面向智能物联网的资源高效模型推理综述

袁 牧 张 兰 姚云昊 张钧洋 罗溥晗 李向阳
(中国科学技术大学计算机科学与技术学院 合肥 230026)

摘 要 从智慧城市到工业自动化,智能物联网在越来越多的场景中得到了广泛应用.模型推理作为实现智能决策和响应的核心技术,在智能物联网系统中扮演着举足轻重的角色.然而,智能物联网设备通常在计算能力、通信带宽、内存容量和电池寿命等资源上高度受限.这使得智能物联网中的模型推理资源开销成为一个关键技术挑战.本综述总结了在智能物联网场景中优化模型推理资源开销的相关技术,对当前在智能物联网应用中使用的主流模型推理优化技术进行概述,并深入分析它们在资源效率方面的优势和不足.本文从推理涉及的三大模块(传感器数据、智能模型、物联网硬件)和五类关键资源的角度出发设计新的技术分类,并首次提出了一套针对智能物联网模型推理的通用的优化流程,能够帮助相关研发人员定位和优化推理效率瓶颈.最后,本文讨论了智能物联网推理效率相关的四个未来研究方向.

关键词 智能物联网;模型推理;资源效率;输入过滤;协同推理
中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2024.02247

Resource-Efficient Model Inference for AIoT: A Survey

YUAN Mu ZHANG Lan YAO Yun-Hao ZHANG Jun-Yang LUO Pu-Han LI Xiang-Yang
(*School of Computer Science and Technology, University of Science and Technology of China, Hefei* 230026)

Abstract From smart cities to industrial automation, AIoT is widely used in more and more scenarios. Model inference, as the core technology for realizing intelligent decision-making and response, plays a pivotal role in AIoT systems. However, AIoT devices are usually highly constrained in resources such as computing, communication, memory, and battery. This makes model inference resource overhead in AIoT a key technical challenge. This review summarizes related technologies for optimizing model inference resource overhead in AIoT scenarios, provides an overview of mainstream model inference optimization techniques currently used in AIoT applications, and deeply analyzes their advantages and disadvantages in terms of resource efficiency. This paper designs a new taxonomy, which is classified from the three modules involved in inference (sensor data, intelligent model, IoT hardware) and five key resources, and proposes the first general optimization workflow for AIoT model inference, which can help relevant R&D personnel locate and optimize inference efficiency bottlenecks. Finally, this paper discusses four future research directions related to the inference efficiency of AIoT.

Keywords Artificial Intelligent of Things; model inference; resource efficiency; input filtering; collaborative inference

---

收稿日期:2023-10-20;在线发布日期:2024-06-28.本课题得到国家重点研发计划项目(2021ZD0110400)、科技创新 2030-"量子通信与量子计算机"重大项目(2021ZD0302900)、国家自然科学基金项目(62132018,62231015,623B2093)、浙江省"尖兵""领雁"研发攻关计划(2023C01029,2023C01143)资助.**袁 牧**,博士研究生,研究方向为网络系统中的模型推理. E-mail: ym0813@mail.ustc.edu.cn. **张 兰**,博士,教授,研究领域为移动计算、隐私保护、数据共享和交易. **姚云昊**,博士研究生,研究方向为端边系统中数据隐私和安全. **张钧洋**,博士研究生,研究方向为异构网络中高效神经网络推理. **罗溥晗**,博士研究生,研究方向为物联网场景中自适应模型压缩和推理加速. **李向阳**(通信作者),博士,教授,ACM/IEEE 会士,ACM 杰出科学家,研究领域为智能物联网、隐私与安全、数据共享和交易. E-mail: xiangyangli@ustc.edu.cn.

1 引 言

人工智能(Artificial Intelligence,AI)和物联网(Internet of Things,IoT)[1]的融合产生了智能物联网(AIoT).人工智能技术为物联网赋予了自动化分析数据的智能算法,物联网则为人工智能输送了可供分析的海量传感器数据.AIoT 将智能算法与庞大的互联设备网络相融合,在诸多领域实现了创新性应用,包括智慧城市、工业自动化、医疗看护、智慧家居、智能穿戴设备等.据 GMI 统计预估[2],全球 AIoT 市场将以超过 20%的复合率增长,在 2032 年达到 250 亿美元.金山云发布的中国智能物联网白皮书[3]也预计中国的物联网连接数将在 2025 年达到 200 亿个,华为全球产业愿景报告[4]预测全球物联网设备数将在 2025 年达到 1000 亿个.随着物联网规模的增大,AIoT 的研究和应用也将日益深入,未来有着广阔的发展前景.

人工智能算法、模型在完成训练后,部署到物联网设备上对传感器采集的数据进行处理的过程称为模型推理(Model Inference).模型推理作为实现智能决策和响应的核心技术,在 AIoT 中扮演着举足轻重的角色.在 AIoT 场景中,通常有三种模型推理的部署方式,即端侧推理、边侧推理(计算卸载)和模型切分推理,如图 1 所示.端侧推理,顾名思义,是将全部的智能模型部署在数据采集设备端,例如摄像头上进行行人检测.其优点在于无需进行数据传输,但会导致端侧计算负载较重,带来过高的能耗和延迟.边侧推理,是将推理计算任务全部卸载到边缘服务器上,例如运动传感器将信号发送给边缘服务器进行人物行为识别.由于计算从端侧卸载到边侧,这种部署方式能够降低端侧能耗,但也引入较高的通信开销.模型切分推理则是权衡了上述两种部署方式,通过将深度神经网络(Deep Neural Network,DNN)模型切分为两部分(或多个部分),端侧计算前半部的推理并将中间结果传输给边缘服务器,在边侧完成后半部的推理计算.在合适的切分策略下,这种方式能够在一定的通信开销下显著降低推理延迟和端侧能耗.

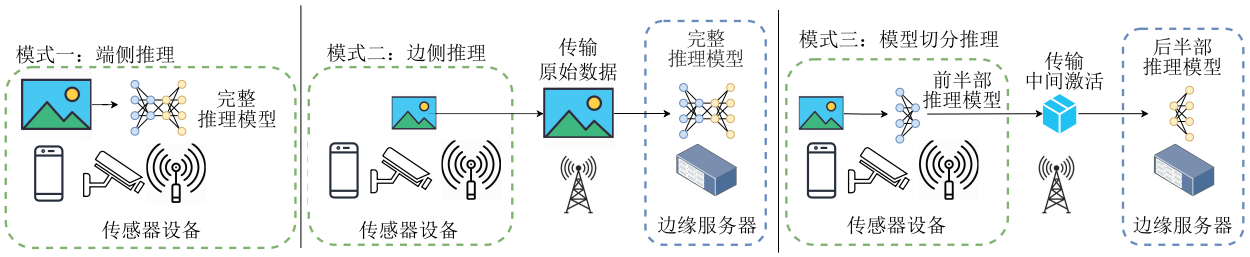

图 1 智能物联网中三种典型的模型推理部署模式:端侧推理、边侧推理以及模型切分推理

资源开销问题是模型推理面临的一项挑战.随着人工智能模型能力的不断提高,高精度模型的参数量也日益增大.例如流行的计算机视觉模型 ResNet-50 有 2.3 亿个参数,基于视觉 Transformer 架构的 ViTLarge 模型则有 3.3 亿个参数,开源的语音识别模型 DeepSpeech 也拥有 1.4 亿参数.而智能物联网设备通常具有资源受限的特点,如有限的计算能力、通信带宽、内存容量和电池寿命.这使得智能物联中的模型推理资源开销成为一个关键技术挑战.

具体地,我们主要关注如下五类开销:

(1)时间.包括端到端响应时延和多处理器累加的计算时间.端到端延迟开销直接影响着用户的 AIoT 服务体验,例如智能音箱的语音助手(例如 Siri 和 Google Assistant)服务一般能够接受的延迟是在 500 ms 范围内.而累加计算时间则决定了 AIoT 服务的运维成本,包括多处理器硬件以及持续工作的能源成本.

(2)内存.智能物联网设备可分为传感器节点和边缘计算节点.① 传感器节点.对于低端传感器,如温度湿度传感器等,其内存一般在几 KB 到几十 KB 之间,而对于智能家居传感器,其内存通常在几 MB 到几十 MB 之间.工业物联网传感器通常需要处理更复杂的场景,其内存可能超过 100 MB;② 边缘计算节点.AIoT 边缘节点通常具有更强大的功能,如视频解码和智能数据处理.其内存一般在几百 MB 到几 GB 之间,一些高端深度学习工作站可能装配拥有几 GB 到十几 GB 显存的独立显卡.模型推理的内存开销直接影响 AIoT 服务的硬件部署成本,从另一个角度,这决定着相同硬件资源下能够部署的模型性能.

(3)带宽.带宽指数据传输速率的大小,通常用每秒传输的数据量来衡量,单位为千比特/秒(Kbps)或兆比特/秒(Mbps).智能物联网中设备的带宽一般情况下是相对较低的,这是因为许多设备被设计

为资源受限的嵌入式系统，其通信需求相对简单而且功耗要很低. 为了提高能源效率，物联网设备通常会使用低功耗通信技术，如 LoRaWAN、NB-IoT 等，这些技术在提供相对较低带宽的同时，可以降低设备的能耗. 对于简单传感器节点如温湿度传感器，通常只需要几 Kbps 到几百 Kbps，对于智能家居传感器如摄像头和音箱，由于涉及多媒体传输，其带宽一般在几 Mbps 到几十 Mbps 之间. 工业物联网设备通常需要更高的带宽，一些复杂传感器的数据传输带宽可能会超过几千 Mbps.

(4) 能量. AIoT 设备通常需要长时间运行，且部分嵌入式系统由电池供电，因此为了延长设备的使用寿命和减少维护频率，物联网设备通常采用低功耗设计. 低端传感器节点通常只执行简单的任务，如收集环境数据并发送给边缘设备. 这些设备能耗通常较低，可以实现长时间的运行，通常以微瓦级的功耗计算. 工业物联网设备通常需要处理更大量的数据和更复杂的场景，但同样会优化设计以降低功耗. 其功耗通常在几百毫瓦到几瓦之间，有时甚至更高，具体取决于设备的复杂性和通信需求.

(5) 存储. 物联网设备的存储一般较小，通常用于存储少量的配置信息、设备状态等必要的数据. 低端传感器节点通常只需要存储少量的配置信息和传感器数据. 其存储容量通常在几到几十 KB 之间. 智能家居设备可能需要存储一些音频、视频和图像数据，用于本地处理或传输到服务器端. 其存储容量通常在几到几十 MB 之间. 物联网设备通常会定期将采集的数据传输到云端或边缘服务器进行存储和处理，以释放设备的本地存储空间.

为了实现高效的智能决策和响应，我们必须寻求推理精度和资源开销的最佳平衡点，以确保在满足性能要求的同时尽可能降低能源消耗，延长设备的使用寿命. 本综述旨在探讨在智能物联网场景中优化模型推理资源开销的相关技术. 我们将对当前在智能物联网应用中使用的主流模型推理优化技术进行概述，并深入分析它们在资源效率方面的优势和不足. 通过对这些方法进行比较和评估，我们希望为智能物联网应用中的模型推理资源管理提供全面的认识，为未来设计高效的智能物联网系统提供有益的参考和指导. 在智能物联网时代到来之际，解决模型推理资源开销问题是我们所面临的紧迫任务. 只有通过创新的技术手段和综合的资源管理策略，我们才能使智能物联网真正成为一个高效、智能、可持续发展的现实. 本综述的研究内容将有助于加速智能物联网技术的演进，推动其在各行各业的广泛应用.

如表 1 所示，本文以智能物联网中模型推理的流程上的三个模块(传感器数据、智能模型、物联网硬件)为划分角度，将现有的优化技术根据其主要优化的资源(时间、内存、带宽、能量、存储)分为八类. 从传感器数据的角度，我们总结了优化时间的输入过滤技术、优化带宽的数据编码技术以及优化存储的存储管理技术. 从智能模型的调度，我们介绍优化时间的自适应配置技术和优化内存的模型压缩技术. 从物联网硬件的角度，我们将相关方法分类为优化时间的计算图优化技术、优化带宽的协同推理技术以及优化能量的加速器技术. 考虑到很多技术都不是对资源进行孤立优化的，例如模型压缩能够同时优化时间、内存、能量和存储资源，我们将总结的八类技术对五类资源的优化效果列于表 1 中. 我们希望本文提出的方法分类能够帮助到相关研究人员，在遇到特定场景中资源瓶颈时能够参考本文的方法加以优化. 最后，我们分析了智能物联网模型推理在资源效率方面仍存在的挑战以及一些可能的未来研究方向.

表 1　智能物联网推理流程的三个模块(数据-模型-硬件)中存在八种优化方法，每种方法标注了所优化的资源

模块	方法	资源				
		时间	内存	带宽	能量	存储
传感器数据	输入过滤	★		✓	✓	
	数据编码	✓		★		
	存储管理	✓				★
智能模型	自适应配置	★				
	模型压缩	✓	★		✓	✓
物联网硬件	计算图优化	★	✓		✓	
	协同推理	✓		★		
	加速器		✓		★	

注：★表示主要优化的资源.

1.1　相关综述

由于本文涉及人工智能和通信网络两个领域，我们将相关综述按领域分为两类介绍.

(1) 人工智能通用场景. Cheng 等人[5]介绍了深度神经网络的模型压缩和加速技术，并将相关技术划分为四大类，即参数剪枝和量化、低秩分解、紧凑卷积过滤器以及知识蒸馏. Hoefler 等人[6]则针对深度模型的稀疏性介绍了提高推理和训练效率的相关技术，包括实现模型稀疏性的训练策略以及降低推理阶段资源开销的方法. 此类工作涉及的技术是本文第 4.2 节的一部分，本文选择性介绍一些针对智能物联网场景的模型压缩和加速方法. Han 等人[7]将能够在推理时根据输入数据动态调整结构或

参数的神经网络称为“动态神经网络”(Dynamic Neural Networks),并从三个角度-样本、空间、时间-介绍了相关方法. Matsubara 等人[8]则专门为动态神经网络中的拆分计算和早退方法撰写了更为细致的综述. 与本文专注于模型推理效率不同,Han 等人[7]的工作介绍了动态神经网络在除了精度-效率权衡以外多个角度上的优势,包括表达能力和可解释性. Thiruvathukal 等人[9]撰写了关于提高计算机视觉效率的书籍,广泛介绍了包括量化、剪枝、知识蒸馏、硬件加速、神经网络架构优化等针对视觉模型的技术. 本文考虑广泛的 AIoT 场景,不局限于计算机视觉任务.

(2) 移动计算、边缘计算、物联网场景. Chen 等人[10]以及 Murshed 等人[11]调研了边缘计算和深度学习的结合,包括在边缘侧的深度学习典型应用(主要是视觉和自然语言处理任务)、模型轻量化以及硬件加速技术、结合端边设备进行协同推理的方法,以及跨多个边缘节点进行分布式模型训练的机制. Zhang 等人[12]介绍了智能物联网的架构和应用,包括计算机视觉类(图像分类、物体检测、物体追踪、语义分割、人脸识别、行人重识别、人体姿态识别、即时定位与地图构建等)、音频分析类(语音识别、说话人识别等)、自然语言处理类(机器翻译等)以及多模态分析类应用. 同时该综述也介绍了智能物联网相关的机器学习技术,包括无监督和半监督学习、迁移学习和域自适应、零样本和小样本学习、强化学习以及联邦学习等技术. 相较于此工作,本文聚焦于智能物联网场景中的模型推理的资源效率问题,旨在提供一个针对性的技术综述,而非全面的介绍. 吴吉义等人[13]总结了 AIoT 技术背景和应用场景,提出了一种云边端融合 AIoT 架构,并介绍了包括数据采集、事件处理及协同、云边端融合、安全及隐私保护等方面的研究现状. 杨铮等人[14]总结了在边缘计算场景下针对视频流上的模型推理技术,从端侧、端边/云协同、边/云侧三个角度介绍了相关方法. 本文在视频分析相关方法上与该工作有一定的重叠,但与该工作专注视频分析应用不同,本文涵盖了智能物联网中的各种应用场景,且提出了不同的技术分类方法. Liu 等人[15]对智能物联网系统中端侧训练/推理、分布式训练/推理以及应用进行了综述整理,介绍了与模型训练和推理相关的资源-精度权衡的技术. 本文在模型推理相关方法上与该工作相交,但不同之处在于本文并不讨论训练相关的技术,且在推理方面给出了更丰富和广泛的技术介绍.

表 2 总结了相关综述和本文的对比,总而言之,本文第一次系统性地调研针对通用物联网场景下的、优化模型推理效率的技术.

