面向神经网络量化的计算机体系结构研究进展

郭 聪 冷静文 过敏意
(上海交通大学计算机学院 上海 200240)

摘 要 近年来,随着人工智能算法的突破和革新,深度神经网络模型在多元应用场景中展现出卓越性能。然而,神经网络规模的快速扩张远超硬件加速器的发展速度,其庞大的参数量与计算需求导致现有加速器难以有效承载。因此,为弥合人工智能算法与计算机硬件之间日益扩大的差距,神经网络量化技术已成为模型商业化部署中最为有效的压缩解决方案。为同时提升神经网络加速效率与模型精度,研究者们开发了面向深度神经网络量化的专用体系结构。这类创新架构通过定制应用专用集成电路(ASIC),在计算机体系结构的多个层级实现基于神经网络量化的加速优化,最终实现运行效率提升、能耗降低、部署成本控制与模型精度保障的综合目标,有力促进人工智能产业的可持续发展。本文系统梳理多维度视角下的神经网络量化新型架构,深度解析各类方案的技术特征,并前瞻性探讨未来量化体系结构的演进方向。

关键词 深度神经网络;量化;计算机体系结构;硬件加速器
中图法分类号 TP391 **DOI号** 10.11897/SP.J.1016.2026.01459

Research Advance on Computer Architecture for Neural Network Quantization

GUO Cong LENG Jing-Wen GUO Min-Yi
(Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240)

Abstract In recent years, breakthroughs and innovations in artificial intelligence algorithms have led to significant successes for deep neural network models across various task scenarios. However, the exponential growth in the scale of neural network models has outpaced the development of existing hardware accelerators. The vast number of model parameters and computational demands make deploying neural networks on current accelerators challenging. To bridge the gap between AI algorithms and computer hardware advancements, model quantization has emerged as one of the most effective compression strategies for commercial applications and the proliferation of neural network models. To accelerate neural network models more efficiently and achieve higher accuracy, numerous hardware architecture designers have developed new architectures specifically for deep neural network model quantization. These innovative architectures, designed for model quantization, involve application-specific integrated circuits (ASICs) and focus on accelerating and optimizing neural network model quantization across various levels of computer architectures. This approach not only enhances operational efficiency and model accuracy but also reduces energy consumption and deployment costs, thereby promoting the green and sustainable development of the AI industry. This paper explores these new architectures from different perspectives, summarizing the features of various schemes and providing insights into potential future directions for novel quantization architecture development.

收稿日期:2024-11-08;在线发布日期:2026-03-06。本课题得到国家自然科学基金项目(No. 62532006)资助。**郭 聪**,博士,主要研究领域为计算机体系结构、高性能计算。E-mail:congguo.ai@gmail.com。**冷静文(通信作者)**,博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为构建智能且高效的人工智能系统。E-mail:leng-jw@sjtu.edu.cn。**过敏意**,博士,教授,中国计算机学会(CCF)会士,主要研究领域为并行/分布式计算、编译器优化、嵌入式系统、普适计算、大数据和云计算。

Keywords deep neural network; model quantization; computer architecture; hardware accelerator

1 引 言

当前,人工智能技术已全面渗透至社会生活、科技创新、商业运营及国家安全等关键领域,以前所未有的广度和深度推动社会进步,现已成为我国国家战略层面的核心技术。人工智能技术的研发与应用是当下最具价值的研究方向之一,其中发展最为迅猛的当属深度学习(Deep Learning)[1]领域。在深度学习的研究中,深度神经网络(Deep Neural Network,DNN)[1]作为最具代表性的人工智能模型,其架构设计受到生物神经系统的启发。模型由海量互联的"神经元"单元构成,这些神经元按层次化结构组织,典型架构包含输入层、隐含层和输出层。通过迭代优化神经元间的连接权重,深度神经网络能够自主提取数据中的深层特征与复杂关联,最终在多种任务中实现自动化决策。

深度神经网络的训练与部署高度依赖于海量数据和高性能计算资源,通过大规模数据训练可显著提升模型精度。目前,其在图像识别[2]、目标识别与检测[3]、语义分割[4]、图像生成[5]、自然语言处理[6]等多个领域的性能已超越人类水平。特别值得关注的是,基于 Transformer 架构[7]的大语言模型近年来取得突破性进展,在机器翻译、信息检索、智能问答和文本摘要等自然语言处理任务中表现卓越。2022 年底,OpenAI 推出的 ChatGPT[8]标志着自然语言处理技术的重大突破,该工具仅用两个月就实现月活跃用户破亿。2023 年,国内深度求索(DeepSeek)公司相继推出 DeepSeek 系列大模型,包括 2024 年发布的 DeepSeek-V3[9]和 2025 年推出的 DeepSeek-R1[10]等产品,在多项基准测试中展现出与国际先进模型相当的性能。这些进展共同推动了中文大模型生态的发展,展现了人工智能技术的广阔应用前景。

然而,随着神经网络技术的快速发展,模型规模的不断扩大导致了数据存储需求激增和算力能耗大幅攀升等问题。与此同时,摩尔定律的逐渐失效使得计算机硬件性能提升难以匹配神经网络模型日益增长的存储与计算需求。研究表明[11],2018—2022 年间,大规模语言模型的参数量每两年增长达 410 倍,而同期硬件算力增速仅为 3 倍,这种指数级的差距给神经网络的实际部署带来了严峻挑战。在此背景下,开发高能效的深度神经网络架构显得尤为重要。当前,由于参数量庞大和计算复杂度高,深度神经网络的计算效率受到严重制约,模型轻量化已成为推动深度学习技术发展的关键突破口。

2020 年,《麻省理工科技评论》将轻量化人工智能(Tiny AI)评为年度十大突破性技术[12],引发了学术界对 AI 轻量化问题的广泛关注。该报告指出,当前 AI 研究过度依赖数据和算力的堆叠,不仅导致严重的碳排放问题,还制约了算法性能的进一步提升。轻量化 AI 技术通过模型压缩(Model Compression)技术[13-14]有效解决了这一困境,其主要技术路径包括知识蒸馏、网络剪枝、量化压缩和高效网络设计等。这些方法能显著减小模型规模,降低存储需求和计算负载,使深度神经网络得以在移动终端、物联网设备等资源受限的环境中高效部署。随着边缘计算的普及,模型压缩技术为实现绿色 AI 发展提供潜在支撑。

知识蒸馏[15]:采用"教师—学生"学习框架,通过软目标传递大模型的知识表征能力,使轻量化模型在参数量显著减少的情况下仍能保持接近原模型的预测精度。网络结构设计[16]:针对边缘计算场景设计的专用网络结构,通过深度可分离卷积等创新算子,在保持模型性能的同时大幅降低计算复杂度。模型剪枝[17]:基于权重重要性评估,移除神经网络中冗余的连接和参数。虽然能有效压缩模型规模,但由于其产生的非结构化稀疏模式,在实际部署中需要专用硬件或软件框架支持。神经网络量化[18]将模型参数和激活值从浮点型(如 FP32)转换为低位宽定点表示(如 INT8),可减少 75%以上的存储开销,并利用现代处理器的 SIMD 指令实现计算加速,是目前商业化部署最广泛的轻量化技术。

在上述轻量化技术中,知识蒸馏、轻量网络设计和模型剪枝主要侧重于算法层面的创新。值得注意的是,模型剪枝由于会生成非结构化的稀疏模式,难以有效适配现有人工智能加速器的计算架构。因此,本研究将重点放在深度神经网络量化技术的体系结构创新上,从硬件设计的角度系统性地总结面向量化计算的新型加速器架构。深度神经网络量化的核心在于将网络中的高精度参数(如 32 位或 16 位浮点数)转换为低位宽表示(如 8 位或 4 位整数)。

这一转换过程能够显著降低模型的存储需求和计算资源开销,从而提升推理速度和能效比。量化技术本质上是在模型精度和计算效率之间寻求平衡——通过可控的精度损失换取更高的运行效率,使其特别适合部署在移动设备和嵌入式系统等资源受限的环境中。

人工智能应用的迅速发展依赖于芯片物理基础的快速发展。人工智能算法实现、大规模数据获取和存储,以及计算能力,都与芯片技术密不可分。为了支持深度神经网络的特性,芯片厂商通常在硬件层面设计相应的应用专用的集成电路(ASIC),以适应新的算法特性,例如英伟达的张量核心(Tensor Core)[19]和谷歌的 TPU 加速器[20]。同样,随着量化技术的进一步发展,研究人员和芯片厂商越来越注重面向深度神经网络量化的新型体系结构的研发工作。新型面向神经网络量化的人工智能加速芯片可以在更小的物理空间内提供更强的计算能力,并以更低的能源消耗支持人工智能应用的训练和执行。基于低精度(量化)模型的人工智能芯片已经被广泛部署在消费者设备中。

本文系统性地研究神经网络量化技术在计算机体系结构领域的创新,从多方面探讨面向量化计算的加速器架构设计方案。如图 1 所示,研究内容主要分为以下部分。

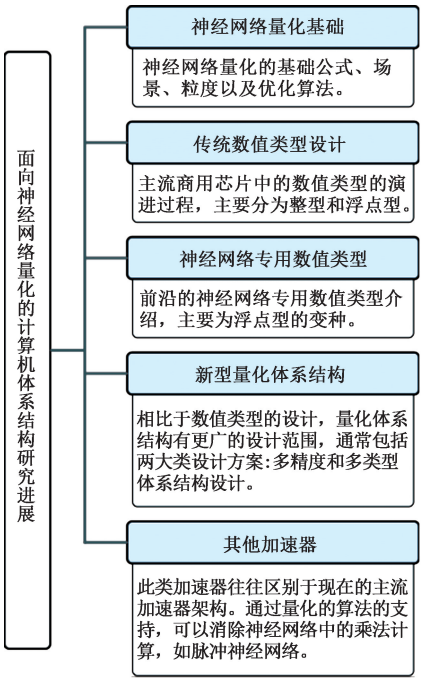

图 1　综述内容概述

第 2 节详细阐述神经网络量化的研究基础,包括量化场景、粒度划分以及优化算法的基本原理;第 3 节分析主流 AI 芯片支持的传统数值类型(整型与浮点型)及其计算特性;第 4 节重点讨论新型专用数值类型的设计方法,这类定制化数值表示通常需要重新设计计算单元,与商用硬件存在兼容性挑战;第 5 节深入探究量化体系结构的创新设计。区别于单一计算单元的优化,量化体系结构需要从存储访问到计算核心进行全局重构,以支持多样化的量化方案。本文将其归纳为多精度量化和多类型量化两大方向,为未来架构设计提供理论参考;第 6 节介绍新型神经网络设计范式。以脉冲神经网络[21]为例,这类方案不仅将网络表示为脉冲信号(可视为极致的二值量化),更需要全新的模拟电路加速器设计,在延续量化思想的同时开创了全新的体系结构研究方向;第 7 节对面向深度神经网络的计算机体系结构研究进行了全面总结。通过系统分析各类通用及专用加速器的架构特征,本研究揭示了神经网络加速器的发展规律与设计范式。基于对现有技术的深入剖析,我们进一步提出了量化专用加速器的未来发展方向,为后续研究提供了重要的理论指导和技术路线参考。

