计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202110867
60(2): 398−414, 2023

面向边缘智能的协同推理综述

王 睿[1,2] 齐建鹏[1] 陈 亮[1] 杨 龙[1]
[1](北京科技大学计算机与通信工程学院 北京 100083)
[2](北京科技大学顺德研究生院 广东佛山 528300)
(wangrui@ustb.edu.cn)

Survey of Collaborative Inference for Edge Intelligence

Wang Rui[1,2], Qi Jianpeng[1], Chen Liang[1], and Yang Long[1]
[1](*School of Computer and Communication Engineering*, *University of Science and Technology Beijing*, *Beijing* 100083)
[2](*Shunde Graduate School of University of Science and Technology Beijing*, *Foshan*, *Guangdong* 528300)

Abstract At present, the continuous change of information technology along with the dramatic explosion of data quantity makes the cloud computing solutions face many problems such as high latency, limited bandwidth, high carbon footprint, high maintenance cost, and privacy concerns. In recent years, the emergence and rapid development of edge computing has effectively alleviated such dilemmas, sinking user demand processing to the edge and avoiding the flow of massive data in the network. As a typical scenario of edge computing, edge intelligence is gaining increasing attention, in which one of the most important stages is the inference phase. Due to the general low performance of resources in edge computing, collaborative inference through resources is becoming a hot topic. By analyzing the trends of edge intelligence development, we conclude that collaborative inference at the edge is still in the increasing phase and has not yet entered a stable phase. We divide edge-edge collaborative inference into two parts: Intelligent methods and collaborative inference architecture, based on a thorough investigation of edge collaborative inference. The involved key technologies are summarized vertically and organized from the perspective of dynamic scenarios. Each key technology is analyzed in more detail, and the different key technologies are compared horizontally and analyzed on the application scenarios. Finally, we propose several directions that deserve further studying in collaborative edge inference in dynamic scenarios.

Key words edge computing; edge intelligence; machine learning; edge collaborative inference; dynamic scenario

摘 要 近年来,信息技术的不断变革伴随数据量的急剧爆发,使主流的云计算解决方案面临实时性差、带宽受限、高能耗、维护费用高、隐私安全等问题.边缘智能的出现与快速发展有效缓解了此类问题,它将用户需求处理下沉到边缘,避免了海量数据在网络中的流动,得到越来越多的关注.由于边缘计算中资源性能普遍较低,通过资源实现协同推理正成为热点.通过对边缘智能发展的趋势分析,得出边缘协同推理目前仍处于增长期,还未进入稳定发展期.因此,在对边缘协同推理进行充分调研的基础上,将边缘协同推理划分为智能化方法与协同推理架构2个部分,分别对其中涉及到的关键技术进行纵向归纳整理,并从动态场景角度出发,对每种关键技术进行深入分析,对不同关键技术进行横向比较以及适用场景分析.最后对动态场景下的边缘协同推理给出值得研究的若干发展方向.

关键词 边缘计算;边缘智能;机器学习;边缘协同推理;动态场景

中图法分类号 TP391

收稿日期:2021−08−26;**修回日期**:2022−04−15

基金项目:国家自然科学基金项目(62173158,72004147)

This work was supported by the National Natural Science Foundation of China (62173158,72004147).

Gartner 指出 2022 年将有 75% 的企业数据在边缘侧产生[1], IDC 预测 2025 年将有 416 亿个边缘侧设备实现互联数据量达 79.4 ZB[2]. 由于云计算的实时性差、带宽受限、高能耗、维护费用高、隐私安全等问题[3-7], 将不能应对边缘侧如此海量的设备与数据,使对用户服务的有效提供面临严峻挑战, 促使边缘计算的蓬勃发展[8]. 边缘计算将计算、存储等资源下沉至用户侧, 以其低时延、动态性、移动性以及位置感知等特征,在智能医疗、智能家居、军事及农业等领域发挥了重要作用[9-13]. 随着通用设备的不断普及、专用设备的不断下沉、虚拟化及中间件技术的飞速发展、设备性能的不断提升以及基础设施运营商的大力投入等[13], 催生出大量可在边缘侧进行训练与推理的边缘协同技术架构, 如 FATE,Paddle-Lite,TensorFlow Lite 等[14], 借助多样化的边缘设备协同能力, 使单设备下的多业务场景、多设备下的复杂智能业务场景发展迅速. 各种云、边、端间协同技术的不断进步[15-16] 与各种机器学习模型训练优化技术、轻量化技术[17] 的深入研究, 促进了边缘智能(edge intelligence, EI)相关方向的飞速发展.边缘智能或智慧边缘计算是指借助边缘侧辅助实现机器学习模型的训练与推理的一系列智能化方法, 使智能更加高效、贴近用户、解决人工智能“最后一公里”问题[18-22]. 边缘协同智能则指在边缘智能的基础上进一步通过边缘节点间的协同, 融合边缘计算资源(网络、计算、存储、感知、应用等)核心能力实现的智能. 从其生命周期来看, 可划分为训练阶段、推理阶段以及模型的部署更新, 而本文则聚焦于其中的推理阶段.

本文第 1 节对边缘智能侧重于从协同推理角度的发展简史以及整体过程进行纵向总结, 引出目前边缘协同推理涉及到的关键技术, 描绘出边缘协同推理在边缘协同智能中的整体位置, 并给出 2 个分类标准, 将边缘协同推理的智能化方法与边缘协同推理的整体架构, 与已有文献进行比较, 突出本文贡献; 第 2 节从协同角度根据分类标准对已有的协同推理阶段、部署更新等问题进行总结并结合边缘计算资源特点对不同技术进行横向比较与分析; 第 3 节聚焦于对边缘协同推理在动态场景下的挑战进行总结, 本文提到的动态场景[23] 包括有地理位置带来的环境变化、边缘计算资源发散、网络拓扑的变化、服务性能波动大、节点频繁交换导致服务或系统吞吐量的动态变化以及应用、网络、设备等带来的不确定性, 标准不统一的场景.并对未来值得关注的研究方向进行初步探讨; 第 4 节总结本文.

1 边缘协同推理概述

1.1 边缘协同智能发展

图 1 展示的是在谷歌学术上使用“edge intelligence”关键词进行检索, 得到的关于边缘智能的文献数量与年份对应的发展趋势. 从图 1 来看, 截止 2020 年, 边缘智能的发展正处于爆发期, 还未达到平稳发展期.

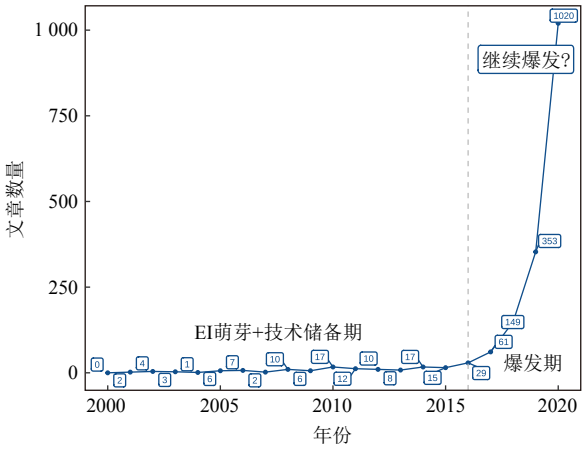


Fig. 1 Edge intelligence developmental trend

图 1 边缘智能发展趋势

进一步, 通过对 Web of Science 中以关键词“edge intelligen”检索得到的 239 篇文献进行分析, 本文将边缘协同智能划分为 3 个阶段: 萌芽期、技术储备期以及爆发期.本文关注的边缘协同推理在图 2 中同时给出了所涉及到的推理过程中使用的核心技术及其在边缘智能发展过程中的关键时间节点, 从中也可以观察到, 人工智能领域涌现出的新技术会极快的同步应用到边缘智能中(图 2 中相同形状图标表示同一类型技术), 如神经网络架构搜索(neural architecture search, NAS)与计算能力受限的硬件资源结合后演变为神经网络架构实现搜索(neural architecture and implementation search, NAIS).技术突破的关键问题是如何在资源受限的边缘计算场景中应用新技术.这些协同推理关键技术极大促进了边缘智能的发展, 下文将在边缘智能的基础上, 结合图 1、图 2, 重点对这些关键技术进行归纳总结并同时从协同角度给出分析.

边缘协同智能萌芽期(2003—2012 年). “edge intelligence”一词最初源自有线网络传输场景, 主要目标是提升网络的可靠性、智能性[24-25]. 随着移动通信技术的发展,多媒体为代表的各类服务在网络中呈爆发态势, 用户的移动性给各类服务的管理带来挑战,边缘协同智能被应用于移动用户的管理,主要目标是提升服务管理效率、降低时延[26-27]. 而此时对如

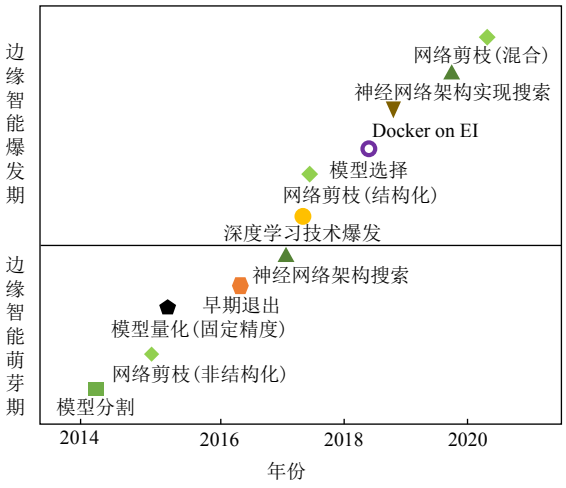


Fig. 2 Emerging time of key techniques in collaborative inference

图 2 协同推理关键技术出现时间

何利用分布式网络中的资源还未形成系统性认识，多为简单的智能算法相互组合，节点间协同性不强，未涉及到与深度学习技术的结合，能支持的智能业务有限.

边缘协同智能技术储备期(2012—2016 年). 该阶段随着深度学习技术的迅速发展，大量引人注目的成果不断涌现，如 2012 年的 AlexNet[28] 和 2015 年的 Inception[29] 分别在 ImageNet 比赛中取得了令人瞩目的成果.随后，更加轻量化的深度学习模型处理技术被不断提出，并应用到边缘设备上，使边缘设备具有了智能化的推理能力[30].随着深度学习与边缘计算不断融合，基于边缘协同的智能处理方案开始显现，如在跌倒检测应用中在边缘设备和边缘服务器之间对智能检测算法进行简单拆分[31].同时，云原生的虚拟化容器技术，如 Docker，也开始迅速发展.此时与深度学习结合的边缘智能开始成为主流，一些中间件、虚拟化技术也飞速发展，但还未出现以边缘计算资源特点与深度学习模型结构融合的推理方法.