表 2 相关综述对比

文献	年份	总结	是否针对通用物联网场景	是否针对推理流程	效率优化角度		
					数据	模型	硬件
Cheng 等人[5]	2017	深度神经网络压缩		√		√	
Hoefler 等人[6]	2021	深度神经网络稀疏化				√	
Han 等人[7]	2021	动态神经网络			√	√	
Matsubara 等人[8]	2022	拆分计算和早退		√		√	
Thiruvathukal 等人[9]	2022	计算机视觉模型加速				√	√
Chen 等人[10],Murshed 等人[11]	2019,2021	边缘计算+深度学习	√			√	√
Zhang 等人[12]	2020	智能物联网架构和应用	√				
吴吉义等人[13]	2021	智能物联网云边端融合	√				√
杨铮等人[14]	2022	边缘计算+视频分析		√	√	√	
Liu 等人[15]	2023	智能物联网训练推理及应用	√			√	√
本文	2023	智能物联网模型推理	√	√	√	√	√

1.2 调研方法

我们在 Google Scholar、DBLP、IEEE Xplore 和 ACM Digital Library 上进行关键词搜索来收集与智能物联网推理相关论文,搜索了包括如下关键词(以及对应的中文翻译):

(1) (Inference|Neural Network)+(Embedded|IoT|Mobile|Edge);

(2) (Inference|Neural Network)+Resource Efficiency;

(3) Neural Network+(Query|Analytics|Storage);

(4) (Neural Network|Model)+(Compression|Binary|Scheduling|Acceleration).

我们基于是否同时满足以下条件对论文进行选择:

(1) 优化目标是否是模型推理的资源效率?

(2) 所用技术能否应用于智能物联网场景?

除了通过关键词检索相关论文以外,我们还通过与合作企业(包括蔚来汽车、大全能源、三一重工等)交流、调研实际系统等方式,总结业界对于智能

物联网模型推理的技术需求以及常用优化方法.

2　智能物联网模型推理

2.1　场景特点

AIoT 场景通常涉及大量的传感器设备和边缘节点，这些设备采集的数据需要进行实时或近实时的推理和处理. 相较于数据中心和移动计算场景，AIoT 场景中的特色应用包括智慧城市、智慧家居、工业自动化、智能监控监测等. 表 3 对比总结了不同网络场景下应用、数据来源、计算和通信以及模型推理的主要挑战.

表 3　不同网络场景对比

场景	示例特色应用	任务数据	计算硬件	通信环境	模型推理关键挑战
数据中心	大语言模型问答 大数据分析挖掘	大规模数据集	高性能服务器	千兆/万兆高速局域网	高吞吐、鲁棒性
移动计算	增强/虚拟现实 位置感知和导航	用户及设备本地数据	移动设备	移动网络/WiFi	低功耗、隐私保护
智能物联网	智慧城市/家居 工业自动化	传感器实时数据	嵌入式设备 边缘服务器	低功耗广域网 蓝牙/Zigbee/NFC	低延迟(实时性) 异构性、可扩展性

(1) 设备异构性. 数据中心中大规模使用的高性能服务器为了可靠性和扩展性通常是同构的，而移动计算场景的一个应用一般只涉及单一移动设备，例如运行位置感知和导航的移动手机以及虚拟现实眼镜. 很不同的，智能物联网的应用往往都涉及大量高度异构的设备，从数据采集层面，智能物联网设备连接各种不同类型的传感器，如摄像头、温度传感器、加速度传感器等，这些传感器产生的数据格式和数据模态是多样的. 从硬件方面，智能物联网中的设备来自各种不同的厂商，因此其硬件架构以及处理能力可能差异很大. 软件方面，AIoT 设备可能使用各种不同的操作系统，如 Linux、实时操作系统(RTOS)、嵌入式系统等.

(2) 离线可推理. 在智能物联网场景中，设备可能会遇到网络不稳定、断网或处于远程区域等情况，这时无法依赖云端的服务和数据传输. 因此，为了保证智能设备的功能稳定性和可用性，在一些应用场景下，设备需要具备在离线状态下进行推理的能力. 这种能力对于一些关键应用场景，如远程地区监控、无人驾驶车辆、智能家居等来说尤为重要，因为它们需要能够在任何环境下独立地做出决策和行动.

(3) 动态自适应. 智能物联网环境中的数据通常是实时生成的，并且可能受到多种因素的影响而不断变化，例如环境条件、设备状态、用户行为等. 传感器采集的数据可能会受到概念漂移(Concept Shift)的影响，即数据分布随时间发生变化，为了保持模型的性能和准确性，模型推理需要能够自适应地识别和适应这些变化.

2.2　技术挑战

由于 AIoT 环境的特性，包括有限的计算资源、异构设备、通信带宽和能耗限制等，在智能物联网场景下的模型推理，面临如下几个与资源效率相关的挑战：

(1) 实时分析推理算不完. 智能物联网应用通常需要在实时或近实时的情况下对传感器数据进行分析和推理. 然而，现代深度学习模型通常非常复杂，需要大量计算资源来进行推理，这在算力有限的物联网设备上是一个挑战. 实时推理涉及高效的模型设计和优化以及硬件加速和高度优化的推理引擎，以确保模型能够在有限的时间内完成推理过程.

(2) 内存有限模型放不进. 智能物联网设备的内存通常受到限制，特别是边缘节点设备. 复杂的深度学习模型可能会非常庞大，导致无法将整个模型放入设备的内存中进行推理. 因此，需要采取一些方法来减小模型的尺寸，以降低内存占用，并尽可能保持模型的准确性.

(3) 通信带宽受限传不出. 在智能物联网中，传感器通常负责采集环境数据，而有时这些数据可能非常庞大. 对于边缘设备而言，带宽通常是有限的，无法将大量数据直接传输到服务器进行处理. 因此，需要进行数据预处理和过滤，只传输必要的信息，或者利用协同推理来降低数据传输需求.

(4) 能耗过高设备撑不住. 智能物联网设备通常由电池供电，因此能耗管理是一个关键挑战. 复杂的深度学习模型需要大量的计算资源，导致设备能耗过高，影响设备的续航时间. 为了应对这个挑战，需要采取能效优化措施，例如利用低功耗硬件、模型量化、节能策略等，以最大限度地减少设备的能耗.

(5) 边缘节点数据存不下. 边缘节点通常具有有限的存储容量，而一些复杂的深度学习模型和以及传感器传输来的数据可能无法完全存储在边缘设

备上.为了解决这个问题,需要针对性地优化数据存储结构,以及利用端边协同的方式,将数据存储和处理任务进行合理划分,使得边缘节点数据能够得到有效管理和优化利用.总体而言,智能物联网中的模型推理面临多方面的技术挑战,需要在计算资源有限、数据传输受限、能耗管理和存储容量有限的情况下,寻找合适的方法和策略来优化模型推理过程,以满足实时性和能效性的要求.这需要综合考虑算法优化、硬件优化和系统设计等多个方面,为智能物联网应用提供高效、可靠的智能支持.

2.3 技术分类原则

如图 2 所示,本综述对技术分类原则是根据技术所属的模块进行划分的.智能物联网推理流程中存在三个关键模块,即传感器数据、智能模型和物联网硬件.三个模块的关联在于,传感器数据是推理流程的起点,数据会通过内存(本地部署的模型)或网络(部署于其他设备的模型)传输给智能模型;智能模型对接收到的数据进行推理计算,完成相应的智能物联网功能;物联网硬件,包括传感设备和计算设备,则作为物理基础承载了数据感知和模型计算的任务.三个模块统一构成了完整的智能物联网模型推理系统.在传感器数据层(第 3 节),我们主要介绍存储管理技术,以及存在于数据传输给模型过程中的输入过滤和数据编码优化机会.在智能模型层(第 4 节),我们重点介绍了自适应配置和模型压缩方法,分别关注模型的静态架构和动态运行时的优化.在物联网硬件层(第 5 节),我们讨论了基于编译器的计算图优化、多设备协同推理以及针对神经网络设计的专用加速器硬件.

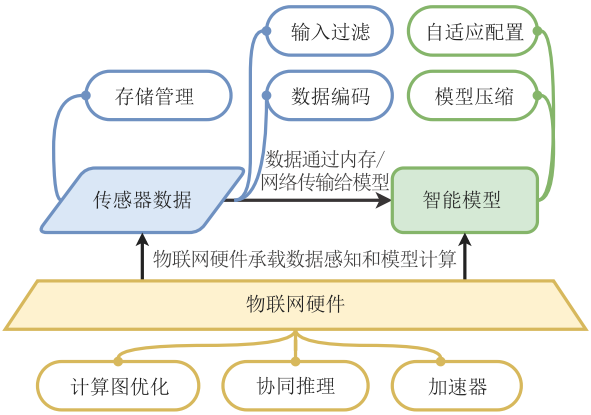

图 2 根据技术所属智能物联网模型推理流程中的模块(数据-模型-硬件)进行的方法分类

3 传感器数据层

传感器数据作为智能物联网模型推理的基础模块,其资源效率的优化可以从消除冗余数据、提高数据编码压缩率以及优化存储管理等多个角度入手,从而提升整体模型推理的性能.

3.1 时间资源优化:输入过滤与推理重用

值得注意的是,在传感器数据上进行推理并不总能产生有效或有价值的结果.以仓库监控视频为例,运行异常行为检测模型在没有出现人物的画面上,其预测结果将为空.类似地,对用户语音指令进行语音识别和语义分析,在用户发表与指令无关的内容时,进行的推理同样毫无意义.因此,在执行推理任务之前,过滤掉冗余输入数据具有重要意义.我们将这类方法称为“输入过滤”,表 4 从应用场景、关键技术和过滤依据等方面总结了典型的输入过滤与推理重用方法.

表 4 输入过滤和推理重用方法总结

方法	年份	应用场景	关键设计	过滤依据
粗粒度输入过滤				
Canel 等人[16](FilterForward)	2019	卷积神经网络推理	预训练特征抽取网络+分类器	置信度阈值
Li 等人[17](Reducto)	2020	视频分析	连续帧低级特征帧差	帧差阈值
Tchaye-Kondi 等人[18](SmartFilter)	2022	视频分析	特征帧差+分类器	置信度阈值
Yuan 等人[19](InFi)	2022	全模态模型推理	端到端可学的全模态过滤器	置信度阈值
细粒度输入过滤				
Jiang 等人[20](Remix)	2021	图像物体检测	图像切分为多个区域	图像子区域语义
Zhang 等人[21](Elf)	2021	视频分析	端侧进行粗粒度物体检测	图像子区域语义
粗粒度推理重用				
Chen 等人[22](Glimpse)	2016	图像物体识别	支持物体追踪的视频帧缓存	场景改变比例阈值
Guo 等人[23](Potluck)	2018	图像分析	数据特征向量缓存	相似度阈值
Guo 等人[24](FoggyCache)	2018	图像/音频分析	自适应局部敏感哈希缓存数据特征	近邻同质分数阈值
细粒度推理重用				
Ning 等人[25](Deep Reuse)	2019	卷积神经网络推理	局部敏感哈希度量特征图相似度	相似度阈值
Xu 等人[26](DeepCache)	2018	视频分析	视频帧切分为多个区域进行缓存	匹配区域搜索
Wu 等人[27](DREW)	2022	卷积神经网络推理	局部敏感哈希+多粒度聚类	聚类簇心

3.1.1　粗粒度数据过滤

在边缘视频分析任务领域，Canel 等人[16]提出了一种名为 FilterForward 的视频帧过滤方法. 该方法利用预训练的特征抽取器中的中间层激活作为输入，设计了一种轻量级微分类器来决定是否将特定视频帧传送至后端计算集群. 具体而言，FilterForward 提出了三种微分类器架构，包括完整帧物体检测、局部二分类器以及窗口化局部二分类器，以适应不同的分析任务. 在推理阶段，FilterForward 通过手动设置微分类器的置信度阈值来进行帧过滤. 值得注意的是，FilterForward 支持多种分析任务共用同一个特征抽取器，尽管不同任务可能选择不同的中间层激活. 针对实时视频分析任务，Li 等人[17]提出了一种名为 Reducto 的摄像头端视频帧过滤方法. 其核心思想是在摄像头端计算视频帧的变化程度，从而过滤掉变化较小的帧，避免将其传输至推理计算设备. 该工作表明，不同低级特征（如角落、边缘、面积和像素特征）适用于不同视频分析任务的帧变化程度测量. 例如，面积特征适用于物体计数任务，而边缘特征适用于物体检测任务. 通过计算连续视频帧之间的低级特征变化值，并采用轻量级聚类方法自适应地估计过滤阈值，Reducto 成功地在仅损失少量推理精度的情况下，节省了高达 51%～97%的计算开销. Tchaye-Kondi 等人[18]提出结合了帧特征差和二分类模型的帧过滤方法 SmartFilter，将特征差作为二分类模型的输入，使用推理反馈训练过滤决策. 输入过滤技术的研究不仅限于视频数据模态，Yuan 等人[19]提出了一种端到端可学的输入过滤框架 InFi，将输入过滤建模为二分类（冗余和非冗余）任务，支持多种输入模态（包括图片、视频、音频、文本、模型中间层特征等）和多种部署模式（包括端上推理、卸载推理和模型切分）. 此外，基于函数族复杂性度量，该工作给出了“可过滤性”的定义，并分析了三类常见推理任务的可过滤性，包括二分类、多分类以及回归任务.

3.1.2　细粒度数据过滤

之所以称上述方法为粗粒度数据过滤，是因为这些方法都是以整个数据作为单位来进行决策过滤与否. 自然地，我们可以考虑将输入数据拆分为多个子输入，例如图片的不同区域，来进行更细粒度的数据过滤. 图 3 左半部展示了粗粒度和细粒度数据过滤技术的区别.

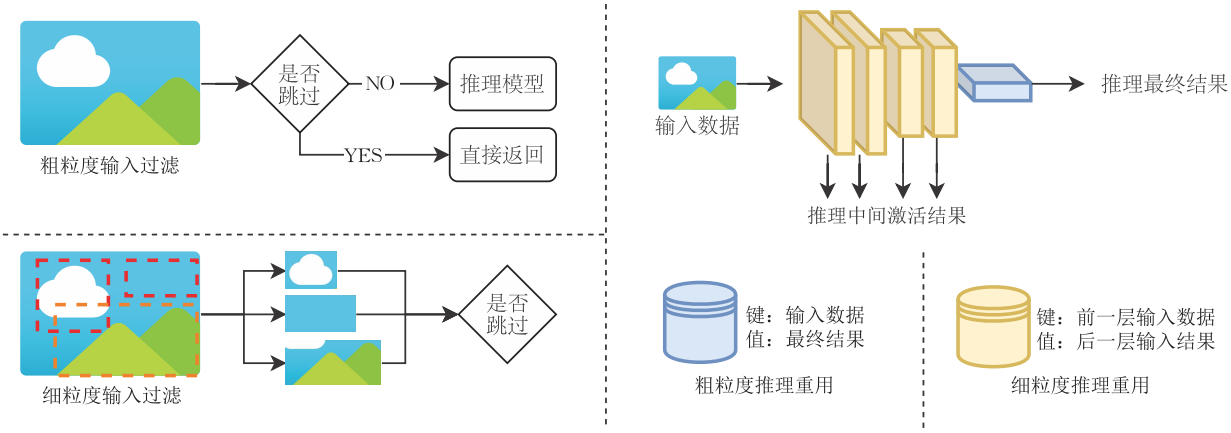

图 3　粗/细粒度的输入过滤和推理重用四个子类方法

Jiang 等人[20]针对物体检测任务设计了一种图像切分过滤的方法 REMIX，通过将图像切分为多个区域，不同区域根据延迟限制选择推理效率不同的模型进行处理（例如对于行人检测任务，行人较密集的区域使用较大的模型推理，而天空、建筑物等区域使用轻量模型或者直接跳过）. Zhang 等人[21]针对移动视觉任务提出了一种视频帧切分和推理并行卸载方法（称为 Elf），在移动设备上检测粗粒度物体并将对应区域切分，将切分后的多个部分视频帧并行发送给边缘节点进行推理卸载，实现对通信带宽的节省. 具体地，Elf 使用一个循环区域提议网络（Recurrent Region Proposal Network）来预测视频帧内的粗粒度物体区域（例如对于人体姿态检测任务，单个人物的检测框即为粗粒度物体区域），并且通过估计边缘节点的资源来动态分配切分的视频帧以实现负载均衡.

3.1.3　粗粒度推理重用

待推理的数据中除了那些结果无意义的冗余，还包含另一种称为可重用性的冗余，即该数据的推理结果与之前已经计算过的结果一致，如果能够高

效地重用之前的结果则能提高资源效率.例如,对于一个智能手环上的用户动作分类任务,如果两段运动信号输出的动作分类结果一致,则可以重用前一段数据的推理结果,来进行快速识别.