2　神经网络量化的研究基础

本节介绍了关于神经网络量化[14]的基础方法和定义,为后文对新型体系结构的总结提供知识背景支撑。

2.1　量化公式

对于量化后的元素(值)$\hat{w}$,其量化和反量化的函数可以概括为以下方程式:

$$q = Clamp\left[Quant\left(\frac{w}{s}\right), \min, \max\right],$$

$$\hat{w} = s \cdot Dequant(q)。$$

其中,w 是原始的数值,s 是量化比例系数(简称比例系数),比例系数可以将原始的高精度数值的范围转化到可量化的数值范围,而 min 和 max 分别是数值的量化范围中的下限和上限。$Quant$ 是量化函数,量化的主要操作就是将通过比例系数缩放后的数值“量化”到目标低精度数值类型的格式,这一过程会产生精度损失。q 为经过量化后的低精度数值,通常可以用低精度(如 4 比特)整型或浮点型表示。例如,将缩放后的浮点数值 5.826 量化到整数 6,是典型的整型数值类型的量化操作。对于均匀量化(即 INT 整型量化),量化函数实际上是一个四舍五入函数。而对于浮点数,以及复杂的自定义数值

类型来说,需要开发人员设计对应的量化函数以及反量化函数。

Clamp 是截断函数,该函数可以将所有数值截断在上限和下限之间,满足量化后的数值表示的要求。例如,对于 8 比特有符号整型量化的数值上限是 127,下限是 −128。在后文的公式中,本文通常省略表示截断函数,默认量化函数中包含了截断函数且满足量化数值类型的设计要求。

反量化操作函数 Dequant 可以将已量化的值解码回原始高精度类型 $\hat{w}$,例如 32 位 Float 数值类型。这对于量化的实现较为关键,可以快速衡量量化操作对于模型精度的影响,并且也是现有量化框架中必要的操作,可以在量化空间和非量化空间进行转换,拓展了量化的应用场景。

2.2 量化场景

面向神经网络的量化常常应用在通用矩阵乘法(General Matrix Multiplication,GEMM)运算中。图 2 展示了矩阵乘法运算。首先,乘法运算的公式为 $\boldsymbol{C}=\boldsymbol{A}\times\boldsymbol{B}$,其中 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 分别代表两个二维矩阵,大小分别为 $(M\times K)$ 和 $(K\times N)$。对于神经网络中的全连接层(Fully Connected,FC)来说,$\boldsymbol{A}$ 矩阵通常代表输入(Input)矩阵,而 $\boldsymbol{B}$ 矩阵可以表示权重(Weight)矩阵。很显然,根据矩阵乘法的计算规则,$\boldsymbol{C}$ 矩阵对应于输出(Output)矩阵,大小为 $(M\times N)$。

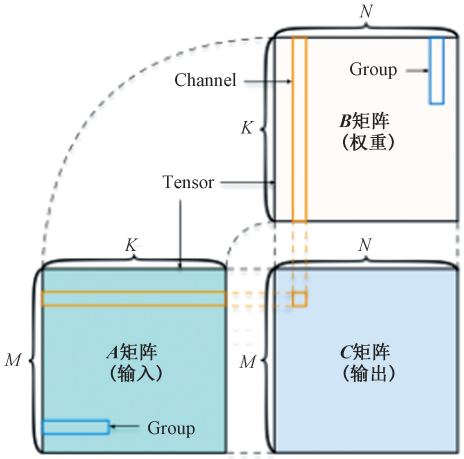

图 2 矩阵乘法的量化场景和粒度

2.3 量化粒度

矩阵乘法的量化粒度在很大程度上会影响量化的效率。通常来说,量化一般分为以下几种模式:张量(Tensor-wise)量化、通道(Channel-wise)量化以及分组(Group-wise)量化,如图 2 所示,这三个层级的量化粒度依次减小。

(1)张量量化

对于矩阵乘法来说,张量(Tensor-wise)量化就是对整个矩阵量化,这也是量化方案中最基础的方式。对于输入矩阵 $\boldsymbol{A}$ 和权重矩阵 $\boldsymbol{B}$ 分别使用一个比例因子和量化函数,因此不需要额外的硬件开销是最基础的量化方式。其量化公式如下:

$$\hat{\boldsymbol{A}}=Quant\left(\frac{\boldsymbol{A}}{s_A}\right),\hat{\boldsymbol{B}}=Quant\left(\frac{\boldsymbol{B}}{s_B}\right),$$

$$\hat{\boldsymbol{C}}=Dequant(\hat{\boldsymbol{A}}\times\hat{\boldsymbol{B}}),$$

$$\boldsymbol{C}=s_A\cdot s_B\cdot\hat{\boldsymbol{C}}。$$

其中,$\hat{\boldsymbol{A}}$ 和 $\hat{\boldsymbol{B}}$ 分别表示量化后的两个矩阵,s_A 和 s_B 分别表示两个矩阵的比例系数。经过低精度的矩阵乘法运算 $\hat{\boldsymbol{A}}\times\hat{\boldsymbol{B}}$ 得到低精度结果矩阵 $\hat{\boldsymbol{C}}$ 后,再通过反量化函数和比例系数得到最终结果 $\boldsymbol{C}$ 矩阵,其中,$\hat{\boldsymbol{C}}$ 矩阵的比例系数 $s_A\cdot s_B$ 通常由量化算法给出,可以由输入矩阵 $\boldsymbol{A}$ 和权重矩阵 $\boldsymbol{B}$ 直接计算而来。在上述公式中可以看出,在计算出低精度矩阵 $\hat{\boldsymbol{C}}$ 的结果后,只需要通过一个逐元素(Element-wise)的乘法操作即可转换为原始的高精度数值,因此这个操作对于现有硬件(如 GPU)非常友好。而量化后的低精度矩阵乘法运算可以大幅降低访存和计算代价,因此,张量(Tensor-wise)量化是最基础的量化方式。

(2)通道量化

通道(Channel-wise)量化[22],通常是针对权重矩阵 $\boldsymbol{B}$ 的特殊量化方案。如图 2 所示,通过对矩阵 $\boldsymbol{B}$ 的每一个通道给定一个独立的比例系数和量化函数,进而可以进行细粒度的量化操作。其量化公式如下:

$$\hat{\boldsymbol{A}}=Quant\left(\frac{\boldsymbol{A}}{s_A}\right),\hat{\boldsymbol{B}}_i=Quant\left(\frac{\boldsymbol{B}_i}{s_{B_i}}\right),$$

$$\hat{\boldsymbol{C}}_i=Dequant(\hat{\boldsymbol{A}}\times\hat{\boldsymbol{B}}_i),$$

$$\boldsymbol{C}_i=s_A\cdot s_{B_i}\cdot\hat{\boldsymbol{C}}_i。$$

其中,$\hat{\boldsymbol{B}}_i$ 和 $\hat{\boldsymbol{C}}_i$ 分别代表矩阵 $\boldsymbol{B}$ 和矩阵 $\boldsymbol{C}$ 的第 i 列(通道),如图 2 所示,矩阵 $\boldsymbol{B}$ 和矩阵 $\boldsymbol{C}$ 包含有 N 列通道。而对比原始的张量量化,对于权重矩阵的通道量化只是将权重矩阵的比例系数从一个数值的标量,变为包含多个数值的向量,该向量大小等于通道的数量,即向量长度为 N。权重矩阵的通道一般特指权重矩阵的列,对于每一列权重都需要和每一行的输入进行点积(dot-product)操作,因此每一列权重可以有一个独立的比例系数而不影响量化后的矩

阵乘法运算。只需要在反量化时,对矩阵 $\boldsymbol{C}$ 的每一列(通道)乘上不同的比例系数即可,这种操作也可以被现有的硬件支持。通道量化因为其较小的粒度往往有较好的量化效果,而且对硬件十分友好,而被广泛应用。

需要指出的是,通道量化一般只应用在权重矩阵中。对于输入矩阵通常采用张量层级的量化,并且其比例系数在运行前已进行预处理,在运行时是一个固定参数。主要原因在于,输入矩阵通常在运行时动态生成,其每一行的输入都是一个独立的输入样本,对于每一个样本在运行时进行量化系数的动态分析以及量化范围的确定是非常耗时的[18]。而且,如果输入矩阵和权重矩阵同时采用量化,会导致输入矩阵的比例系数也变为向量形式,而在反量化处理时,$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 两矩阵相关的比例系数向量,就需要一次额外的外积矩阵乘法,计算代价较高。因此,大多数量化框架对输入矩阵($\boldsymbol{A}$)采用张量量化,而对权重矩阵($\boldsymbol{B}$)采用通道量化。

(3)分组层量化

目前,一些量化算法[23]提出了分组层次(Group-wise)的量化方法,也称为分块层次(Block-wise)。如图 2 所示,这种量化本质上是进一步降低了量化的粒度,对于每一个分组设置一个独立的比例因子和量化方案。然而,这种量化方法是无法在现有硬件上计算的。如果权重矩阵或输入矩阵中的一个通道存在两种不同的分组,每个分组具有不同的比例系数,这种两个分组将无法直接进行计算。因为在这两个分组中,同样的量化后数值有着不同的实际值,如果直接计算并累加,其计算结果将没有意义。因此,分组量化往往需要较为复杂的特定硬件支持和设计。而在现有的硬件中,这些分组量化方案[23]往往只能通过实时反量化,将分组量化的值转换为原始高精度数值,才能进行矩阵乘法计算,并不能节省计算的开销,所以这种方法更类似于一种内存压缩方案。

2.4　量化优化算法

训练后量化(PTQ)[22-26]和量化感知训练(QAT)[18,27-31]。量化感知训练(QAT)是重新训练网络并减轻量化引入的精度下降的最有前途的技术之一。然而,训练过程耗时且成本高。QAT 需要在训练过程中模拟量化,这将耗费大量的时间进行重新训练和超参数调整。相比之下,PTQ 直接将训练好的模型进行量化。因此,PTQ 由于不需要任何微调或重新训练过程而备受关注,但其精度下降比较严重。总的来说,PTQ 对现有大语言模型的量化较为友好,被广泛使用。

3　传统量化方案的问题和挑战

随着神经网络量化的发展,各大硬件厂商都在商用芯片中对低精度神经网络模型做了有效支持。在人工智能发展初期,早期的深度学习模型通常使用单精度(FP32)数值类型对模型进行计算。2015 年前后,众多学者发现[17],在现有的深度神经网络模型中存在显著的参数冗余性,因此可以使用低精度数值类型对神经网络进行量化而不损失精度。例如,2016 年,Benoit Jacob 等学者提出端到端的整型卷积神经网络可以从输入到输出只使用 8 比特整型数值类型来进行计算[18],是神经网络量化的早期代表性工作。自此,神经网络的量化便受到众多研究人员的关注,从而也引起了硬件厂商的关注。由于硬件设计的兼容性,现有商用芯片往往都是基于传统的数值类型设计新型数值类型以减少对现有硬件和软件框架的修改。本节将介绍传统的数值类型,以及其在现有的神经网络量化中所遭遇的挑战。

