, Wang L, Pei Q Y, et al. HiTDL: High-throughput deep learning inference at the hybrid mobile edge[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 4499-4514.

[128] Yang X, Xu Z K, Qi Q, et al. PICO: Pipeline inference framework for versatile CNNs on diverse mobile devices[J]. IEEE Transactions on Mobile Computing, 2024, 23(4): 2712-2730.

[129] Wu J, Wang L, Jin Q R, et al. Graft: Efficient inference serving for hybrid deep learning with SLO guarantees via DNN re-alignment[J]. IEEE Transactions on Parallel and Distributed Systems, 2024, 35(2): 280-296.

[130] Hanyao M X, Jin Y B, Qian Z Z, et al. Edge-assisted online on-device object detection for real-time video analytics[C]. Proceedings of the IEEE Conference on Computer Communications. Vancouver, 2021: 10.

[131] Ran X K, Chen H L, Zhu X D, et al. DeepDecision: A mobile deep learning framework for edge video analytics[C]. Proceedings of the IEEE Conference on Computer Communications. Honolulu, 2018: 1421-1429.

[132] Galanopoulos A, Ayala-Romero J, Leith D, et al. AutoML for video analytics with edge computing[C]. Proceedings of the IEEE Conference on Computer Communications. Vancouver, 2021: 10.

[133] Luo G Y, Shao C Z, Cheng N, et al. EdgeCooper: Network-aware cooperative LiDAR perception for enhanced vehicular awareness[J]. IEEE Journal on Selected Areas in Communications, 2024, 42(1): 207-222.

[134] Zhang M, Zhu Y F, Shen L F, et al. OmniSense: Towards edge-assisted online analytics for 360-degree videos[C]. Proceedings of the IEEE Conference on Computer Communications. New York, 2023: 1-10;

[135] Yin L, Sun J, Zhou J L, et al. ECFA: An efficient convergent firefly algorithm for solving task scheduling problems in cloud-edge computing[J]. IEEE Transactions on Services Computing, 2023, 16(5): 3280-3293.

[136] Ghimire D, Kil D, Kim S H. A survey on efficient convolutional neural networks and hardware acceleration[J]. Electronics, 2022, 11(6): 945.

[137] Song C H, Liu S, Han G J, et al. Edge-intelligence-based condition monitoring of beam pumping units under heavy noise in industrial internet of things for industry 4.0[J]. IEEE Internet of Things Journal, 2023, 10(4): 3037-3046.

[138] 任姚丹珺, 戚正伟, 管海兵, 等. 工业互联网边缘智能发展现状与前景展望[J]. 中国工程科学, 2021, 23(2): 104-111.
(Ren Y D J, Qi Z W, Guan H B, et al. Development and prospect of edge intelligence for industrial internet[J]. Strategic Study of CAE, 2021, 23(2): 104-111.)

[139] Hudson N. Smart decision-making via edge intelligence for smart cities[M]. Lexington: University of Kentucky, 2022: 16-18.

[140] 张开元, 桂小林, 任德旺, 等. 移动边缘网络中计算迁移与内容缓存研究综述[J]. 软件学报, 2019, 30(8): 2491-2516.
(Zhang K Y, Gui X L, Ren D W, et al. Survey on computation offloading and content caching in mobile edge networks[J]. Journal of Software, 2019, 30(8): 2491-2516.)

[141] 杨维永, 刘苇, 崔恒志, 等. SG-Edge: 电力物联网可信边缘计算框架关键技术[J]. 软件学报, 2022, 33(2): 641-663.
(Yang W Y, Liu W, Cui H Z, et al. SG-edge: Key technology of power internet of Things trusted edge computing framework[J]. Journal of Software, 2022, 33(2): 641-663.)

[142] Hudson N, Hossain M J, Hosseinzadeh M, et al. A framework for edge intelligent smart distribution grids via federated learning[C]. International Conference on Computer Communications and Networks. Athens, 2021: 1-9.

[143] Zhou H, Zhang Z Y, Li D W, et al. Joint optimization of computing offloading and service caching in edge computing-based smart grid[J]. IEEE Transactions on Cloud Computing, 2023, 11(2): 1122-1132.

[144] Shen Y F, Shi Y M, Zhang J, et al. Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis[J]. IEEE Journal on Selected Areas in Communications, 2021, 39(1): 101-115.