Chen 等人[22]针对移动设备和边缘服务器协同进行物体识别任务提出了 Glimpse 方法,通过维护一个视频帧的缓存实现对物体的高效追踪,并基于场景改变比例采样关键视频帧(较大可能得到与端上追踪结果不同的帧)传输给边缘节点进行物体识别,从而降低延迟和带宽开销. Guo 等人[23]提出一种跨应用的近似重用方法 Potluck 用以优化移动设备的智能任务,通过缓存输入数据的特征向量(例如对于图片可以使用经典的 SIFT 或 SURF 特征)和对应的推理结果,在新数据到达时计算和缓存数据的相似度,若超过设定的相似度阈值(该工作提出了一种基于神经网络的阈值自动调整算法)则直接返回缓存的结果,否则重新调用对应的推理函数. Guo 等人[24]观察到在近距离的多个端设备上调用的模型推理任务有着高度相似的上下文数据,其推理结果常常相同,进而设计了一种跨设备的近似计算重用方法 FoggyCache 用以减少冗余推理计算.具体地,FoggyCache 使用自适应局部敏感哈希将输入数据的特征进行降维,并针对性地为 K-近邻算法补充设计了"同质分数",用以判断缓存是否命中(同质分数高于某个阈值时认为命中,从缓存中直接返回之前的推理结果,否则认为不命中,重新调用推理模型计算).

3.1.4 细粒度推理重用

由于深度神经网络的推理一般由顺序的多层前向计算组成,自然地可以将重用的思想应用在模型的各个层.对于某一层或多层组成的块而言,以其输入作为缓存的键,以其输出的激活作为缓存的值,我们能够构建一种更细粒度的计算重用架构.图 3 右半部分展示了粗粒度和细粒度推理重用技术在缓存的键值对上的区别.

Ning 等人[25]针对卷积神经网络推理提出了 Deep Reuse 方法用以重用单个输入内的以及跨输入间的激活特征图. Deep Reuse 测试了多种聚类方法后发现局部敏感哈希方法最适合于检测激活特征图之间的相似度,并且确定角余弦距离相较于欧氏距离更适合.值得注意的是,该工作支持推理中间任意阶段的计算重用而不仅限于输入,我们考虑到和其他近似重用方法的相关性将此工作划分在此. Xu 等人[26]针对移动设备上的视频推理任务提出了 DeepCache 方法,通过利用视频流的时序局部性重用卷积计算结果,从而提高推理效率.具体地,DeepCache 将视频帧切分成多个区域,以输入的帧作为缓存的键,以卷积计算的特征图作为缓存的值. DeepCache 借鉴了经典的视频压缩方法里的匹配区域搜索算法,通过视频运动启发式模式匹配分析视频的内部结构,从而发现可重用的输入图像区域. Wu 等人[27]针对 Winograd 卷积[28]提出一种计算重用技术(称为 DREW),用以加速弱计算设备上的卷积模型推理.具体地,DREW 为 Winograd 卷积里涉及的局部敏感哈希投影和桶映射操作设计了专用的重用流程,并且将聚类粒度扩展到了多个通道,实现对效率-精度的可调节权衡.

3.2 带宽资源优化:数据编码

传统的数据编码压缩方法一般旨在保障人类感知质量的前提下降低数据大小,例如图像压缩算法考虑的失真指标 PSNR 和人眼视觉感知是相关的.当数据是为了人类的识别(例如观看视频),这些数据编码压缩算法已经得到了充分的研究.但当我们考虑数据是为了人工智能模型的识别(例如在视频中进行人物检测),传统的数据压缩算法的设计的针对性很弱,潜在很大的优化空间[29].

Xie 等人[30]认为传统的图像压缩(编解码)方法是为人类的感知(例如人眼视觉)设计的而非深度神经网络,进而提出一种神经网络感知的图像压缩方法 Grace,通过从空间频率和颜色对深度神经网络感知进行建模,为卸载到边缘节点的模型推理生成优化的压缩策略. Grace 基于深度神经网络相对于输入图像的频率和颜色的梯度来估计感知模型,之后以最小化文件大小为目标,以构建的感知模型的损失为限制,优化图像编解码的参数(包括量化表和 RGB 到 YUV 颜色空间转换权重). Lu 等人[31]提出了第一个端到端的视频压缩的深度模型 DVC,使用基于卷积神经网络的光流估计来获取运动信息(用于减少视频帧序列的时序冗余)并重建当前帧,通过自动编码器神经网络压缩对应的运动和残差信息,所有模块(运动估计、运动补偿、残差压缩、运动压缩、量化、比特率估计等)都可以通过单个损失函数端到端的训练. Hu 等人[32]针对物联网应用中图像传输对丢包的脆弱性问题,设计了 Starfish 图像压缩算法,能够很好地处理丢包的同时实现更高的压缩比进而节省通信带宽.具体地,Starfish 使用一个

(基于神经架构搜索自动生成的)小型神经网络生成非结构的图像压缩表示,相较于传统的 JPEG 使用的结构化表示,非结构化表示将图像信息均匀地分布于神经网络表示中,重建图像需要完整地表示而非特定的部分数据,因而对丢包更具有弹性. Deng 等人[33]针对无人机和边缘节点协同进行物体检测推理设计了 Geryon 方法,利用雷达数据辅助相机视频数据进行重点画面区域抽取和编码,减少了与边缘节点的带宽开销和卸载延迟. Li 等人[34]指出预测编码(基于预测帧生成和残差编码)使用简单的减法运算来消除跨帧冗余,是一种次优的解决方案,因而提出一种条件编码方法 DCVC,将上下文信息作为编码器、解码器和熵模型输入的一部分,在特征域使用运动估计和补偿来学习编码的条件信息. Du 等人[35]设计了一种视频编码方法 AccMPEG,通过一个轻量卷积神经网络来估计编码的各个宏块(Macro-Block)对推理精度的影响,进而优化宏块的编码质量参数,实现低延迟高精度的视频推理. Xiao 等人[36]针对语义分割模型推理任务提出了 STAC 视频压缩方法,利用深度神经网络的梯度作为空间敏感度量(具体地,梯度量化了由压缩每个像素引起的损失函数变化)进行自适应压缩,通过光流将压缩策略和分割结果传播到多个视频帧,进而降低带宽开销. 关于基于神经网络的图像和视频压缩,可参考 Ma 等人[37]于 2020 年发表的综述文章.

3.3　存储资源优化:存储管理

从数据存储的角度,相较于传统数据存储管理,智能物联网模型推理任务也存在可观优化空间. 以视频存储为例,用于人工查验的视频数据库往往以 H.264 之类的编码算法全帧存储视频文件;显然地,这里的存储存在大量的冗余(推理不会产生有价值的结果)视频帧. 针对推理任务设计存储管理策略能够显著提高存储资源效率. Poms 等人[38]针对视频分析任务中的像素数据存取提出了 Scanner 系统,将视频集合以及视频衍生栅格数据(包括深度图、激活图、流场等)组织为专为压缩视频优化后的数据存储中的表,实现许多对视频处理有用的功能(例如视频帧稀疏采样、访问视频帧的时间窗口以及跨连续帧计算的状态传播). 除了视频存取之外,Scanner 还将分析任务组织为数据流图,并设计了将计算任务自适应调度至异构硬件(包括 CPU、GPU 以及媒体处理 ASIC 等)的算法. Haynes 等人[39]提出了一种视频存储系统 VSS 用以将高级的视频操作和低级的存储细节解耦合,为了提高资源效率,VSS 将视频分解为一系列独立可解码的帧集合,并将物理上邻近的摄像机拍摄的视频(往往存在画面重叠)进行协同压缩(仅存储重叠部分一次)以减少存储开销. VSS 基于原始视频和缓存的表达来自动地选择最高效地生成目标格式和区域的数据的方法,支持三大类限制的读写,包括时间(例如起止时间和帧率)、空间(例如分辨率和兴趣区域)以及物理(例如压缩编码器、视频质量)参数. Daum 等人[40]设计了一种基于图块(视频编码器里的概念)的视频存储管理器 TASM,提供了空间维度的视频随机访问功能,进而优化子帧选择查询(例如查询包含某物的裁剪后的视频片段)的性能. TASM 使用一种语义索引的方法维护视频内容的元数据,语义信息由标签(包括物体类别和其他的性质例如颜色)和区域框组成,语义索引使用 B-树对三元组(视频,标签,时间)聚类实现. Hu 等人[41]针对视频分析系统提出了一个在查询和存储之间独立的检索层 Video-zilla,使用与推理任务相关的特征向量来表示每一路视频(例如画面内物体的分布特征用于物体识别任务),并基于此构建同一视频流内和跨多个视频流的层次化的视频索引,实现将跨视频流视频分析的时间复杂度和视频流数的关系从线性降为次线性.

4　智能模型层

智能模型的计算是智能物联网模型推理流程中的核心模块,现有工作探索了自适应配置和模型压缩的方法优化时间资源和内存资源效率.

4.1　时间资源优化:自适应配置

智能模型的推理计算本身依赖于很多“配置”,以视频上进行物体检测任务为例,输入相关配置包括视频分辨率、帧率、图像批大小,以及计算相关配置包括检测算法、算法框架内使用的具体骨干神经网络、多模型调度顺序等等. 图 4 展示了通常考虑的输入相关和计算相关的配置. 在推理过程中,根据当前的负载、数据内容等动态性进行自适应地调整相关配置的方法,我们称为自适应配置. 更细粒度的,我们从调度、查询、级联三个角度的自适应性进行方法的分类介绍.

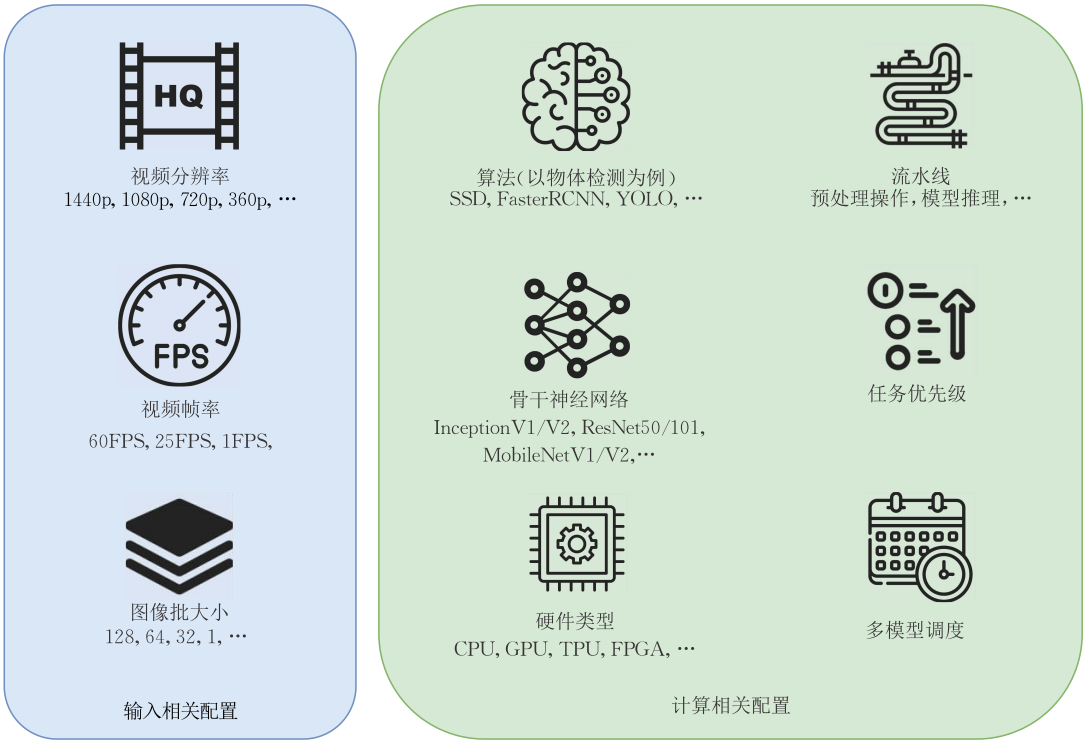

图 4 自适应调度通常考虑的输入相关和计算相关配置

(1)自适应调度. Zhang 等人[42]针对实时视频分析任务提出了 VideoStorm 系统,对于有着不同的资源-质量配置的视频分析查询,VideoStorm 高效地生成配置文件并统一地进行调度. 该工作考虑视频分析的配置包括视频分辨率、帧率,以及分析算法的参数(例如物体检测的滑动窗口大小). VideoStorm 包含两个阶段:① 离线阶段通过贪心搜索确定配置空间里的 Pareto 最优解(在多目标优化问题中,如果在不牺牲至少一个其他目标的情况下无法改进任何目标,则该解被称为 Pareto 最优解);② 在线阶段调度器仅考虑这些 Pareto 最优配置,通过计算一个考虑质量和延迟的效用函数,对所有的分析查询进行统一调度(最大化最低效用或整体效用). Jiang 等人[43]针对基于神经网络的视频物体检测流程,考虑了包括帧率、图像尺寸、物体检测算法(FasterRCNN 和 SSD)、物体检测骨干模型(InceptionResNet、ResNet101、ResNet50、InceptionV2 和 MobileNetV1)在内的可调节配置,提出一种视频自适应配置方法(称为 Chameleon). 具体地,Chameleon 利用影响物体检测最优配置的因素(包括物体速度和大小)所蕴含的时空关联性,将最优配置的搜索空间大幅缩小并摊销到多个视频源上,在相同精度下实现 2 到 3 倍的加速效果. Ali 等人[44]针对模型推理服务可能遇到突发请求(进而带来高延迟)的问题设计了一种自适应批量化(Adaptive Batching)的方法 BATCH,根据推理请求负载和延迟目标,预估延迟分布并自适应地调整两个参数:批大小(Batch Size)和超时(Timeout,等待后续请求以组成一批数据的最大时间). Crankshaw 等人[45]考虑模型推理流水线中通用的配置参数(包括数据批处理大小、硬件加速器的选择、推理请求达到过程等)设计了 InferLine 系统,包含两个主要组件,即组合规划器(低频率运行)和自动缩放器(高频率运行). 组合规划器根据逻辑计算图自动为每个计算任务选择硬件类型(CPU、GPU、TPU、FPGA 等)、并行副本数、数据批处理大小,而自动缩放器则根据在线的请求到达过程的变化(请求率)来缩放推理流水线中的各个阶段(以满足时延限制). 这里的缩放操作指根据组合规划器选择的硬件结果,增加或减少并行处理任务的对应硬件. Kang 等人[46]观察到视觉分析系统的端到端瓶颈在于数据预处理(包括解码和调整大小操作)阶段,并针对此开发了 SMOL 系统,主要包含了两个优化技术:一是通过显式地使用数据增强来训练低分辨率模型,并使用部分解码方法降低解码开销;二是将预处理操作和模型推理计算进行流水线化,并针对性地优化了内存管理和线程调度. Liu 等人[47]考虑实时推理系统中存在的优先级倒置(指由于顺序执行逻辑导致的低优先级的任务在高优先级任务

之前执行，这里的优先级指例如在自动驾驶中，识别一辆车道上失控的车要比检测周围建筑更重要，而简单的顺序推理逻辑可能会导致优先级倒置）问题，提出了一种对输入数据进行优先级区域划分以及在延迟限制下的推理调度算法. Li 等人[48]提出了一种能够自适应于能耗的多出口神经网络（Multi-Exit Neural Network），由具有不同精度和计算配置的两个出口组成，为智能物联网异构设备上的部署提供了灵活性，可以针对不同的应用需求在精度和处理时间之间进行权衡. Yuan 等人[49-50]针对多模型推理任务提出了一种基于深度强化学习的自适应模型调度框架，给定一个输入数据，将其标注状态作为强化学习的观察输入，使用 Q-价值网络预估各个模型的执行价值，最后基于预估的执行价值和资源限制决定调度策略. 该工作提出在动作空间（每个模型作为一个动作）中加入“结束”动作，实验结果表明此方法能够有效地提高强化学习收敛速度. 该工作考虑两类资源限制，即一维时延限制和二维时延-内存限制，两类问题皆为 NP 难问题，为了调度的高效性文中分别给出了启发式的调度算法. 不同于经典的静态神经网络，动态神经网络[7]能够根据不同的输入数据调整神经网络结构和参数，因此在推理的资源效率上具有技术优势. Huang 等人[51]为图像分类任务提出了一种多尺度的卷积网络结构，其由多个处理不同分辨率（尺度）特征图的子网络组成，能够快速生成适合分类的特征. Wang 等人[52]基于跳层推理的思想提出了 SkipNet 卷积神经网络，使用门控网络根据前一层的激活选择性地跳过卷积层. 类似的，Campos 等人[53]为递归神经网络设计了 SkipRNN，通过在每一步中更新控制信号，以确定是否更新或复制上一步的隐藏状态，从而实现可跳层推理. Wang 等人[54]设计了一种多分支可扩展的动态神经网络架构，MBSNN，该架构具有多个子网来提供不同精度的推理结果，并提出了一种基于阈值选择的自适应推理机制，该机制可以动态地选择 MBSNN 的阈值，以在给定的时延约束下实现最佳精度.