边缘协同智能爆发期(2017 年以后). 得益于上个阶段的技术储备，该时期大量的深度学习模型与边缘计算结合，众多边缘协同智能处理方案迅速增长，如边缘计算场景下的分布式协同训练[32]、联邦学习[33]；基于模型切割、早期退出等技术的分布式协同推理[34-35]、浅层的宽度学习系统[36] 与虚拟化技术的结合，使边缘协同智能的快速落地成为可能，极大减轻了不同设备间的运维成本[37-38]. 现阶段用于边缘协同推理的场景复杂多样，用于推理的基础服务或架构还未形成稳定发展的趋势，表现出应用层所具有的多样化特点.

结合边缘智能当前所处的阶段可以看出，其目前正处快速发展阶段，这也说明边缘协同推理方向存在众多问题与挑战，由于目前未对边缘协同推理形成整体概貌，有必要对其进行进一步分析，对相关技术进行归类划分，指明未来的发展方向.

1.2 边缘协同推理的整体过程

边缘协同推理利用节点间的协同，通过不同的训练优化手段，获取用于协同推理的模型并结合场景的资源特点等信息在训练节点或边缘设备等资源上完成部署. 如图 3 所示，按照边缘协同推理在边缘协同智能中的生命周期位置来看，推理阶段与训练阶段相互结合，是一个不断往复循环、不断提升的过程. 根据推理业务需求，其中还可能涉及模型的更新部署(本文“更新”特指替换服务中的推理模型，而非训练阶段的梯度等的更新).

通过训练得到的模型往往不能直接应用到推理场景，需要额外的模型处理步骤.根据模型的生命周期以及推理场景的资源特点，训练阶段可采用的优化手段有模型选择、模型量化、早期退出、神经网络

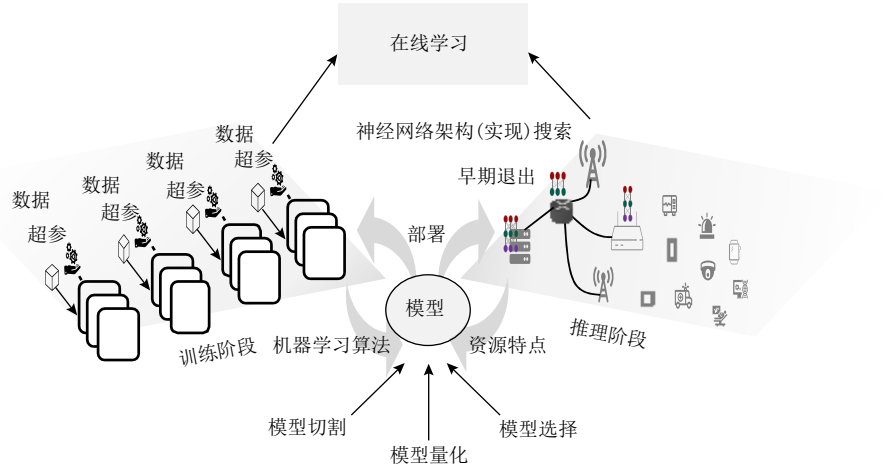


Fig. 3 Key techniques, processes and application scenarios of edge collaborative inference

图 3 边缘协同推理关键技术、过程及应用场景

架构(实现)搜索等.推理阶段可对模型进行2次处理，结合早期退出、模型量化、神经网络架构(实现)搜索等方式对模型进行进一步修改，以及在支持节点协同的情况下,采用模型切割、模型选择等技术对模型进行2次无损处理；在推理阶段的后期，由于时间、业务、数据等的变化涉及到模型的更新，此时多借助重新训练模型、在线学习等技术.

除此之外，通过图3还可看出，边缘计算场景的资源异构性会涉及到不同的推理架构适用性问题.目前来看推理架构的选择通常与特定的推理模型处理技术相关.本文将从推理的智能化方法与架构2个角度对现有研究展开讨论.值得一提的是，目前边缘计算方面有着众多优秀的工作，由于篇幅所限，本文主要针对边缘协同中推理阶段的模型处理、更新部署以及运行推理的架构进行阐述.

1.3 与已有综述研究的比较以及本文的贡献

由于推理过程中所用机器学习模型规模大、复杂性高，限制了推理关键技术在边缘计算场景下的训练与应用.目前，有研究者针对模型的推理内容进行了总结.文献[39]探讨了分布式机器学习的发展，讨论了其系统性挑战以及利用高性能计算加速和提高可扩展性的方法，描述了一种分布式机器学习的参考架构,基于此给出了各个阶段的常用算法.文献[40]总结了分布式协同优化中的热点研究方向，包括加速优化算法、非凸优化算法和无梯度优化算法，并结合实际应用场景进行了展望.文献[41]回顾了物联网场景中能够支持机器学习的模型在低性能硬件上执行的主要技术，比较了深度学习、RNN、GAN等可在边缘设备上实现的机器学习算法和这类算法下沉到边缘的过程；作为底层支持，还介绍了基于边缘服务器的架构、支持AI的物联网无线标准、卸载技术、隐私问题处理和训练阶段的边缘实现.文献[42]在深度学习的基础知识和最新技术的基础上，分析了在边缘设备上执行深度神经网络训练和深度学习模型推理的方法和架构，并探讨了在边缘部署深度学习的未来与挑战.文献[43]在文中讨论了加速深度学习推理的不同结构与方法，分为仅设备端执行、仅边缘服务器执行和中间方案3种，以及在边缘设备上训练深度学习模型，重点关注了多设备和隐私处理的分布式训练.文献[18]在概述了深度学习和边缘智能的基本概念与定义后，针对深度学习模型在网络边缘进行训练/推理的总体架构、框架和新兴关键技术作出了介绍.

文献[18, 39−43]关注深度神经网络模型压缩与加速技术，并针对边缘设备中的训练和推理作出了总结，与之不同的是，本文充分调研了近几年来边缘智能相关的文献，从动态场景角度，重点关注节点间协同的推理处理方式，分别从推理模型与架构的角度进行描述，总结了用于边缘协同推理的核心技术发展，对边缘智能中训练和推理相关技术面临的机遇与挑战做出了展望.

本文贡献有5个方面：

1)按照关键技术的发展脉络，对边缘协同智能发展史进行了梳理；

2)对边缘协同推理进行分类，将其划分为边缘协同推理的智能化方法与架构；

3)对边缘协同推理中的模型轻量化技术进行归纳整理及分析，并横向比较了不同技术的适用范围；

4)对边缘协同推理中的架构进行归纳整理，分析了不同架构的优缺点及适用场景讨论；

5)除了对每种技术进行单独分析外，本文还对边缘协同推理的共性问题进行了分析，并指出其值得发展的研究方向.

2 边缘协同推理核心智能化方法与架构

将边缘智能中的协同推理阶段现有研究划分为2个方面：推理智能化方法与架构.智能化方法指的是与推理任务相关的智能化方法，本文主要针对深度学习(或深度神经网络).架构指运行推理任务的底层网络拓扑结构.通过对智能化方法的观察与分析，归纳出用于边缘协同推理的核心技术；通过对已有工作在架构角度的抽象，归纳出目前协同推理在部署及运行时的特点、适用场景及不足之处.

2.1 边缘协同推理的智能化方法

主流的深度神经网络模型大小通常为几兆字节甚至几百兆字节，计算量较高给低配置的边缘节点带来了挑战[44]，因此需要考虑如何在边缘节点上对模型进行部署.一般而言，边缘计算中的各个资源指标是有限的，这就导致在给定的约束条件下会存在多个可行解，需要考虑对存储、计算、通信、能耗、隐私等的“折中”方案，实现“折中”的过程涉及模型的处理方式，主要是对模型进行轻量化的优化技术.本节就目前相关主流技术进行归纳整理，并在协同推理角度进行分析.

2.1.1 模型切割

深度神经网络模型多具有良好的内部结构，如图4所示，按照模型的内部结构可通过纵切、横切及混切等方式将模型切分成不同粒度且具有相互依赖关

系的模型切片[45], 之后将切片按照依赖关系分别部署在云及边缘端. 如采用纵切方式的 DeepThings[46-47], 横切方式的 Neurosurgeon[34]、MoDNN[48]、Cogent[49]. 混切方式的 DeepX[50]、AOFL[51]、CRIME[52]、DeepSlicing[53], 以模型切割为主压缩等其他轻量化方法为辅的 Edgent[54], ADCNN[55] 等, 通过优化资源(能耗、通信或计算等)的代价函数对模型内部的切割点进行枚举, 以寻找满足用户或系统需求的切割方案.模型切割技术在保证模型推理精度不变的前提下, 能更好的适应边缘计算. 但由于边缘计算中涉及资源分布广泛、性能不一, 尤其是在动态场景下资源地理分布范围广, 造成所面临的环境时刻发生变化且不唯一; 网络规模变大造成网络拓扑变化; 边缘计算场景中计算资源不集中, 靠近用户侧, 资源发散; 应用、网络、设备的异构性等边缘计算资源类型多; 由于同一节点运行多种类型服务的情况, 资源分配困难使得网络拥堵, 服务器波动大; 故障频率高、性能波动大、节点协调困难、服务调整趋于被动, 存在滞后性并需要分布式思维解决等问题.切割技术中需要引起关注的是切割的整体过程, 其中包括切割的执行者、切割时的参考依据如何获取、切片的依赖关系映射、切片更新时间及频率等, 这决定了切割的方案在边缘计算环境中的适用性及稳定性, 表 1 给出了不同方法的模型切割过程中涉及的关键步骤的比较.

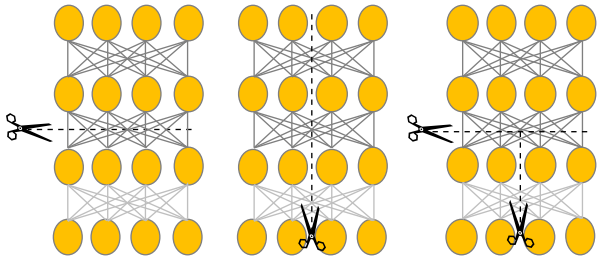


Fig. 4 Model partition methods

图 4 模型切割方式

Table 1 Comparison of Model Partition Methods

表 1 模型切割方法比较

方法	模型切割执行者	切割的依据收集方式	切片依赖关系处理/服务发现方式	切片更新方式	优化目标	运行时涉及的切片数量
DeepThings[46-47]	网关(gateway)	周期性收集节点状态	网关统一调度	节点拥有完整模型	内存、通信	≥2
ADCNN[55]	中心节点(central node)	基于历史任务执行时延估计	中心节点调度	节点拥有完整模型	时延、通信	≥2
Neurosurgeon[34]	客户端	实时观测当前网络、能耗状态	IP 绑定(固定)	节点拥有完整模型	能耗(时延)	2
MoDNN[48]	中心节点(group owner)	节点注册到中心节点时获取	中心节点调度	部署一次,无更新	时延	≥2
DeepX[50]	中心节点(execution planner)	实时收集与线性回归预测	中心节点调度	每次运行推断重新生成执行计划	能耗、内存	≥2
AOFL[51]	云端或中心节点	周期性收集节点状态	IP 绑定(固定)	重新部署	时延、通信	≥2
CRIME[52]	任意节点	节点实时交互	直接邻居集合	节点拥有完整模型	时延、能耗	≥2
DeepSlicing[53]	主节点调度(master)	基于历史任务执行时延估计	中心节点调度	节点拥有完整模型	时延、内存	≥2
Edgent[54]	主节点(边缘服务器)	观测的历史网络数据	IP 绑定 (固定)	重新部署	准确率、时延	2
文献 [45]	中心节点	实时收集节点状态	IP 绑定(固定)	节点拥有完整模型	内存	≥2
Cogent[49]	中心节点(DDPG adgent)	周期性收集节点状态	Kubernetes 提供的静态虚拟 IP 绑定(固定)	重新部署	准确率、时延	2
文献 [56−57]	边缘服务器(server)	根据模型及优化目标折中分析	IP 绑定(固定)	重新部署	计算、通信时延	2