[145] 张雪晴, 刘延伟, 刘金霞, 等. 面向边缘智能的联邦学习综述[J]. 计算机研究与发展, 2023, 60(6): 1276-1295.
(Zhang X Q, Liu Y W, Liu J X, et al. An overview of federated learning in edge intelligence[J]. Journal of Computer Research and Development, 2023, 60(6): 1276-1295.)

[146] 朱力, 龚泰源, 梁豪, 等. 边缘智能在轨道交通中的应用: 前景与展望[J]. 电子与信息学报, 2023, 45(4): 1514-1528.
(Zhu L, Gong T Y, Liang H, et al. Application of edge intelligence in rail transit: Prospects and future outlook[J]. Journal of Electronics & Information Technology, 2023, 45(4): 1514-1528.)

[147] Li H, Ota K, Dong M X. Learning IoV in 6G: Intelligent edge computing for Internet of vehicles in 6G wireless communications[J]. IEEE Wireless Communications, 2023, 30(6): 96-101.

[148] Muhammad G, Alshehri F, Karray F, et al. A comprehensive survey on multimodal medical signals fusion for smart healthcare systems[J]. Information Fusion, 2021, 76: 355-375.

[149] Alex K. AI and iot-based technologies for precision medicine[M]. Pennsylvania: IGI Global, 2023: 36-59.

[150] Ning Z L, Dong P R, Wang X J, et al. Mobile edge computing enabled 5G health monitoring for Internet of medical things: A decentralized game theoretic approach[J]. IEEE Journal on Selected Areas in Communications, 2021, 39(2): 463-478.

[151] Li X H, Chen T, Cheng Q F, et al. Smart applications in edge computing: Overview on authentication and data security[J]. IEEE Internet of Things Journal, 2021, 8(6): 4063-4080.

[152] Nour B, Cherkaoui S, Mlika Z. Federated learning and proactive computation reuse at the edge of smart homes[J]. IEEE Transactions on Network Science and Engineering, 2022, 9(5): 3045-3056.

[153] IDC. IDC: 2024年中国智能家居市场十大洞察[EB/OL]. (2024-01-01)[2024-01-20]. https://www.idc.com/getdoc.jsp?containerId=prCHC51606924.

[154] Pan G J, Zhang H, Xu S G, et al. Joint optimization of DNN inference delay and energy under accuracy constraints for AR applications[C]. Proceedings of the Global Communications Conference. Rio de Janeiro, 2022: 2230-2235.

[155] Sukhmani S, Sadeghi M, Erol-Kantarci M, et al. Edge caching and computing in 5G for mobile AR/VR and tactile internet[J]. IEEE MultiMedia, 2019, 26(1): 21-30.

[156] He Y, Ren J, Yu G, et al. Optimizing the learning performance in mobile augmented reality systems with CNN[J]. IEEE Transactions on Wireless Communications, 2020, 19(8): 5333-5344.

[157] Wang C, Zhang S, Chen Y, et al. Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics[C]. Proceedings of the Conference on Computer Communications. Toronto, 2020: 257-266.

[158] Mendez J, Bierzynski K, Cuéllar M P, et al. Edge intelligence: Concepts, architectures, applications, and future directions[J]. ACM Transactions on Embedded Computing Systems, 2022, 21(5): 1-41.

作者简介

侯祥鹏(1999−), 男, 博士生, 从事边缘智能、移动边缘计算等研究, E-mail: xphou@njust.edu.cn;

兰兰(2002−), 女, 硕士生, 从事边缘智能、能量采集系统等研究, E-mail: Lanlan@njust.edu.cn;

陶长乐(2000−), 男, 硕士生, 从事边缘智能、移动边缘计算等研究, E-mail: cltao@njust.edu.cn;

寇小勇(1992−), 男, 博士生, 从事边缘计算、侧信道安全等研究, E-mail: kouxy@njust.edu.cn;

丛佩金(1994−), 女, 副教授, 博士, 从事边缘计算、云计算、物联网等研究, E-mail: cpj@njust.edu.cn;

邓庆绪(1970−), 男, 教授, 博士, 从事信息物理系统、实时嵌入式系统等研究, E-mail: dengqx@mail.neu.edu.cn;

周俊龙(1988−), 男, 副教授, 博士, 从事边缘智能、云计算、嵌入式实时系统等研究, E-mail: jlzhou@njust.edu.cn.