（2）自适应查询. Lu 等人[55]针对视频分析查询设计了 Optasia 数据流系统，将视频分析模块化为一些通用模块（例如车辆类型分类、跨摄像头的车辆重识别、车牌识别等）并以数据流的形式进行组织. Optasia 使用关系型查询优化技术（包括谓词下推和多查询合并）来对数据流进行去重和并行化. Kang 等人[56]针对特定类别物体是否出现的视频查询任务（例如查询包含公交车的视频帧）设计了 NoScope 系统，通过在查询阶段搜索并训练一个针对特定查询的级联模型（包括一个视频帧差异检测器和二分类器）加速原始的全帧处理工作流. NoScope 搜索训练出的针对性模型放弃了物体判别的通用性，而只关注目标查询的物体类别，因而能够大幅模型降低参数量和计算开销（实验测试速度是原始神经网络的 340 倍）. Hsieh 等人[57]针对离线视频集“事后查询”任务（例如查询包含某类物体的视频帧）提出了 Focus 系统，将事后查询任务拆分为两个阶段：摄取阶段和查询阶段. Focus 使用一个轻量的卷积网络在摄取阶段对视频帧构建一个关于可能包含的物体类别的近似索引，在查询阶段自适应地利用使用原本的物体检测模型补偿近似索引的低精度，实现低延迟高精度的事后查询，注意，Focus 提出的方法中的摄取阶段应该划分为本文所提出的“输入预处理”类，但是由于该工作的主要创新在于自适应地在查询阶段利用近似索引这一“输出”，我们将其划分在此章节中. Kang 等人[58]针对视频分析结果的聚合（平均每帧中的车辆个数）和限制（包含至少五辆车的帧）查询任务提出了 BlazeIt 系统，使用一种基于神经网络的方法快速回答近似的聚合查询，并为限制类查询设计了一种基于代理模型的搜索算法，相较于传统的过滤和采样方法能够更好地平衡不同物体出现频率下的查询表现. Romero 等人[59]关注视频分析任务中多种谓词和推理模型的相关查询，指出人工创建执行计划往往是开销高且不准确的，提出一种声明性的接口，称为关系提示，允许用户根据其领域知识来建议推理模型的关系（包括两类：可替代和可过滤）. 基于关系提示接口，该工作提出 VIVA 系统，包括用于确定哪些关系提示适用于查询的提示验证器，用于通过模型替换、数据过滤和谓词重排序生成替代执行计划的规划器以及用于权衡性能和精度的优化器. Xu 等人[60]考虑一种称为探索式视频分析的任务，其特点是用户会从一个较广泛的查询开始，迭代式地细化查询直到得到目标分析结果，例如在交通视频数据库中，从查询“夜间的卡车”开始，在后续查询中迭代式地加入关于车身颜色、车牌信息等条件. 该工作设计了 EVA 系统，使用符号方法来分析谓词并识别查询之间的重叠程度，并基于重用机会的可能性设计了谓词重排序和模型选择方法. Chen 等人[61]关注视频数据上一种称为时空约束排序检索的任务，其特点是用户会对视频中物体的时空关系提出约束. 该工作使用图来编码物体的属性（例如类别和颜色）和物体间的时空关系，并基于物

体追踪算法为每个帧内的物体赋予 ID 信息，并利用时空信息匹配和早退机制加速检索. Kang 等人[62]针对非结构数据(例如文本和图像)分析查询设计了一种语义索引方法 TASTI，基于后端推理模型提供的输出相近程度函数，TASTI 为每个非结构化数据记录生成嵌入向量，并根据嵌入向量和有标签数据点(视作聚类的代表点)为查询高效地自动生成近似结果. Cao 等人[63]设计了 FiGO 来改进视频分析查询优化，使用包含多个权衡了吞吐和精度模型的集成模型来代替单一的轻量化模型，将视频切分成不同大小的块序列，并自适应地为各个块选择合适的模型(重要的块使用较慢但精确的模型，不相关的块则使用快速但不精确的模型).

(3) 自适应级联. Shen 等人[64]观察到视频推理应用处理的数据分布往往是高度偏斜的(指局限在例如人脸识别仅需要处理同一房间的数个人物，而不需要像在训练阶段一样考虑泛化性)，而这些高度偏斜的分布数据可以用更为简单轻量的模型正确识别，但直接地用简单模型替换可能会导致精度严重下降，因此提出了一种基于多臂老虎机的在线决策算法，用以自适应地从多种轻量模型和一个参考模型(精度最高但开销也大)中选择对当前数据进行推理. Anderson 等人[65]针对视觉分析结果查询任务提出 Tahoma 系统，通过统一地优化级联推理架构和输入数据表达提高查询速度. 具体地，Tahoma 会预先以不同的神经网络超参数(例如卷积层数)和输入表达参数(例如 RGB 三通道表达或灰度单通道表达)训练多个专用的二分类模型，然后根据这些专用模型构建多个级联分类器，最后根据用户对于速度和精度的要求选择出 Pareto 最优级联方案. Chakrabarti 等人[66]考虑一种边缘计算场景，其终端设备上运行的推理模型(相对较弱)在执行完成后，会决策是否需要将当前数据卸载至边缘节点模型(相对较强)以进行更精确的推理. 具体地，该工作关注通信(由数据卸载带来的)资源限制和推理精度的权衡，形式化了一种基于马尔可夫决策过程的优化问题，并设计了基于模型输出置信度和令牌桶(令牌桶算法[67]是网络速率限制的一种常用算法，能够在限制数据的平均传输速率的情况下，允许一定程度的突发流量)状态的决策算法. Yuan 等人[68]提出 MLink 系统，通过构建异构模型之间的语义关联(称为模型链接)来提高多模型推理效率，具体分为两个阶段：模型链接构建阶段和多模型协同推理阶段. MLink 将异构模型的输出分为两类，即向量和序列，基于此设计了四类输出空间映射模型(即模型链接). 直观而言，模型链接的目的是利用一个模型的输出去预测其他模型的输出，在多模型推理阶段，MLink 自适应地根据资源限制和模型链接的性能选择部分模型执行推理计算，其他未被选择的模型则利用模型链接预测它们的输出，从而在相同资源开销下提高全集标签的召回率. Hwang 等人[69]针对视频分析系统中的级联推理方法在：① 解码瓶颈和 ② 缺少空间查询支持两个局限性提出了改进系统 CoVA，将计算分为压缩数据域(即解码前的视频)和像素数据域(即解码后的图像)，通过在解码前对压缩数据进行移动物体检测(基于编码元数据例如宏块类型和运动矢量)来选择必要的帧进行解码，进而有效地缓解解码瓶颈. Ghosh 等人[70]针对边-云协同的视频分析任务提出 REACT 方法，选择性地将图像发送给云端，和边侧的识别结果进行融合，利用一定的云侧冗余计算提高了边侧的识别精度，同时保持了边侧识别的低延迟特性.

4.2 内存资源优化：模型压缩

模型压缩技术能够显著提高内存资源效率，相关综述[5-6]已经给出较为全面的介绍. 为了完整性，本文在此从轻量化模型设计、网络剪枝(如图 5 所示)和参数量化三个角度着重介绍一些尤其适用于物联网设备的模型压缩技术.

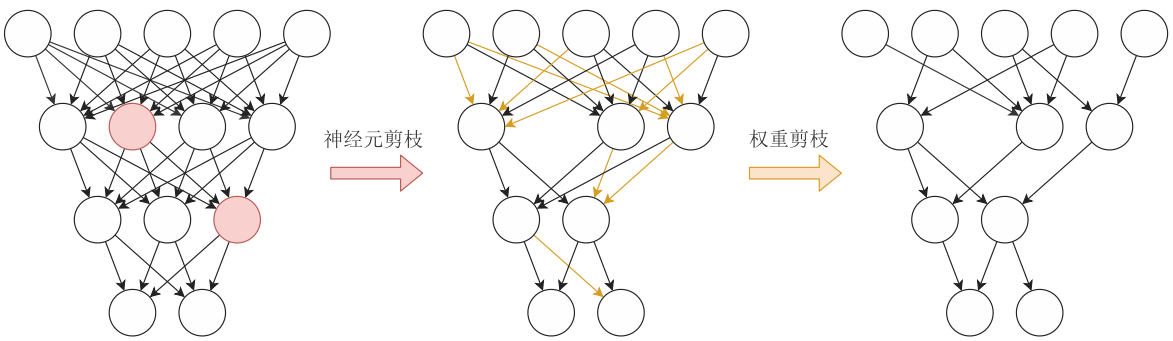

图 5 网络剪枝技术的两种基础思路：神经元剪枝和权重剪枝

(1)轻量化模型设计. Zhang 等人[71]为计算能力非常有限的移动物联网设备设计了一种计算效率极高的 CNN 轻量级新架构 ShuffleNet. 通过引入分组卷积技术降低深度可分离卷积模块中计算量较大的 Pointwise-Conv(逐点 1×1 卷积)开销，N 组卷积可将计算量降低为原结构的 $1/N$. 同时使用通道混洗技术来实现组间特征信息的交互，防止不同通道特征组之间的信息流阻塞. ShuffleNet 将模型计算量压缩至 10～150 Mflops 同时实现了同计算量级下最高的精度. Ma 等人[72]在 ShuffleNet 的基础上提出了 ShuffleNet V2，针对只优化计算量的深度可分离卷积(Depthwise Separable Convolution)系列模型实际加速效果差等问题，总结了轻量级模型设计的关键准则. 包括降低访存量、控制分组卷积组数、减少网络碎片化和关注元素级算子等，并根据这些准则设计出 ShuffleNet V2. 该工作深入分析了模型结构计算量和访存量对实际推理速度的影响，发现的设计规律对后续轻量级压缩模型构建具有重要指导意义. Lin 等人[73]针对低功耗物联网设备上的智能计算任务提出了 MCUNet，一种融合了高效的神经网络架构搜索方法 TinyNAS 和轻量级推理引擎 TinyEngine 的系统-算法协同设计框架. TinyNAS 是一种两阶段的 NAS(神经架构搜索)算法，首先设置不同的输入分辨率和通道缩放比例来优化 NAS 搜索空间，使其适应各种微小、不同的资源约束，随后在优化后的搜索空间内进行 NAS 搜索. TinyEngine 是针对 MCU 设计的高效推理引擎，其重点优化了代码生成、内存调度、卷积循环展开、算子融合等操作，大幅降低了推理内存开销. MCUNet 在多个图片、视频、音频任务上超越了现有轻量级网络的精度，使得弱设备上的微型机器学习成为可能. Lin 等人[74]在 MCUNet 的基础上进一步提出了 MCUNetV2，通过优化模型计算流从而大幅削减模型峰值内存使用量. 基于推理过程中层间内存使用不均衡的观察，在内存密集阶段，利用逐块推理取代逐层推理，单次只计算特征图的小部分区域，在略微提升计算量的前提下节省 4～8 倍内存. 同时利用重分布感受野技术，降低逐块推理部分感受野的同时提升逐层推理部分的感受野，使得减少重复计算的同时保持模型精度.

(2)网络剪枝. Liu 等人[75]设计了一种智能物联网应用需求驱动的模型压缩框架. 根据给定的模型推理精度和时延需求，使用设备存储、能量等指标作为约束条件. 将优化目标和约束条件整合进强化学习奖励函数中，基于 DQN 构建强化学习代理来为每层搜索特定的压缩动作(剪枝、张量分解、算子替换等)，使得压缩后模型在满足硬件约束的前提下最优化用户需求. Gao 等人[76]设计了一种运行时自适应剪枝框架，在保留完整模型的基础上通过选择性执行部分通道来压缩计算量. 基于特征图重要性高度依赖输入的观察，提出了特征增强和抑制 FBS(Feature Boosting and Suppression)模块，在运行时根据不同输入样本选择性放大显著卷积通道并跳过(剪枝)不重要的通道，既保持了模型完整性和高精度也能够提升推理速度. Cai 等人[77]针对不同计算能力的物联网设备构建了一个多分支的超网络，将部署阶段的压缩重训练成本集中到超网络离线训练阶段. 超网络中包含设定好的不同卷积核尺寸、通道数、层数的子模块，通过渐进收缩的训练方式，逐步训练好超网络下不同层级的子模块. 在部署阶段，使用一个预先训练好的模型精度-时延预测器快速搜索出超网络中符合当前硬件资源限制的子网络. 该方法将网络搜索和网络训练解耦合，使得只需单次训练便可用于多种场景和设备，大幅降低了部署时压缩的成本. Liu 等人[78]针对智能物联网应用运行阶段可能出现由于设备能量变化、资源利用率变化导致实际推理速度不符合预期的问题，综合考虑了移动应用程序部署上下文的动态性，设计了一种上下文自适应和自进化的 DNN 放缩框架. 使用一个无需微调的自进化网络集成训练算法，以集成多种备选的 DNN 压缩配置(即压缩架构和权重)，同时引入运行时搜索策略快速搜索最合适的压缩配置并演化相应的权重来适配动态的设备上下文环境.

(3)参数量化. 二值神经网络(Binary Neural Network)[79]指参数权重限制为两个值(例如 0 或 1)的神经网络，相较于常见的单精度(使用 32 位浮点数表达参数)神经网络，能够降低 32 倍的内存开销，且推理所需的乘法累加算子可以用更快的简单累加替代，但在大幅提升资源效率的同时二值化也会导致严重的信息损失. 探索如何在保证精度的情况下实现神经网络二值化推理，对于智能物联网是一个有前景的方向. Lin 等人[80]将卷积神经网络的权重限制为−1 或+1，使得卷积计算可以仅通过加减法而无需乘法来实现，当激活值也是二值化的时候则可以通过按位运算实现，并使用多个二值权重的线性组合来近似全精度权重，减轻了精度下降的同时显著减少推理时间和功耗. Rastegari 等人[81]提出用于图像分类的 XNORNet，其权重和输入都是二值化的(−1 或+1)，通过 XNOR(异或 XOR 操作后进行非 NOT 操作)和 Bitcount 位运算操作来近似

原本的卷积计算,实现在 CPU 上的 58 倍加速. Zhu 等人[82]提出 XORNet 使用异或 XOR 操作代替异或非 XNOR,从而避免了 NOT 操作使得位操作能够在一个周期内完成,对于具有缩放因子的二值网络,XOR-Net 则将缩放因子矩阵的乘法移动到下一层,并将常数移动到权重的缩放因子,进而进一步减少全精度操作. Lin 等人[83]针对量化误差中的(除了常模误差之外的)角度偏差提出旋转(Rotated)二值神经网络 RBNN,通过学习一个将全精度权重向量转化至二元超立方体的几何顶点的旋转矩阵来进行角度对齐,此外针对优化中可能出现的局部最优问题,RBNN 在训练阶段动态调整旋转权重以进行二值化. Chen 等人[84]针对移动手机端上的 GPU 设计了二值神经网络推理引擎 PhoneBit,为训练后的 BNNs 开发了一组运算符级优化,包括位置友好的数据布局、带矢量化的位打包和用于高效二进制卷积的层集成方法. Zhang 等人[85]指出输入层的浮点权重和激活会导致在 FPGA 设备上计算难以并行化,提出利用部分激活(计算额外的系数二值卷积层来用两个比特更新部分特征)来提高精度的模型 FracBNN 并为其设计了基于 FPGA 的支持分数激活的加速器. Wang 等人[86]针对 FPGA 上运行基于 XNOR 的二值神经网络推理设计了 LUTNet,使用 FPGA 原生的 K-LUT 组件(可以执行任意 K 个输入的布尔运算)替代 XNOR 运算,基于拉格朗日差值多项式将二值权重实现为 LUT 的配置掩码,在相近的精度情况下实现了更高的硅面积效率和能量效率.

5 物联网硬件层

模型推理计算的最后一个模块便是使用具体的硬件实际地执行计算. 从物联网硬件的角度,相关方法探索了优化时间资源的计算图优化技术、针对端边通信带宽优化的协同推理技术以及设计优化能量资源的特殊硬件加速器. 需要注意的是,本章并不是要对比物联网的不同硬件,而是讨论的由于物联网硬件层的特点(包括网络环境和设备异构等)带来的优化空间.

5.1 时间资源优化:计算图优化

计算图(Computation Graph)是用于表示计算过程的图结构,其中节点表示操作(算子或函数),边表示操作之间的数据流动. 计算图通常用于描述深度学习模型的前向传播(推理)过程,其中输入数据经过一系列操作后得到输出结果. 在计算图中,每个节点都表示一个具体的计算步骤(比如加法、乘法、卷积、激活函数等),而边表示数据在这些计算步骤之间的传递. 对计算图在预处理阶段进行优化可以帮助减少资源的消耗、降低推理延迟,从而提高模型在生产环境中的性能.

