DOI: 10.19650/j.cnki.cjsi.J2311004

深度神经网络的自适应联合压缩方法

姚博文,彭喜元,于希明,刘连胜,彭 宇
(哈尔滨工业大学测控工程系 哈尔滨 150080)

摘 要:现有模式单一且固定的深度神经网络压缩方法受限于精度损失,而难以对模型进行充分压缩,致使压缩后模型在实际部署时仍需消耗大量成本高昂且容量有限的存储资源,对其在边缘端的实际应用造成严峻挑战。针对该问题,本文提出一种可同时对模型连接结构和权重位宽进行自适应联合优化的压缩方法。与已有组合式压缩不同,本文充分融合稀疏化和量化方法进行联合压缩训练,从而全面降低模型规模;采用层级自适应的稀疏度和数据表征位宽,缓解因固定压缩比导致的精度次优化问题。通过使用本文提出方法对 VGG、ResNet 和 MobileNet 在 CIFAR-10 数据集上的实验表明,精度损失分别为 1.3%、2.4% 和 0.9% 时,参数压缩率达到了 143.0×、151.6×和 19.7×;与 12 种典型压缩方法相比,模型存储资源的消耗降低了 15.3×~148.5×。此外,在自建的遥感图像数据集上,该方法仍能在达到最高 284.2×压缩率的同时保证精度损失不超过 1.2%。

关键词: 深度神经网络;模型压缩;联合优化;稀疏化;量化
中图分类号: TP183 TH89 **文献标识码**: A **国家标准学科分类代码**: 520.2060

Adaptive joint compression method for deep neural networks

Yao Bowen, Peng Xiyuan, Yu Ximing, Liu Liansheng, Peng Yu
(*Department of Test and Control Engineering, Harbin Institute of Technology, Harbin 150080, China*)

Abstract: Deep neural network compression methods with a single and fixed pattern are difficult to compress the model sufficiently due to the limitation of accuracy loss. As a result, the compressed model still needs to consume costly and limited storage resources when it is deployed, which is a significant barrier to its use in edge devices. To address this problem, this article proposes an adaptive joint compression method, which optimizes model structure and weight bit-width in parallel. Compared with the majority of existing combined compression methods, adequate fusion of sparsity and quantization methods is performed for joint compression training to reduce model parameter redundancy comprehensively. Meanwhile, the layer-wise adaptive sparse ratio and weight bit-width are designed to solve the sub-optimization problem of model accuracy and improve model accuracy loss due to the fixed compression ratio. Experimental results of VGG, ResNet, and MobileNet using the CIFAR-10 dataset show that the proposed method achieves 143.0×, 151.6×, and 19.7× parameter compression ratios. The corresponding accuracy loss values are 1.3%, 2.4%, and 0.9%, respectively. In addition, compared with 12 typical compression methods, the proposed method reduces the consumption of hardware memory resources by 15.3×~148.5×. In addition, the proposed method achieves maximum compression ratio of 284.2× whilemaintaining accuracy loss within limited range of 1.2% on the self-built remote sensing optical image dataset.

Keywords: deep neural network; model compression; joint optimization; sparsity; quantization

0 引 言

当今的深度神经网络以其卓越的任务精度被广泛应用于测试测量、工业制造、医疗监测等众多领域。然而,随着神经网络模型复杂度的快速增长,在智能传感、工业视觉检测、工业物联网等边缘计算需求密集场景下,有限且昂贵的高性能硬件存储资源已在传输速率、功耗和容量等多个方面呈现应用瓶颈[1]。为进一步提升深度神经网络在边缘端硬件上部署和推理的实际性能,需要解决的首要问题便是模型参数规模与硬件平台间的访存资源瓶颈问题。已有研究发现,神经网络模型推理时因大量的参数访存产生的能量消耗远高于有效计算过程的开销,从而对硬件实际推理效率和能效也造成严重影响[2]。

收稿日期:2023-01-28 Received Date: 2023-01-28

为解决该问题,神经网络模型压缩已经成为一种有效而必要的技术手段。有研究表明,深度神经网络模型通常存在大量的冗余结构和过参数化现象[3]。从实际应用中的任务精度需求和部署资源条件出发,通过对神经网络中冗余部分的压缩,既可获得更加轻量化的模型以缓解硬件部署资源压力,同时又可实现与压缩前近似的精度,以保证其在实际应用中的有效性[4]。因而,模型压缩对解决边缘端有限资源条件下,模型部署时硬件资源、模型规模和任务精度之间的不平衡问题有巨大的研究价值和应用潜力[5]。

目前神经网络模型的压缩方法主要包括稀疏化和量化等。传统的模型压缩过程,通常是采用单一的压缩方法来获得满足硬件部署和计算需求的轻量级模型。但随着模型推理过程向前端转移,单方面的模型压缩已经难以满足存储和算力都极度受限的边缘端硬件平台对部署模型规模提出的新要求和挑战[6]。因此,通过组合或联合多种模型压缩方法,从不同角度同时对模型结构和参数进行压缩,已经成为进一步减轻模型体量、提高推理性能和适应轻小型硬件部署模型需求的新型压缩范式。

联合压缩方法的优势在于可以从不同角度对模型冗余性进行发掘,以获得综合压缩性能的提升。针对决定深度神经网络模型存储资源占用的两项主要因素,连接数量和权重位宽,采用相适应的压缩方式,以获得模型规模的充分缩减。在已有研究中,连接结构的压缩主要通过删除不重要的模型权重,已达到模型稀疏化的目的,从而获得更紧凑的模型结构[7];权重参数方面则是通过将高位宽连续数据映射为低位宽离散数据的近似计算方式,降低模型计算复杂度和存储需求[8]。稀疏化和量化两种方法的联合应用,相较于单一压缩方法可以获得压缩率的大幅提升,进而有效降低模型部署时对硬件存储资源的消耗[9]。