3.1　传统数值类型介绍

在计算机体系结构的发展中,运算单元中的数值计算是计算机设计中最重要的部分之一。其中以整型数(INT)和浮点数(Float)最为常见。

表 1 为常见的整型(INT)数值类型和表示范围,从早期的 8 位计算机(如 8080 处理器[32])到目前 64 位(如英特尔酷睿系列处理器[33])计算机可以支持的计算字长和比特位宽越来越大,目前可以支持 64 位的整型数计算。整型也是在计算机中最基础的计算和数值类型,其可以精确地表示整数。但整型的表示范围较小,无法表示小数和大数。

表 1　整型数表示方法

比特数	无符号数		有符号数	
	下限	上限	下限	上限
8	0	255	−128	127
16	0	65535	−32768	32767
32	0	$2^{32}-1$	-2^{31}	$2^{31}-1$
64	0	$2^{64}-1$	-2^{63}	$2^{63}-1$

浮点(Float)数值类型目前也是现代计算机主要支持的数值类型之一。目前,IEEE 已经为浮点数指定了一系列的标准,即 IEEE 二进制浮点数算术标准(IEEE 754)标准[34],该标准是 20 世纪 80 年

代以来使用最广泛的浮点数运算标准，为众多的商用计算机的浮点运算器所采用。浮点型可以用以下公式表示：

$$\text{Float} = sign \times 2^{exp-bias} \times 1.mant。$$

其中，$sign$ 表示浮点数的符号位；exp 表示指数值(exponent)，也称为阶码；$mant$ 表示小数值(mantissa)，称为尾码，表示为二进制小数；$bias$ 则是IEEE 754 标准中所强调的指数偏移量，该偏移量对于某一种浮点数值类型来说是一个常量。浮点数的实际值等于符号位(sign)乘以指数减去偏移值(exponent-bias)的二次幂，再乘以小数值(1. mantissa)，其中 IEEE 标准规定每个浮点数中蕴含有一个隐藏位，计算时需要将尾码值视作小数，在小数点前增加一位 1。详细的浮点数介绍可参考相应的公开标准，IEEE 754 标准的数值统计如表 2 所示，其中所有数值的符号位都只有 1 位。

表 2 浮点数类型

比特数	名称	符号位	阶码位	尾码位
16	半精度	1	5	10
32	单精度	1	8	23
64	双精度	1	11	52
128	四倍精度	1	15	112

其中，单精度(FP32)和双精度(FP64)浮点数最为常用，在现有的商用处理器上被广泛采用。IEEE 754 标准在 2008 年有了一个较大的更新[35]，在标准中明确定义了半精度浮点数(FP16)和四倍精度浮点数(FP128)。其中，半精度浮点数(FP16)现在已经被广泛用于神经网络的训练和推理过程。

随着神经网络的不断发展，特别是大语言模型的出现，神经网络的巨大参数量使得模型难以在现有的硬件中运行。因此，许多新型低精度数值类型不断涌现。相比于整型数值类型的单一设计空间，浮点数因此具有较大的设计空间而被新型数值类型设计采用。在本文中以 ExMy 表示基于浮点类型数设计的数值类型，其中 x 表示阶码比特宽度，y 表示尾码比特宽度，常见的数值类型如表 3 所示。

表 3 新型浮点数类型

比特数	名称	阶码位	尾码位	机构
4	FP4	2	1	NVIDIA
8	FP8 E4M3	4	3	NVIDIA
8	FP8 E5M2	5	2	NVIDIA
16	BrainFloat (BF16)	8	7	Google
16	DeepFloat	6	9	IBM
19	TensorFloat (TF32)	8	10	NVIDIA
24	FP24	7	16	AMD
24	PXR24	8	15	Pixar

除了浮点数数值类型，现有硬件也支持低精度的整型数值类型，例如 INT4 和 INT2，甚至是 1 比特数值，部分上述新型数值类型只有在学术报告中提出，尚未大规模商用。

3.2 商用加速器低精度数值类型设计

随着人工智能技术的快速发展，特别是在深度学习领域，芯片厂商对低精度计算的研究与优化已成为关键，以应对庞大的数据处理需求并显著提升计算效率。表 4 总结了现有大规模商用的人工智能芯片中主要使用的低精度数值类型的研究，涵盖了英特尔公司的酷睿系列 CPU[33]、AMD 公司面向人工智能应用所开发的 MI 系列 GPU[36]、英伟达公司的 GPU[37]，以及谷歌公司神经网络训练研发的专用加速器 TPU[20]。其中，“首次”代表该系列处理器是首次支持该数值类型；“支持”代表该处理器支持该类型，但不是该系列的首次支持；“—”代表该处理器不支持该数值类型；“取消”代表该处理器上一代支持该类型，但是这一代中不再支持。表 4 中只列出了处理器系列的关键节点，新增和移除了某种数值类型，并未将所有的处理器型号列出。

(1)英特尔 CPU

CPU 虽然并不是现有深度神经网络的主流运行平台，但是英特尔公司仍然面向人工智能领域的低精度计算特征进行了特定的优化和设计。英特尔在 AI 领域对低精度计算的优化表现在其酷睿系列 CPU[33]及 AVX512 指令集[38]的不断迭代上。从 Sky Lake 开始，英特尔引入了 INT8 和 INT16 类型的支持，并首次在 Cooper Lake 中引入 BF16 类型以及在 Alder Lake 中引入 FP16 类型，展现了其对提高 AI 计算性能的重视。这些技术进步不仅提高了处理速度，也减少了能耗，使得英特尔的产品能在 AI 应用中更高效地运行，同时也为未来更高效的芯片设计奠定了基础。

(2)AMD GPU

AMD 以其 MI 系列 GPU[36]为人工智能市场提供强大的计算支持，从 MI50 到 MI210，该系列对低精度计算的支持持续增强。自 MI50 开始，该系列就专为人工智能应用设计，因此原生支持多种低精度数值类型，例如 INT8 和 FP16 数值类型。在 MI100 中，AMD 首次支持了 BF16，而 MI210 首次支持了 INT4，为 AI 应用提供了更高效的计算选项。AMD 的 GPU 通过支持广泛的数值精度类型，使其可以在多种场景下获得更高的运行效率。AMD 通过这些技术进展，提高了其产品的市场竞争力，为用

表 4　部分芯片所支持的数值类型汇总

处理器/架构	发布时间	峰值性能 Tops	INT 32	INT 16	INT8	INT4	INT1	FP32	FP16	BF16	TF32	FP8	FP4
Intel Sky Lake	2015 年 8 月	—	支持	首次	首次	—	—	支持	—	—	—	—	—
Intel Cooper Lake	2020 年 6 月	—	支持	支持	支持	—	—	支持	—	首次	—	—	—
Intel Alder Lake	2021 年 11 月	—	支持	支持	支持	—	—	支持	首次	支持	—	—	—
AMD MI50	2018 年 11 月	53(INT8)	支持	—	首次	—	—	支持	支持	—	—	—	—
AMD MI100	2020 年 11 月	92.3(INT4)	支持	—	支持	首次	—	支持	支持	首次	—	—	—
AMD MI210	2022 年 3 月	181(INT4)	支持	—	支持	支持	—	支持	支持	支持	—	—	—
NVIDIA Pascal (P100) CUDA Core	2016 年 4 月	21.2(FP16)	支持	—	—	—	—	支持	首次	—	—	—	—
NVIDIA Pascal (P4) CUDA Core	2016 年 9 月	21.8(INT8)	支持	首次	首次	—	—	支持	—	—	—	—	—
NVIDIA Volta (V100)CUDA Core	2017 年 5 月	15.7(FP32)	支持	取消	取消	—	—	支持	—	—	—	—	—
NVIDIA Volta (V100) Tensor Core	2017 年 5 月	125(FP16)	—	—	—	—	—	—	首次	—	—	—	—
NVIDIA Turing (2080Ti)Tensor Core	2018 年 8 月	113.8(FP16) 455.4(INT4)	—	—	首次	首次	—	—	支持	—	—	—	—
NVIDIA Ampere (A100)Tensor Core	2020 年 5 月	312(FP16) 1248(INT4) 4992(Binary)	—	—	支持	支持	首次	—	支持	首次	首次	—	—
NVIDIA Hopper (H100)Tensor Core	2022 年 9 月	1000(FP16) 2000(INT8)	—	—	支持	取消	取消	—	支持	支持	支持	首次	—
NVIDIA Blackwell (B100)Tensor Core	2024 年 3 月	7000 (FP8/INT8) 14000(FP4)	—	—	支持	—	—	—	支持	支持	支持	支持	首次
Google TPUv1	2016 年 5 月	92(INT8)	—	—	首次	—	—	—	—	—	—	—	—
Google TPUv2	2017 年 5 月	46(BF16)	—	—	取消	—	—	—	—	首次	—	—	—
Google TPUv3	2018 年 5 月	123(BF16)	—	—	—	—	—	—	—	支持	—	—	—
Google TPUv4i	2021 年 5 月	138(BF16/INT8)	—	—	支持	—	—	—	—	支持	—	—	—

注:峰值性能:Intel 官方不再公布算力数据。不考虑稀疏算力。

户提供了更多的灵活性和选择,以应对不同的计算需求。

(3)英伟达 GPU CUDA Core

英伟达的 CUDA Core[39]一直是其 GPU 的核心组成部分,是一种通用计算核心。从 Pascal 架构开始,英伟达就通过 CUDA Core 在各种精度级别上进行了优化,特别是在 Pascal(P100)架构中首次引入对 FP16 的支持以及 Pascal(P4)中首次引入对 INT16 和 INT8 的支持。这一优化提升了神经网络训练和推理的处理速度,使得英伟达的 GPU 在执行深度学习应用时,能够提供更快的响应速度和更高的数据吞吐量。然而在 2018 年,Volta 相比于上一代的 Pascal 架构甚至取消了对 INT16 和 INT8 的支持。从这个现象我们也可以发现,英伟达公司对 CUDA Core 的定位已从面向神经网络的主要计算单元变为对于 Tensor Core 的辅助计算单元,而 Tensor Core 已经变为英伟达公司面向 AI 应用的主要计算核心。

(4)英伟达 GPU Tensor Core

英伟达的 Tensor Core[19]技术是其近年来最新的技术突破,专为深度学习和 AI 计算设计。自 Volta 架构引入以来,Tensor Core 通过专门的硬件加速器优化了深度学习的矩阵乘法运算。如前文所述,矩阵乘法正是大多数神经网络结构的基础,占据绝大部分的运行时间。在第一代 Volta 架构的 Tensor Core 设计中仅支持 FP16 类型计算。

在随后的 Turing 架构中,Tensor Core 进一步增强了对整型计算 INT8 和 INT4 的多种精度的支持。而在 Ampere 架构中又对新型数值类型 INT1(Binary)、BF16 和 TF32 等格式的支持,显著提升了计算效率和硬件利用率。而在 Hopper 架构中,英伟达首次支持了 FP8 格式,并具有 E5M2 和 E4M3 两种类型。而在最新的 Blackwell 架构中则支持了 FP4 数值类型,这是商业芯片中最早支持 FP4 类型的加速器。英伟达通过不断创新使得其 Tensor Core 技术成为市场上最为先进的解决方案之一,为 AI 研发提供了巨大的动力。