(1)算子融合. 算子融合(Operator Fusion,或称为内核融合和层融合)是许多模型推理框架中提高效率的重要方法,主要解决普遍存在的两个关键性能问题,即内存墙和并行墙问题. 图 6 展示了一个卷积神经网络结构中进行算子融合的示例. 针对内存墙问题,通过对计算图上存在数据依赖的算子进行融合能够使得中间数据访存退化为局部变量甚至寄存器变量,进而提升访存效率. 以 TVM[87]、XLA[88]和 MLIR[89]等为代表的 AI 编译框架能够自动地对相邻的存在数据依赖的算子进行融合. 关于 AI 编译优化,可参考 Li 等人[90]发表的相关综述. Zheng 等人[91]针对内存密集型算符开发了 AStitch 编译器,考虑多个维度的优化目标并系统化地抽象出四种算符拼接范式,分别是独立范式、局部范式、区域范式和全局范式. 具体地,AStitch 使用了层次化数据重用来解决两层复杂依赖性并且能够扩大融合范围,基于自适应线程映射技术实现对于不同输入张量形状的支持. 对于由单个算子节点并行度与多处理器不匹配导致的并行墙的问题,Ma 等人[92]提出 Rammer 编译框架,针对算子间并行和算子内并行的相互影响,通过将计算图算子进行并行编排来提高整体并行度. 不同于之前的两层调度(深度学习编译器负责算子间调度,在此之下还存在一层算子内并行调度器)架构,Rammer 将计算图分解为更小的调度单元,在编译时直接生成调度规划并静态映射到硬件上,从而减少运行时调度带来的开销. Niu 等人[93]提出 DNNFusion,应用代数化简(转化为数学上等价的计算)优化,包括手工设计的结合律、分配律和交换律,之后根据算子属性对算子进行类别(一对一、一对多、多对多、重组、打乱五类)划分并按类融合,其中对于是否可以融合的判断基于手工设计的规则实现. Zhao 等人[94]同时考虑内存墙和并行墙问题,设计了 Apollo 编译器,提出多层规约融合优化技术,使用一个统一的框架支持循环融合、拼接融合、并行融合等多个优化. 具体地,Apollo 包含三层融合,分别通过多面体调度进行循环融合、通过识别数据依赖进行拼接融合以及通过识别无依赖融合算子进行并行融合. Cai 等人[95]针对嵌入式物联网设备上的访存瓶颈设计了 Optimus 算子融合框架,包括一种用于调度融合后的算子的精确内存开销模型,以及基于有向无环图的最优算子融合算法.

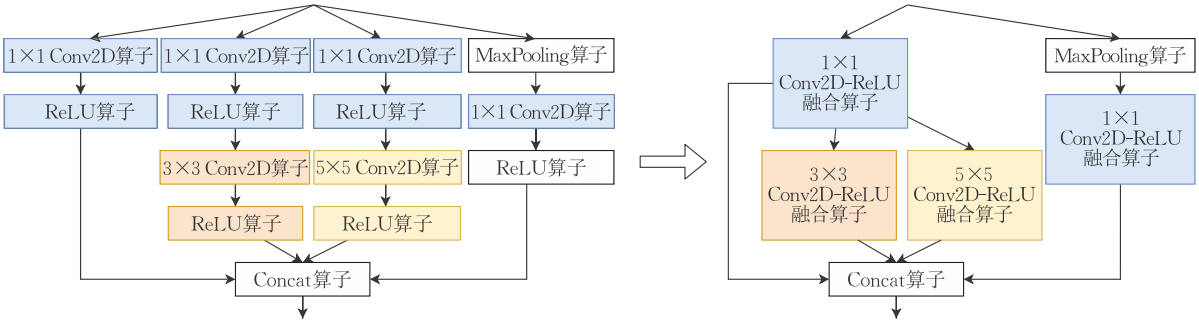

图 6　计算图优化中的算子融合示例

(2)算子优化. 深度神经网络推理的最常见的基础算子是矩阵乘法，NVIDIA 开发的 cuBLAS 和 cuDNN 软件库优化了针对 GPU 的矩阵乘法算子. 使用快速傅里叶变换(Fast Fourier Transform)能够更进一步地将乘法运算的次数从 $O(M^2N^2)$ 降低至 $O(M^2\log M)$，其中 M、N 分别是矩阵和卷积核的大小. 由算法复杂度可见，基于 FFT 的卷积计算对于较大的卷积核(尤其是 $N>M$ 时)而言是高效的，但常用的卷积神经网络中往往使用较小的卷积核(例如 3×3 和 1×1)，针对这一问题，Lavin 等人[28]提出了一种利用 Winograd 最小过滤算法的卷积方法. 相较于直接卷积，该算法能够将卷积层的算术复杂度降低多达 4 倍. Google 为嵌入式设备上模型推理开发的 TensorFlow Lite Micro[96]，基于专门优化(包括内核优化、删减依赖、静态规划)后的解释执行器实现将神经网络模型部署到微控制器上进行推理. Fawzi 等人[97]提出 AlphaTensor，经矩阵乘法转化为张量分解任务，通过深度强化学习来发现高效的分解算法，实现了对经典的 Strassen 算法的复杂度优化.

5.2　带宽资源优化：协同推理

使用物联网端侧设备和边侧设备共同进行推理计算的方法称为协同推理. 自然地，根据神经网络的拆分方式，我们从层间拆分、层内拆分以及层间和层内拆分结合三个角度(如图 7 所示)介绍相关技术. 值得注意的是，模型协同推理技术通常既考虑切分设备之间的通信开销，又考虑智能模型的结构，因此严格来说这一类技术应该属于“智能模型”和“物联网硬件”交叉的一层. 本综述考虑到模型切分方法都会重点考虑待部署硬件的计算和通信资源，将此节安排在“物联网硬件”一章.

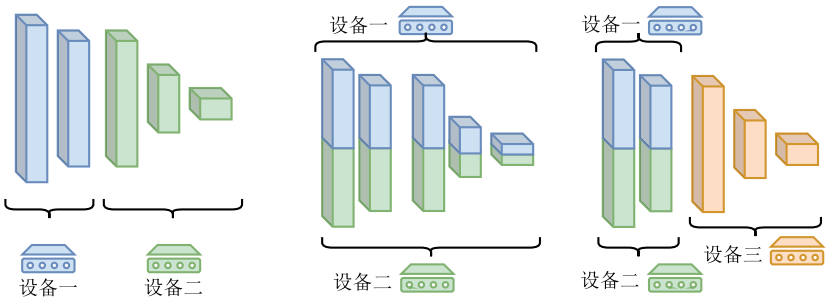

图 7　协同推理三种神经网络模型拆分方式：层间、层内、层间和层内结合

(1)层间拆分部署. 对深度神经网络的层间拆分基于这样一个基本认识：“网络前半部分的卷积层通常具有较大的输出数据规模和较低的计算需求，而网络后半部分的全连接层则恰好相反”. 通过合理选择 DNN 模型的拆分位置，在降低通信开销的同时充分利用端边设备的计算能力，能够达到降低推理时间/能量开销的目的. Kang 等人[98]针对链式 DNN 模型设计了一种二拆分策略 Neurosurgeon，拆分后模型的前半部分在端侧运行并上传中间结果，由模型的后半部分在边侧完成最终结果的推理. Neurosurgeon 使用回归模型预测不同类型层(如卷积层、池化层、全连接层)的计算开销和输出数据传输开销(时间开销，能量开销)，之后线性搜索总代价最小的二拆分方案. Li 等人[99]针对链式 DNN 模型，设计了一种能够权衡推理时延和推理精度的二拆分策略 Edgent. 借助早退出策略，即使存在严格的推理时延约束，Edgent 仍然能够通过小幅度牺牲精度得到合理的拆分方案. Edgent 在模型训练阶段在 DNN 不同位置增加提前退出推理过程的分支，并在推理阶段尝试对所有早退出分支线性搜索总代价最小的二拆

分方案,在无法满足推理时延约束的情况下,允许启用更早的退出分支. Eshratifar 等人[100]针对链式 DNN 模型设计了一种多拆分策略 JointDNN,允许将 DNN 模型中任何一部分连续的层部署在端侧或边侧运行. 该工作考虑到现有深度学习框架会对 DNN 执行算子融合优化(即连续执行两个 DNN 层时间小于分别执行的时间之和),因此对链式 DNN 进行了特殊的图建模并在图上寻找最短路径以得到延迟最低的拆分方案. Hu 等人[101]针对有向无环图结构的 DNN(DAGNN)模型设计了一种二拆分策略 DADS,相较于 Neurosurgeon 具备更好的泛用性. 该工作首先设计了一种对 DAGNN 端边协同推理的图建模方法 ECDI,DAGNN 中每个层为图中的一个顶点,层的输入输出关系由有向边表示,边的权重为中间数据的传输时延. 通过在上述有向无环图上进行最小割,DADS 能够得到总时延最小的二拆分方案. Xu 等人[102]设计了一种在可穿戴设备(端)和智能手机(边)上对 DAGNN 进行二拆分协同加速的策略 DeepWear. 该工作发现 DADS 利用有向无环图的最小割进行的 DAGNN 模型的拆分,在模型规模较大时可能会产生难以接受的决策时延,而且这些 DAGNN 中通常具有大量重复的子图结构. 因此,DeepWear 通过 GRAMI[103]快速挖掘 DAGNN 中的频繁出现的子图结构,将其整体看作有向无环图中的一个节点以大幅度缩小问题规模(例如,GoogLeNet 的 DAGNN 结构从 1096 个节点缩小到 35 个). 最后,DeepWear 也通过最小割寻找推理时延最小(非流式输入数据)或吞吐量最大(流式输入数据)的二拆分方案. Laskaridis 等人[104]针对链式 DNN 设计了一种推理时动态二拆分部署框架 SPINN,在离线时构建模型的静态性能档案,在运行时结合系统状态预测模型的动态性能,进行自适应的 DNN 拆分部署. SPINN 允许用户自定义一系列硬约束(例如,时延小于某个阈值,吞吐高于某个阈值)和软约束(例如,最小化端侧/边侧能量开销),通过检测系统当前状况,根据优先级逐个满足硬约束,在无法满足的情况下将硬约束转化为软约束进行优化,最终得到一个尽可能好的二拆分点. Zhang 等人[105]针对 DAGNN 模型设计了一种相较于 DADS[101]效率更高的二拆分策略 QDMP,优化了 DADS 的端边协同推理的图建模方法,同时考虑了算子融合对模型推理速度的影响. DADS 证明了对有向无环图(DAG)的最小割一定在两个连续的割点之间,从而缩小了基于最小割的二拆分方法的搜索空间. Yao 等人[106]针对层间拆分设计了一种基于压缩感知(Compressive Sensing)的方法,在端侧模型输出的特征图之后增加一个轻量编码器,并在边侧模型之前增加一个对应的解码器,实现了对端边通信效率的优化,且在理论上给出了无损推理的保障. Banitalebi-Dehkordi 等人[107]设计了一种适应用户需求的 DAGNN 二拆分部署框架 Auto-Split,在优化推理时延的基础上,允许用户自定义内存和出错率的约束. 该工作在 QDMP[105]对 DAGNN 端边协同推理的图建模的基础上,引入模型参数量化. 对 DAGNN 的每一层,Auto-Split 允许在满足出错率约束的条件下,尽可能量化参数,减少模型计算时延和数据传输时延,并基于对 DAGNN 图建模的最小割,Auto-Split 选择全部中间数据满足内存约束的,且总推理时延最小的方案作为最终的拆分方案. 表 5 从关键设计和拆分方案生成两个角度总结了层间拆分推理方法.

表 5 层间拆分方法总结

方法	年份	关键设计	拆分方案生成
Kang 等人[98](Neurosurgeon)	2017	回归模型预测链式神经网络开销	线性搜索
Li 等人[99](Edgent)	2019	增加神经网络的早退分支	线性搜索
Eshratifar 等人[100](JointDNN)	2019	考虑算子融合对拆分的影响	图上最短路径
Hu 等人[101](DADS)	2019	考虑有向无环图结构的神经网络	图上最小割
Xu 等人[102](DeepWear)	2019	挖掘图中重复子图结构,缩小拆分问题规模	图上最小割
Zhang 等人[105](QDMP)	2020	理论证明最小割的节点条件,缩小搜索空间	图上最小割
Laskaridis 等人[104](SPINN)	2020	离线构建性能档案,在线自适应拆分	多约束优化
Banitalebi-Dehkordi 等人[107](Auto-Split)	2021	考虑模型参数量化的影响	图上最小割

(2)层内拆分部署. 对 DNN 的层内拆分是基于张量运算的可并行性,通过将原本由单个设备进行的一个 DNN 层的计算,卸载到多个设备并行完成,从而加速 DNN 模型的推理过程. Mao 等人[108]针对 DNN 中的卷积层和全连接层计算,提出了一种并行加速方案 MoDNN. 对于全连接层的计算,MoDNN 设计了一种基于稀疏性的权重矩阵拆分策略,将全连接层的输入输出看作带权无向图的节点,权重矩阵看作邻接矩阵,通过谱聚类拆分其为一系列不相交的子图,对应权重矩阵中一系列行列不相交的非

稀疏子矩阵. MoDNN 保留这些非稀疏部分在本地计算，剩余的稀疏部分根据负载情况卸载到工作节点，由于卸载部分的稀疏性，数据传输量可以显著减少. Mao 等人[109]针对卷积层计算，还提出了一种更细粒度的并行加速方案 MeDNN. 该工作提出了一种对卷积层计算的二维递归拆分方案，遍历当前可用工作节点的全部二组合，选择特征图宽和高中较小的一维作为基准，根据两组工作节点的总计算能力将特征图按比例拆分为两个部分交由两组工作节点计算. MeDNN 递归地按上述过程继续在每组工作节点中进行特征图的拆分和分配，直到每组只剩下一个工作节点. Zhao 等人[110]提出了一种对卷积神经网络(CNN)进行整体并行加速的方案 DeepThings，该工作以 CNN 最后一层的输出数据作为拆分依据，逆向推理出影响这部分输出结果的所有前驱层输入数据的范围，将这些计算分布到多台端设备进行并行计算. 但是，对于输出数据的相邻部分，在前驱层计算中影响它们的数据必定出现重叠部分，并且随着网络深度的增加，将会导致越来越多的重复计算. 因此，DeepThings 通过一个中心节点收集和分发前驱层的重叠部分计算结果进行数据重用，减少了重复计算的时间开销. 但数据重用会导致不同端设备的计算产生数据依赖，这在一定程度上限制了并行性. Du 等人[111]针对组卷积神经网络(GCNN)提出了一种通道维度的并行加速方案 GWPM，同时设计了一种松散耦合的卷积层结构，进一步提升了模型推理的并行性. 该工作基于 GCNN 的组内卷积的无关性，将不同组(每个组包含部分通道的特征图)分配给不同端设备实现并行加速. 考虑到GCNN 无法提取组间特征导致的精度降低，GWPM 引入了三次通道乱序(即对通道重新分组)来缓解上述问题. 此外，该工作还提出了一种基于 PGConv(卷积核大小为 1 的卷积运算，提取通道间特征)和 DWConv(每个组只包含一个通道数据的组卷积，提取单张特征图内的特征)的松散耦合卷积层结构 LCS，使组内卷积能够进一步并行. Du 等人[112]对 GWPM 进行了进一步的优化，提出了 DeCNN. 除了通过 GCNN 对 CNN 进行解耦，实现不同组的并行加速，以及通过三次通道乱序缓解模型精度降低，DeCNN 还通过扩大卷积核的滤波器数量，进一步提升模型的精度. 考虑到三次通道乱序会引入三次数据同步，而且这个同步时间比大多数层的执行时间长，因此 DeCNN 引入样本间的并行，通过两个连续样本的推理重叠，提升整个模型推理的吞吐. Zhang 等人[113]考虑到对卷积层进行层内拆分导致的重复计算，提出了一种基于局部卷积和渐进式训练的，针对 CNN 的并行加速方案 ADCNN. 卷积层的层内拆分并行会导致两种可能的后果：① 特征子图的边缘部分在前驱层计算中存在重叠部分，导致重复计算；或 ② 对重叠部分进行数据重用减少重复计算，但引入设备间数据依赖，限制并行性. 该工作设计了一种完全可分解的空间分区方案 FDSP，舍弃重叠部分的计算结果，对每个特征子图的边缘使用零填充，使得卷积层的计算能够拆分为完全独立的子任务，不存在重复计算和数据依赖. 由此引发的精度损失，ADCNN 通过在原有模型基础上进行渐进式再训练进行恢复，并对输出结果量化以进一步加速推理. Zeng 等人[114]在 MoDNN 对卷积层的一维线性拆分方案 BODP 的基础上，提出了一种适配设备内存和计算能力，并优化拆分边缘数据传输的卷积层计算拆分方案 CoEdge. CoEdge 的优化目标是获得模型每层的层内一维拆分并行方案，最小化整个模型运行的能量开销. 该工作证明了在执行时延约束、每个设备内存约束和上述填充约束下，最小化整个模型运行的能量开销的问题 $P1$ 是一个 NP-hard 问题，并证明忽略填充约束时，问题退化为一个线性规划问题 $P2$. 因此，CoEdge 可以通过检查 $P2$ 的解是否满足填充约束，迭代逼近 $P1$ 的解. 表 6 从关键设计和拆分方案生成两个角度总结层内拆分推理方法.