联合压缩方法的关键是根据参数和结构的冗余特点对压缩方法进行合理的分配与组合,使方法间形成互补,促进综合压缩效果的提升。针对稀疏化和量化联合方法,研究者们提出了阶段式的组合优化方式对模型结构和参数位宽分别进行独立压缩,从而获得更高的压缩比[10-12]。然而,在不同压缩阶段下模型本身冗余特点将发生改变,因此无法做出及时的适应性调整。阶段式的组合压缩方法难以与模型本身产生最优的适配结果。同时,相较于使用单一方法的压缩过程,联合压缩策略具有更高的复杂度,往往需要依赖人工设计经验对各阶段进行手动微调,不仅会造成较高的人力成本开销,而且易引入人工经验误差导致次优化。

针对上述问题,本文提出一种基于稀疏化和量化协同优化(sparse and quantized synergy, SQS)的自适应模型联合压缩方法。该方法将稀疏化和量化操作融合并内嵌于模型的训练过程,在不需要复杂调参和额外人工经验的前提下,随模型训练过程,对模型层级的连接冗余和位宽冗余进行自适应迭代学习和压缩,从而有效避免因分阶段压缩导致的次优化现象和额外调试成本开销。SQS 采用即插即用的节点操作形式,可以方便地嵌入不同模型结构中,在完成常规模型训练的同时完成高倍率的模型参数压缩,获得低存储资源消耗的轻量级模型,并通过低成本的整体微调恢复模型精度至压缩前的近似水平。

本文通过将稀疏化和量化操作融入模型训练过程,以端到端的方式实现对神经网络连接数量和参数位宽进行联合压缩,提升方法间的协同作用优势,避免了分阶段式压缩的复杂策略设计;构建具有稀疏化因子和位宽选择因子的联合压缩框架,根据模型层的冗余特点进行自适应学习,从而在训练和压缩过程中提升压缩策略对模型参数与结构变化的适应性,改善精度和压缩率的不平衡关系;在无需修改原始模型结构的情况下,可快捷而高效的通过插入操作节点构建联合压缩训练框架。通过对 VGG[13]、ResNet[14] 和 MobileNet[15] 主流神经网络模型的实验,验证了本文所提出方法在降低模型存储资源消耗方面的有效性和性能优越性。

1 相关研究

1.1 模型稀疏化

模型稀疏化是深度神经网络常用的压缩方法之一,其主要原理是在不造成严重精度损失的情况下,通过对“不重要”的连接结构的裁剪获得更加紧凑的网络模型,因此,通常也被称为模型剪枝。Lin 等[16] 通过广泛的统计计算发现特征图的秩可以用于对特征信息含量进行评价,进而间接确定对应连接结构的重要性。此外,Han 等[7] 基于“权重越小,对模型精度贡献越小”的假设,通过将小于特定阈值的权重置为 0 的方式获得稀疏连接的网络模型,并在 AlexNet[17]、VGG[13] 等模型上成功验证了该方法的有效性。类似的,研究者们还提出了使用 ℓ_1 范数[18]、信息熵[19] 以及 BN 层 γ 因子[20] 等参数的量值作为连接重要性评价指标的剪枝方法。然而,为取得压缩率和精度平衡,在剪枝过程中通常需要消耗大量人力成本进行基于经验的设置和调优。为解决该问题,Xiao 等[21] 和 Kusupati 等[22] 引入了可学习参数进行自动化剪枝,通过迭代训练使模型稀疏度自动适应网络参数冗余特点,同时提升稀疏模型的鲁棒性。

1.2 参数量化

参数量化是通过仿射变换从数据表示精度方面对计算过程进行近似,例如将浮点 32 位数据映射为整型 8 位数据,获得更低存储资源消耗的同时降低计算复杂度。

谷歌公司在 2018 年的研究中,对使用整型参数代替浮点参数进行神经网络推理的有效性进行了验证[8]。由于神经网络不同层间对数据位宽的敏感性不同,其所适应的量化位宽也不尽相同。因此,Dong 等[23]、Huang 等[24]和 Wang 等[25]相继提出了可对位宽敏感性进行自适应的混合精度量化方法,以有效平衡精度和低位宽计算间的矛盾,为模型在资源受限的嵌入式平台中部署提供了更大的潜力。

1.3　稀疏化和量化联合压缩

稀疏化和量化方法可以被同时应用于神经网络模型以获得更高的压缩率。Han 等[9]提出了一种三阶段式(剪枝,量化和熵编码)的联合压缩方法,用以降低深度神经网络模型的存储消耗,但需要依靠人工经验对压缩超参数进行调试。为解决该问题,Yang 等[10]提出了基于约束优化的自动化联合压缩方法,在无需人工设置超参数的情况下在压缩率和精度之间取得了平衡。此外,Park 等[26]针对训练过程中的稀疏化和量化问题,提出了基于梯度优化的联合压缩方法。Gonzalez-Carabarin 等[11]提出了一种灵活的剪枝策略,同时可以联合量化方法以获得满足硬件需求的压缩模型。联合压缩方法为追求模型极至压缩率提供了巨大潜力,同时也已成为权衡深度神经网络模型在应用场景、压缩比、精度和硬件可用性方面的重要方法。

2　自适应联合压缩方法

2.1　方法框架

深度神经网络复杂且难以解释的连接结构和参数分布,对压缩程度具有独特且多样的敏感性特点。该特点表明,采用单一且相对固定的压缩模式,往往难以对模型不同维度的结构和参数冗余度进行匹配,进而容易获得“过饱和”或“欠饱和”的压缩模型,使模型处于有效信息难以保留而无效信息压缩不充分的不良状态,从而降低模型鲁棒性和任务完成能力,对实际部署中硬件资源利用率和模型推理效率的发挥造成严重影响。针对该问题,本文提出一种基于模型稀疏化和量化联合的层级自适应压缩方法框架,如图 1 所示。与采用两阶段压缩的联合方式不同,本文基于原始模型结构构建并行的联合压缩操作节点,并通过设置可训练的压缩参数使参数压缩过程随模型权重训练同步进行。同时为匹配模型不同层之间对于压缩敏感性的差别,采用层级自适应的压缩参数训练方法,即通过端到端的训练使模型自动学习稀疏阈值和量化位宽,从而降低因人为设定压缩目标而引入的人工误差和精度损失。