(5)谷歌 TPU

谷歌的 TPU 加速器[20]专为优化神经网络训练和推理而设计。从 TPUv1 到 TPUv4 的演变中,谷歌不断提升对低精度操作的支持,尤其是在 TPUv2 中首次提出并支持 BF16 数值[20]类型。TPU 通过高度优化的硬件设计,使得大规模的神经网络训练更加高效和经济,特别是在处理大型模型如 GPT[8]和 BERT[40]时,TPU 显著提升了训练效率。

通过分析这些领先企业的产品线和技术,可以看到低精度数值类型在人工智能芯片中的应用正在快速发展,旨在解决深度学习模型庞大的参数和计算量所带来的挑战。这些技术的持续进步不仅推动了人工智能硬件的发展,也为复杂的人工智能应用提供了坚实的技术支持。可以发现,对于低精度的支持是目前神经网络量化的一个重要趋势,也从侧面说明神经网络中存在大量的冗余特性。

3.3 现有数值类型存在的问题和挑战

现有商用芯片中所采用低精度数值类型设计仍然局限在浮点数和定点数这两种类型。从本质上来讲,上述两类数值类型设计方案的前提仍然是需要兼容现有商业芯片,以最小的设计和制造成本生产大规模可商用的处理器。因此,基于整型或者浮点型的简单设计仍然存在一定的局限性并不能很好地适应深度神经网络的量化过程。

多项量化工作都已经提出整型(INT)数值类型并不适合神经网络的量化。在深度神经网络中的张量的数值分布大部分服从高斯分布,而最常用于量化的 INT 类型则是典型的符合均匀分布的数值类型,这个现象已经得到普遍共识。发表在 MICRO 会议上的工作 ANT[41]对这个问题给出了理论分析。许多量化工作将原模型和量化模型之间的均方误差(MSE)作为优化目标。这种方法在数字图像处理领域是一种常见的度量方法,MSE 度量方法可以由以下公式定义:

$$\begin{aligned}\mathrm{MSE}&=E\left[(x-\hat{x})^2\right]\\&=\int(x-\hat{x})^2p(x)\mathrm{d}x。\end{aligned}$$

其中,x 和 $\hat{x}$ 分别表示原始值和量化值,$p(x)$是张量的概率密度函数。很显然,MSE 和张量的分布有直接关联。因此,如果其量化的数值类型的数值分辨率的分布类似于张量分布,就可以显著降低量化前后的 MSE,这也是许多非整型量化方法的核心依据[31,42-44]。也正因如此,英伟达最新的 GPU H100 中就已经增加了 FP8 的量化方式以取代之前的 INT8 量化,从而得到更好的量化效果,甚至支持了模型的训练过程。

Float 类型在高精度(如 32 比特)情况下有着较好的表示范围。但同样,类似于整型量化,在低精度情况下其表示能力受限。正如表 3 所示,FP4,FP8 的阶码的表示范围依然存在较大局限。如果简单使用固定的浮点表示方式,低精度的浮点数只能表示固定的数值范围,而这种范围并不能很好地反映神经网络中数值的分布情况。因此,大量的研究在浮点类型的基础上,面向神经网络的数值分布进行了进一步的改造和设计。

4 面向神经网络的数值类型设计

类似于神经网络专用加速器的设计理念,众多研究人员提出面向深度神经网络量化的专用数值类型设计,这些设计适应了神经网络的数值特征。因此,这些设计不是对整型和浮点型的简单拓展,往往具有更复杂的设计,并且需要一定的硬件和软件的支持。本节归纳和总结了部分的人工智能专用的数值类型设计。目前,面向神经网络的特征,研究人员设计了多种新型数值类型,并被广泛引用在各种应用当中。这些新型数值类型旨在优化数值表示,以减少计算复杂度、降低功耗并提高推理速度。以下将详细介绍部分典型的面向神经网络专有化新型数值类型设计,及其特点和应用场景。

新型数值类型设计主要侧重于对数值类型内部表示的区域进行重新构造。图 3 是一个典型的数值类型设计范式[45]。首先,符号位、阶码位、尾码位三个区域和传统的浮点类型的编码方案类似。不同的是,现有的新型数值类型往往会增加一个控制(Regime)位的比特区域,该区域的设计可以引入更多的信息,可以根据这些信息确定后续阶码和尾码的长度。从而可以得到内部可变长的数值类型设计方案,而整体上又是定长的方案。后文将根据这种区域的定义介绍新型数值类型设计。

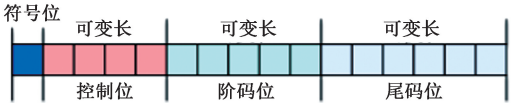

图 3 新型数值类型的比特位定义

Power-of-Two Quantization (PoT)[43]是一种将数值限制在 2 的幂次方的量化方法,例如 1,2,4,8 等。这也意味着 PoT 只保留了符号位和阶码位。

这种方法在数字电路中非常适合硬件加速,因为乘以 2 的幂次方可以通过移位操作实现,而无需复杂的乘法运算。PoT 特别适用于资源极端受限的嵌入式系统和移动设备。然而,这种量化方式可能导致神经网络模型在精确度上产生很大的损失。为了改善 PoT 的精度限制,Additive Power-of-Two Quantization (APoT)[42] 在 PoT 的基础上进行了扩展,它通过累加若干个 PoT 值来表示数值,从而实现更精细的量化级别。这种方法结合了高效计算的优点和更精细的量化要求,能够在保持神经网络较高精度的同时实现高效计算。

Adaptive Float[44] 采用了一种新颖的浮点数表示方法,它为每个张量配置了一个偏移量,并动态调整浮点数的指数部分和尾数部分。这种自适应调整使得 Adaptive Float 能够根据数值的实际分布情况在不同计算阶段或数值块之间调整数值精度。虽然它在面积上比传统的定点数运算器稍大,但提供了更高的灵活性并保证了较高的模型精度。

Posit[45] 是一种新兴的数值表示形式,旨在取代传统的浮点数表示,以提供更高效的精度和计算性能。Posit 保留了全部的符号位、控制位、阶码位以及尾码位。它是由 John L. Gustafson 于 2017 年提出,属于一种定长的数值类型,与 IEEE 754 浮点数相比,Posit 提供了更大的动态范围,并且在接近 1 的数值范围内拥有更多的有效位(但对于非常大或非常小的值,可能有效位会减少)。因此,有很多研究将 Posit 应用在神经网络量化中[46-47]。

HiFloat8[48] 是华为公司在 2024 年提出的新型数值类型方法,该类型类似于 Posit,增加了一个控制位,这个区域采用动态长度的编码。这种编码是一种前缀码(prefix code),即表示某个特定的二进制码永远不会是表示任何其他二进制码的前缀,这样就可以用不同长度的控制码表示不同信息。通过提前规定阶码和尾码的长度,可以实现多种数值类型组合。

类似地,Flint[41] 也是一种采用了前缀码的数值类型。Flint 通过将前缀码直接作为阶码。换句话说,利用前缀码的设计将浮点型的阶码转换为不定长编码,同时尾码也是不定长编码,但保持总体编码长度为定长。

Normal Float[31] 数值类型采用了分位数量化方法,这是一种信息论上最优的方式,确保每个量化区间中包含相同数量的输入张量值。分位数量化通过估算输入张量的分位数来实现。然而,这一过程的主要挑战在于分位数估算的计算成本较高。为此,采用了如 SRAM 分位数的快速分位数近似算法来进行估算。但由于这些算法的近似特性,在处理异常值时可能会导致较大的量化误差。

分块浮点数(Block Floating Point,BFP)[49] 则是一种应用更广泛的特殊类型的浮点类型数,在应用于神经网络之前就已经有较长的发展历史。BFP 通过在一组数值(称为“块”)中共享指数部分,简化了浮点数的表示,减少了存储需求,并提高了计算效率。此方法与传统浮点数表示不同,传统方式中每个数值都有独立的指数和尾数部分,而 BFP 中,一组数值(如神经网络中的激活值或权重)共享一个共同的指数值,每个数值项则保留各自的尾数部分。由于多个数值共享一个指数,这种数值表示形式显著减少了存储需求,尤其在指数变化较小的数据集中,共享一个指数不会引入过大的误差。

BFP 的优点在于其存储效率和计算效率。整个数值块共享一个指数大大减少了需要存储的指数部分,这在大规模数据处理中可以显著节省存储空间。此外,由于所有操作都在相同的指数基准下进行,许多如乘法和加法的操作可以简化,减少了指数调整的次数从而提高了计算效率。其计算可以通过简单的定点运算实现,从而对硬件加速器如 FPGA 或 ASIC 等定制硬件非常有利。

BFP 已经在一些深度学习框架中得到应用。例如,FlexFloat[50] 是最早在深度神经网络中应用 BFP 的思想的研究之一,它根据不同区块的数值大小分别设置偏移量,为深度学习训练提供了一种灵活的数值格式。此外,FAST[51] 是一种针对训练场景的 BFP 加速器,也是较早的 BFP 硬件架构设计,由 H. T. Kung 教授在 2022 年提出,通过多种精度的 BFP 设计和随机舍入技术优化了 DNN 训练的效率和模型精度。

而 Microsoft Floating Point(MSFP)[52] 方案是目前最为知名的 BFP 方案之一,被广泛应用于大语言模型(LLM)的量化之中。MSFP 是微软公司提出的一种高效推理用自定义数值类型,是一种针对神经网络推理优化的低精度浮点格式,它在保持精度的同时,极大地提升了计算效率和缩小了模型大小。通过软硬件协同设计方案,MSFP16 相比 BF16 类型降低了 66% 的硬件成本,而 MSFP12 相比 INT8 降低了 75% 的硬件成本,同时提供了相同或更好的模型精度。报告显示,MSFP 对模型精度的影响微乎其微(<1%),且无需对模型拓扑结构进行

更改，可以与成熟的云生产流程集成，适合在高性能计算、云计算及广泛的物联网设备中应用。2023 年底，微软联合 AMD、英特尔、Meta、英伟达、高通共同改进了 MSFP，将其升级为 Microscaling 数值类型（MXFP）[53]，同样基于 BFP 思想，设计 MXFP 数值类型。

发表于 MICRO 2023 会议的 Bucket Getter[54] 方案对 MSFP 做了进一步的优化，专门设计了用于处理低比特块浮点数（BFP）格式的深度神经网络加速引擎。该引擎采用桶式数值组织方式，将数值划分为多个“桶”，每个桶内数值共享一个指数，这种设计减少了指数变化带来的处理复杂度，提高了数值处理速度和效率。相比于现有的 MSFP 方案，Bucket Getter 能够在维持数值局部性的同时，减少内存访问频率，从而降低延迟，提高吞吐量。这种架构特别适合执行深度学习中关键的复杂矩阵运算，如卷积和矩阵乘法，有效支持深度学习模型的高效运行。

5 基于新型量化方案的体系结构设计

本节主要归纳和总结了基于神经网络量化的新型计算机体系结构研究。不同于单一的数值类型设计，这些量化体系结构对加速器的硬件修改较大，需要不同维度的软硬件协同设计和支持。相比于数值类型设计，在现有人工智能加速器上支持新型量化框架，往往涉及更多层次、更大范围的硬件改动。

5.1 基于混合数值精度的量化体系结构

在深度学习领域，多精度量化架构的发展已成为提升模型训练和推理效率的关键技术方向。自英伟达公司在 2017 年首次提出混合精度训练方案[55]以来，这一技术逐渐演化，并在多种硬件平台中得到落地实现。特别是在 2018 年推出的 Turing 架构[56]中，英伟达进一步集成了支持多精度计算的 GPU 张量核心，验证了混合精度技术在实际部署中的可行性与优势。