表 6　层内拆分方法总结

方法	年份	关键设计	拆分方案生成
Mao 等人[108](MoDNN)	2017	基于稀疏性的权重矩阵拆分	子图谱聚类
Mao 等人[109](MeDNN)	2017	二维递归拆分卷积层	算力比例拆分
Zhao 等人[110](DeepThings)	2018	卷积输出逆向推出前驱层输入范围	融合平铺分区
Du 等人[111](GWPM)	2020	提高并行性的松耦合卷积结构	卷积通道分组
Du 等人[112](DeCNN)	2020	通道乱序以及扩大卷积核数量	卷积通道分组
Zhang 等人[113](ADCNN)	2020	局部卷积和渐进式训练	特征空间分区
Zeng 等人[114](CoEdge)	2020	增加对特征子图填充的约束	线性规划

(3)层间及层内拆分部署. 一些方法提出了同时考虑层间拆分和层内拆分的并行加速策略. Yang 等人[115]考虑了存在多个异构端设备的场景(例如,智慧家庭),针对链式 DNN 模型设计了一种流水线并行加速方案 PICO,通过将 DNN 划分为多个阶段(DNN 中的连续的几层),每个阶段卸载到一个设备子集并行执行,从而最大化系统吞吐. PICO 首先考虑每个设备的计算能力相同的情况,也即为全部设备计算能力的均值,得到一个初步的阶段划分(层间划分),再基于贪心的策略,将设备按计算能力由大到小迭代地分配给每个阶段,形成一系列"阶段-设备集"对,最后对每个阶段进行层内拆分并行. Zhang 等人[116]则是在多个异构端设备的场景下,针对 DAGNN 设计了一种基于同步点的层间多拆分方案 PSS,并对拆分出的每组连续的 DNN 层,设计了一种跨层的特征图拆分方案 AIR. 该工作考虑到对于诸如 ResNet、GoogleNet 的 DAGNN,以层为单位进行多设备并行会引入频繁的同步等待,因此 PSS 搜索网络中的全部同步点,根据同步点拆分 DAGNN 为一系列 DNN 块. AIR 以 DNN 块为单位,根据计算能力拆分 DNN 块最后一层的输出,再递归地搜索所有前序层中相关的特征子图,分配到每个端设备进行多设备并行. Huang 等人[117]利用可解释人工智能(Explainable AI)技术显式地控制模型特征的稀疏性,进而为部署阶段的切分留出更大的计算和通信优化空间,并提出了一种名为 AgileNN 的技术,在推理阶段弱设备侧只需要本地处理少量重要的特征,大量不重要的特征则被传输到边缘节点. 显然 AgileNN 的推理模式同时考虑了层间拆分和层内拆分方法,其核心创新点在于从特征重要性角度进行端边拆分,并在训练阶段为拆分优化做准备. 表 7 从关键设计和层间、层内拆分方案生成三个角度总结了结合层内和层间拆分的协同推理方法.

表 7 层间、层内拆分方法总结

方法	年份	关键设计	层间拆分方案生成	层内拆分方案生成
Yang 等人[115](PICO)	2021	按连续层划分为多阶段	按全部算力均值划分	按设备算力贪心迭代
Zhang 等人[116](PSS&AIR)	2021	按网络内的同步点分块	按网络分块划分	递归搜索前序相关特征子图
Huang 等人[117](AgileNN)	2022	利用XAI控制模型稀疏性	重要特征划分至端侧,不重要特征划分至边侧	

5.3 能量资源优化:加速器

为智能推理计算设计特殊的硬件加速器受到了广泛的关注,Wu 等人[118]给出了基于 FPGA 的模型推理相关工作的综述,Moolchandani 等人[119]对基于 ASIC(Application-Specific Integrated Circuit)的卷积神经网络推理给出了综述. 本综述在此将相关技术按其可应用的神经网络类型进行细粒度划分,从通用神经网络、卷积神经网络、图神经网络以及较为前沿的光子神经网络四个角度分别介绍.

(1)通用神经网络. 针对通用机器学习模型加速器,中国科学院计算技术研究所提出了采用流式处理乘加树架构的 DianNao[120]、DaDianNao[121]和 PuDianNao[122],以及采用类脉动阵列架构的 ShiDianNao[123]系列工作. DianNao 是这一系列工作的架构基础,主要包含输入神经元缓冲区、输出神经元缓冲区、突触权重缓冲区、神经计算单元、控制逻辑以及直接内存访问(Direct Memory Access,DMA)模块. DaDianNao 提出了一种将 DianNao 扩展为多芯片的架构,支持将较大的模型载入片上内存,在功耗和硅面积效率均表现出相较于 GPU 的优势. ShiDianNao 通过将加速器与传感器直接相连,减少了内存通信的开销,更适合于边缘推理任务. PuDianNao 将加速神经网络模型的架构设计扩展为支持更通用机器学习模型. Han 等人[124]观察到从 DRAM 取权重操作是模型推理的能耗的主要因素,因此提出针对压缩后的模型的推理硬件 EIE,将网络权重载入 SRAM 并对稀疏性和权重共享进行针对性优化,相较于 CPU 和 GPU 实现了 24000 和 3400 倍的能量效率. Abdelfattah 等人[125]针对 FPGA 硬件上的模型推理设计了一种覆盖(Overlay,用于 FPGA 虚拟化的方法之一),并实现了一个将神经网络映射至覆盖的计算图编译器. 该工作基于超长指令字(Very-Long Instruction Word)网络将 FPGA 的可配置参数分为运行时和编译时两类,因此支持在编译时优化架构性能以及在运行时加速不同的神经网络模型(只需对覆盖进行重编程而无需重新配置 FPGA).

(2)卷积神经网络. Zhang等人[126]和Niu等人[127]探索了基于 FPGA 对于频/谱域卷积神经网络的加速. 基于快速傅里叶变换,能够将原本在空间域的卷积计算转化成更轻量化的频域 Hadamard 乘积,从而加速模型推理. Wei 等人[128]设计了一种脉动阵列(Systolic Array,由紧耦合的数据处理单元组成的同构网络)加速卷积神经网络在 FPGA 上的推理,

并且实现了从 C 语言代码生成 FPGA 卷积神经网络设计的自动化. Zhang 等人[129]为卷积神经网络在 FPGA 上的推理设计了 Caffeine，将卷积层和全连接层统一为卷积矩阵乘法表达并根据神经网络配置应用以权重或输入为主的映射方法. Caffeine 优化了卷积计算的多级数据并行和流水线并行，同时组合使用了片上和片外数据重组以提高内存打款利用率. Choudhury 等人[130]针对 FPGA 上的卷积神经网络设计了一种覆盖，能够通过对每个层的控制字来即时配置，并且利用了多种卷积层并行（包括过滤器并行、卷积核并行、通道并行）的优化.

(3) 图神经网络. Yan 等人[131]为图卷积神经网络（Graph Convolutional Networks，GCNs）设计了一种混合硬件框架 HyGCN，将图卷积推理抽象为聚合（聚合邻居节点的特征）和组合（神经网络对聚合后的特征进行转化）两个阶段，分别设计了优化数据重用和并行矩阵乘法的聚合和组合加速器. Zhang 等人[132]针对 FPGA 上运行图卷积神经网络推理设计了 BoostGCN，提出一种以图节点为中心的内存效率更高的特征聚合方法，并设计了支持流水线执行推理的 FPGA 硬件架构.

(4) 光子神经网络. Sludds 等人[133]提出了一种光子神经网络推理的边缘架构 NetCast，基于云的智能收发器（用于数字-模拟信号转换）将权重数据流式传输到边缘设备，从而实现超高效的光子神经网络推理.

6　讨　论

在上面的三个章节中，我们从三个模块（传感器数据、智能模型、物联网硬件）详细介绍了优化五类资源（时间、内存、带宽、能量、存储）的八类技术. 这些方法提供了丰富的优化“点”，但我们仍缺少将他们之间的关联起来的逻辑“线”. 因此，本节首先提出了一个通用的优化流程，将已经介绍过的技术进行串联，给出了易于实践的工作流图（见图 8）. 之后，我们根据当前技术现状，分析出四个仍存在较大技术挑战的研究方向，希望能够为相关研究人员提供有价值的科研思路参考.

6.1　通用优化流程

基于对现有方法的分析以及我们的实际优化经验，我们给出一个四步骤的通用优化流程，如图 8 所示. (1) 首先我们需要决定部署方式，这一过程通过判断两个条件完成. 其一是数据能否从采集端侧传出，这一条件一方面依赖应用场景的隐私要求，如果隐私要求较高则只能进行端侧的推理部署. 另一方面，如果网络波动较严重，数据无法传出且要求离线可推理的情况下，也需要进行端侧的推理部署. 如果数据能够正常传出，我们需要考虑端侧设备能否满足推理所需的性能要求，例如延迟是否在限制内以及内耗对于端侧是否可接受等. 如果端侧可以满足性能要求，则仍优先考虑端侧推理部署. 之后我们需要考虑第三个条件，即模型能否进行拆分. 这一条件依赖于模型的使用权限，包括黑盒、灰盒和白盒权限. 如果应用开发者仅能够部署使用黑盒推理 API，则只能考虑边侧推理部署. 只有两个条件都满足时，我们才能使用切分推理部署的方式；(2) 第二步，我们称为预处理阶段的优化，即在执行推理程序之前. 在这一阶段，切分推理部署方式下可以应用协同推理技术，以决定具体的（静态或动态的）模型切分方式；切分推理部署和边侧推理部署方式都可以应用数据编码技术，进一步降低端边通信的开销. 后续的优化对于三类部署方式是相同的，我们首先可以应

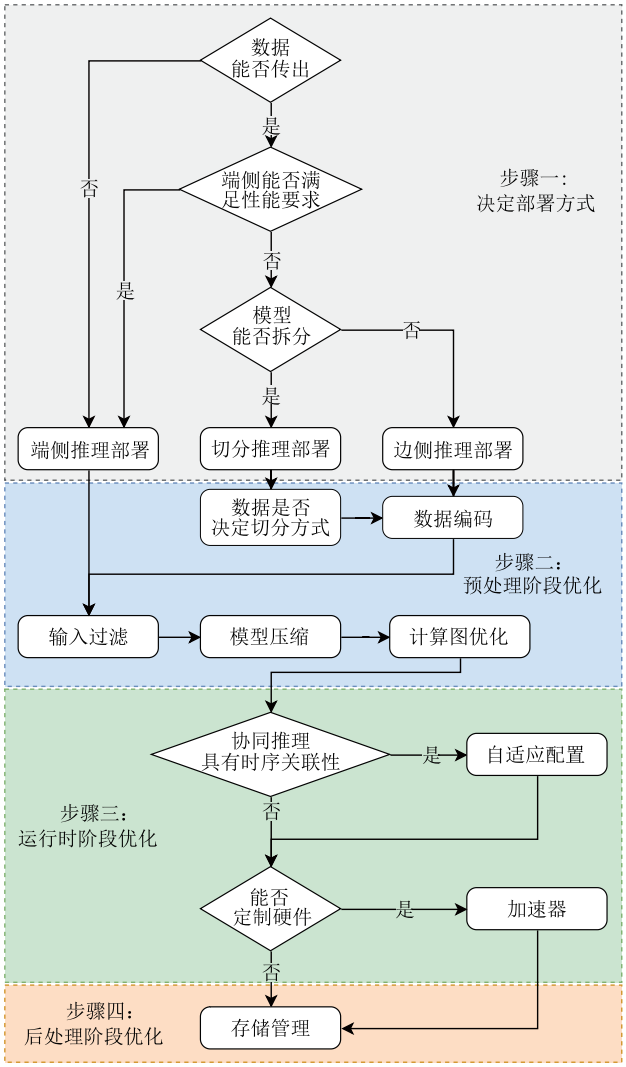

图 8　通用优化流程

用输入过滤技术对待处理的数据(端侧推理处理本地采集的数据、切分推理处理端侧部分模型传输来的中间激活、边侧推理处理端侧传输的数据)进行筛选.完成对于“输入数据”的优化之后,我们开始对执行计算的“推理模型”进行优化.首先可以尝试对模型的静态架构和参数进行压缩,得到轻量化版本的神经网络模型.关于模型能否进行有效压缩,通常我们需要考量参数的稀疏性以及任务分布的局部性.再之后,则可以应用计算图优化技术,对优化后的静态架构和参数生成进一步加速的“运行时”可执行模型推理程序;(3)第三步,在运行时阶段进行的在线推理优化.这一阶段首先我们要判断待推理的数据是否具有时序上的关联性,即推理模式是否具有规律性.如果该条件成立,一般而言我们才能应用自适应推理技术,动态地根据数据在线调整推理工作流.第二个条件是判断能否定制硬件,如果可能,则可以考虑设计特定加速器硬件,以提高推理的能量效率;(4)第四步,在后处理阶段的优化,这一阶段我们可以应用存储管理技术优化存储资源效率.值得注意的是,将多种优化方法结合使用当前尚无广泛的经验,例如在应用了新的数据编码技术后,后续的输入过滤技术需要做对应的适配;又比如对于需要对模型进行切分的协同推理,模型压缩技术也需要相应的针对性设计.总结出一整套实用的技术方案需要更多的组合式的测试验证.

6.2 未来研究方向

通过组合使用现有优化技术,能够解决许多推理系统的效率瓶颈问题.但我们发现,仍存在四个尚未充分探索并亟须研究的方向.如表 8 所示,分别是考虑如何扩展更多的传感器数据源、如何协调多个边缘节点进行推理分析、在后处理阶段进行更多的优化以及大模型在智能物联网中的推理.

表 8 未来研究方向总结

研究方向	潜在的挑战	可能的方案	应用的前景
传感器可扩展性	数据处理和存储压力异构传感器集成	自适应数据过滤和压缩	智慧城市 农业智能监测
多边缘协同推理	分布式协作算法复杂性 数据异构性和集成	异构数据融合处理	智慧交通 个性化医疗康养
后处理阶段优化	环境动态性高 数据分布漂移问题	推理模式预测 增量式推理 推理模型终身学习	工业自动化 全天候实时决策
大模型推理	网络带宽与数据传输压力 多智能体协同复杂性	数据深度压缩与编码 端边云协同分层推理	具身智能机器人 移动端智能助手

(1)传感器可扩展性.在智能物联网场景中,传感器可扩展性指的是推理系统能够在需要时高效地增加接入的传感器数量.随着物联网的发展,连接设备和传感器的数量不断增加,导致产生的数据量呈指数级增长,这带来了数据处理和存储的巨大压力以及集成异构传感器的技术挑战.表 9 总结了常见物联网场景中传感器的类型和数量规模.为了有效地监测、分析和应用这些数据,系统必须能够支持更多数量的传感器.在近日的一项工作中,Yuan 等人[134]发现在智能视频分析系统中增加传感器(摄像头)时,视频解码模块会成为端到端并发度的瓶颈,并研究了如何通过筛选解码前的视频数据包来提高视频分析系统对摄像头数量的扩展性.然而,物联网应用广泛,涵盖从智能城市和工业自动化到健康监测和农业领域.不同应用场景对传感器类型和数量的需求各不相同,我们需要研究不同传感器、多样的分析任务下的可扩展性问题.例如对于音视频、无线信号等传感数据,我们可以利用其天然的时序关联性对数据进行自适应过滤.以及利用推理任务在特定物联网数据域中分布的狭窄性(相较于原始的较广的训练数据域),对数据进行较传统方案更大程度的压缩.探索如何提高传感器可扩展性对提高智慧城市、农业智能监测等 AIoT 应用的规模起着关键作用.

表 9 常见智能物联网场景中传感器类型和数量规模

智能物联网场景	传感器类型	传感器数量
车联网	车载摄像头、运动传感器、雷达、激光雷达、超声波传感器、IMU、GPS	几十/辆
智慧康养	IP 摄像头、穿戴式运动传感器、生物医学(心率、氧气水平等)/温湿度传感器	数十/房间
智慧楼宇	IP 摄像头、能耗监测传感器、烟雾/火灾传感器、运动传感器、温湿度传感器	数百/建筑
工业自动化	IP 摄像头、温湿度/电流/电压/振动传感器/化学/液位传感器	数百-数千/工厂
园区安防	IP 摄像头、红外传感器、运动传感器、烟雾/火灾传感器	数千/园区
环境监控	无人机摄像头、空气传感器、温湿度传感器、水质传感器	数千-数万/城市
交通管理	IP 摄像头、雷达、地磁传感器、停车传感器、噪声传感器	数万-数十万/城市

(2)多边缘协同推理. 现有的端边协同工作大多考虑一种多个传感器和单个边缘节点的计算架构,然而在大规模 AIoT 部署中,将所有数据发送到中心边缘节点可能会导致系统可靠性降低. 多边缘节点协同推理指的是将分布在不同位置的边缘节点(包括传感器、设备等)通过合作进行推理,以提高整个系统的智能性和效率. 但另一方面,不同边缘节点可能携带着异构的数据,例如图像、声音、传感器数据等. 一些不同类型的数据需要被整合和综合,方能提供更丰富、全面的信息基础,用于更准确的分析. 实现这一目标需要解决潜在的分布式协作算法复杂性高,以及数据异构性带来的集成困难等技术挑战. 一种可能的解决方案是通过异构数据融合实现多边协同处理,即将多个边缘节点的数据在特征空间上进行融合,进而为多边算力协同提供优化机会. 实现多边协同推理可以增强系统的容错性和可靠性,即使某些节点失效,其他节点仍然可以合作进行推理,确保系统的正常运行,例如在城市级智慧交通中,多边缘协同推理能够提高交通管理效率和可靠性. 通过在多个边缘节点上进行协同推理,也可以实现更智能的决策和行为. 多边协同推理技术的研究将对于城市级智慧交通以及个性化医疗康养等涉及多边的 AIoT 场景大有帮助.