如图 1 所示,自适应联合压缩框架主要分为联合压缩节点构建、联合压缩训练和压缩模型微调 3 部分。

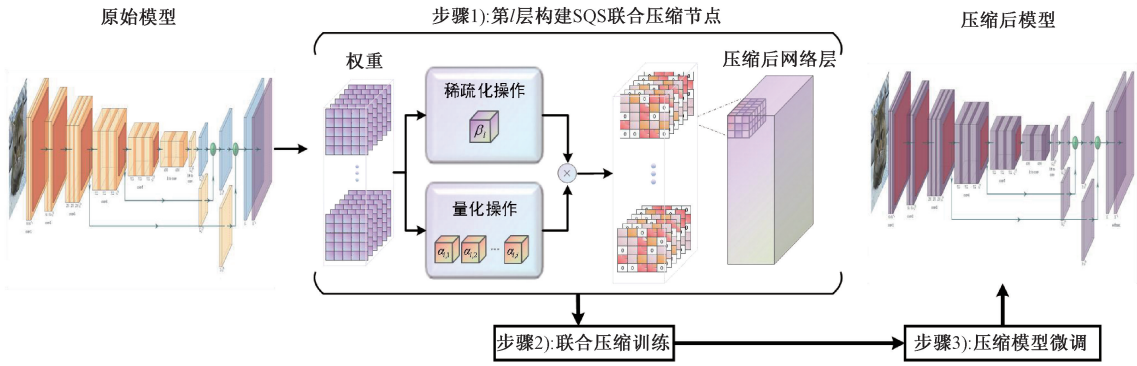


图 1　自适应联合压缩方法框架

Fig. 1　Adaptive joint compression method framework

1)联合压缩节点构建。在原始模型的基础上通过在层间插入可学习联合压缩节点,将稀疏化阈值和量化位宽重要性因子作为可训练参数引入模型训练过程,构建具有联合压缩误差感知能力的神经网络模型。

2)联合压缩训练。在无需预训练的情况下,使用目标数据集直接对构建后的模型进行迭代训练,分别对神经网络权重和压缩参数执行层级的自适应优化和更新,促使模型训练在达到目标精度的同时使压缩率最大化。

3)压缩模型微调。基于训练获得的模型和压缩参数,使用少量目标数据对其进行微调和测试,得到最终的压缩模型。

自适应联合压缩方法具体实现过程如算法 1 所示。其中,算法分别对稀疏化和量化定义了可学习的压缩参数 β_l 和 $\alpha_{l,i}$。该参数的学习过程可跟随神经网络的训练和测试过程进行同步迭代,降低压缩过程中超参数的设计与调试难度,并且压缩过程与训练过程深度融合,无需设计额外且复杂的损失函数,大幅降低技术使用难度,具有即插即用和简单高效等优势。

算法 1 自适应联合压缩方法

输入:训练数据 $\boldsymbol{D}$,神经网络基础结构 *model*,稀疏因子 β_l,位宽选择因子 $\alpha_{l,i}$,最大迭代次数 *iteration*。

输出:微调后压缩模型。

1: 向 *model* 结构中插入联合压缩节点

2: 初始化模型权重 $\boldsymbol{w}_{l,i}$、β_l 和 $\alpha_{l,i}$

3: for 1,2,…, *iteration* do

4: for 1,2,…,L do

5: 通过式(3)~(6)和(14)更新权重 $\boldsymbol{w}_{l,i}$

6: end for

7: 加载数据 $\boldsymbol{D}$,通过式(1)、(2)计算 *model* 损失误差

8: 用梯度下降法更新模型权重,β_l 和 $\alpha_{l,i}$

9: end for

10: repeat

11: 删除多余量化分支结构,进行模型微调

12: until 达到预期模型精度

13: end 输出模型

本文所提出的自适应联合压缩方法框架,旨在通过构建即插即用的联合压缩节点,将模型稀疏化和量化过程快速而有效地融合至模型训练过程中,对结构和数据位宽两方面的冗余性进行深度挖掘,同时充分发挥稀疏化和量化互补优势,对不同压缩敏感性的网络层进行层级自适应学习,从而在保证神经网络精度的同时进一步提升模型压缩率。

2.2 联合优化

联合优化的目的是通过训练迭代的方式在使网络收敛同时获得具有稀疏结构和低位宽表示权重的轻量级模型,因此对于给定神经网络模型的目标函数为:

$$\min \quad \mathcal{L}_{train} = E(\boldsymbol{Y}, f(\boldsymbol{X}, \boldsymbol{W}^*)) + \lambda \sum_{l \in L, i \in N} R(\boldsymbol{w}_{l,i}) \tag{1}$$

式中:$\mathcal{L}_{train}$ 表示模型的在训练数据上的损失误差,由交叉熵函数 $E(\cdot)$ 和正则化项 $\sum R(\boldsymbol{w}_{l,i})$ 计算得出。其中,交叉熵函数中变量 $\boldsymbol{Y}$ 和 $f(\cdot)$ 分别表示目标值和预测值;$\boldsymbol{X}$ 为输入数据;λ 为正则化因子,通常为常数。为在神经网络训练过程中实现模型稀疏化和量化的联合压缩操作,将 $f(\cdot)$ 中的权重向量修改为联合压缩向量 $\boldsymbol{W}^*$,即将权重分别输入至稀疏化操作节点和量化操作节点,稀疏化操作节点根据权重分布和自适应阈值获得对应的权重掩码向量,同时量化操作节点通过多分支的量化位宽搜索完成权重参数由浮点数据向整型数据的映射变换。最终两者结果的阿达玛乘积作为可用于训练和实际推理的权重参数。联合压缩操作原理框图如图 2 所示。