这种技术的引入不仅改变了训练深度神经网络的方式，也为推理过程提供了性能优化的可能。通过动态调整计算过程中的数值精度，混合精度技术能够在减少计算资源消耗的同时，保持或甚至提升模型的性能表现。这种平衡是通过精心设计的硬件支持和算法优化共同实现的。

除了英伟达公司，学术界和工业界的其他研究团队也在这一领域做出了显著的贡献。例如，发表在 ISCA 2018 会议的 Bit Fusion[57] 首次提出了一种创新的位级动态组合架构，它能够根据网络层的具体需求动态调整处理单元的位宽。这种灵活的设计使得加速器能够更有效地处理不同类型的计算任务，从而提高了整体的能效和性能，后续有多个研究[58-59]采用了类似的思想。同年，同样发表在 ISCA 2018 会议上的 Outlier-aware Accelerator（OLAccel）[60] 通过引入基于异常值感知的低精度量化方法，进一步优化了神经网络加速器的能效。这种方法利用了神经网络中数值的非均匀分布特性，通过降低对最终结果影响较小的数值的量化精度，从而减少了不必要的计算量和能源消耗，同时保留极少量（约为 3%）的高精度异常值，从而兼顾了神经网络模型的精度和运行效率。

2020 年，发表在 MICRO 2020 会议的 GOBO[61] 则专注于量化自然语言处理模型中的注意力机制，同样通过基于异常值感知的低精度量化方法，降低了模型的延迟并提高了推理效率。这一技术特别适用于需要快速响应的应用场景，如实时语言翻译和交互式对话系统。同年，发表在 ISCA 2020 会议上的 Dynamic Region-based Quantization（DRQ）[62] 通过在不同的网络区域应用不同的量化级别，灵活地调整精度，以达到既保持高性能又降低计算复杂性的目标。这种区域化的策略为精度和性能之间的权衡提供了更多的控制空间。

BiScaled-DNN[63] 通过实现双尺度量化，允许在网络的不同层使用不同的缩放比例系数计算，这样既能保证关键层的精度，也能在不太敏感的层中节省计算资源。这种策略特别适合于资源受限的环境，如移动设备和嵌入式系统。类似的，发表在 ISCA2024 会议的 Tender[64] 针对大模型存在异常值的问题，巧妙地将正常值和异常值的量化比例系数分别处理，可以有效减少硬件代价。FILM-QNN[65] 专为 FPGA 平台设计，它通过层内混合精度量化来优化深度神经网络的硬件加速。这种设计充分利用了 FPGA 的可重配置性，实现了对不同深度学习任务的高效定制。Sibia[66] 则引入了带符号的位切片架构，该架构通过利用 DNN 计算中的比特位级稀疏性来加速处理。这种方法不仅提高了计算效率，还降低了存储需求，非常适合处理大规模的深度学习模型。同样地，发表在 MICRO 2024 会议的 BBS[67] 也是利用了位切片的比特位级别的稀疏性，首先保证了 50% 的稀疏性，并且利用整型数值补码的特性可以将 8 比特量化几乎无损地降低到 4 比特量化，进一步降低了神经网络量化的计算量。

多精度量化不仅用于推理,也已在训练阶段广泛应用。RaPiD[68]加速器采用超低精度技术,极大地提高了 AI 模型的训练和推理速度。这种设计通过极端地降低数值的位宽,为高效计算提供了新的可能。FlexBlock[69]加速器提供了多模式块浮点支持,通过灵活切换不同的精度模式,可以根据训练阶段的不同需求调整硬件的运行状态,这在复杂的深度学习训练任务中非常有用。FPRaker[70]通过设计新的处理单元,专门用于加速神经网络训练的前向和反向传播过程,其优化的数据流和计算方式显著提升了训练的效率和速度。2-in-1 Accelerator[71]展示了在硬件层面上支持精度的动态切换,为模型提供了更高的鲁棒性,同时保持了计算效率。这种加速器特别适合于需要频繁调整计算精度的场景,例如训练场景。Cambricon-Q[72]则通过其混合架构设计,在保持高效率的同时,支持复杂的模型训练。这种设计不仅适应了不同训练阶段的需求,还能在保持高精度的同时提供高效的训练性能。

这些多精度量化架构的研究展示了如何通过技术创新有效地提升深度神经网络的性能和效率。随着人工智能技术的不断进步和应用领域的不断扩展,预计未来将有更多的创新解决方案涌现,以满足日益增长的计算需求和能效挑战。

5.2　混合数值类型的量化体系结构

在深度学习的量化研究领域,对不同张量和矩阵的不同分布的特点,采用不同的数值类型进行量化已成为提高效率与降低硬件资源消耗的重要方法。本节探讨了在同一数值精度条件下,不同类型数值量化的多种体系结构,涵盖了近期的几项创新技术。这些工作区别于简单的数值类型设计,往往采用多种数值类型,通过新型的体系结构设计,以支持不同的数值类型进行计算和存储,侧重于软硬件协同设计支持,相比于数值类型设计有更多层次和更大范围的硬件修改。

首先,Adaptive Numerical Data Type (ANT)[41]技术专注于低比特深度神经网络的量化,通过灵活地组合不同的数值类型,如 Flint、整型(INT)以及 2 的幂次(PoT),实现了对不同数值特征的适应性调整。其中,Flint 是 ANT 所提出的新型数值类型,其编码本身就已经包含了不定长的指数和小数位数,但是总体上 Flint 是定长的。这种方法不仅优化了数值的表示方式,还对所有类型使用相同的精度,提高了内存访问效率,且透明于用户和内存系统。由于 ANT 采用了多种不同的数值类型,其对于神经网络中的张量和矩阵的量化误差较小,从而得到更高的模型精度。然而,不同类型数值无法直接在同一计算单元中执行运算。因此,ANT 设计出 TypeFusion 体系结构支持多种不同数值类型计算,从而获得更高的运行效率。

其次,OliVe 技术[73]通过硬件友好的异常值—剪裁值对(Outlier-Victim Pair)的量化策略,为大型语言模型提供加速。这种策略识别数值中的异常值并将其与剪裁值进行两两配对,以优化存储和计算过程。OliVe 的设计针对处理大规模模型时常见的数值不均匀性问题,有效提升了处理速度和精确度,尤其是在大型语言处理模型如 Transformer 中表现突出。

此外,英伟达的多种 FP8 格式[74]在保持统一精度的同时,通过设计不同的数值表示形式来适应不同的计算需求。英伟达的这些 FP8 格式通过调整尾数和指数部分的位宽,使得开发者可以根据具体的应用需求选择最合适的格式,从而在维持精度的前提下最大化性能和效率。在英伟达的设计中主要支持 E5M2 和 E4M3 两种格式。

通过这些创新的量化技术,研究人员和工程师们能够更好地在精度、性能和资源消耗之间找到平衡。这些技术在优化 AI 硬件的同时,也为深度模型的广泛部署提供关键支撑。

5.3　量化体系结构的统一对比框架

在此,本文提出了一种关于新型量化体系结构的对比框架,如下图 4 所示。

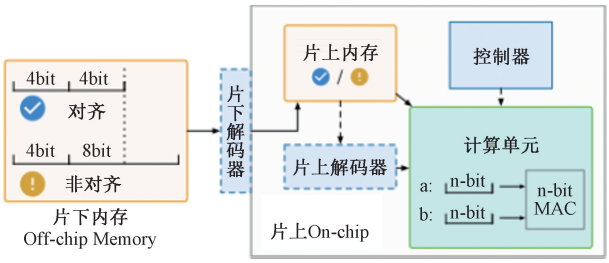

图 4　量化体系结构的统一化对比框架

首先,该对比框架较为简洁地描述了加速器中的三个主要层级:片下内存、片上内存以及记忆计算单元。如表 5 所示,对于众多的量化方案来说,可以总结为三种量化模式:传统量化,混合精度量化以及混合类型量化。其中传统量化通常采用现有的数值类型及其变种,例如最为常见的 INT8、INT4 和浮点类型 FP8 等数值类型。因此,这种数值类型对硬件非常友好可以在现有的加速器中实现。基于传统量化的变种大多是基于浮点类型的分块浮点数

(Block Floating Point,BFP),这种 BFP 兼顾了硬件友好性以及量化的性能,是目前较为实用的发展方向,在未来的几年内有着较好的发展前景。

表 5 多种量化方案的对比分析

量化类型	特征	片下内存	片上内存	计算
传统量化	硬件友好 模型精度较差 如:INT8,FP8	对齐	对齐	高精度同构
混合精度量化	硬件不友好 模型精度较好 如:OLAccel、GOBO	非对齐	对齐	高精度同构
混合类型量化	硬件友好 模型精度较好 如:ANT、OliVe	对齐	对齐	低精度异构

第二种为混合精度量化。这种量化方式根据模型的算法特征设计新型架构,在同一个架构中同一个张量内部支持多种不同的量化精度,可以降低模型的整体内存量,极大地压缩模型大小,但是其缺点也是显而易见的。这种量化方式往往采用非对齐的存储方式,导致其访存极不规则,导致大量的访存开销。并且为了兼容片上内存的访问,只能在片下将非对齐的内存提前处理,片上保持对齐。因为其采用多种不同精度,在片上往往解码为其最大的精度,例如 GOBO 将 3 比特和 16 比特的混合数据统一转化为 16 比特的 FP16 数值类型,OLAccel 将 4 比特和 16 比特的数值统一转化为 16 比特 INT 类型,保证了片上的访存和计算的对齐。但因此会造成片上的访存和计算精度较高,硬件代价较大。

第三种为混合类型量化。这种量化方式充分考虑了量化内存的开销,为了降低访存的开销,设计思路由不同长度的数值精度转换为不同类型的数值类型,但保持了相同的数值长度。因此,这种类型对访存非常友好,在片下和片上的访存对内存系统是透明的,但是这种量化方式需要对计算单元进行重构,重新设计异构的计算单元的构造以适应新型的数值类型,因此难以与现有的硬件兼容。但因其较好的量化效果,在一个较长远的芯片开发周期来看,混合类型量化在硬件支持后有较为实用的应用前景。

5.4 面向大语言模型的量化研究进展

近年来,大语言模型(Large Language Model,LLM)凭借强大的表示能力在自然语言处理、对话生成、信息检索等领域取得了前所未有的成功。然而,模型规模的指数级增长带来的庞大计算量和显著的内存需求,使其在资源受限环境下的高效部署面临巨大挑战。因此,针对大语言模型的量化技术成为当前研究的热点。

大语言模型的特性与传统卷积神经网络(CNN)有显著差异,主要体现在以下两个方面:(1)自回归机制:大语言模型的推理计算通常是逐步预测下一个词元的过程,从矩阵-矩阵乘法转变为矩阵-向量乘法,这种计算模式使得仅权重量化(Weight-only)成为一种高效的部署策略;(2)异常值(Outlier)特性:大语言模型的权重分布往往存在少量极大的异常值,这些异常值对模型精度影响巨大,因此如何高效地处理和保留异常值成为量化技术中的关键问题。这些方法在前文已有介绍,本节将从 LLM 角度重新梳理其技术路线。