通过表 1 可以看出, 由于边缘计算资源的可用性会发生变化, 因此涉及切片的重新部署. 主流方式多采用中心节点对其他边缘节点状态进行收集, 从而作为模型重新切割的依据, 之后将模型切片下发到边缘节点. 这种收集节点状态的形式默认对所有的边缘节点具有感知能力, 但这个前提并不适用于边缘计算尤其是动态场景[55,58]. 在模型的重新切割及部署上, 涉及 2 个关键问题: 1)模型或数据可支持的并行度对推理速度产生重要影响, 由于模型的内部结构采用横切的方式, 在并行度上很难提升; 纵切虽然可极大提升并行度, 但一般会带来数据依赖的问题, 因此设计适合纵切的模型是一个重要的研究方向, 如考虑计算复用、设计可用于无数据依赖切割的模型等[45,55]. 2)部署方案更新的频率较高会带来过高的代价, 既包括切片本身也包含所涉及的其他组件; 此外, 切片更新过程中可能会面临无法提供服务问题带来的服务抖动, 此时可参考经典的任务冗余(副本)技术解决. 节点拥有整个模型是避免针对同一模型

重新切割带来的反复下发调度，提升协同推理灵活性的关键. 但对于运行内存不足的节点而言，拥有整个模型存在较大阻碍，此时可借助其他轻量化技术对模型进行压缩或考虑辅助存储、网络存储的形式减少频繁更新切片带来的额外代价，此外，在线学习、感知学习[59]等实时更新模型的学习方式也值得关注.

除了关注模型切片的部署之外，推理时切片间依赖关系的处理也值得关注，如根据网络状态自适应决定相邻切片间数据或张量(tensor)的压缩率[56-60]以应对紧张的通信资源. 在切片间的依赖关系路由或服务发现上，目前多数都采用固定的 IP 映射的方式，或采用解析节点进行，当部署的切片所涉及的节点数量过多时，固定的 IP 映射的方式尤其不适合于动态的边缘计算环境，采用解析节点则引入了额外的解析时间[61]，数据命名网络可为此方面问题提供解决方案[62]. 此外，多用户方面的协同推理还有所欠缺，目前的解决方案多针对单个场景，涉及的应用不多，用户量不多，默认的是边缘服务器资源充足的场景. 当面临多用户、多应用时，由于边缘计算的地理分布特点，云计算中资源的按需扩展很难在边缘侧实现，这会导致资源紧张，使服务满足不了用户或企业需求，此时可参考计算复用的思想，如多场景共用通用模型的部分切片[63]. 虽然依据模型的内部结构对模型进行切分的研究成果较为丰富，但多数模型切片部署方案针对的是边缘计算资源状态相对稳定或基于瞬时状态的静态场景，其稳定性易受动态场景影响而出现系统瓶颈，可以参考的方案是从模型的鲁棒性入手，当存在某个瓶颈节点时可直接跳过部分子切片或神经元[64-65]. 在具体的场景中，由于节点的资源是相对有限的，节点算力存在差异而且处于动态变化中，这增大了对于恰当的模型分割点的选择难度. 无论横切还是纵切的切割方式，划分计算任务时如果不能有效结合节点算力差异，导致任务分配下发不合理，那么节点间的协同效率也会受到较大影响. 另外在节点协同处理问题的情境中，网络状态的处理也十分重要，不仅会直接影响到模型切割的结果，在执行推理的过程中，产生的通信延迟也会明显影响推理服务质量[66]. 同时现有的模型切割方式存在策略的选择不够全面，为了实现协同，难以兼顾推理效果、推理延迟、服务能耗等问题.

2.1.2 模型压缩[67]

由于边缘节点内存、计算能力、能耗等有限，模型的鲁棒性、稀疏性等允许我们通过张量分解方法对张量降秩处理[68]；通过剪枝剔除影响小的参数对模型进行压缩[17,64]；通过量化方法降低权重和中间计算结果的位宽[69-70]，进一步降低模型在内存与计算量上的需求，文献 [17,64,68−70] 方法属于软件优化，在其应用于具体的边端设备时，由于模型的多样性与一些加速芯片的架构特点并不匹配，甚至可能存在“内存墙”问题，在模型的推理速度及能耗提升上还存在阻碍[71]，为了进一步有效利用资源，结合资源特点进行定制化的压缩，软硬件协同优化[72-73]也值得关注.

在软件优化方面，除了与训练过程结合生成低稀疏性的紧致模型外，还包括对已有模型的处理，处理技术主要包含剪枝与量化 2 种技术. 对剪枝而言，从模型的结构出发，可分别对滤波器(filter)[74]、通道(channel)[75-76]、神经元[77]等分别或混合[78]进行压缩处理. 在剪枝粒度上，主要包含非结构化剪枝与结构化剪枝 2 种. 非结构化剪枝删除任意位置的权重，其特点是粒度细、压缩率高，如在多次迭代过程中删除冗余参数的知识蒸馏方法[79]. 但非结构化剪枝并不能显著降低计算量且因存在的稀疏性带来额外开销，需要定制化加速器才能完成计算加速. 结构化剪枝中剪枝粒度大，具有良好的加速效果，如通过后继层对前驱层的重要性反馈删除影响小的通道[75]，但此类方法压缩率相对较低. 这促进了对混合粒度的剪枝方法研究，如满足一定结构规则性的基于模式的剪枝方案对卷积内核进行修剪以满足特定模式[80]. 剪枝操作通常会给推理精度带来不利影响，目前主流方法多采用重新训练的方式解决这个问题[74]，但对重新训练而言，由于计算代价大，重新训练比较适用于精度及效率具有重要意义的场景[70]，即需要评估模型部署后所带来的收益与重新训练的代价后才考虑是否选择重新训练.

对量化而言，由于模型的参数量巨大,低位宽的数据表示方法可以极大压缩模型尺寸，提升推理速度. 根据取值范围，可将量化分为 2 值量化、3 值量化[81]、线性量化、非线性量化[82]，其本质是多对一的映射问题. 2 值量化方法中，主要是将权重映射为 1 和−1[69]，将激活值映射为 1 和 0[83]；3 值量化主要是在 2 值量化基础上引入额外的 0 来增强所能表达的状态空间[84-85]；线性量化则主要将原始权重数据量化为连续的对硬件友好的定点[86-90]；非线性量化通常没有特定的映射规则,也有学者称其为参数共享[91]，如使用不同的哈希映射对网络每一层进行压缩[92]；使用 k-means 聚类实现相近参数的压缩[67,93]，将最近邻居量化到相同位宽[94]等. 由于数据表示的精度上存在损失，因此，量化方法会对推理精度带来一定影响. 同样的，也可通

过重新训练提升推理精度[94], 进一步地, 为了充分贴合硬件的计算性能, 可通过编译器的指令优化使算法级压缩与硬件资源优化相融合[80], 从而达到在可移植的基础上进一步提升推理速度的目的.

在软硬件协同优化方面, 硬件敏感的神经结构搜索(hardware-aware neural architecture search, HNAS)成为热点, 不同于软件层面的模型压缩方法, HNAS将硬件资源与模型同时考虑在内实现定制化压缩. 定制化的压缩方案极大提升了推理性能. 但有些硬件的结构是可重塑或定制化的, 如FPGA. 因此, 好的模型压缩方法还需要考虑如何在一个可变的硬件上进行充分压缩, 如NAIS, NAIS方法所面对的搜索空间既包含模型本身又包含硬件特点, 实现最优的部署需要较高的代价, 针对此问题, 文献[95]设计了一种可微分的方法来加速该过程; 同时, 由于模型稀疏性普遍存在,设计高效的跳零架构, 直接跳过冗余的零值计算.加速计算过程[96]以及存内计算(processing in memory)[97]等也值得关注.

现有大多数方案都基于固定或基于历史数据分析的资源分配, 在资源不稳定的情况下, 推理运行时的效果也值得关注. 该方面可结合硬件进行实时压缩, 考察激活值的稀疏性、动态的压缩激活值. 如, 通过区分图像敏感区域, 自适应地选择不同的激活值和权重量化位宽[98], 通过对输入数据按照取值范围编/解码, 实现多通道自适应压缩[99]等. 这方面的研究目前不多, 尤其是在边缘计算场景下值得关注.

小结与分析: 可以看出, 模型压缩涉及范围广泛, 从模型的内部结构出发, 包含了模型不同的组成部分, 从使用的场景出发包含了与硬件资源结合时的优化, 单独一个方法很难适用于不同的场景, 方法的适用性对边缘计算场景而言至关重要. 由于涉及的资源分布广泛, 节点性能存在很大差异, 仅靠人工设计网络结构及压缩的方式很难普及, 通过软硬件协同的网络压缩自动化方法值得我们进一步挖掘.然而, 目前对模型进行压缩及更新替换的多数方法皆通过高性能服务器或云等远端进行[100], 在不具备这种条件的场景, 如野外、战场、隐私安全要求高的环境或通信代价大的场景, 需要设备间的协同来完成更新. 但由于边缘计算场景资源异构性的存在, 当前压缩方法及精度表示多种多样, 在不同设备间移植的兼容性需要进一步考虑.

就量化方法而言, 虽然保留了完整的模型, 但不同硬件由于功耗上的设计使其可表示的数值精度存在很大差异[101], 固定位宽的表示方式可能并不适用于临近的所有协同节点, 这种情况下接收到的模型与资源的匹配度无法达到最优, 2.1.2节提到的多种优化方法也就无法发挥作用, 再次压缩处理的模型是否满足需求甚至能否再次训练值得关注. 值得一提的是在边缘协同推理场景, 受限资源的分配及调度粒度至关重要, 该问题将决定“腾出资源做合适的任务”等资源调度及优化的发展. 除此之外, 还需要注意的是, 边缘节点的可用资源是不同的、动态变化的, 现有的压缩方法对此的适应性存在空白, 亟待解决.

2.1.3 模型选择/早期退出

如图5和图6所示, 模型选择指首先训练具有不同尺寸大小的模型, 之后结合推理场景, 自适应地选择合适的模型用于离线推理. 早期退出(early exit)与模型选择相似, 不同之处在于早期退出除了最终的输出层之外, 还可通过中间层输出结果,避免数据流经整个网络, 并可以实现参数共享[18].