联合压缩操作原理可表示为:

$$\boldsymbol{W}_l^* = M(\boldsymbol{W}_l) \circ Q(\boldsymbol{W}_1, bit) \tag{2}$$

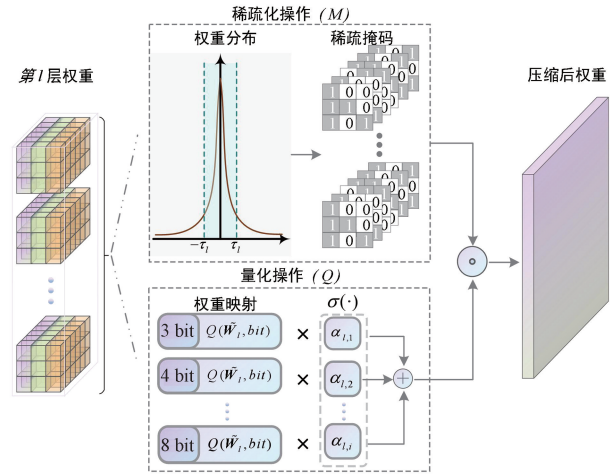


图 2 联合压缩操作原理

Fig. 2 Principle of joint compression operation

式中:$M(\cdot)$ 表示为稀疏化函数,用于计算权重掩码矩阵;$Q(\cdot)$ 为量化函数,用于在 *bit* 位量化区间内进行权重的仿射变换。然而,在压缩过程中必然会对模型的原始结构造成一定破坏,使模型连接结构和权重分布呈现不稳定状态,影响最终模型的推理精度。因此,为保证模型能够快速而稳定收敛,一方面,通过将权重向量 $\boldsymbol{W}$ 替换为联合压缩向量 $\boldsymbol{W}^*$ 而同时引入稀疏化误差和量化误差,使模型在训练过程中完成对压缩误差的适应和优化,从而缓解压缩过程对模型精度的不良影响,提高压缩后轻量级网络的鲁棒性;另一方面,借助 ℓ_1 正则化项对模型的训练权重进行约束,使其趋于稀疏而紧凑的分布状态,从而降低训练时稀疏化的难度,并缩小参数映射空间,以在一定程度上缓解因直接将稀疏权重置为零值和量化分辨率不足时导致的模型损失误差震荡,进而降低模型压缩难度,加快收敛速度。

1) 自适应阈值稀疏化

针对一个具有 L 层的神经网络模型,每层的权重总数为 N_l,对于第 l 层的第 i 个权重可以表示为 $\boldsymbol{w}_{l,i}$,其中,$l \in \{1,2,\cdots,L\}$,$i \in \{1,2,\cdots,N_l\}$。基于阈值的权重稀疏化方法通常可以表示为:

$$\widetilde{\boldsymbol{w}}_{l,i} = \boldsymbol{w}_{l,i} \times M_l(\boldsymbol{w}_{l,i}) \tag{3}$$

$$M_l(\boldsymbol{w}_{l,i}) = \begin{cases} 1, & \boldsymbol{w}_{l,i} > \tau \\ 0, & \text{其他} \end{cases} \tag{4}$$

式中:$\widetilde{\boldsymbol{w}}_{l,i}$ 为稀疏化后的权重值;$M_l(\cdot)$ 为赫维赛德阶跃函数,用于计算权值掩码矩阵;τ 为预先设定的权重阈值。由于不同网络层的权重在特征提取过程中的作用不同,有效权重和非必要权重的分布比例呈现较大的差异化,因此,依靠人工经验预先设定的固定阈值难以对层间稀疏度的差异性进行充分适配,极易产生过度或过轻压缩的次优化结果。

为解决该问题,本文采用可训练的连续函数代替固定阈值,通过迭代训练对神经网络权重进行自动化学习,以获得适应于权重压缩敏感性的最优化阈值。同时,根据该阈值对每层的权重数据进行区域动态划分和稀疏化操作。具体方法如下。

(1)前向推断

首先,定义层级稀疏化因子 $\beta_l \in \{\beta_1,\cdots,\beta_L\}$,利用 $sigmoid(\cdot)$ 将其映射到(0,1) 区间,从而获得该层的稀疏率和阈值:

$$\rho_l = sigmoid(\beta_l) \tag{5}$$

其中,$\rho_l \in (0,1)$ 为 l 层的稀疏率。之后,通过阈值计算函数 $T(\cdot)$ 得出当前层的阈值:

$$\tau_l = T(\boldsymbol{w}_l, N_l \times \rho_l) \tag{6}$$

阈值 $\tau_l \in \{\tau_1,\cdots,\tau_L\}$,$\min(\boldsymbol{w}_{l,i}) \leqslant \tau_l < \max(\boldsymbol{w}_{l,i})$;$T(\cdot)$ 的输入为 l 层的权重和当前 ρ_l 值的权重索引。

最终,根据阈值 τ_l 对权重进行划分,当 $|\boldsymbol{w}_{l,i}| > \tau_l$ 时,权重作为可量化数据输入量化操作模块;当 $|\boldsymbol{w}_{l,i}| \leqslant \tau_l$ 时,权重作为可删减数据输入稀疏操作模块。其中,对于稀疏化操作,利用式(3) 和(4) 获得权值掩码矩阵 $\boldsymbol{mask}_l$,并与权重矩阵进行对位相乘,获得稀疏化后的权重矩阵 $\widetilde{\boldsymbol{W}}_l \in \{\widetilde{\boldsymbol{W}}_1,\cdots,\widetilde{\boldsymbol{W}}_L\}$。