围绕大语言模型的上述特性,目前已有多种创新的量化方法提出并取得较好效果。Normal Float[31]数值类型通过分位数量化方法确保每个量化区间包含相同数量的输入张量值,有效缓解了模型量化中异常值带来的误差问题。但由于近似算法的限制,处理极端异常值时可能出现较大的误差。

针对大模型中特有的异常值处理问题,Microscaling(MXFP)[53]作为一种先进的分块浮点数(Block Floating Point,BFP)量化技术被提出,采用软硬件协同优化,以块为单位共享指数,大幅提升了硬件计算效率,降低了硬件实现成本。同时,Microscaling 通过在模型内部不同区域动态调整量化范围,能够高效地保留异常值并确保较低的精度损失,因而在大语言模型部署中具有突出优势。

进一步地,OliVe[73]提出了异常值-剪裁值对(Outlier-Victim Pair)的量化方案。OliVe 巧妙地利用数值的分布特性,通过将异常值与剪裁值进行配对量化,以低硬件开销实现高效的异常值管理,极大降低了 4 比特量化带来的精度损失。然而,尽管 OliVe 方案显著提升了对异常值的处理效率,但在直接应用于超大规模 LLM 时仍然面临一定的挑战,表明针对大模型的专用化改进仍有进一步优化的空间。

除此之外,SmoothQuant[75]、AWQ[26]等方法也体现了对激活值量化敏感性问题的针对性处理。SmoothQuant 通过引入通道级别的动态缩放(per-channel scaling)策略,有效降低了量化误差。AWQ 则进一步通过激活感知(activation-aware)缩放,显著提升了模型在低比特量化情况下的稳定性和准确性。

随着大语言模型部署的普及,针对其特有的自回归机制和异常值特性而专门设计的量化方法,已逐渐成为主流研究趋势。未来在量化算法创新、数值类型设计以及软硬件协同体系结构开发方面,将持续围绕这些关键特性展开深入探索,以进一步提升大语言模型的高效部署和应用能力。

6　新兴近似计算与神经形态加速器

硬件近似计算[76](Approximate Computing,AC)技术在深度神经网络(DNN)加速器设计中受到广泛关注。该技术利用 DNN 在推理过程中的容错特性,通过引入可控的计算误差,显著降低能耗和硬件复杂度。与传统的低精度量化方法不同,近似计算不仅关注数值精度的降低,还包括在电路层面进行的结构性简化,例如设计近似加法器和乘法器、简化存储单元等。这些方法通常需要对加速器的整体架构进行重新设计,以适应新的计算模式和数据流。

文献[76]对现有的硬件近似计算技术进行了系统的分类和分析,涵盖了从电路级别的近似单元设计到系统级别的容错机制。研究表明,尽管引入了近似计算可能会带来一定的精度损失,但通过合理的设计和优化,可以在保持模型性能的同时,实现显著的能效提升。此外,近似计算还在提升系统可靠性和安全性方面展现出潜力。与前文讨论的量化方法相比,近似计算提供了一种从硬件层面优化 DNN 加速器的新途径,值得在特定应用场景中进一步探索和应用。

6.1　超低精度近似计算加速器

(1)算法和理论支持

二值化神经网络(Binarized Neural Network,BNN)[77]显著提高人工智能架构的运行效率,它通过将权重和激活值限制为+1 或-1 大幅简化了计算和存储需求。这种方法从根本上改变了从传统的浮点运算到简单的二进制运算的神经计算动态,减少了内存使用,并加速了操作,便于在资源受限的设备上部署深度学习模型,如手机和嵌入式系统。

因此,由于二值化的产生,使得计算可通过 XNOR 运算实现。例如,XNOR-Net[78]是专为在大规模数据集(如 ImageNet)上使用二值卷积神经网络而设计的一种特殊 BNN 架构。通过将标准的卷积操作修改为二进制操作,XNOR-Net 显著减少了计算负担和模型大小,同时保持了竞争性的准确率。这是通过引入缩放因子来实现的,这些因子补偿了二进制表示固有的信息损失,从而在效率和性能之间提供了平衡。此外,BiBench[79]框架为网络二值化提供了全面的基准测试和分析工具。它为评估不同的二值化技术提供了一个结构化的环境,评估它们在各种任务和条件下的性能。这种分析对于理解二值化中涉及的权衡至关重要,特别是不同策略如何影响神经网络的速度、准确性和资源利用。通过使用 BiBench,研究人员和开发者可以深入了解 BNN 的实际限制和潜力,指导进一步的优化和创新。

三元神经网络[80](Ternary Neural Network,TNN)则是另一种类似的优化方案。与二值化神经网络类似,TNN 将权重和激活值限制在三个可能的值:-1、0、+1。TNN 在减少内存使用和功耗方面表现出色,同时仍能提供良好的模型性能。因为,TNN 通过三个值表示神经网络,在保持模型精简的同时,比 BNN 拥有更强的表达能力,因此能更好地适应不同的数据特征和结构。

综合来看,BNN 和 TNN 这些研究为开发和部署在资源受限环境中的高效神经网络模型提供了重要的技术支持。这些模型通过采用创新的近似数值表示和优化计算方法,不仅促进了 AI 技术在多样化设备上的应用,还推动了整个行业朝着更加节能和成本效益的方向发展,这些网络设计也促进了近似计算加速器的发展。

(2)超低精度加速器

超低精度量化神经网络特别是二/三值神经网络(BNN/TNN)因其高效的计算和存储需求而受到广泛关注。在众多研究中,BinaryConnect[81]探索了在传播过程中使用二进制权重训练深度神经网络的可能性,显著降低了模型的复杂度和运算成本。此外,基于 DRAM 的加速器[82]架构为二值化网络提供了新的硬件优化路径,如用于手机的 PhoneBit 加速器[83]。Hyperdrive[84]和 Xcel-RAM[85]等则通过多芯片系统来实现超低功耗的近似计算推理。在超低功耗和高吞吐量计算方面,新型的 BNN 加速器,如 BRein Memory[86]和 FINN[87]框架,通过在单芯片上重新配置内存和网络结构,通过重构数据路径和访存结构,实现对 BNN/TNN 的高效加速。这些研究不仅表明了在资源受限的移动设备和嵌入式系统上部署深度学习模型的可行性,还推动了基于 FPGA 的二值卷积神经网络加速器架构的发展[88]。

总的来说,这些研究展示了通过硬件和软件的协同优化,可以显著提升 BNN 和 TNN 的性能和能效。这对于推动深度学习技术在能源效率和计算速

度方面的进一步发展具有重要意义。

6.2 神经形态加速器

(1)脉冲神经网络

脉冲神经网络[89](Spiking Neural Network,SNN)被誉为神经网络的第三代,旨在更加真实地模仿生物大脑神经网络的行为。尽管其结构与传统的深度神经网络相似,SNN 在几个关键方面进行了创新,使得它们在处理方式和信息传递机制上更接近生物神经系统。

SNN 的主要特点在于它们传递信息的方式。不同于 DNN 中连续的浮点数值,SNN 中的信息传递是通过一系列的二进制脉冲(spike)来实现的。这些脉冲在时间序列中传播,每一个脉冲都代表了一个信息的发送。此外,SNN 不使用传统的激活函数,而是通过更新神经元的膜电位(Membrane Potential,MP)来调控神经元的激活状态,这一过程更加贴近生物神经元的电生理特性。在 SNN 中,神经元的 MP 根据接收到的输入电流进行调整,当 MP 达到特定的阈值时,神经元便生成一个新的脉冲,并将其发送到网络的下一层。这种基于时间的处理机制使得 SNN 在时间信息的编码和处理上具有天然的优势。

SNN 的另一个特点在于其信息处理是分时间步进行的。在每个时间步中,神经元根据前一层神经元传来的脉冲和相应的权重计算输入电流,然后更新自身的 MP。这种分步处理的方式使得 SNN 在处理动态和时序数据时更为高效,例如在语音识别和视频处理等应用中表现出色。SNN 中使用的脉冲神经元模型多种多样,其中最为著名的是 Leaky Integrate-and-Fire(LIF)模型[90]。LIF 模型以其简洁性而广受欢迎。此模型不仅能精确模拟神经元的电活动,还能通过其参数设置来适应不同的模拟需求,使其在多种科学研究和工程应用中都有广泛的应用。

SNN 的这些独特属性使得它们在动态信息处理场景中展现出显著潜力。从自动驾驶车辆的实时决策系统到高级机器人的感知处理系统,再到下一代智能设备的能效管理,SNN 的应用前景广阔。因此有众多基于脉冲神经网络的相关架构的设计。

(2)脉冲神经网络架构设计

在学术研究领域,对 SNN 架构的探索主要集中在提高网络的计算效率和减少数据传输上。例如,SpinalFlow[91]架构是首个专为 SNN 设计的体系架构,其通过时间分批处理来减少数据移动,这在理论上能够显著提高处理效率。然而,SpinalFlow 的设计仅针对时间编码的 SNN,一次只能发射一个神经元计算,这种算法上的限制降低了其普适性。

PTB 架构[92]提出了一个基于脉动阵列的新型架构,不限制神经元发射的次数,能够并行处理结构化的稀疏脉冲数据。它采用时间分组技术,通过在时间窗口内对脉冲信息进行分组处理以提高系统的利用率。尽管如此,PTB 在处理短步长 SNN 时的资源利用率仍存在问题。Stellar[93]架构是一种算法硬件协同设计的脉冲神经网络加速器,采用了基于脉动阵列的处理方式,通过增加脉冲稀疏性,从而达到较高的并行处理能力。

在工业界中,IBM TrueNorth[94]和英特尔 Loihi[95]已经开发出具体的 SNN 硬件实现,推动了 SNN 技术的商业化进程。IBM 推出的 TrueNorth 芯片使用了大规模并行架构,模拟生物神经元的并行计算模式。该芯片通过其高度优化的神经元和突触配置,显著降低了能耗,适用于需要大量并行处理且能耗敏感的应用。

英特尔 Loihi 芯片在模拟生物神经行为的基础上,实现了在线学习的功能。它不仅支持复杂的 SNN 模型,还可以在不依赖外部系统的情况下进行自适应学习,这使得 Loihi 在边缘计算等场景中具有广泛的应用潜力。高通的 Zeroth 处理器则专注于在移动设备上实现高效的 SNN 计算。通过优化 SNN 的计算架构,Zeroth 旨在为智能手机和其他便携设备提供实时的智能处理能力。

这些先进的商业化芯片展示了 SNN 在处理效率、实时性及能效方面的优势,预示着 SNN 技术在未来智能系统中的广泛应用前景。总之,无论是在学术研究还是工业应用中,SNN 的架构设计都在不断进步,其模拟人脑处理信息的能力已经显示出巨大的潜力和商业价值。

7 总结与展望

本文系统性地回顾并总结了近年来面向神经网络量化的计算机体系结构的研究进展。面对深度学习模型快速扩张带来的计算资源挑战,文章强调了量化技术作为实现深度神经网络高效部署和能源优化的关键手段。通过全面的梳理和分析,本文聚焦于数值类型创新、量化体系结构设计以及适配不同神经网络特性的专用体系结构,探讨了当前量化技术的最新进展和未来的发展趋势。