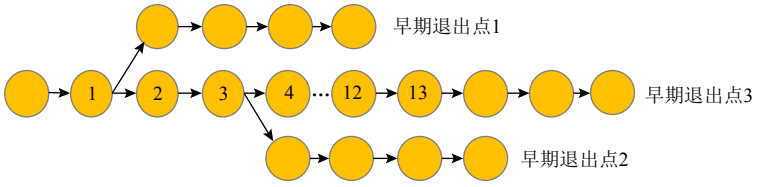


Fig. 5 Early exit pattern

图5 早期退出模式

早期退出(典型的如BranchyNet[102])通过增加多个推理分支提升模型推理速度. 由于多分支的加入, 相比模型切割带来了更多可能性, 如文献[32]针对云边协同场景提出分布式深度神经网络(distributed deep neural network, DDNN), 以分支为切割点, 将切割点的两侧分别部署在边缘端与云端. 与模型选择类似, 由于具有多个推理分支, 退出点的选择是决定边缘协同推理在云端还是边缘端进行推理的关键. 早期退出通过分支选择器计算分支可信度来决定推理的退出点. 可信度可通过计算Softmax输出层的熵大小或额外增加可信度决策模型得到, 如文献[103]提出AO(authentic operation)模块, 为每种类型的推理任务建立了个性化的决策阈值, 文献[104−105]针对连续多推理请求场景(视频分析), 考虑可满足时延

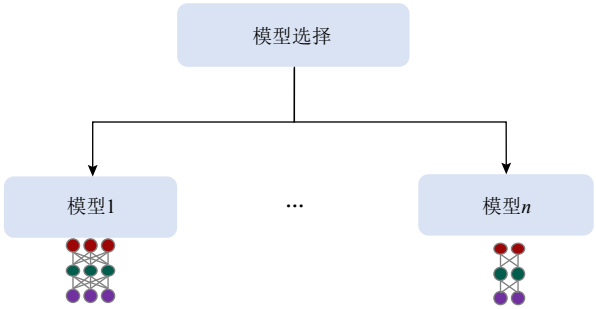


Fig. 6 Model selection pattern

图 6 模型选择模式

要求的退出点，以是否满足时延要求设计收益函数，并以此作为分支选择依据，按照请求的先后次序进行推理，并给出了按先后次序(ordered scheduling structure)调度可以得到的一个最优结果的证明.

在验证方面，如何快速地对所提早期退出算法进行验证是实现快速迭代的前提，文献 [106] 提出了一款基于 MATLAB 的 DeepFogSim 仿真平台，DeepFogSim 以能耗-时延为优化目标，模拟云—边—端协同场景，并支持部分资源动态性. 我们前期对目前已有的大量开源边缘计算仿真平台进行了归纳整理，或可为快速验证想法提供有效的技术候选方案[14]. 同时，针对动态场景特点，我们也研发了开源的仿真平台 EasiEI[107].

在模型选择方面，简单的形式是同时准备 2 种尺寸的模型分别部署在边缘侧与云端，边缘侧用于初步的推理，当推理结果的可信度不满足给定的阈值时，则选择云端进行推理[108-109]. 由于模型训练的架构、网络结构不同，在推理时延、能耗以及精度等有所差异，在同一场景下，如何在不同的模型之间进行选择是首先需要考虑的问题，即模型选择器(model selector)，如以降低推理时延为目标，通过前置专用的预测模型[110-112] 识别输入数据，自动选择合适的推理模型.在不同场景下，尤其是在资源紧张的边缘计算场景下，单终端节点可能面临多任务推理场景或多租户场景. 此时不同任务之间的竞争、调度等易导致服务质量下降. 这方面的研究主要是对不同领域的模型进行智能选择或模型融合[100,106,108,113]. 如文献 [63] 结合迁移学习共享了部分计算用于支持不同场景的推理任务.

小结与分析：可以看出，模型选择与早期退出都涉及到如何根据给定的优化目标进行决策的问题，在决策过程中需要针对目标设置合理的阈值.然而，优化目标一般是推理精度与能耗、时延等的折中. 由于边缘计算场景众多、资源类型不一，自动化地确定阈值是该类轻量化方法能否适应不同场景的关键，尤其是在推理过程中，阈值的确定应当根据当前的运行状况进行动态改变.文献 [54] 考虑了节点协同时网络的动态性，提前计算出不同的早期退出点与切割点候选方案，在推理过程中结合网络变化实时调整推理方案，利用多节点协同共同保证推理精度，满足了一定的实时性.但早期退出方法的退出点数量受限于模型层数，在资源已经受限的边缘协同环境中，由于节点算力差异与变化，按照退出点进行切割然后采用协同推理的方式无法提供更加灵活、更细粒度的资源分配控制，这方面还需要进一步提升. 与其他技术，如模型压缩结合或可提供潜在的解决方案. 在模型选择方面，通过协同方式有效获取节点特点、节点运行状态是选择合适模型的关键. 一种可行的方式是结合不同节点的历史数据与推理模型的算子、结构等来训练用于评估推理效果的模型，近期研究 nn-Meter[114] 根据不同类型边缘节点的特点，对在不同节点上进行推理的时延进行预测取得了不错的效果，该研究或可为在边缘节点上如何选择更好的模型提供支持. 除时延指标外，还有能耗等指标值得考虑.

适用于边缘计算场景的轻量化技术还需要根据场景需求对多种技术进行融合.如在时延及推理精度都具有高要求的工业物联网领域，关注的是如何将轻量级模型运行在单个节点或尽可能地降低多节点协同推理时带来的网络代价，此时可将早期退出与模型切割结合，将一部分推理请求提前过滤[115]；在计算代价大的场景，可考虑将量化与早期退出结合，压缩计算量[94]；当支持节点协同时，可考虑将模型切割、量化与重训练结合[55]，从模型尺寸、推理精度方面一起优化，实现更加高效的分布式协同推理；对于动态场景而言，由于节点负载、网络负载的不断变化会导致部署时所采用的优化目标不满足资源约束，多数解决方案通过将完整的模型存储在每个节点上，通过观测当前的系统状态实时调整协同推理的部署方案，以此来满足资源约束.

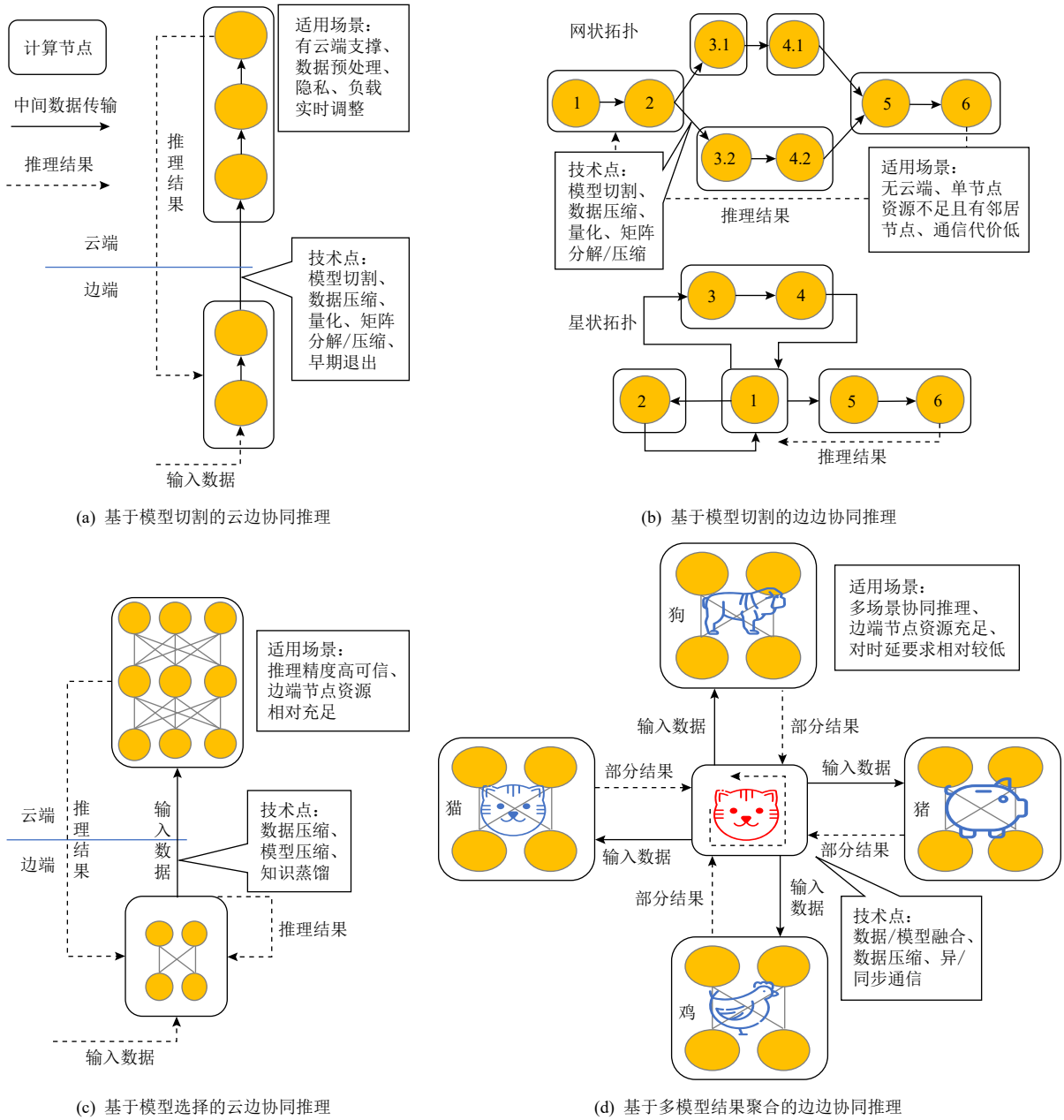
2.2 整体架构

仅通过云与端实现的推理场景受多方面制约，难在复杂的业务场景中发挥作用，主要表现在对带宽密集型业务的原始数据或中间数据传输代价大、隐私安全要求高的业务数据传输敏感；所处环境恶劣的业务与云端连接不稳定或限制上行带宽；传输链路过长容易出现故障；端侧能耗限制较强等. 这在智能驾驶、在线交易以及军事等领域十分普遍. 边缘协同推理通过云、边等资源联邦，克服数据传输、隐私安全、运行环境等存在的问题，对推理任务“就近”

解决. 文献 [8] 指出，“边缘”是一个连续统，那么，协同推理的架构主要关注点则是如何调集连续统中的资源. 本文按照资源及数据的协同处理方式，从云与边的角度对边缘协同推理进行分析.

从中心化计算与否出发，为了便于在逻辑上描述，此处将云端或边缘服务器端统称为云端，将终端节点或具有计算能力的边端节点统称为边端. 如图 7 所示，目前主流的用于进行边缘协同推理的框架主要包含 4 类. 采用模型切割方式的云边协同推理(图 7(a))与边边协同推理(图 7(b))、基于模型选择的云边协同推理(图 7(c))以及基于多个不同任务场景的多模型结果聚合的边边协同推理(图 7(d)). 其中图 7(b)基于模型切割的边边协同推理架构根据任务处理的流程及网络拓扑结构又可分为 2 类：根据模型切割时子切片依赖图形成的网状拓扑与以协调节点或推理请求者为中心的呈放射状的星状拓扑.