(2)反向传播

可进行梯度反向传播是实现自动化学习阈值的关键。式(5)使用 $sigmoid$ 函数对各层稀疏率进行求解从而获得阈值,因此,对阈值的训练学习被转化为对稀疏因子 β_l 的迭代更新求解过程。根据梯度反向传播原理,定义稀疏因子反向传播更新公式为:

$$\beta'_l = \beta_l - lr \times \frac{\partial \rho_l}{\partial \beta_l} \times \sum_i^{N_l} \frac{\partial E_l}{\partial \boldsymbol{w}_l} \times \boldsymbol{mask}_l \tag{7}$$

式中:β'_l 为迭代更新后稀疏因子;lr 为学习率。根据式(5) 可知稀疏率对稀疏因子 β_l 的梯度为:

$$grad(\beta_l) = \partial \rho_l / \partial \beta_l = \mathrm{e}^{-\beta_l}/(1+\mathrm{e}^{-\beta_l})^2 \tag{8}$$

为提升稀疏因子在训练过程中对权重精度感知能力,在 $grad(\beta_l)$ 的基础上乘以 l 层权重的梯度和。具体的,在式(7) 中,计算 l 层误差 E_l 对权重 $\boldsymbol{w}_l$ 的偏导数,并利用掩码矩阵 $\boldsymbol{mask}_l$ 屏蔽求保留权重的梯度和。

最终,通过式(7)和(8),使用随机梯度下降方法实现对稀疏因子 β_l 的迭代更新,从而完成对稀疏化阈值的层级自适应训练。

自适应阈值稀疏化方法通过引入稀疏化因子构建具有自动学习能力的稀疏阈值,并通过设计反向传播方法实现渐进式的层级权重稀疏化,使权重在训练过程中随模型稀疏结构变化而进行动态更新,从而使模型权重和稀疏结构间取得良好的适应性。

2)多分支混合精度搜索量化

由于神经网络权重通常呈现以零点为均值的正态分布,因此,为尽可能覆盖其数据分布特点,以对称均匀量化方法为基础,将浮点 32 bit 权重向更低 bit 的整型数值进行仿射变换。权重参数的量化过程可表示为:

$$Q(\boldsymbol{W}_l, bit) = clamp\left(\left\lceil \boldsymbol{W}_l \times \frac{(2^{bit-1}-1)}{(R_{\max}-R_{\min})} \right\rceil, Q_n, Q_p\right) \tag{9}$$

式中:$Q(\boldsymbol{W}_l, bit)$ 为量化函数,输入为第 l 层权重向量 $\widetilde{\boldsymbol{W}}_l$ 和期望量化位宽 bit;$clamp(\cdot, Q_n, Q_p)$ 函数为截断函数,即分别以 $Q_n = -2^{bit-1}$ 和 $Q_p = 2^{bit-1}-1$ 为最小值和最大值量化边界;$\lceil\cdot\rceil$ 为向上取整函数;$R_{\max}$ 和 $R_{\min}$ 分别为浮点权重的最大值和最小值。

与稀疏化类似,由于神经网络每层敏感性的差异,致使在采用不同数据位宽进行表示时也呈现出不同的冗余性。因此,需要对不同层匹配不同的量化位宽,以提高特征和数据表示精度之间的适应性。

本文参考神经网络架构搜索思路,构建具有多分搜索结构的混合精度量化节点,并基于量化感知训练方法实现对量化位宽的自动化学习。其中,为表征不同分支下位宽的重要性,引入量化位宽选择因子 $\alpha_{l,i}$,其量化表达式为:

$$\hat{\boldsymbol{W}}_l = \sum_i \sigma(\alpha_{l,i}) \times Q(\boldsymbol{W}_l, bit_i) \tag{10}$$

式中:$\hat{\boldsymbol{W}}_l \in \{\hat{\boldsymbol{W}}_1,\cdots,\hat{\boldsymbol{W}}_L \mid \hat{\boldsymbol{W}}_l \in [Q_{n_i}, Q_{p_i}]\}$ 为量化后权重。为限制量化因子搜索可行解的空间范围,对 $\alpha_{l,i}$ 利用函数 $\sigma(\cdot)$ 进行归一化处理:

$$\sigma(\alpha_{l,i}) = \mathrm{e}^{\alpha_{l,i}} \Big/ \sum_{c=1}^{C} \mathrm{e}^{\alpha_{l,c}} \tag{11}$$

式中:C 为量化分支节点个数,$C \in \mathbb{Z}^+$;函数 $\sigma(\cdot)$ 将 $\alpha_{l,i}$ 转换为在[0,1] 且和为 1 的概率分布,用于表征各量化分支的重要性概率。如式(10) 所示,将乘以重要性概率的量化值进行求和,并利用梯度下降法求解,即获得重要性概率最高的量化分支。

为保证搜索过程模型收敛的稳定性,量化感知训练采用伪量化形式,即将浮点权重量化为整型数值后,再映射回浮点类型。如此,既可以将权重量化误差引入训练过程,同时也可保证训练时前向推理和反向传播的精确性。伪量化操作公式为:

$$Q_{Fake}(\boldsymbol{W}_l, bit) = \left(\frac{R_{\max}-R_{\min}}{2^{bit-1}-1}\right) \times Q(\boldsymbol{W}_l, bit) \tag{12}$$

因此,式(10)可被改写为:

$$\hat{\boldsymbol{W}}_l = \sum_i \sigma(\alpha_{l,i}) \times Q_{Fake}(\boldsymbol{W}_l, bit_i) \tag{13}$$

在反向传播过程中,为解决量化映射函数的不连续问题,可采用主流的直通式梯度评估器方法对量化过程的梯度进行近似。因此,根据链式求导法则可知,式(13)中目标函数对伪量化后权重的参数更新公式可