(3)后处理阶段优化. 从我们整理出的通用优化流程图中可以看到,预处理阶段和运行时阶段都有丰富的研究工作,优化了时间、内存、带宽、能量等关键资源,但在后处理阶段只有存储管理这一类技术. 在后处理阶段进行推理优化研究,主要涉及的技术挑战包括高度动态的智能物联网环境,以及数据分布漂移带来的模式改变问题. 我们认为后处理阶段实际上还有许多值得研究的方向,包括:① 推理模式预测,即研究如何基于推理分析结果预测计算的模式和规律,进而对未来的任务执行调度进行优化;② 增量式推理,即考虑在已有数据基础上增量地进行推理,而不是每次都重新分析所有数据,这可以显著减少计算负担,适用于实时决策和大规模数据集(例如工业自动化系统中持续产生的生产数据);③ 推理模型终身学习,即针对不断变化的环境和数据,研究如何快速对模型进行增量式或在线的重训练,以提高模型的适应性,进一步挖掘模型轻量化的潜能. 探索后处理阶段优化方案,会提高 AIoT 推理系统对于多变环境及数据分布的场景,将更好地赋能包括工业自动化以及全天候实时决策等关键应用.

(4)大模型推理. 智能物联网场景下大模型推理任务主要面临的网络带宽与数据传输压力以及多智能体协同复杂性这两大挑战. 在网络带宽与数据传输方面,未来的研究应着眼于数据深度压缩和编码技术的创新,通过开发高效的数据压缩算法和智能编码机制,显著减少在设备和云端之间传输的数据量,从而缓解网络带宽的压力. 例如,新的压缩算法可以在保持数据精度的同时大幅减少数据体积,而智能编码技术则能够优化数据传输过程中的效率和可靠性. 这些技术的应用将在具身智能机器人和移动端智能助手中尤为重要,这些设备需要在带宽受限的环境中快速传输和处理大量数据. 与此同时,多智能体协同复杂性问题将通过端边云协同分层推理来解决. 未来的研究需要探索如何在这种架构下有效分配计算任务,使得具身智能机器人和移动端智能助手能够在不同层次上高效协同. 具体来说,轻量级的初级推理任务将被部署在边缘设备上,以实现低延迟和高响应速度,而复杂和资源密集型的推理任务则通过云端进行处理,确保计算资源的最优使用和推理结果的高准确性. 在具身智能机器人应用中,这意味着机器人能够在本地实时处理环境数据和执行基本任务,同时通过云端进行复杂的决策和学习. 在移动端智能助手中,这种架构可以使助手在本地快速响应用户需求,同时利用云端的强大计算能力进行深度分析和个性化服务. 为了实现这些目标,未来的研究将需要在算法优化、系统架构设计和通信协议创新等多个层面上进行深入探索. 通过综合运用数据深度压缩和编码技术以及端边云协同分层推理架构,不仅能够大幅提升大模型在智能物联网中的应用性能,还能显著改善用户体验,为具身智能机器人和移动端智能助手等关键应用提供更智能、更高效的解决方案. 这将推动智能物联网技术的进一步发展和普及,开创更加智能互联的未来.

7　总　结

智能物联网已经广泛应用于智慧城市和工业自动化等多个领域. 在这些应用中,模型推理是实现智能决策和响应的核心技术. 本文综述了在智能物联网中优化模型推理资源开销的相关技术,并深入探讨了它们在资源效率优化方面的特点. 通过从传感器数据、智能模型和物联网硬件三大模块及五类关键资源的角度出发,本文提出了新的技术分类方法,并首次引入了一套通用的优化流程. 这套流程旨在帮助研发人员定位并优化推理过程中的效率瓶颈.

最后,本文还讨论了智能物联网推理效率的未来研究方向,指出了四个值得关注的领域,包括传感器可扩展性、多边协同推理、后处理优化以及大模型在智能物联网中的推理.这些研究方向将为进一步提升智能物联网系统的资源效率和应用深度与广度提供有价值的参考.

参 考 文 献

[1] Wang Xiao-Jing, Zhang Jin. A review of Internet of Things research. Journal of Liaoning University: Natural Sciences Edition, 2010, 37(1): 37-39(in Chinese)
(王晓静, 张晋. 物联网研究综述. 辽宁大学学报:自然科学版, 2010, 37(1): 37-39)

[2] AIoT Market. https://www.gminsights.com/industry-analysis/aiot-market

[3] Kingsoft Cloud, iResearch. China Intelligent Internet of Things (AIoT) White Paper. 2020(in Chinese)
(金山云, 艾瑞咨询. 中国智能物联网(AIoT)白皮书. 2020)

[4] Huawei. GIV (Global Industry Vision) Report. 2021(in Chinese)
(华为. 全球产业愿景报告. 2021)

[5] Cheng Yu, Wang Duo, Zhou Pan, Zhang Tao. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282, 2017

[6] Hoefler T, Alistarh D, Ben-Nun T, et al. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. Journal of Machine Learning Research, 2021, 22(1): 1-124

[7] Han Yizeng, Huang Gao, Song Shiji, et al. Dynamic neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(11): 7436-7456

[8] Matsubara Y, Levorato M, Restuccia F. Split computing and early exiting for deep learning applications: Survey and research challenges. ACM Computing Surveys, 2022, 55(5): 1-30

[9] Thiruvathukal G K, Lu Y-H, Kim J, et al. Low-Power Computer Vision: Improve the Efficiency of Artificial Intelligence. Boca Raton, USA: CRC Press, 2022

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

[11] Murshed M G S, Murphy C, Hou D, et al. Machine learning at the network edge: A survey. ACM Computing Surveys, 2021, 54(8): 1-37

[12] Zhang Jing, Tao Dacheng. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. IEEE Internet of Things Journal, 2020, 8(10): 7789-7817

[13] Wu Ji-Yi, Li Wen-Juan, Cao Jian, et al. A review of research on intelligent Internet of Things (AIoT). Telecommunications Science, 2021, 37(8): 1-17(in Chinese)
(吴吉义, 李文娟, 曹健等. 智能物联网 AIoT 研究综述. 电信科学, 2021, 37(8): 1-17)

[14] Yang Zheng, He Xiao-Wu, Wu Jia-Hang, et al. Edge computing technologies for streaming video analytics. Scientia Sinica Informationis, 2022, 52(1): 1-53(in Chinese)
(杨铮, 贺骁武, 吴家行等. 面向实时视频流分析的边缘计算技术. 中国科学: 信息科学, 2022, 52(1): 1-53)

[15] Liu Sicong, Guo Bin, Fang Cheng, et al. Enabling resource-efficient AIoT system with cross-level optimization: A survey. IEEE Communications Surveys & Tutorials, 2024, 26(1): 389-427

[16] Canel C, Kim T, Zhou G, et al. Scaling video analytics on constrained edge nodes. Proceedings of Machine Learning and Systems, 2019, 1(1): 406-417

[17] Li Yuanqi, Padmanabhan A, Zhao Pengzhan, et al. Reducto: On-camera filtering for resource-efficient real-time video analytics//Proceedings of the ACM SIGCOMM Conference. Virtual Event, USA, 2020: 359-376

[18] Tchaye-Kondi J, Zhai Yanlong, Shen Jun, et al. SmartFilter: An edge system for real-time application-guided video frames filtering. IEEE Internet of Things Journal, 2022, 9(23): 23772-23785

[19] Yuan Mu, Zhang Lan, He Fengxiang, et al. InFi: End-to-end learnable input filter for resource-efficient mobile-centric inference//Proceedings of the 28th Annual International Conference on Mobile Computing and Networking (MobiCom'22). Sydney, Australia, 2022: 228-241

[20] Jiang Shiqi, Lin Zhiqi, Li Yuanchun, et al. Flexible high-resolution object detection on edge devices with tunable latency//Proceedings of the 27th Annual International Conference on Mobile Computing and Networking. New Orleans, USA, 2021: 559-572

[21] Zhang Wuyang, He Zhezhi, Liu Luyang, et al. Elf: Accelerate high-resolution mobile deep vision with content-aware parallel offloading//Proceedings of the 27th Annual International Conference on Mobile Computing and Networking (MobiCom'21). New Orleans, USA, 2021: 201-214

[22] Chen T Y-H, Balakrishnan H, Ravindranath L, Bahl P. Glimpse: Continuous, real-time object recognition on mobile devices//Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. Seoul, Republic of Korea, 2016: 155-168

[23] Guo Peizhen, Hu Wenjun. Potluck: Cross-application approximate deduplication for computation-intensive mobile applications//Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'18). Williamsburg, USA, 2018: 271-284

[24] Guo Peizhen, Hu Bo, Li Rui, Hu Wenjun. FoggyCache: Cross-device approximate computation reuse//Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom'18). New Delhi, India, 2018: 19-34

[25] Ning Lin, Shen Xipeng. Deep reuse: Streamline CNN inference on the fly via coarse-grained computation reuse//Proceedings of the ACM International Conference on Supercomputing. Phoenix, USA, 2019: 438-448

[26] Xu Mengwei, Zhu Mengze, Liu Yunxin, et al. DeepCache: Principled cache for mobile deep vision//Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom'18). New Delhi, India, 2018: 129-144

[27] Wu Ruofan, Zhang Feng, Guan Jiawei, et al. DREW: Efficient Winograd CNN inference with deep reuse//Proceedings of the ACM Web Conference 2022. Berlin, Germany, 2022: 1807-1816

[28] Lavin A, Gray S. Fast algorithms for convolutional neural networks//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 4013-4021

[29] Liu Chuan-Hong, Guo Cai-Li, Yang Yang, et al. Semantic communication method for intelligent tasks in artificial intelligence-based Internet of Things. Journal on Communications, 2021, 42(11): 97-108(in Chinese)
(刘传宏，郭彩丽，杨洋等. 人工智能物联网中面向智能任务的语义通信方法. 通信学报, 2021, 42(11): 97-108)

[30] Xie Xiufeng, Kim Kyu-Han. Source compression with bounded DNN perception loss for IoT edge computer vision//Proceedings of the 25th Annual International Conference on Mobile Computing and Networking(MobiCom'19). Los Cabos, Mexico, 2019: 1-16

[31] Lu Guo, Ouyang Wanli, Xu Dong, et al. DVC: An end-to-end deep video compression framework//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). 2019: 11006-11015

[32] Hu Pan, Im Junha, Asgar Z, Katti S. Starfish: Resilient image compression for AIoT cameras//Proceedings of the 18th Conference on Embedded Networked Sensor Systems. 2020: 395-408

[33] Deng Kaikai, Zhao Dong, Han Qiaoyue, et al. Geryon: Edge assisted real-time and robust object detection on drones via mmWave radar and camera fusion. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2022, 6(3): 1-27

[34] Li Jiahao, Li Bin, Lu Yan. Deep contextual video compression. Advances in Neural Information Processing Systems, Virtual Event, 2021, 34(1): 18114-18125

[35] Du Kuntai, Zhang Qizheng, Arapin A, et al. AccMPEG: Optimizing video encoding for accurate video analytics. Proceedings of Machine Learning and Systems, 2022, 4(1): 450-466

[36] Xiao Xuedou, Zhang Juecheng, Wang Wei, et al. DNN-driven compressive offloading for edge-assisted semantic video segmentation//Proceedings of the IEEE INFOCOM 2022. 2022: 1888-1897

[37] Ma Siwei, Zhang Xinfeng, Jia Chuanmin, et al. Image and video compression with neural networks: A review. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(6): 1683-1698

[38] Poms A, Crichton W, Hanrahan P, Fatahalian K. Scanner: Efficient video analysis at scale. ACM Transactions on Graphics, 2018, 37(4): 1-13

[39] Haynes B, Daum M, He Dong, et al. VSS: A storage system for video analytics//Proceedings of the 2021 International Conference on Management of Data. 2021: 685-696

[40] Daum M, Haynes B, He Dong, et al. TASM: A tile-based storage manager for video analytics//Proceedings of the 2021 IEEE 37th International Conference on Data Engineering. 2021: 1775-1786

[41] Hu Bo, Guo Peizhen, Hu Wenjun. Video-zilla: An indexing layer for large-scale video analytics//Proceedings of the 2022 International Conference on Management of Data. 2022: 1905-1919

[42] Zhang Haoyu, Ananthanarayanan G, Bodik P, et al. Live video analytics at scale with approximation and delay-tolerance //Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17). Boston, USA, 2017: 377-392

[43] Jiang Junchen, Ananthanarayanan G, Bodik P, et al. Chameleon: Scalable adaptation of video analytics//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. 2018: 253-266

[44] Ali A, Pinciroli R, Yan Feng, Smirni E. BATCH: Machine learning inference serving on serverless platforms with adaptive batching//Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis(SC20). 2020: 1-15

[45] Crankshaw D, Sela G-E, Mo Xiangxi, et al. InferLine: Latency-aware provisioning and scaling for prediction serving pipelines//Proceedings of the 11th ACM Symposium on Cloud Computing. 2020: 477-491

[46] Kang D, Mathur A, Veeramacheneni T, et al. Jointly optimizing preprocessing and inference for DNN-based visual analytics. Proceedings of the VLDB Endowment, 2020, 14(2): 87-100

[47] Liu Shengzhong, Yao Shuochao, Fu Xinzhe, et al. On removing algorithmic priority inversion from mission-critical machine inference pipelines//Proceedings of the 2020 IEEE Real-Time Systems Symposium(RTSS). 2020: 319-332

[48] Li Yuyang, Wu Yawen, Zhang Xincheng, et al. Energy-aware adaptive multi-exit neural network inference implementation for a millimeter-scale sensing system. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2022, 30(7): 849-859

[49] Yuan Mu, Zhang Lan, Li Xiang-Yang, Xiong Hui. Comprehensive and efficient data labeling via adaptive model scheduling //Proceedings of the 2020 IEEE 36th International Conference on Data Engineering. 2020: 1858-1861

[50] Yuan Mu, Zhang Lan, Li Xiang-Yang, et al. Adaptive model scheduling for resource-efficient data labeling. ACM Transactions on Knowledge Discovery from Data, 2022, 16(4): 1-22

[51] Huang Gao, Chen Danlu, Li Tianhong, et al. Multi-scale dense networks for resource efficient image classification//Proceedings of the International Conference on Learning Representations. Vancouver, Canada, 2018

[52] Wang Xin, Yu Fisher, Dou Zi-Yi, et al. SkipNet: Learning dynamic routing in convolutional networks//Proceedings of the European Conference on Computer Vision. 2018: 409-424

[53] Campos V, Jou B, Giro-i-Nieto X, et al. Skip RNN: Learning to skip state updates in recurrent neural networks//Proceedings of the International Conference on Learning Representations. Vancouver, Canada, 2018

[54] Wang Haizhou, Li Liying, Cui Yangguang, et al. MBSNN: A multi-branch scalable neural network for resource-constrained IoT devices. Journal of Systems Architecture, 2023, 142(1): 1-10

[55] Lu Yao, Chowdhery A, Kandula S. Optasia: A relational platform for efficient large-scale video analytics//Proceedings of the ACM Symposium on Cloud Computing (SoCC'16). 2016: 57-70

[56] Kang D, Emmons J, Abuzaid F, et al. NoScope: Optimizing neural network queries over video at scale. Proceedings of the VLDB Endowment, 2017, 10(11): 1586-1597

[57] Hsieh K, Ananthanarayanan G, Bodik P, et al. Focus: Querying large video datasets with low latency and low cost //Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18). Carlsbad, USA, 2018: 269-286

[58] Kang D, Bailis P, Zaharia M. BlazeIt: Optimizing declarative aggregation and limit queries for neural network-based video analytics. Proceedings of the VLDB Endowment, 2019, 13(4): 533-546

[59] Romero F, Hauswald J, Partap A, et al. Optimizing video analytics with declarative model relationships. Proceedings of the VLDB Endowment, 2022, 16(3): 447-460