其中图 7(a)基于模型切割的云边协同推理依据模型切割技术，将模型分解为具有先后以来关系的不同切片，分别部署在边端与云端. 边端将部分中间结果经过处理后发送至云端完成后续的计算，最终由云端返回推理结果.考虑的指标主要包含隐私安全、通信代价与计算代价的联合优化. 边端除了对数据



(a) 基于模型切割的云边协同推理

(b) 基于模型切割的边边协同推理

(c) 基于模型选择的云边协同推理

(d) 基于多模型结果聚合的边边协同推理

Fig. 7 Mainstream of collaborative inference framework in edge computing

图 7 主流的边缘计算协同推理架构

预处理[115]、计算一部分中间输出结果外，还可与早期退出结合[54,116]，将一部分具有推理功能的分支部署在边端，当边端推理结果不满足需求时，通过云端更深层次的模型进行推理.该类架构从逻辑上看仅存在一个云与边之间的切割点. 如文献 [34,49,54,56−57] 等所述，模型主要切分为 2 部分，分别部署在边端与云端. 由于切片数量的减少，简化了网络拓扑，利于观测网络中的动态变化和协同推理的实时调整. 如文献 [54] 根据可用带宽自适应地划分移动设备和边缘服务器之间的深度神经网络（DNN）计算，由于推理速度受到移动设备上其他计算任务的限制，不足以满足严格的时延要求，因此又引入了早退机制来进一步降低延迟. 除带宽外,节点负载状况也需要关注[116]，文献 [117] 基于动态场景，包括传输带宽变化与节点计算能力变化，提出建立可靠性指标，评估了时延约束下模型的可靠性.

其中图 7（b）基于模型切割的边边协同推理在不具备云端支持，同时单个节点资源不足的场景下，可将模型切片映射或动态调整到到不同的节点，通过节点间的协同完成推理. 该类方式可自发地通过协同交互处理任务，多涉及资源性能受限节点、节点间具有网络连接，关注的指标主要包括能耗、时延、内存等，较适用于动态场景（资源、环境动态变化）下的单方资源受限、无云端控制或控制存在一定阻碍等场景.该类架构从逻辑上看模型存在多个切割点，所涉及的切片数量多大于 2 个，较基于模型切割的云边协同推理更为灵活. 根据调度方式的不同，逻辑上的网络拓扑可分为网状拓扑与星状拓扑. 网状拓扑的部署方式一般事先固定，按照切片的先后依赖顺序查找后继节点，应对复杂的动态场景能力弱；星状拓扑在部署上一般通过中心协调点（或推理请求节点）对切片按照先后依赖顺序不断收集中间结果并转发到其他节点进行调度（参见表 1 中的切片依赖关系处理/服务发现方式），灵活性较网状拓扑更强，但调度频繁易使代价过高.

其中图 7（c）基于模型选择的云边协同推理与模型切割与早期退出结合的云边协同类似，基于模型选择的云边协同推理在边端具有完整的推理能力，不同的是边端所运行的推理模型是完整的. 该类架构下一般是针对同一场景训练 2 个大小不同尺寸的模型，如借助知识蒸馏技术分别部署在云端与边端. 由于边端的模型尺寸小、推理能力相对较弱，因此在推理精度不可靠的情况下会将推理请求转发至云端进行更为精确的推理[118].该类架构边端与云端的链接是否可靠取决于边端的模型推理精度是否满足需求，本质上由边端的性能决定. 如文献 [119] 提出在边端运行小模型（SNN），云端（边缘服务器）运行 DNN，由边端根据自身运行状态（模型复杂度、推理精度、数据质量、计算能力和通信带宽等）选择是否通过云端获取推理结果.

其中图 7（d）基于多模型结果聚合的边边协同推理与多数将推理请求发往云端思路不同，该类架构在概念上与决策级的信息融合相近，中心节点（或推理请求节点）将推理请求下发给多个边端，每个边端具有某一特定任务或领域的推理能力，之后通过汇总来自多个边端的推理结果得出最终结论.不同于基于模型选择的云边协同推理中的场景有限，基于多模型结果聚合的边边协同推理能适应的场景更加多样，这种类似于任务冗余的形式能够提供较高粒度的可靠性及并行性支持.如文献 [120] 提出基于竞争机制和选择性学习方法，在边端每个节点上运行其所擅长的不同领域的模型，利用多节点的协同，实现多边端协同推理.但由于一个推理请求会同时广播给多个边端，边边协同推理主要关注的是推理精度及速度，在能耗及通信方面值得进一步提升，可通过额外的选择器对多个边端是否执行进行提前筛选[121].

小结与分析：基于 2.2 节分类及分析可以看出，从推理结果最终的出处来看，边缘计算协同推理架构可分为 2 类.一类是用户发出推理请求，多节点通过处理协同推理过程中的中间数据，最终由云端或边端中的单个节点得出推理结果，该类思想与向云端卸载任务一致，如图 7（a）~（c）所述.另一类是由边端汇总来自其他多个边端或云端的多个推理结果，之后对比评判得出最终的推理结果，该类思想可参考云端将推理任务卸载到边端或众包，如图 7（d）所述.从整体云边端角度分析，第 1 类涉及对模型的压缩、剪枝、量化、切割等过程，第 2 类则涉及多边端推理结果融合. 表 2 给出了不同的架构所采用的关键技术、针对的问题及其适用场景的比较.

对于云计算，目前的分布式架构所利用的资源丰富，推理效果好（推理精度高且推理速度快）是关注的重点，而能耗并非核心关注问题.边缘协同推理由于其资源特点，在推理效果好的前提下同时期望得到低代价的部署方案，其所处的复杂环境决定了这是一个“折中”问题. 节点状态、环境及用户等会对问题的解产生动态性影响，使资源的状态发生变化. 为了更好地提升资源利用率，同时确保服务的有效性，就需要对资源进行重新调度或部署.然而，边缘协

Table 2 Comparison of Different Architectures

表 2 不同架构的比较

序号	名称	关键结合技术	针对的问题	适用场景
1	基于模型切割的云边协同推理	模型切割、数据压缩、量化、矩阵分解/压缩、早期退出	边端设备能耗、算力有限、能耗与时延“折中”	有云端支撑、数据预处理、隐私、负载实时调整
2	基于模型切割的边边协同推理	模型切割、数据压缩、量化、矩阵分解/压缩	与云端链接不可靠、单节点资源受限、能耗与时延“折中”	无云端支撑、单节点资源不足且有邻居节点、通信代价低
3	基于模型选择的云边协同推理	数据压缩、模型压缩、知识蒸馏	边端设备推理精度不可靠	推理精度高可信、边端节点资源相对充足
4	基于多模型结果聚合的边边协同推理	数据/模型融合、数据压缩、异/同步通信	协同推理并行度低、推理精度不可靠	多场景协同推理、边端节点资源充足、对时延要求相对较低

同推理属于计算及带宽密集型的复杂业务，所处环境多样，用户不一，具有极大的不确定性[122-123]，不同的调度方式在复杂性及时效性或准确性上存在差异，产生的运维代价不同，服务有效性与资源利用率存在冲突.

在资源调度与控制方面，由于边端设备类型众多，资源涉及多方且参与意愿不同，在协同推理时若非采用资源预分配的手段，很难达到实时的最优协同推理.而资源预分配恰恰是需要统一调度的，这意味着会存在资源上的过度消耗、竞争，调度周期也不好确定，无法及时有效处理多方请求. 因此，对所有资源实现统一调度与控制是不现实的，这导致了推理过程中资源募集能力不一，加剧了边缘协同推理的难度.

从涉及的网络计算资源来看，2 节点是分水岭，节点数量影响了调度服务对网络环境、节点负载等的感知能力及处理能力，最终反映在任务调度的实时性上. 单节点自身资源可以支持推理服务的运行，但可靠性差；2 节点协同交互简单，方便管理，但不够健壮，处理任务时可调动的资源有限；多节点鲁棒性强，适合动态场景，但模型复杂，管理及优化困难. 目前，2 节点主要应用在云边端协同场景，多节点更多应用于 IoT 场景[45].

因此，边缘协同推理与众多应用层的服务类似，服务表现碎片化、多角度，很难提供一个通用的基础架构. 如智慧城市场景，单独的轨迹数据处理就涉及到乘客、司机、城市规划、交通等诸多角度[124]. 可以看出，对于边缘协同推理的框架而言，调度是否灵活、推理过程是否具有弹性、能否支持多设备(用户)并发推理及并行需求是关键. 除此之外，由于在推理过程中可能涉及到不同类型信息的融合，是否可快速移植支持异构平台，如借助虚拟化技术；是否融合异构网络，如智能家居中的各种传感器网络[125]，也值得关注.随着时间的推移，还会涉及到推理模型的更新问题. 目前来看，这方面主要依据联邦学习[126]或重新部署的形式，更新的频率及更新的代价与更新后能否带来好的收益值得关注.

边缘计算表现出广泛的异构性、动态性，使不同的优化技术应用在“连续统”中，不同类型的边缘节点上存在巨大挑战[127]，这些节点往往运行多种基础服务. 目前多数方案都仅关注在模型角度，完整的协同推理过程除了 AI 模型外还涉及其他服务，如可靠性保障、数据中间传输、数据存储、日志追踪等的基础支撑服务，尽管多数轻量化技术在时延方面可以满足一些场景需求，但涉及的模块过多时推理服务的稳定性值得进一步研究[128].

3 边缘协同推理在动态场景下的挑战与展望

本文从边缘智能出发，简要描述了其发展过程. 着重从边缘协同角度对模型推理阶段涉及到的关键技术进行了归纳总结，并从动态场景的角度分别进行了分析. 截至目前，边缘协同智能依然处于快速发展阶段，其大体分为 2 类：一类是基于原有的智能化方法与边缘计算资源特点不断结合(如深度神经网络架构实现搜索、混合精度量化)；另一类是直接从边缘计算产生的方法(或称边缘原生方法，如模型选择)结合边缘计算资源的特点(地理分布、异构性、动态性等).目前还存在诸多挑战，下面介绍几个值关注与讨论的方向：

1)推理模型与动态性的适应问题. 以往优化算法较适用于云计算中同质化资源，其资源状态变化不大，一般按需分配、按需扩展，利于结合业务对负载实时调控.然而，在应对具有一定规模的呈地理分布的边缘节点时，由于边缘节点相对于云计算节点的可控性不强，且存在资源异构性及动态性，使当前边缘协同推理智能化的主流方法中还存在一些值得进一步关注的问题，如模型切割技术的整体或部分更新问题；模型压缩技术在不同节点间的可移植性及再训练问题；模型选择/早期退出在面对资源变化

时的不同模型问题；分支切换灵活性及资源分配粒度问题. 同时，如 2.2 节所述，尽管目前已经呈现出不同技术的融合态势，但在协同环境下，边缘协同推理智能化方法依旧面临许多共性问题，如已部署模型的更新替换频率及兼容性问题、动态资源变化与所运行模型(或部分)资源需求的匹配问题、额外的中心调度或部署代价问题等.