定义为：

$$\boldsymbol{w}'_{l,i} = \boldsymbol{w}_{l,i} - lr \times \sigma(\alpha_{l,i}) \times \frac{\partial E_l}{\partial \widetilde{\boldsymbol{W}}_l} \tag{14}$$

式中：$\boldsymbol{w}'_{l,i}$ 表示为通过迭代更新后的权重参数；$\partial E_l/\widetilde{\boldsymbol{W}}_l \approx \partial E_l/\partial \boldsymbol{w}_l$，即采用连续的浮点权重梯度近似量化后离散的权重梯度，从而保证量化误差在各层间的连续传递。就网络整体的收敛过程而言，量化产生的截断误差和近似误差是影响收敛效果的主要因素，通过在训练过程对误差的传递可以有效的提升模型权重对误差的感知能力，进而通过循环迭代逐渐纠正误差对最终模型预测结果的影响，使模型可以近似达到原始的收敛状态。

同时，采用对 $\arg\min_{bit} \mathcal{L}_{train}$ 目标函数进行参数空间搜索的方式，在给定的位宽搜索空间内，以 $\alpha_{l,i}$ 表征量化分支重要性，在量化感知训练的基础上，通过循环迭代搜索获得量化计算误差最小的分支，从而获得各层权重的最优化位宽表示，实现自动化的层级量化位宽选择，进而提高数据表示位宽对层级特征表达精度的适应性。但由于 $\sigma(\cdot)$ 函数本身限制，在完成压缩训练后残余的量化分支会对模型实际推理效果造成影响，因此需在训练后进行适当微调，以消除冗余分支。

3 实验与结果分析

为验证本文所提出的自适应联合压缩方法有效性，针对当前主流的神经网络模型，RseNet[14]、VGG[13]和 MobileNet[15]，在机器视觉领域图像分类任务中主流基准测试数据集 CIFAR-10[27]上进行实验，并对模型的精度和存储资源消耗情况进行对比分析。

3.1 实验设置

实验硬件环境为 AMAX 服务器，主要配置包括：Intel Xeon E5-2640 CPU 和 NVIDIA Tesla P100（16 GB）GPU×2；软件环境配置为 Python 3.10.8 和 Pytorch 1.13.0 深度学习框架。实验数据集 CIFAR-10：数据集包含 10 个类别，共 60 000 张彩色图像，其中包含 50 000 张训练图像和 10 000 张测试图像。

实验所用网络模型基础结构由 torchvision 库提供，构建网络模型后对权重进行随机初始化。通过文献[10]对超参数的设置方法，并根据实验环境的实际情况，对本文所用超参数进行如下设置：采用随机梯度下降和自适应矩估计优化算法分别对权重和稀疏因子 β_l、位宽选择因子 $\alpha_{l,i}$ 进行更新，其中，优化器学习率初始化值分别设置为 0.1 和 0.01，并采用 warmup 策略对学习率进行调整；量化位宽搜索空间中候选 *bit* 位宽设置为[3, 4, 5, 6, 7, 8]；训练数据批大小为 1 024；最大训练轮数为 120。

实验引用近年来在模型压缩领域的先进方法与本文提出方法进行对比，包括 APRS[28]、Hrank[16]和 ALE[19]3 类剪枝方法；MXQN[24]、DSQ[29]、SQ[30]、APOT[31]、HAWQ[23]和 Unified INT8[32]6 类量化方法；以及 DPP[11]、HFPQ[12]和 SQL[10]3 类联合压缩方法。为保证实验结果的公平性，本文参考上述对比方法中的评价指标，采用如下 4 项指标对本文提出方法性能进行分析。

1）准确率损失[10]，用于评价压缩前后的模型精度的损失情况，即用原始模型精度减去压缩后模型精度，表示为“*Acc.* ↓”。准确率损失如为正值，则表示压缩后模型精度下降；反之，则表示模型精度上升。

2）稀疏度[11]，用于评价模型的稀疏化程度，即为压缩后零值权重数量与权重总数的比值，表示为“S_p”。稀疏度值越大，表示模型压缩比越高。

3）平均权重位数[10]，用于评价模型权重的量化位宽，即为压缩后所有模型权重的量化位宽与权重总数的比值，表示为“*Ave. bits*”。平均权重位数越小，则表示模型量化程度越高，压缩率越大。

4）压缩率[10]，用于评价模型权重所占用存储资源的压缩程度。即原始模型权重字节数与压缩后模型权重字节数的比值，表示为“*Comp. ratio*”。压缩率越高，则表示压缩后模型所消耗的存储空间越小。

3.2 对比分析

在 CIFAR-10 数据集上分别对 VGG16、ResNet 和 MobileNet 模型进行对比实验，其中，ResNet 选择 18、20、56 和 110 这 4 个版本；MobileNet 选择 V1 和 V2 两个版本。实验结果数值均采用四舍五入形式保留一位小数进行对比。其中，VGG16 实验结果如表 1 所示。

表 1 VGG16 在 CIFAR-10 数据集上压缩结果对比

Table 1 Comparison of compression results of VGG16 on CIFAR-10 dataset

压缩方法	*Acc.* ↓/%	S_p/%	*Ave. bits*	*Comp. ratio*
APRS	0.9	80.5	32.0	5.1×
Hrank	−2.7	92.0	32.0	12.5×
MXQN	0.4	0	9.0	3.6×
DSQ	−0.1	0	1.0	32.0×
SQ	0.2	0	5.7	25.1×
DPP-C-F	0.4	84.4	8.0	25.6×
HFPQ	1.1	86.0	5.0	45.7×
SQL	1.5	90.0	1.8	177.8×
SQS	**1.3**	**94.6**	**4.1**	**143.0×**