从本文的分析可以看出,量化技术已逐渐从早期单纯的高精度(FP32/FP16)压缩扩展到极低精度(如 FP4、INT4 甚至三值量化)的探索阶段。尤其是在大语言模型(LLM)的兴起背景下,模型规模的指数级增长已迫使推理和训练均向更低精度转移。特别是模型推理环节,量化技术已广泛成为标准化的部署手段,其研究热点不断转向如何在极低精度的条件下仍然保持稳定的精度和高效能比。

未来,量化技术在神经网络体系结构中的发展趋势体现在以下几个方面:

(1)量化精度的持续降低趋势

随着大语言模型规模持续膨胀、计算资源约束进一步加剧,量化技术向更低精度过渡已成为主流方向。以谷歌 TPU 与英伟达 Tensor Core 的发展历程为例,这一趋势可划分为两个主要阶段:

第一阶段(2015～2018 年),谷歌与英伟达相继在商用芯片中首次支持低精度 INT8 量化,这标志着业界逐步认可并积极推进量化体系结构的发展。在此期间,学术界针对神经网络量化算法进行了大量深入的研究[14],为后续面向量化的计算机体系结构设计奠定了坚实基础。这一阶段的研究工作主要围绕传统数值类型的支持展开,出现了基于 INT 类型量化的代表性体系结构设计,例如 BitFusion[57] 与 OLAccel[60]。

第二阶段(2018～2024 年),众多新型数值类型不断涌现,各大厂商纷纷重新审视并探索新的数值类型设计及量化技术路线,进一步将神经网络量化推向更低精度(如 4 比特)时代。这一阶段,英伟达公司在 4 比特量化的支持态度上经历了较为复杂的演变:2018 年,Turing 架构的 Tensor Core 首次引入了对 INT4 量化的支持,但 2022 年推出的 Hopper 架构则取消了 INT4 支持,转而采用 FP8 量化方案。最终在 2024 年发布的 Blackwell 架构中,重新支持了更低精度的 FP4 量化。同样地,谷歌的 TPU 也经历了类似的变化过程:其第二代和第三代产品中曾一度放弃 INT8 量化,转向 BF16 数值类型,直到第四代 TPU 才重新支持 INT8 量化。这一演变过程体现出业界在追求更低精度与保持计算精度之间的持续权衡和技术迭代。

总体而言,从 FP32 到 BF16、再到 FP8 和 FP4,量化精度不断降低已成为不可逆转的技术趋势。未来的研究将集中于探索更加稳定且高效的超低比特(如 2 比特或三值)量化方案,并进一步通过软硬件协同优化来减轻因精度降低而带来的性能损失,这将成为该领域的重要发展方向。

(2)数值类型设计的多样化与专用化

随着量化精度的降低,传统的整型(INT)和浮点型(FP)已无法在精度、动态范围和编码灵活性上满足极低精度场景下的模型需求,因此未来量化数值类型的设计必将呈现出高度多样化的趋势。本文分析的 Posit、Flint、Adaptive Float 等新型数值类型都反映出这种趋势。这些新型数值类型通过更灵活的编码和专门的硬件支持,更有效地匹配神经网络数值分布,极大提升了模型精度和计算效率。这意味着未来在大语言模型量化领域,专门针对模型特征设计的数值类型将获得更加广泛的关注和应用。

(3)面向大语言模型特性的体系结构优化

当前深度神经网络模型已从传统 CNN 架构扩展到以 Transformer 为代表的大语言模型。这类模型独特的自回归计算机制和异常值特征,使得针对传统 CNN 设计的加速器和量化技术不再适用。因此,如何设计适配于 LLM 特性的量化体系结构成为未来重要的研究方向。例如,仅权重量化(Weight-only)技术的应用以及对异常值(Outlier)的高效处理机制的研究,已展现出巨大的潜力。未来研究应进一步深入挖掘 LLM 的独特特征,开发更加针对性的量化加速方案,这不仅将提高模型的整体性能,还能在降低计算资源消耗与能耗方面展现巨大的潜力。此外,作为突破冯·诺依曼瓶颈的另一重要方向,面向多领域融合计算的新型数据流架构通过软硬件协同设计在性能与能效上展现出超越传统通用处理器的潜力[96],有望与量化体系结构的演进相结合,共同构成未来高效能人工智能硬件的重要技术路线。

综上所述,本文通过对当前量化体系结构研究成果的系统总结和归纳,明确指出了量化技术未来的三个重要发展趋势:量化精度持续降低、数值类型专用化设计以及针对大语言模型特性的专用架构优化。通过进一步推动这些方向的研究发展,量化技术将在未来的人工智能硬件发展中继续扮演核心角色,有效支撑 AI 产业的绿色和可持续发展。

参　考　文　献

[1] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015,521(7553):436-444

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

[3] Guo Y, Liu Y, Georgiou T, et al. A review of semantic seg-

mentation using deep neural networks. International Journal of Multimedia Information Retrieval, 2017,7(2):87-93

[4] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. Communications of the ACM, 2020,63(11):139-144

[5] Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. International Journal of Computer Vision (IJCV), 2015,115(3):211-252

[6] Nadkarni P M, Ohno-Machado L, Chapman W W. Natural language processing: an introduction. Journal of the American Medical Informatics Association, 2011,18(5):544-551

[7] Vaswani A. Attention is all you need. arXiv preprint arXiv: 1706.03762, 2017

[8] Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023

[9] Liu A, Feng B, Xue B, et al. DeepSeek-v3 technical report. arXiv preprint arXiv:2412.19437, 2024

[10] Guo D, Yang D, Zhang H, et al. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025

[11] Gholami A, Yao Z, Kim S, et al. AI and memory wall. IEEE Micro, 2024,44(4):1-5

[12] Lichfield G. 10 breakthrough technologies 2020: Tiny AI// https://www.technologyreview.com/10-breakthrough-technologies/2020/, 2020

[13] Cheng Y, Wang D, Zhou P, et al. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282, 2017

[14] Deng L, Li G, Han S, et al. Model compression and hardware acceleration for neural networks: A comprehensive survey. Proceedings of the IEEE, 2020,108(4):485-532

[15] Gou J, Yu B, Maybank S J, et al. Knowledge distillation: A survey. International Journal of Computer Vision, 2021,129(6):1789-1819

[16] Elsken T, Metzen J H, Hutter F. Neural architecture search: A survey. Journal of Machine Learning Research, 2019,20(55):1-21

[17] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv preprint arXiv:1510.00149, 2015

[18] Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA, 2018: 2704-2713

[19] Choquette J, Gandhi W, Giroux O, et al. NVIDIA A100 tensor core GPU: Performance and innovation. IEEE Micro, 2021,41(2):29-35

[20] Jouppi N P, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit//Proceedings of the 44th Annual International Symposium on Computer Architecture. New York, USA, 2017:1-12

[21] Ghosh-Dastidar S, Adeli H. Spiking neural networks. International Journal of Neural Systems, 2009,19(4):295-308

[22] Nagel M, Baalen M, Blankevoort T, et al. Data-free quantization through weight equalization and bias correction//Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, Republic of Korea, 2019:1325-1334

[23] Frantar E, Ashkboos S, Hoefler T, et al. GPTQ: Accurate post-training quantization for generative pre-trained transformers. arXiv preprint arXiv:2210.17323, 2022

[24] Gupta S, Agrawal A, Gopalakrishnan K, et al. Deep learning with limited numerical precision//Proceedings of the International Conference on Machine Learning. Lille, France, 2015:1737-1746

[25] Guo C, Qiu Y, Leng J, et al. Squant: On-the-fly data-free quantization via diagonal hessian approximation. arXiv preprint arXiv:2202.07471, 2022

[26] Lin J, Tang J, Tang H, et al. AWQ: Activation-aware weight quantization for LLM compression and acceleration. arXiv preprint arXiv:2306.00978, 2023

[27] Wang Z, Lu J, Tao C, et al. Learning channel-wise interactions for binary convolutional neural networks//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, USA, 2019:568-577

[28] Zhuang B, Tan M, Liu J, et al. Effective training of convolutional neural networks with low-bitwidth weights and activations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021,44(10):6140-6152

[29] Chen M, Shao W, Xu P, et al. EfficientQAT: Efficient quantization-aware training for large language models. arXiv preprint arXiv:2407.11062, 2024

[30] Liu Z, Oguz B, Zhao C, et al. LLM-QAT: data-free quantization aware training for large language models. arXiv preprint arXiv:2305.17888, 2023

[31] Dettmers T, Pagnoni A, Holtzman A, et al. QLORA: efficient finetuning of quantized LLMs//Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA, 2023:10088-10115

[32] Mazor S. Intel 8080 CPU chip development. IEEE Annals of the History of Computing, 2007,29(2):70-73

[33] Doweck J. Inside intel ® core microarchitecture//Proceedings of the 2006 IEEE Hot Chips 18 Symposium (HCS). Stanford, USA, 2006:1-35

[34] Kahan W. IEEE standard 754 for binary floating-point arithmetic. Lecture Notes on the Status of IEEE, 1996, 754(94720-1776):1-30

[35] Markstein P. The new IEEE-754 standard for floating point arithmetic//Cuyt A, Krämer W, Luther W, Markstein P, eds. Numerical Validation in Current Hardware Architectures: Dagstuhl Seminar Proceedings, Volume 8021. Dagstuhl, Germany: Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2008:1-3

[36] Smith A, James N. AMD instinct? MI200 series accelerator

and node architectures//Proceedings of the 2022 IEEE Hot Chips 34 Symposium (HCS). Cupertino, USA, 2022:1-23

[37] Choquette J, Giroux O, Foley D. Volta: Performance and programmability. IEEE Micro,2018,38(2):42-52

[38] Boemer F, Kim S, Seifu G, et al. Intel HEXL: Accelerating homomorphic encryption with Intel AVX512-IFMA52//Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography. New York, USA, 2021:57-62

[39] Choquette J, Gandhi W. NVIDIA A100 GPU: Performance & innovation for GPU computing//Proceedings of the 2020 IEEE Hot Chips 32 Symposium (HCS). IEEE Computer Society. Palo Alto, USA, 2020:1-43

[40] Devlin J. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805, 2018

[41] Guo C, Zhang C, Leng J, et al. Ant: Exploiting adaptive numerical data type for low-bit deep neural network quantization//Proceedings of the 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO). Chicago, USA, 2022:1414-1433

[42] Li Y, Dong X, Wang W. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. arXiv preprint arXiv:1909.13144, 2019

[43] Przewlocka-Rus D, Sarwar S S, Sumbul H E, et al. Power-of-two quantization for low bitwidth and hardware compliant neural networks. arXiv preprint arXiv:2203.05025, 2022

[44] Tambe T, Yang E Y, Wan Z, et al. Adaptivfloat: A floating-point based data type for resilient deep learning inference. arXiv preprint arXiv:1909.13271, 2019

[45] Gustafson J L, Yonemoto I T. Beating floating point at its own game: Posit arithmetic. Supercomputing frontiers and innovations, 2017,4(2):71-86

[46] Yu J, Prabhu K, Urman Y, et al. 8-bit transformer inference and fine-tuning for edge accelerators//Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. La Jolla, USA, 2024,3:5-21