2)在边缘协同推理验证方面，动态场景建模将促进协同推理相关方法的良性发展. 动态场景下的边缘计算普遍存在硬件故障、系统及软件的负载变化、人为与环境因素的影响、地理分布广泛以及服务状态复杂多变易受影响等特点，运维成为一大挑战.纵观已有研究，在验证思路时多数通过实际的生产场景，或较为简单的代价评估模型模拟，缺乏动态场景特点上的考虑. 造成这种现状的原因，一是生产场景对于广泛的研究人员并不容易可及，二是模拟场景考虑的影响因素不够全面，在边缘协同推理的有效性评估上还存在困难，给边缘协同智能的落地带来一定阻碍. 除此之外，边缘协同推理作为完整应用，还涉及各种网络、计算中间件的运行，这些中间件无一不需要大量的资源来维护. 因此，提供一个可信的动态仿真场景值得研究.

3)在边缘原生方法方面，在线学习与边缘计算的结合或可为边缘协同智能提供更广阔的适用场景. 目前多数研究或工作将训练与推理分开，界线清晰，较适合于资源丰富且动态性不强的场景. 但对于涉及计算、网络等资源存在限制且动态的场景，单独的训练过程并不适用，如何利用有限的资源对推理模型进行在线更新值得研究，相关研究领域或可参考感知计算、触觉网络等.

4 结　　语

边缘协同推理具有极大的应用价值，目前，正处于快速发展期，但清晰而又统一的处理方法尚未形成，值得我们重点研究. 本文对边缘协同智能的发展历史进行了简要回顾，对推理过程中涉及到的关键技术进行了归纳整理. 通过对不同关键技术的纵向总结、适用场景分析以及技术间的对比等，重点从动态场景角度提出了边缘协同推理存在的挑战与值得发展的方向. 整体来看，边缘协同推理目前还有极大的发展空间，我们未来的研究工作重点将放在动态场景建模以及动态场景下的边缘协同推理可靠性保障方面.

作者贡献声明：王睿设计了论文框架、调研文献、指导论文写作并修改论文；齐建鹏和陈亮负责文献调研、撰写及修改部分论文；杨龙补充完善论文.

参　考　文　献

[1] David S, David C, Nick J. Top 10 strategic technology trends for 2020[EB/OL]. (2019-10-20)[2022-02-05].https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020

[2] Carrie M, David R, Michael S. The growth in connected IoT devices is expected to generate 79.4ZB of data in 2025, according to a new IDC forecast[EB/OL]. (2019-06-18) [2022-02-15].https://www.businesswire.com/news/home/20190618005012/en/The-Growth-in-Connected-IoT-Devices-is-Expected-to-Generate-79.4ZB-of-Data-in-2025-According-to-a-New-IDC-Forecast

[3] Xiao Yinhao, Jia Yizhen, Liu Chunchi, et al. Edge computing security: State of the art and challenges[J]. Proceedings of the IEEE, 2019, 107(8): 1608−1631

[4] Kevin M, Amir E. AWS customers rack up hefty bills for moving data[EB/OL]. (2019-10-21)[2022-02-15].https://www.theinformation.com/articles/aws-customers-rack-up-hefty-bills-for-moving-data

[5] Jin Hai, Jia Lin, Zhou Zhi. Boosting edge intelligence with collaborative cross-edge analytics[J]. IEEE Internet of Things Journal, 2020, 8(4): 2444−2458

[6] Xiang Chong, Wang Xinyu, Chen Qingrong, et al. No-jump-into-latency in China's Internet! toward last-mile hop count based IP geo-localization[C/OL] //Proc of the 19th Int Symp on Quality of Service. New York: ACM, 2019[2021-03-15].https://doi.org/10.1145/3326285.3329077

[7] Jiang Xiaolin, Shokri-Ghadikolaei H, Fodor G, et al. Low-latency networking: Where latency lurks and how to tame it[J]. Proceedings of the IEEE, 2018, 107(2): 280−306

[8] Shi Weisong, Zhang Xingzhou, Wang Yifan, et al. Edge computing: Status quo and prospect[J]. Journal of Computer Research and Development, 2019, 56(1): 69−89 (in Chinese)
(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)

[9] Zamora-Izquierdo MA, Santa J, Martínez JA, et al. Smart farming IoT platform based on edge and cloud computing[J]. Biosystems Engineering, 2019, 177(1): 4−17

[10] Xiao Wenhua, Liu Bixin, Liu Wei, et al. A review of edge computing for harsh environments[J]. Journal of Command and Control, 2019, 5(3): 181−190 (in Chinese)
(肖文华, 刘必欣, 刘巍, 等. 面向恶劣环境的边缘计算综述[J]. 指挥与控制学报, 2019, 5(3): 181−190)

[11] Stojkoska BLR, Trivodaliev KV. A review of Internet of things for smart home: Challenges and solutions[J]. Journal of Cleaner Production, 2017, 140(3): 1454−1464

[12] Wan Shaohua, Gu Zonghua, Ni Qiang. Cognitive computing and wireless communications on the edge for healthcare service

robots[J]. Computer Communications, 2020, 149(1): 99−106

[13] Lü Huazhang, Chen Dan, Fan Bin, et al. Standardization progress and case analysis of edge computing[J]. Journal of Computer Research and Development, 2018, 55(3): 487−511 (in Chinese)
(吕华章, 陈丹, 范斌, 等. 边缘计算标准化进展与案例分析[J]. 计算机研究与发展, 2018, 55(3): 487−511)

[14] Qi Jianpeng. Awesome edge computing[EB/OL]. (2003-06-02)[2022-03-15]. https://github.com/qijianpeng/awesome-edge-computing#engine

[15] Cheol-Ho H, Blesson V. Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms[J]. ACM Computing Surveys, 2019, 52(5): 1−37

[16] Zeng Peng, Song Chunhe. Edge computing[J]. Communications of China Computer Federation, 2020, 16(1): 8−10 (in Chinese)
(曾鹏, 宋纯贺. 边缘计算[J]. 中国计算机学会通讯, 2020, 16(1): 8−10)

[17] Gao Han, Tian Yulong, Xu Fengyuan, et al. Overview of deep learning model compression and acceleration[J]. Journal of Software, 2021, 32(1): 68−92 (in Chinese)
(高晗, 田育龙, 许封元, 等. 深度学习模型压缩与加速综述[J]. 软件学报, 2021, 32(1): 68−92)

[18] Zhou Zhi, Chen Xu, Li En, et al. Edge intelligence: Paving the last mile of artificial intelligence with edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1738−1762

[19] Li Kenli, Liu Chubo. Edge intelligence: Status quo and prospect[J]. Big Data, 2019, 5(3): 69−75 (in Chinese)
(李肯立, 刘楚波. 边缘智能: 现状和展望[J]. 大数据, 2019, 5(3): 69−75)

[20] Tan Haisheng, Guo Deke, Zhang Chi, et al. Development and challenges of cloud-edge-device collaborative intelligent edge computing[J]. Communications of China Computer Federation, 2020, 16(1): 38−44 (in Chinese)
(谈海生, 郭得科, 张弛, 等. 云边端协同智能边缘计算的发展与挑战[J]. 中国计算机学会通讯, 2020, 16(1): 38−44)

[21] Zhang Xingzhou, Lu Sidi, Shi Weisong. Research on collaborative computing technology in edge intelligence[J]. Artificial Intelligence, 2019, 5(7): 55−67 (in Chinese)
(张星洲, 鲁思迪, 施巍松. 边缘智能中的协同计算技术研究[J]. 人工智能, 2019, 5(7): 55−67)

[22] Wang Xiaofei. Smart edge computing: The bridge from the Internet of everything to the empowerment of everything[J]. People 's Forum·Academic Frontiers, 2020(9): 6−17 (in Chinese)
(王晓飞. 智慧边缘计算: 万物互联到万物赋能的桥梁[J]. 人民论坛·学术前沿, 2020(9): 6−17)

[23] Fan Zhenyu, Wang Yang, Fan Wu, et al. Serving at the edge: An edge computing service architecture based on ICN[J]. ACM Transactions on Internet Technology, 2021, 22(1): 1−27

[24] Jennings A , Copenhagen R V , Rusmin T. Aspects of network edge intelligence[R/OL]. 2001 [2022-03-16]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.6997&rep=rep1&type=pdf

[25] Romaniuk R S. Intelligence in optical networks[G] //Proceedings of SPIE 5125: Proc of the Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments. Bellingham, WA: SPIE, 2003: 17−31

[26] Okagawa T, Nishida K, Yabusaki M. A proposed mobility management for IP-based IMT network platform[J]. IEICE Transactions on Communications, 2005, 88(7): 2726−2734

[27] Liang Ye. Mobile intelligence sharing based on agents in mobile peer-to-peer environment[C] //Proc of the 3rd Int Symp on Intelligent Information Technology and Security Informatics. Piscataway, NJ: IEEE, 2010: 667−670

[28] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84−90

[29] Szegedy C, Liu Wei, Jia Yangqing, et al. Going deeper with convolutions[C/OL] //Proc of the 28th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2015 [2022-03-16]https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html

[30] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size[EB/OL]. (2016-11-04) [2022-03-16]. https://arxiv.org/abs/1602.07360

[31] Cao Yu, Chen Songqing, Hou Peng, et al. FAST: A Fog computing assisted distributed analytics system to monitor fall for Stroke mitigation[C] //Proc of the 10th IEEE Int Conf on Networking, Architecture and Storage. Piscataway, NJ: IEEE, 2015: 2−11

[32] Teerapittayanon S, McDanel B, Kung H T. Distributed deep neural networks over the cloud, the edge and end devices[C] //Proc of the 37th IEEE Int Conf on Distributed Computing Systems. Piscataway, NJ: IEEE, 2017: 328−339

[33] Wang Xiaofei, Han Yiwen, Wang Chenyang, et al. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning[J]. IEEE Network, 2019, 33(5): 156−165

[34] Kang Yiping, Johann H, Gao Cao, et al. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge[J]. ACM SIGARCH Computer Architecture News, 2017, 45(1): 615−629

[35] Li En, Zhou Zhi, and Chen Xu. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy[C] //Proc of the 2018 Workshop on Mobile Edge Communications. New York, ACM, 2018: 31−36

[36] Li Yikai, Zhang Tong, Chen Junlong. Wide twin networks for edge computing applications[J]. Acta Automatica Sinica, 2020, 46(10): 2060−2071 (in Chinese)
(李逸楷, 张通, 陈俊龙. 面向边缘计算应用的宽度孪生网络[J]. 自动化学报, 2020, 46(10): 2060−2071)

[37] Al-Rakhami M, Alsahli M, Hassan M M, et al. Cost efficient edge intelligence framework using docker containers[C] //Proc of the 16th IEEE Int Conf on Dependable, Autonomic and Secure Computing. Piscataway, NJ: IEEE, 2018: 800−807

[38] Al-Rakhami M, Gumaei A, Alsahli M, et al. A lightweight and cost effective edge intelligence architecture based on containerization technology[J]. World Wide Web, 2020, 23(2): 1341−1360