[60] Xu Zhuangdi, Kakkar G T, Arulraj J, Ramachandran U. EVA: A symbolic approach to accelerating exploratory video analytics with materialized views//Proceedings of the International Conference on Management of Data. 2022: 602-616

[61] Chen Yueting, Koudas N, Yu Xiaohui, Yu Ziqiang. Spatial and temporal constrained ranked retrieval over videos. Proceedings of the VLDB Endowment, 2022, 15(11): 3226-3239

[62] Kang D, Guibas J, Bailis P D, et al. TASTI: Semantic indexes for machine learning-based queries over unstructured data//Proceedings of the 2022 International Conference on Management of Data (SIGMOD'22). 2022: 1934-1947

[63] Cao Jiashen, Sarkar K, Hadidi R, et al. FiGO: Fine-grained query optimization in video analytics//Proceedings of the 2022 International Conference on Management of Data (SIGMOD'22). 2022: 559-572

[64] Shen Haichen, Han S, Philipose M, Krishnamurthy A. Fast video classification via adaptive cascading of deep models//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017: 2197-2205

[65] Anderson M R, Cafarella M, Ros G, Wenisch T F. Physical representation-based predicate optimization for a visual analytics database//Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE). 2019: 1466-1477

[66] Chakrabarti A, Guérin R, Lu Chenyang, Liu Jiangnan. Real-time edge classification: Optimal offloading under token bucket constraints//Proceedings of the 2021 IEEE/ACM Symposium on Edge Computing (SEC). 2021: 41-54

[67] Heinanen J, Guérin R. A single rate three color marker. Technical Report: RFC 2679, USA, 1999

[68] Yuan Mu, Zhang Lan, Li Xiang-Yang. MLink: Linking black-box models for collaborative multi-model inference. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(9): 9475-9483

[69] Hwang J, Kim M, Kim D, et al. CoVA: Exploiting compressed-domain analysis to accelerate video analytics//Proceedings of the USENIX Annual Technical Conference. Carlsbad, USA, 2022: 707-722

[70] Ghosh A, Iyengar S, Lee S, et al. REACT: Streaming video analytics on the edge with asynchronous cloud support//Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation (IoTDI). 2023

[71] Zhang Xiangyu, Zhou Xinyu, Lin Mengxiao, Sun Jian. ShuffleNet: An extremely efficient convolutional neural network for mobile devices//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018: 6848-6856

[72] Ma Ningning, Zhang Xiangyu, Zheng Hai-Tao, Sun Jian. ShuffleNet V2: Practical guidelines for efficient CNN architecture design//Proceedings of the European Conference on Computer Vision. Munich, Germany, 2018: 116-131

[73] Lin Ji, Chen Wei-Ming, Lin Yujun, et al. MCUNet: Tiny deep learning on IoT devices. Advances in Neural Information Processing Systems, 2020, 33(1): 11711-11722

[74] Lin Ji, Chen Wei-Ming, Cai Han, et al. MCUNetV2: Memory-efficient patch-based inference for tiny deep learning. arXiv preprint arXiv:2110.15352, 2021

[75] Liu Sicong, Lin Yingyan, Zhou Zimu, et al. On-demand deep model compression for mobile devices: A usage-driven model selection framework//Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services. Munich, Germany, 2018: 389-400

[76] Gao Xitong, Zhao Yiren, Dudziak Ł, et al. Dynamic channel pruning: Feature boosting and suppression. arXiv preprint arXiv:1810.05331, 2018

[77] Cai Han, Gan Chuang, Wang Tianzhe, et al. Once-for-all: Train one network and specialize it for efficient deployment. arXiv preprint arXiv:1908.09791, 2020

[78] Liu Sicong, Guo Bin, Ma Ke, et al. Adaspring: Context-adaptive and runtime-evolutionary deep model compression for mobile applications. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2021, 5(1): 1-22

[79] Yuan Chunyu, Agaian S S. A comprehensive review of binary neural network. Artificial Intelligence Review, 2023, 56(11): 1-65

[80] Lin Xiaofan, Zhao Cong, Pan Wei. Towards accurate binary convolutional neural network. Advances in Neural Information Processing Systems, 2017, 30(1): 1-9

[81] Rastegari M, Ordonez V, Redmon J, Farhadi A. XNOR-Net: ImageNet classification using binary convolutional neural networks//Proceedings of the Computer Vision (ECCV 2016). Amsterdam, The Netherlands, 2016: 525-542

[82] Zhu Shien, Duong L H K, Liu Weichen. XORNet: An efficient computation pipeline for binary neural network inference on edge devices//Proceedings of the 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS). 2020: 124-131

[83] Lin Mingbao, Ji Rongrong, Xu Zihan, et al. Rotated binary neural network. Advances in Neural Information Processing Systems, 2020, 33(1): 7474-7485

[84] Chen Gang, He Shengyu, Meng Haitao, Huang Kai. PhoneBit: Efficient GPU-accelerated binary neural network inference engine for mobile phones//Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2020: 786-791

[85] Zhang Yichi, Pan Junhao, Liu Xinheng, et al. FracBNN: Accurate and FPGA-efficient binary neural networks with fractional activations//Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2021: 171-182

[86] Wang E, Davis J J, Cheung P Y K, Constantinides G A. LUTNet: Rethinking inference in FPGA soft logic//Proceedings of the 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). 2019: 26-34

[87] Chen Tianqi, Moreau T, Jiang Ziheng, et al. TVM: An automated end-to-end optimizing compiler for deep learning//Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18). Carlsbad, USA, 2018: 578-594

[88] Sabne A. XLA: Compiling machine learning for peak performance. Google Research, 2020

[89] Lattner C, Amini M, Bondhugula U, et al. MLIR: Scaling compiler infrastructure for domain specific computation//Proceedings of the IEEE/ACM International Symposium on Code Generation and Optimization. 2021: 2-14

[90] Li Mingzhen, Liu Yi, Liu Xiaoyan, et al. The deep learning compiler: A comprehensive survey. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(3): 708-727

[91] Zheng Zhen, Yang Xuanda, Zhao Pengzhan, et al. AStitch: Enabling a new multi-dimensional optimization space for memory-intensive ML training and inference on modern SIMT architectures//Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'22). 2022: 359-373

[92] Ma Lingxiao, Xie Zhiqiang, Yang Zhi, et al. Rammer: Enabling holistic deep learning compiler optimizations with rTasks//Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation. 2020: 881-897

[93] Niu Wei, Guan Jiexiong, Wang Yanzhi, et al. DNNFusion: Accelerating deep neural networks execution with advanced operator fusion//Proceedings of the ACM SIGPLAN International Conference on Programming Language Design and Implementation. 2021: 883-898

[94] Zhao Jie, Gao Xiong, Xia Ruijie, et al. Apollo: Automatic partition-based operator fusion through layer by layer optimization. Proceedings of Machine Learning and Systems, 2022, 4(1): 1-19

[95] Cai Xuyi, Wang Ying, Zhang Lei. Optimus: An operator fusion framework for deep neural networks. ACM Transactions on Embedded Computing Systems, 2022, 22(1): 1-26

[96] David R, Duke J, Jain A, et al. TensorFlow Lite Micro: Embedded machine learning for TinyML systems. Proceedings of Machine Learning and Systems, 2021, 3(1): 800-811

[97] Fawzi A, Balog M, Huang A, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. Nature, 2022, 610(7930): 47-53

[98] Kang Yiping, Hauswald J, Gao C, et al. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. ACM SIGARCH Computer Architecture News, 2017, 45(1): 615-629

[99] Li En, Zeng Liekang, Zhou Zhi, Chen Xu. Edge AI: On-demand accelerating deep neural network inference via edge computing. IEEE Transactions on Wireless Communications, 2019, 19(1): 447-457

[100] Eshratifar A E, Abrishami M S, Pedram M. JointDNN: An efficient training and inference engine for intelligent mobile cloud computing services. IEEE Transactions on Mobile Computing, 2019, 20(2): 565-576

[101] Hu Chuang, Bao Wei, Wang Dan, Liu Fengming. Dynamic adaptive DNN surgery for inference acceleration on the edge//Proceedings of the IEEE INFOCOM. 2019: 1423-1431

[102] Xu Mengwei, Qian Feng, Zhu Mengze, et al. DeepWear: Adaptive local offloading for on-wearable deep learning. IEEE Transactions on Mobile Computing, 2019, 19(2): 314-330

[103] Elseidy M, Abdelhamid E, Skiadopoulos S, Kalnis P. GRAMI: Frequent subgraph and pattern mining in a single large graph. Proceedings of the VLDB Endowment, 2014, 7(7): 517-528

[104] Laskaridis S, Venieris S I, Almeida M, et al. SPINN: Synergistic progressive inference of neural networks over device and cloud//Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. 2020: 1-15

[105] Zhang Shigeng, Li Yinggang, Liu Xuan, et al. Towards real-time cooperative deep inference over the cloud and edge end devices. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2020, 4(2): 1-24

[106] Yao Shuochao, Li Jinyang, Liu Dongxin, et al. Deep compressive offloading: Speeding up neural network inference by trading edge computation for network latency//Proceedings of the 18th Conference on Embedded Networked Sensor Systems. 2020: 476-488

[107] Banitalebi-Dehkordi A, Vedula N, Pei Jian, et al. Auto-Split: A general framework of collaborative edge-cloud AI//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021: 2543-2553

[108] Mao Jiachen, Chen Xiang, Nixon K W, et al. MoDNN: Local distributed mobile computing system for deep neural network//Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE). Lausanne, Switzerland, 2017: 1396-1401

[109] Mao Jiachen, Yang Zhongda, Wen Wei, et al. MeDNN: A distributed mobile system with enhanced partition and deployment for large-scale DNNs//Proceedings of the 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Hammamet, Tunisia, 2017: 751-756

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

[111] Du Jiangsu, Shen Minghua, Du Yunfei. A distributed *in-situ* CNN inference system for IoT applications//Proceedings of the 2020 IEEE 38th International Conference on Computer Design (ICCD). 2020: 279-287

[112] Du Jiangsu, Zhu Xin, Shen Minghua, et al. Model parallelism optimization for distributed inference via decoupled CNN structure. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(7): 1665-1676

[113] Zhang Sai Qian, Lin Jieyu, Zhang Qi. Adaptive distributed convolutional neural network inference at the network edge with ADCNN//Proceedings of the 49th International Conference on Parallel Processing. 2020: 1-11

[114] Zeng Liekang, Chen Xu, Zhou Zhi, et al. CoEdge: Cooperative DNN inference with adaptive workload partitioning over heterogeneous edge devices. IEEE/ACM Transactions on Networking, 2020, 29(2): 595-608

[115] Yang Xiang, Qi Qi, Wang Jingyu, et al. Towards efficient inference: Adaptively cooperate in heterogeneous IoT edge cluster//Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS). 2021: 12-23

[116] Zhang Shuai, Zhang Sheng, Qian Zhuzhong, et al. DeepSlicing: Collaborative and adaptive CNN inference with low latency. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(9): 2175-2187

[117] Huang Kai, Gao Wei. Real-time neural network inference on extremely weak devices: Agile offloading with explainable AI//Proceedings of the 28th Annual International Conference on Mobile Computing and Networking. 2022: 200-213

[118] Wu Ran, Guo Xinmin, Du Jian, Li Junbao. Accelerating neural network inference on FPGA-based platforms: A survey. Electronics, 2021, 10(9): 1025

[119] Moolchandani D, Kumar A, Sarangi S R. Accelerating CNN inference on ASICs: A survey. Journal of Systems Architecture, 2021, 113(1): 1-26

[120] Chen Tianshi, Du Zidong, Sun Ninghui, et al. DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. ACM SIGARCH Computer Architecture News, 2014, 42(1): 269-284

[121] Chen Yunji, Luo Tao, Liu Shaoli, et al. DaDianNao: A machine-learning supercomputer//Proceedings of the Annual IEEE/ACM International Symposium on Microarchitecture. Cambridge, UK, 2014: 609-622

[122] Liu Daofu, Chen Tianshi, Liu Shaoli, et al. PuDianNao: A polyvalent machine learning accelerator. ACM SIGARCH Computer Architecture News, 2015, 43(1): 369-381

[123] Du Zidong, Fasthuber R, Chen Tianshi, et al. ShiDianNao: Shifting vision processing closer to the sensor//Proceedings of the Annual International Symposium on Computer Architecture. Portland, USA, 2015: 92-104

[124] Han Song, Liu Xingyu, Mao Huizi, et al. EIE: Efficient inference engine on compressed deep neural network. ACM SIGARCH Computer Architecture News, 2016, 44(3): 243-254

[125] Abdelfattah M S, Han D, Bitar A, et al. DLA: Compiler and FPGA overlay for neural network inference acceleration//Proceedings of the 2018 28th International Conference on Field Programmable Logic and Applications (FPL). Dublin, Ireland, 2018: 411-4117

[126] Zhang Chi, Prasanna V. Frequency domain acceleration of convolutional neural networks on CPU-FPGA shared memory system//Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2017: 35-44

[127] Niu Yue, Kannan R, Srivastava A, Prasanna V. Reuse kernels or activations? A flexible dataflow for low-latency spectral CNN acceleration//Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2020: 266-276

[128] Wei Xuechao, Yu C H, Zhang Peng, et al. Automated systolic array architecture synthesis for high throughput CNN inference on FPGAs//Proceedings of the Annual Design Automation Conference. 2017: 1-6

[129] Zhang Chen, Sun Guangyu, Fang Zhenman, et al. Caffeine: Toward uniformed representation and acceleration for deep convolutional neural networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 38(11): 2072-2085

[130] Choudhury Z, Shrivastava S, Ramapantulu L, Purini S. An FPGA overlay for CNN inference with fine-grained flexible parallelism. ACM Transactions on Architecture and Code Optimization, 2022, 19(3): 1-26

[131] Yan Mingyu, Deng Lei, Hu Xing, et al. HyGCN: A GCN accelerator with hybrid architecture//Proceedings of the 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). 2020: 15-29

[132] Zhang Bingyi, Kannan R, Prasanna V. BoostGCN: A framework for optimizing GCN inference on FPGA//Proceedings of the 2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines(FCCM). 2021: 29-39

[133] Sludds A, Bandyopadhyay S, Chen Zaijun, et al. Delocalized photonic deep learning on the internet's edge. Science, 2022, 378(6617): 270-276

[134] Yuan Mu, Zhang Lan, You Xuanke, Li Xiang-Yang. PacketGame: Multi-stream packet gating for concurrent video inference at scale//Proceedings of the ACM SIGCOMM 2023 Conference. 2023: 724-737

YUAN Mu, Ph. D. candidate. His research interest is building resource-efficient networking systems for serving machine-learning models.

ZHANG Lan, Ph. D., professor. Her research interests include mobile computing, privacy protection, data sharing & trading.

YAO Yun-Hao, Ph. D. candidate. His research interests are data privacy and security issues in end-edge systems.

ZHANG Jun-Yang, Ph. D. candidate. His research interest is building efficient neural network inference systems in the heterogeneous network environment.

LUO Pu-Han, Ph. D. candidate. His research interest is adaptive model compression and inference acceleration in IoT applications.

LI Xiang-Yang, Ph. D., professor, IEEE fellow. His research interests include Artificial Intelligence of Things (AIoT), privacy and security of AIoT, and data sharing and trading.

Background

This research delves into the burgeoning field of AIoT (Artificial Intelligence of Things) with a particular focus on addressing the critical challenge of resource overhead in model inference. The contemporary landscape witnesses the pervasive integration of AIoT across diverse domains, ranging from smart cities to industrial automation. The efficacy of AIoT systems hinges significantly on model inference, which serves as the linchpin for intelligent decision-making and responses.

The crux of the matter lies in the inherent resource constraints of AIoT devices, spanning computing power, bandwidth, memory, and battery life. Navigating these limitations presents a formidable technical hurdle, particularly concerning the resource overhead incurred during model inference. Consequently, this research seeks to explore and present an in-depth analysis of technologies geared towards optimizing model inference resource overhead in AIoT scenarios.

The scope of this study encompasses a comprehensive review of prevalent model inference optimization techniques employed in contemporary AIoT applications. The analysis dissects the strengths and weaknesses of these techniques, with a specific emphasis on their resource efficiency. To facilitate a systematic understanding, the paper proposes a taxonomy based on three pivotal inference modules (sensor data, intelligent model, IoT hardware) and five key resources (time, memory, bandwidth, energy, and storage).

Within this framework, the research introduces and scrutinizes eight distinct technologies, including input filtering, adaptive configuration, and collaborative inference. Furthermore, a generalized optimization workflow is designed to aid research and development professionals in pinpointing and addressing efficiency bottlenecks within the inference process.

In conclusion, the research not only addresses the contemporary challenges in AIoT but also establishes a foundation for future exploration. By proposing a taxonomy and outlining optimization techniques, this paper contributes to advancing the understanding of efficient model inference in resource-constrained AIoT systems.