[47] Ramachandran A, Wan Z, Jeong G, et al. Algorithm-hardware co-design of distribution-aware logarithmic-posit encodings for efficient DNN inference. arXiv preprint arXiv: 2403.05465, 2024

[48] Luo Y, Zhang Z, Wu R, et al. Ascend HiFloat8 format for deep learning. arXiv preprint arXiv:2409.16626, 2024

[49] Kalliojarvi K, Astola J. Roundoff errors in block-floating-point systems. IEEE transactions on signal processing, 1996,44(4):783-790

[50] Tagliavini G, Marongiu A, Benini L. Flexfloat: A software library for transprecision computing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018,39(1):145-156

[51] Zhang S Q, McDanel B, Kung H T. Fast: DNN training under variable precision block floating point with stochastic rounding//Proceedings of the 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Seoul, Republic of Korea, 2022:846-860

[52] Rouhani B, Lo D, Zhao R, et al. Pushing the limits of narrow precision inferencing at cloud scale with microsoft floating point//Proceedings of the 34th International Conference on Neural Information Processing Systems. Online, Canada, 2020:10271-10281

[53] Rouhani B D, Zhao R, More A, et al. Microscaling data formats for deep learning. arXiv preprint arXiv: 2310.10537, 2023

[54] Lo Y C, Liu R S. Bucket Getter: A bucket-based processing engine for low-bit block floating point (BFP) DNNs//Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture. Toronto, Canada, 2023:1002-1015

[55] Micikevicius P, Narang S, Alben J, et al. Mixed precision training. arXiv preprint arXiv:1710.03740, 2017

[56] Li A, Su S. Accelerating binarized neural networks via bit-tensor-cores in turing GPUs. IEEE Transactions on Parallel and Distributed Systems, 2020,32(7):1878-1891

[57] Sharma H, Park J, Suda N, et al. Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural network//Proceedings of the 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA). Los Angeles, USA, 2018:764-775

[58] Zhou J, Wu J, Gao Y, et al. DyBit: Dynamic bit-precision numbers for efficient quantized neural network inference. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024,43(5):1613-1617

[59] Reggiani E, Pappalardo A, Doblas M, et al. Mix-GEMM: An efficient hw-sw architecture for mixed-precision quantized deep neural networks inference on edge devices//Proceedings of the 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Montreal, Canada, 2023:1085-1098

[60] Park E, Kim D, Yoo S. Energy-efficient neural network accelerator based on outlier-aware low-precision computation//Proceedings of the 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA). Los Angeles, USA, 2018:688-698

[61] Zadeh A H, Edo I, Awad O M, et al. GOBO: Quantizing attention-based nlp models for low latency and energy efficient inference//Proceedings of the 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). Athens, Greece, 2020:811-824

[62] Song Z, Fu B, Wu F, et al. DRQ: dynamic region-based quantization for deep neural network acceleration//Proceedings of the 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA). Valencia, Spain, 2020:1010-1021

[63] Jain S, Venkataramani S, Srinivasan V, et al. Biscaled-DNN: Quantizing long-tailed datastructures with two scale factors for deep neural networks//Proceedings of the 56th Annual Design Automation Conference 2019. New York, USA, 2019:1-6

[64] Lee J, Lee W, Sim J. Tender: Accelerating large language models via tensor decomposition and runtime requantization//Proceedings of the 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA). Buenos Aires, Argentina, 2024:1048-1062

[65] Sun M, Li Z, Lu A, et al. Film-qnn: Efficient FPGA acceleration of deep neural networks with intra-layer, mixed-precision quantization//Proceedings of the 2022 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York, USA, 2022:134-145

[66] Im D, Park G, Li Z, et al. Sibia: Signed bit-slice architecture for dense DNN acceleration with slice-level sparsity exploitation//Proceedings of the 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Montreal, Canada,2023:69-80

[67] Chen Y, Meng J, Seo J, et al. BBS: Bi-directional bit-level sparsity for deep learning acceleration//Proceedings of the 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO). Austin, USA2024:551-564

[68] Venkataramani S, Srinivasan V, Wang W, et al. RaPiD: AI accelerator for ultra-low precision training and inference//Proceedings of the 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). Valencia, Spain, 2021:153-166

[69] Noh S H, Koo J, Lee S, et al. FlexBlock: A flexible DNN training accelerator with multi-mode block floating point support. IEEE Transactions on Computers, 2023,72(9):2522-2535

[70] Awad O M, Mahmoud M, Edo I, et al. FPRaker: A processing element for accelerating neural network training//Proceedings of the 54th Annual IEEE/ACM International Symposium on Microarchitecture. Online , 2021:857-869

[71] Fu Y, Zhao Y, Yu Q, et al. 2-in-1 accelerator: Enabling random precision switch for winning both adversarial robustness and efficiency//Proceedings of the 54th Annual IEEE/ACM International Symposium on Microarchitecture. Online, 2021:225-237

[72] Zhao Y, Liu C, Du Z, et al. Cambricon-Q: A hybrid architecture for efficient training//Proceedings of the 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). Valencia, Spain, 2021:706-719

[73] Guo C, Tang J, Hu W, et al. Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization//Proceedings of the 50th Annual International Symposium on Computer Architecture. Orlando, USA, 2023:1-15

[74] Choquette J. NVIDIA Hopper H100 GPU: Scaling performance. IEEE Micro, 2023,43(3):9-17

[75] Xiao G, Lin J, Seznec M, et al. Smoothquant: Accurate and efficient post-training quantization for large language models//Proceedings of the International Conference on Machine Learning. Honolulu, USA, 2023:38087-38099

[76] Armeniakos G, Zervakis G, Soudris D, et al. Hardware approximate techniques for deep neural network accelerators: A survey. ACM Computing Surveys, 2022,55(4):1-36

[77] Courbariaux M, Hubara I, Soudry D, et al. Binarized neural networks: Training deep neural networks with weights and activations constrained to+1 or -1. arXiv preprint arXiv: 1602.02830, 2016

[78] Rastegari M, Ordonez V, Redmon J, et al. Xnor-net: Imagenet classification using binary convolutional neural networks//Proceedings of the European Conference on Computer Vision. Tel Aviv, Israel, 2016:525-542

[79] Qin H, Zhang M, Ding Y, et al. Bibench: Benchmarking and analyzing network binarization//Proceedings of the International Conference on Machine Learning. Honolulu, USA, 2023:28351-28388

[80] Zhu C, Han S, Mao H, et al. Trained ternary quantization. arXiv preprint arXiv:1612.01064, 2016

[81] Alemdar H, Leroy V, Prost-Boucle A, et al. Ternary neural networks for resource-efficient AI applications//Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN). Anchorage, USA, 2017:2547-2554

[82] Choi H, Lee Y, Kim J J, et al. A novel in-DRAM accelerator architecture for binary neural network//Proceedings of the 2020 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS). Kokubunji, Japan, 2020:1-3

[83] Chen G, He S, Meng H, et al. Phonebit: Efficient GPU-accelerated binary neural network inference engine for mobile phones//Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). Grenoble, France, 2020:786-791

[84] Andri R, Cavigelli L, Rossi D, et al. Hyperdrive: A multi-chip systolically scalable binary-weight CNN inference engine. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2019,9(2):309-322

[85] Agrawal A, Jaiswal A, Roy D, et al. Xcel-RAM: Accelerating binary neural networks in high-throughput SRAM compute arrays. IEEE Transactions on Circuits and Systems I: Regular Papers, 2019,66(8):3064-3076

[86] Ando K, Ueyoshi K, Orimo K, et al. BRein memory: A single-chip binary/ternary reconfigurable in-memory deep neural network accelerator achieving 1.4 TOPS at 0.6 W. IEEE Journal of Solid-State Circuits, 2017,53(4):983-994

[87] Umuroglu Y, Fraser N J, Gambardella G, et al. Finn: A framework for fast, scalable binarized neural network inference//Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays. Monterey, USA, 2017:65-74

[88] Abdelouahab K, Pelcat M, Serot J, et al. Accelerating CNN inference on FPGAs: A survey. arXiv preprint arXiv:

1806.01683, 2018

[89] Maass W. Networks of spiking neurons: The third generation of neural network models. Neural Networks, 1997, 10(9): 1659-1671

[90] Teeter C, Iyer R, Menon V, et al. Generalized leaky integrate-and-fire models classify multiple neuron types. Nature Communications, 2018, 9(1): 709-709

[91] Narayanan S, Taht K, Balasubramonian R, et al. Spinalflow: An architecture and dataflow tailored for spiking neural networks//Proceedings of the 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA). Valencia, Spain, 2020: 349-362

[92] Lee J J, Zhang W, Li P. Parallel time batching: Systolic-array acceleration of sparse spiking neural computation//Proceedings of the 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Seoul, Republic of Korea, 2022: 317-330

[93] Mao R, Tang L, Yuan X, et al. Stellar: Energy-efficient and low-latency SNN algorithm and hardware co-design with spatiotemporal computation//Proceedings of the 2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Edinburgh, UK, 2024: 172-185

[94] Hsu J. IBM's new brain . IEEE Spectrum, 2014, 51(10): 17-19

[95] Davies M, Srinivasa N, Lin T H, et al. Loihi: a neuromorphic manycore processor with on-chip learning. Ieee Micro, 2018, 38(1): 82-99

[96] Leng J W, Guo M Y, Zeng D Z, et al. Dataflow microprocessor: development, trends, and challenges. SCIENCE CHINA Information Sciences, 2025, 55: 452-463(in Chinese)
(冷静文,过敏意,曾德泽,等.数据流芯片的发展现状、趋势与挑战.中国科学:信息科学,2025,55:452-463)

GUO Cong, Ph. D. His main research interests include computer architecture and high-performance computing.

LENG Jing-Wen, Ph. D., professor, Ph. D. supervisor. He is currently interested in building intelligent and robust system for artificial intelligence.

GUO Min-Yi, Ph. D., professor, Ph. D. supervisor, IEEE Fellow. His main research interests include parallel/distributed computing, compiler optimizations, embedded systems, pervasive computing, big data and cloud computing.

Background

Deep neural networks (DNNs) have seen rapid advancements, particularly in achieving superior performance across tasks like image recognition, object detection, semantic segmentation, image generation, and natural language processing. However, as DNN models become more complex, issues related to computational energy consumption and memory demands have emerged. These challenges have led to increased research interest in model compression, with model quantization emerging as a prominent technique.

Various strategies have been explored to address the limitations of DNNs, including knowledge distillation, network architecture design, model pruning, and model quantization. Among these, model quantization has gained notable traction due to its potential to significantly reduce model size and computational load while maintaining acceptable accuracy levels. Recent developments have focused on refining quantization techniques to support efficient DNN execution on resource-constrained devices, aligning with broader trends in AI miniaturization, such as the push toward Tiny AI. Despite these efforts, several challenges persist, especially in designing hardware architectures that fully support quantized models in real-world applications.

This survey aims to summarize and evaluate recent advancements in model quantization techniques and their integration into computer architectures. It specifically emphasizes the design of novel AI accelerators optimized for quantized models. The previous work has laid a foundation in AI model compression, including contributions to low-precision training, network pruning, and the design of quantized neural networks. This background informs the current survey, which offers a comprehensive understanding of the state-of-the-art approaches and identifies opportunities for future research in model quantization and AI hardware co-design.