[39] Verbraeken J, Wolting M, Katzy J, et al. A survey on distributed

machine learning[J]. ACM Computing Surveys, 2020, 53(2): 1−33

[40] Chai Tianyou, Yang Tao. Research status and prospects of distributed collaborative optimization[J]. Scientia Sinica Technologica, 2020, 50(11): 1414−1425 (in Chinese)
(杨涛, 柴天佑. 分布式协同优化的研究现状与展望[J]. 中国科学: 技术科学, 2020, 50(11): 1414−1425)

[41] Merenda M, Porcaro C, Iero D. Edge machine learning for AI-enabled IoT devices: A review[J/OL]. Sensors, 2020, 20(9) [2022-03-18]. https://doi.org/10.3390/s20092533

[42] Véstias M P, Duarte R P, de Sousa J T, et al. Moving deep learning to the edge[J/OL]. Algorithms, 2020, 13(5) [2022-03-18]. https://doi.org/10.3390/a13050125

[43] Chen Jiasi, Ran Xukan. Deep learning with edge computing: A review[J]. Proceedings of the IEEE, 2019, 107(8): 1655−1674

[44] Hong Xuehai, Wang Yang. Research on the development and countermeasures of edge computing technology[J]. China Engineering Science, 2018, 20(2): 28−34 (in Chinese)
(洪学海, 汪洋. 边缘计算技术发展与对策研究[J]. 中国工程科学, 2018, 20(2): 28−34)

[45] Hadidi R, Cao Jiashen, Ryoo M S, et al. Toward collaborative inferencing of deep neural networks on Internet-of-things devices[J]. IEEE Internet of Things Journal, 2020, 7(6): 4950−4960

[46] Zhao Zhuoran, Barijough K M, Gerstlauer A. Deepthings: Distributed adaptive deep learning inference on resource-constrained IoT edge clusters[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(11): 2348−2359

[47] Pnevmatikatos D N, Pelcat M, Jung M. Embedded Computer Systems: Architectures, Modeling, and Simulation[M]. Berlin: Springer, 2019

[48] Mao Jiachen, Chen Xiang, Nixon K W, et al. MoDNN: Local distributed mobile computing system for deep neural network[C] //Proc of the 24th Design, Automation Test in Europe Conf Exhibition. Piscataway, NJ: IEEE, 2017: 1396−1401

[49] Shan Nanliang, Ye Zecong, Cui Xiaolong. Collaborative intelligence: Accelerating deep neural network inference via device-edge synergy[J/OL]. Secrrity and Communication Networks, 2020 [2022-03-16]. https://doi.org/10.1155/2020/8831341

[50] Lane N D, Bhattacharya S, Georgiev P, et al. DeepX: A software accelerator for low-power deep learning inference on mobile devices[C/OL] //Proc of the 15th ACM/IEEE Int Conf on Information Processing in Sensor Networks (IPSN). 2016 [2022-04-06]. https://doi.org/10.1109/IPSN.2016.7460664

[51] Zhou Li, Samavatian M H, Bacha A, et al. Adaptive parallel execution of deep neural networks on heterogeneous edge devices [C] //Proc of the 4th ACM/IEEE Symp on Edge Computing. New York: ACM, 2019: 195−208

[52] Jahierpagliari D, Chiaro R, Macii E, et al. CRIME: Input-dependent collaborative inference for recurrent neural networks[J]. IEEE Transactions on Computers, 2020, 70(10): 1626−1639

[53] Zhang Shuai, Zhang Sheng, Qian Zhuzhong, et al. DeepSlicing: Collaborative and adaptive CNN inference with low latency[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 22(9): 2175−2187

[54] Li En, Zeng Liekang, Zhou Zhi, et al. Edge AI: On-demand accelerating deep neural network inference via edge computing[J]. IEEE Transactions on Wireless Communications, Institute of Electrical and Electronics Engineers, 2020, 19(1): 447−457

[55] Zhang Saiqian, Lin Jieyu, Zhang Qi. Adaptive distributed convolutional neural network inference at the network edge with ADCNN[C/OL] //Proc of the 49th Int Conf on Parallel Processing. 2020 [2022-03-18]. https://doi.org/10.1145/3404397.3404473

[56] Shao Jiawei, Zhang Jun. BottleNet++: An end-to-end approach for feature compression in device-edge co-inference systems[C/OL] //Proc of the IEEE Int Conf on Communications Workshops. Piscataway, NJ: IEEE, 2020 [2022-03-18]. https://doi.org/10.1109/ICCWorkshops49005.2020.9145068

[57] Shao Jiawei, Zhang Jun. Communication-computation trade-off in resource-constrained edge inference[J]. IEEE Communications Magazine, 2020, 58(12): 20−26

[58] Avasalcai C, Tsigkanos C, Dustdar S. Resource management for latency-sensitive IoT applications with satisfiability[J/OL]. IEEE Transactions on Services Computing, 2021 [2022-03-18]. https://doi.ieeecomputersociety.org/10.1109/TSC.2021.3074188

[59] Chen Min, Li Wei, Hao Yiyue, et al. Edge cognitive computing based smart healthcare system[J]. Future Generation Computer Systems, 2018, 86(9): 403−411

[60] Hu Diyi, Krishnamachari B. Fast and accurate streaming cnn inference via communication compression on the edge[C] //Proc of the 5th ACM/IEEE Int Conf on Internet of Things Design and Implementation. Piscataway, NJ: IEEE, 2020: 157−163

[61] Hsu K J, Choncholas J, Bhardwaj K, et al. DNS does not suffice for MEC-CDN[C] //Proc of the 19th ACM Workshop on Hot Topics in Networks. New York: ACM, 2020: 212−218

[62] Campolo C, Lia G, Amadeo M, et al. Towards named AI networking: Unveiling the potential of NDN for edge AI[G] //LNCS 12338: Proc of the 19th Int Conf on Ad-Hoc Networks and Wireless. Cham: Springer, 2020: 16−22

[63] Jiang A H, Wong D L K, Canel C, et al. Mainstream: Dynamic stem-sharing for multi-tenant video processing[C] //Proc of the 2018 USENIX Annual Technical Conf. New York: ACM, 2018: 29−42

[64] Mhamdi E, Guerraoui R, Rouault S. On the robustness of a neural network[C] //Proc of the 36th IEEE Symp on Reliable Distributed Systems. Piscataway, NJ: IEEE, 2017: 84−93

[65] Yousefpour A, Devic S, Nguyen B Q, et al. Guardians of the Deep Fog: Failure-resilient DNN inference from edge to cloud[C] //Proc of the 1st Int Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things. New York: ACM, 2019: 25−31

[66] Hu Chuang, Bao Wei, Wang Dan, et al. Dynamic adaptive DNN surgery for inference acceleration on the edge[C] //Proc of the 38th IEEE Conf on Computer Communications. Piscataway, NJ: IEEE, 2019: 1423−1431

[67] Song Han, Mao Huizi, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[EB/OL]. (2016-02-15) [2022-03-18]. https://arxiv.org/abs/1510.00149

[68] Masana M, van de Weijer J, Herranz L, et al. Domain-adaptive deep network compression[C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 22−29

[69] Courbariaux M, Bengio Y, David J P. BinaryConnect: Training deep neural networks with binary weights during propagations[C] //Proc of the 28th Int Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2015: 3123−3131

[70] Gholami A, Kim S, Zhen Dong, et al. A survey of quantization methods for efficient neural network inference[J]. arXiv preprint, arXiv: 2103.13630, 2021

[71] Cao Qingqing, Irimiea A E, Abdelfattah M, et al. Are mobile DNN accelerators accelerating DNNs?[C] //Proc of the 5th Int Workshop on Embedded and Mobile Deep Learning. New York: ACM, 2021: 7−12

[72] Guo Kaiyuan, Song Han, Song Yao, et al. Software-hardware codesign for efficient neural network acceleration[J]. IEEE Micro, 2017, 37(2): 18−25

[73] Guo Kaiyuan, Li Wenshuo, Zhong Kai, et al. Neural network accelerator comparison[EB/OL]. (2018-01-01) [2022-12-26]. https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator tsinghua.edu.cn/project.html

[74] Li Hao, Kadav A, Durdanovic I, et al. Pruning filters for efficient convnets[J]. arXiv preprint, arXiv: 1608.08710, 2017

[75] Luo Jianhao, Zhang Hao, Zhou Hongyu, et al. ThiNet: Pruning cnn filters for a thinner net[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(10): 2525−2538

[76] He Yihui, Zhang Xianyu, Sun Jian. Channel pruning for accelerating very deep neural networks[C] //Proc of the 16th IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 1398−1406

[77] Hu Hengyuan, Peng Rui, Tai Y W, et al. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures [J]. arXiv preprint, arXiv: 1607.03250, 2016

[78] Wen Wei, Wu Chunpeng, Wang Yandan, et al. Learning structured sparsity in deep neural networks[C] //Proc of the 30th Int Conf on Neural Information Processing Systems. New York: ACM, 2016: 2082−2090

[79] Chen Hanting, Wang Yunhe, Xu Chang, et al. Data-free learning of student networks[C] //Proc of the 17th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 3513−3521

[80] Niu Wei, Ma Xiaolong, Lin Sheng, et al. PatDNN: Achieving real-time DNN execution on mobile devices with pattern-based weight pruning[C] //Proc of the 25th Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2020: 907−922

[81] Qin Haotong, Gong Ruihao, Liu Xianglong, et al. Binary neural networks: A survey [J]. Pattern Recognition, 2020, 105(9): 107281

[82] Lu Ye, Gong Cheng, Li Tao. Challenges and opportunities of deep neural network compression automation[J]. China Computer Society Communications, 2021, 17(3): 41−47 (in Chinese)
(卢冶, 龚成, 李涛. 深度神经网络压缩自动化的挑战与机遇[J]. 中国计算机学会通讯, 2021, 17(3): 41−47)

[83] Hubara I, Courbariaux M, Soudry D, et al. Binarized neural networks[C] //Proc of the 30th Int Conf on Neural Information Processing Systems. New York: ACM, 2016: 4114−4122

[84] Li Fengfu, Liu Bin. Ternary weight networks[J]. arXiv preprint, arXiv: 1605.04711, 2016

[85] Alemdar H, Leroy V, Prost-Boucle A, et al. Ternary neural networks for resource-efficient AI applications[C] //Proc of the 30th Int Joint Conf on Neural Networks. Piscataway, NJ: IEEE, 2017: 2547−2554

[86] Chen Yao, Zhang Kang, Gong Cheng, et al. T-DLA: An open-source deep learning accelerator for ternarized DNN models on embedded FPGA[C] //Proc of the 14th IEEE Computer Society Annual Symp on VLSI. Piscataway, NJ: IEEE, 2019: 13−18

[87] Zhou Shuchuang, Wu Yuxin, Ni Zekun, et al. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients[J]. arXiv preprint, arXiv: 1606.06160, 2018

[88] Wang Peisong, Hu Qinghao, Zhang Yifan, et al. Two-step quantization for low-bit neural networks[C] //Proc of the 31st IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 4376−4384