由表 1 可知，SQS 在权重参数存储空间压缩率达到原模型 143.0 倍的情况下，准确率相对原始网络下降了

1.3%。相比于 Hrank 剪枝方法,在稀疏率大致相同时,压缩率提升了大约 130.5 倍。相较于 HFPQ 联合压缩方法,在精度损失相近的情况下,压缩率提升大约 97.3 倍。SQS 在不采用 1 和 2 bit 量化位宽情况下,与 SQL 联合压缩方法相比,压缩率下降了 34.8 倍,模型精度提升大约 0.2%。

VGG16 在 CIFAR-10 上使用 SQS 方法进行联合压缩的实验结果如图 3 所示,图 3 以模型网络层为单位分别统计了卷积层(convolution, Conv)和全连接层(full connection, Fc)的稀疏度、量化位宽以及压缩率情况。

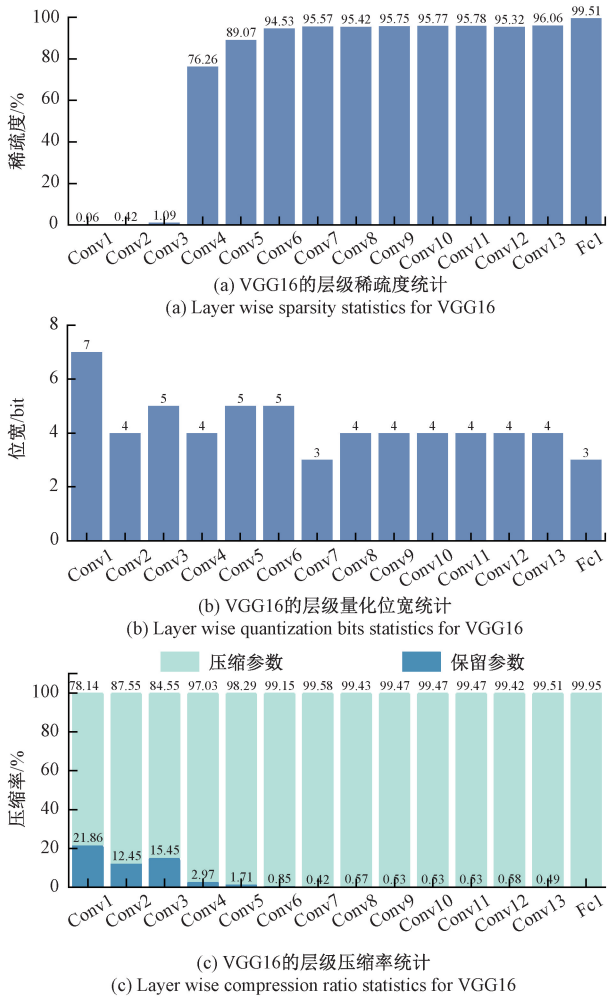


图 3　VGG16 在 CIFAR-10 上使用 SQS 联合压缩结果

Fig. 3　SQS compression results for VGG-16 on CIFAR-10

图 3(a)中,模型稀疏度总体随层级加深呈上升趋势,验证了层间连接冗余性的层级差异,且浅层的压缩敏感性高于深层;由图 3(b)可以看出,为保留更丰富的低维信息,模型第一层自适应的选择了相对高的量化位宽,而对于深层网络则对数据位宽要求较低;综合上述结果,图 3(c)对模型各层消耗存储资源的压缩情况以百分比形式进行了直观的比较。

为更好地验证 SQS 压缩方法在不同网络模型上的有效性。本文分别对 ResNet 18/20/56/110 和 MobileNet V1/V2 进行对比实验,实验结果如表 2 和 3 所示。

表 2　ResNet 在 CIFAR-10 数据集上压缩结果对比

Table 2　Comparison of compression results of ResNet on CIFAR-10 dataset

ResNet	压缩方法	*Acc.* ↓/%	S_p/%	*Ave. bits*	*Comp. ratio*
ResNet 18	DDP-C-F	−0.2	62.7	32.0	3.7×
	SQS	**1.4**	**92.1**	**3.9**	**101.2×**
ResNet 20	DSQ	0.5	0	1.0	32.0×
	APOT	0.6	0	2.0	16.0×
	HAWQ	0.2	0	2.0	13.1×
	SQL	0.1	46.0	1.9	35.4×
	SQS	**2.0**	**89.7**	**4.3**	**72.0×**
ResNet 56	Hrank	2.5	68.1	32.0	3.1×
	SQS	**2.4**	**95.1**	**4.4**	**151.6×**
ResNet 110	APRS	−0.4	69.2	32.0	3.2×
	Hrank	0.9	68.7	32.0	3.2×
	SQS	**1.1**	**93.7**	**4.0**	**127.9×**

表 3　MobileNet 在 CIFAR-10 数据集上压缩结果对比

Table 3　Comparison of compression results of MobileNet on CIFAR-10 dataset

ResNet	压缩方法	*Acc.* ↓/%	S_p/%	*Ave. bits*	*Comp. ratio*
MobileNet V1	DPP-C-F	0.5	63.1	32.0	2.7×
	SQS	**0**	**61.3**	**4.6**	**18.0×**
MobileNet V2	U. INT8	−1.1	0	8.0	4.0×
	ALE	−1.2	90.0	32.0	2.7×
	SQS	**0.9**	**62.6**	**4.3**	**19.7×**

由表 2 和 3 可以看出,使用 SQS 压缩方法可在准确率损失不超过 2.5%的情况下,相较于对比方法大幅提升模型压缩率。如相较于 Hrank,SQS 在 ResNet56 上的压缩率提升了 148.5 倍,而准确率损失相差仅为 0.1%;相较于 DPP-C-F,SQS 在 ResNet18 和 MobileNet V1 上的压缩率分别提升了 97.5 倍和 15.3 倍,而准确率损失最高仅相差 1.6%。

3.3　消融实验

为进一步验证联合方法对模型性能的影响,以模型结构具有广泛代表性的 ResNet 20 为例,基于 CIFAR-10 数据集进行消融实验。在相同模型结构和训练方式的基础上,