[89] Jung Sangli, Son Changyong, Lee Seohyung, et al. Learning to quantize deep networks by optimizing quantization intervals with task loss[C] //Proc of the 32nd IEEE/CVF Conf on Computer Vision and Pattern Recognition . Piscataway, NJ: IEEE, 2019: 4345−4354

[90] Gong Cheng, Li Tao, Lu Ye, et al. μL2Q: An ultra-low loss quantization method for DNN compression[C/OL] //Proc of the Int Joint Conf on Neural Networks. Piscataway, NJ: IEEE, 2019 [2022-04-07]. https://doi.org/10.1109/IJCNN.2019.8851699

[91] Ge Daohui, Li Hongsheng, Zhang Liang, et al. A review of lightweight neural network architecture[J]. Journal of Software, 2020, 31(9): 2627−2653 (in Chinese)
(葛道辉, 李洪升, 张亮, 等. 轻量级神经网络架构综述[J]. 软件学报, 2020, 31(9): 2627−2653)

[92] Shi Lei, Feng Shi, Zhu Zhifang. Functional hashing for compressing neural networks[J]. arXiv preprint, arXiv: 1605.06560, 2016

[93] Wu Junru, Wang Yue, Wu Zhenyu, et al. Deep k-means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions[C] //Proc of the 35th Int Conf on Machine Learning PMLR. New York: ACM, 2018: 5363−5372

[94] Xu Xiaowei, Lu Qing, Wang Tianchen, et al. Efficient hardware implementation of cellular neural networks with incremental quantization and early exit[J]. ACM Journal on Emerging Technologies in Computing Systems, 2018, 14(4): 1−20

[95] Li Yuhong, Hao Cong, Zhang Xiaofan, et al. EDD: Efficient differentiable DNN architecture and implementation co-search for embedded AI solutions[C/OL] //Proc of the 57th ACM/IEEE Design Automation Conf. New York: ACM, 2020 [2022-04-07]. https://doi.org/10.1109/DAC18072.2020.9218749

[96] Aimar A, Mostafa H, Calabrese E, et al. NullHop: A flexible convolutional neural network accelerator based on sparse representations of feature maps[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(3): 644−656

[97] Sebastian A, Le Gallo M, Khaddam-Aljameh R, et al. Memory devices and applications for in-memory computing[J]. Nature Nanotechnology, 2020, 15(7): 529−544

[98] Song Zhuoran, Fu Bangqi, Wu Feiyang, et al. DRQ: Dynamic region-based quantization for deep neural network acceleration[C] //Proc of the 47th ACM/IEEE Annual Int Symp on Computer Architecture. New York: ACM, 2020: 1010−1021

[99] Yang Yixiong, Yuan Zhe, Su Fang, et al. Multi-channel precision-sparsity-adapted Inter-frame differential data Codec for video neural network processor[C] //Proc of the 33rd ACM/IEEE Int Symp on Low Power Electronics and Design. New York: ACM, 2020: 103−108

[100] Tang Yibin, Wang Ying, Li Huawei, et al. MV-Net: Toward real-time deep learning on mobile GPGPU systems[J]. ACM Journal on Emerging Technologies in Computing Systems, 2019, 15(4): 1−25

[101] Chen Shengbo, Shen Cong, Zhang Lanxue, et al. Dynamic aggregation for heterogeneous quantization in federated learning[J]. IEEE Transactions on Wireless Communications, 2021, 20(10): 6804−6819

[102] Teerapittayanon S, McDanel B, Kung H T. BranchyNet: Fast inference via early exiting from deep neural networks[C] //Proc of the 23rd Int Conf on Pattern Recognition. Piscataway, NJ: IEEE, 2016: 2464−2469

[103] Lo C, Su YY, Lee CY, et al. A dynamic deep neural network design for efficient workload allocation in edge computing[C] //Proc of the 35th 2017 IEEE Int Conf on Computer Design. Piscataway, NJ: IEEE, 2017: 273−280

[104] Wang Zizhao, Bao Wei, Yuan Dong, et al. SEE: Scheduling early exit for mobile DNN inference during service outage[C] //Proc of the 22nd Int ACM Conf on Modeling, Analysis and Simulation of Wireless and Mobile Systems. New York: ACM, 2019: 279−288

[105] Wang Zizhao, Bao Wei, Yuan Dong, et al. Accelerating on-device DNN inference during service outage through scheduling early exit[J]. Computer Communications, 2020, 162(10): 69−82

[106] Scarpiniti M, Baccarelli E, Momenzadeh A, et al. DeepFogSim: A toolbox for execution and performance evaluation of the inference phase of conditional deep neural networks with early exits atop distributed Fog platforms[J/OL]. Applied Sciences, 2021, 11(1) [2022-03-18]. https://doi.org/10.3390/app11010377

[107] Su Xiao. EasiEI simulator[CP/OL]. [2022-03-18]. https://gitlab.com/Mirrola/ns-3-dev/-/wikis/EasiEI-Simulator

[108] Park E, Kim D, Kim S, et al. Big/little deep neural network for ultra low power inference[C] //Proc of the Int Conf on Hardware/Software Codesign and System Synthesis. Piscataway, NJ: IEEE, 2015: 124−132

[109] Putra T A, Leu J S. Multilevel Neural network for reducing expected inference time[J]. IEEE Access, 2019, 7(11): 174129−174138

[110] Taylor B, Marco V S, Wolff W, et al. Adaptive deep learning model selection on embedded systems[J]. ACM SIGPLAN Notices, 2018, 53(6): 31−43

[111] Shu Guansheng, Liu Weiqing, Zheng Xiaojie, et al. IF-CNN: Image-aware inference framework for cnn with the collaboration of mobile devices and cloud[J]. IEEE Access, 2018, 6(10): 68621−68633

[112] Stamoulis D, Chin T W, Prakash A K, et al. Designing adaptive neural networks for energy-constrained image classification[C] //Proc of the Int Conf on Computer-Aided Design. New York: ACM, 2018: 1−8

[113] Song Mingcong, Zhong Kan, Zhang Jiaqi, et al. In-Situ AI: Towards autonomous and incremental deep learning for IoT systems[C] //Proc of the 24th IEEE Int Symp on High Performance Computer Architecture. Piscataway, NJ: IEEE, 2018: 92−103

[114] Zhang Li, Han Shihao, Wei Jianyu, et al. nn-Meter: Towards accurate latency prediction of deep-learning model inference on diverse edge devices[C] //Proc of the 19th Annual Int Conf on Mobile Systems, Applications, and Services. New York: ACM, 2021: 81−93

[115] Yue Zhifeng, Zhu Zhixiang, Wang Chuang, et al. Research on big data processing model of edge-cloud collaboration in cyber-physical systems[C] //Proc of the 5th IEEE Int Conf on Big Data Analytics. Piscataway, NJ: IEEE, 2020: 140−144

[116] Wang Huitian, Cai Guangxing, Huang Zhaowu, et al. ADDA: Adaptive distributed DNN inference acceleration in edge computing environment[C] //Proc of the 25th Int Conf on Parallel and Distributed Systems. Piscataway, NJ: IEEE, 2019: 438−445

[117] Chen Liang, Qi Jiapeng, Su Xiao, et al. REMR: A reliability evaluation method for dynamic edge computing network under time constraints[J]. arXiv preprint, arXiv: 2112.01913, 2021

[118] Long Saiqin, Long Weifan, Li Zhetao, et al. A game-based approach for cost-aware task assignment with QoS constraint in collaborative edge and cloud environments[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(7): 1629−1640

[119] Yang Bo, Cao Xuelin, Li Xiangfan, et al. Mobile-edge-computing-based hierarchical machine learning tasks distribution for IIoT[J]. IEEE Internet of Things Journal, 2020, 7(3): 2169−2180

[120] Fang Yihao, Jin Ziyi, Zheng Rong. TeamNet: A collaborative inference framework on the edge[C] //Proc of the 39th IEEE Int Conf on Distributed Computing Systems. Piscataway, NJ: IEEE, 2019: 1487−1496

[121] Fang Yihao, Shalmani SM, Zheng Rong. CacheNet: A model caching framework for deep learning inference on the edge[J]. 2020 (2020-07-03)[2022-03-17]. arXiv preprint, arXiv: 2007.01793, 2020

[122] Tan Chao, Zhang Jingxuan, Wang Tiexin, et al. Uncertainty in complex software systems[J]. Journal of Software, 2021, 32(7): 1926−1956 (in Chinese)
(檀超, 张静宣, 王铁鑫, 等. 复杂软件系统的不确定性[J]. 软件学报, 2021, 32(7): 1926−1956)

[123] Song Chunhe, Zeng Peng, Yu Haibin. Industrial Internet intelligent manufacturing edge computing: Current situation and challenges[J]. ZTE Technology, 2019, 25(3): 50−57 (in Chinese)
(宋纯贺, 曾鹏, 于海斌. 工业互联网智能制造边缘计算: 现状与挑战[J]. 中兴通讯技术, 2019, 25(3): 50−57)

[124] Chen Chao, Zhang Daqing, Wang Yasha, et al. Enabling Smart Urban Services with GPS Trajectory Data[M]. Berlin: Springer, 2021

[125] Huang Qianyi, Li Zhiyang, Xie Wentao, et al. Edge computing in smart home[J]. Journal of Computer Research and Development, 2020, 57(9): 1800−1809 (in Chinese)

(黄倩怡, 李志洋, 谢文涛, 等. 智能家居中的边缘计算[J]. 计算机研究与发展, 2020, 57(9): 1800−1809)

[126] Li Xian, Bi Suzhi, Wang Hui. Optimizing resource allocation for joint AI model training and task inference in edge intelligence systems[J]. IEEE Wireless Communications Letters, 2021, 10(3): 532−536

[127] Trivedi A, Wang Lin, Bal H, et al. Sharing and caring of data at the edge[C/OL]//Proc of the 3rd USENIX Workshop on Hot Topics in Edge Computing. Berkeley, CA: USENIX Association, 2020[2022-04-06]. https://www.usenix.org/conference/hotedge20/presentation/trivedi

[128] Richins D, Doshi D, Blackmore M, et al. AI tax: The hidden cost of AI data center applications[J]. ACM Transactions on Computer Systems, 2021, 37(1-4): 1-32

Wang Rui, born in 1975. PhD, professor. Senior member of CCF. His main research interests include IoT, edge intelligence and smart healthcare.

王　睿, 1975 年生.博士, 教授. CCF 高级会员. 主要研究方向为物联网、边缘智能和智慧医疗.

Qi Jianpeng, born in 1992. PhD candidate. Student member of CCF. His main research interests include edge intelligence and resource management.

齐建鹏, 1992 年生.博士研究生.CCF 学生会员. 主要研究方向为边缘智能和资源管理.

Chen Liang, born in 1997. Master candidate. His main research interest includes edge intelligence and reliability.

陈　亮, 1997 年生.硕士研究生.主要研究方向为边缘智能与可靠性.

Yang Long, born in 1999. Master candidate. His main research interest includes lightweight models and methods

杨　龙, 1999 年生.硕士研究生.主要研究方向为轻量级模型与方法.