分别进行稀疏化、量化和联合压缩方法的对比实验,统计不同方法的训练损失误差。实验采用原始模型的训练过程作为对比基线,在 SQS 框架中分别屏蔽量化和稀疏化过程,以获得只稀疏化和只量化操作的压缩过程,与完整的联合压缩方法 SQS 进行对比,实验结果如图 4 所示。

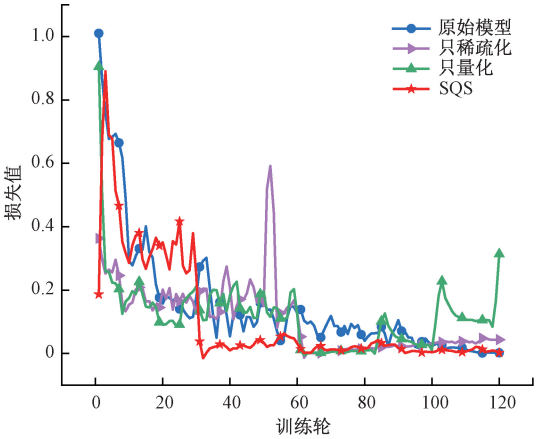


图 4 压缩方法对 ResNet 20 在 CIFAR-10 上的性能影响

Fig. 4 Impact of compression methods on the performance of ResNet 20 on CIFAR-10

由图 4 可以发现,模型损失值随训练轮数的增加整体呈下降趋势。其中,在达到最大设定轮数 120 时,SQS 方法与原始模型的损失值基本相同,而只进行稀疏化和只进行量化的压缩模型损失相对较高,说明 SQS 方法具有更好的模型精度保持能力。

此外,SQS 方法在训练后仍保留了多余的量化分支,需要在量化后通过模型重建对多余分支进行删除,然而,即使删除的分支在推理过程中占比较小,但仍会对模型实际推理效果产生影响。为解决该问题,本文在联合压缩训练基础上采用微调方法对压缩后模型进行精度恢复,直至收敛到满足应用需求状态。为验证微调训练对模型压缩的作用,采用与图 4 中 SQS 方法相同的实验设置,对 ResNet 20 在联合压缩训练和压缩后微调过程中在训练集上的精度随训练轮数的变化情况进行统计,如图 5 所示。

为方便对比分析,联合压缩训练与压缩后微调采用相同训练轮数,且微调过程中始终保持稀疏权重处于失活状态,仅对非零权重进行更新,不损失模型压缩比。由图 5 可知,在联合压缩训练过程中,模型精度随训练轮数增加而不断上升,并最终收敛至 92.87%;在微调过程中,模型虽基于已训练参数,但由于量化分支被去除而引入的计算误差,使其精度下降了约 0.3%,之后随微调过程的进行,在经过约 5 轮循环后模型精度基本恢复,并因其消除了额外的量化权重干扰,促使模型精度随微调轮数的增加而进一步提升,并最终收敛至 93.49%。由此可知,SQS 联合压缩框架在获得高压缩比的情况下,保证了

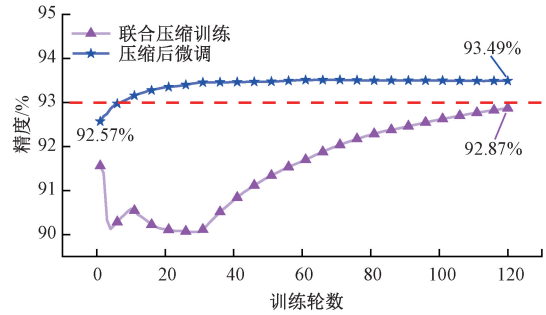


图 5 ResNet 20 联合压缩训练与压缩后微调的精度对比

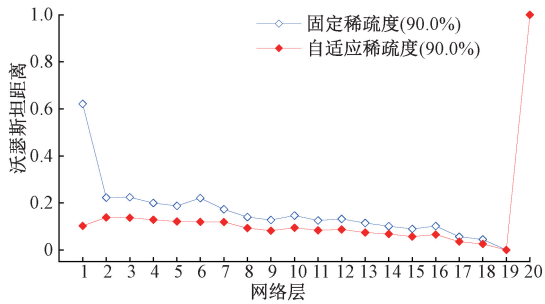
Fig. 5 Accuracy comparison of ResNet 20 joint compression training and post-compression fine-tuning

模型精度损失在可控范围内,对训练数据具有良好而稳定的收敛性。

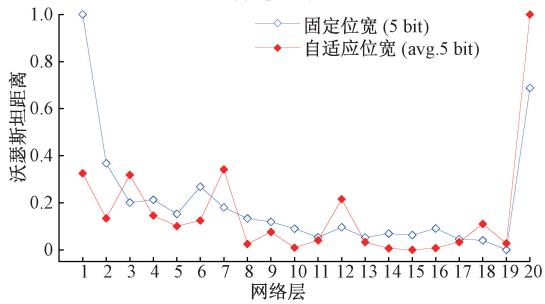
在验证微调方法对网络整体性能影响的基础上,从神经网络层级敏感性角度,分别对本文所提出的自适应稀疏度和自适应位宽方法在模型性能方面的影响进行分析验证。实验主要对比人为设置固定参数的传统压缩方法和自适应压缩方法之间差异,并参考文献[28]使用沃瑟斯坦距离(Wasserstein distance, WD)度量原始模型与压缩后模型参数间的层级分布差异,以间接表示该层的压缩敏感性。实验以 ResNet 20 为例,在 SQS 框架下分别进行稀疏化和量化对压缩敏感性的适应程度实验,结果如图 6 所示。其中,图 6(a)是对固定稀疏度和自适应稀疏度在达到相近的稀疏度(90%)时的层级 WD 值;图 6(b)是对固定位宽和自适应位宽在达到相近位宽(5 bit)时的层级 WD 值。

由图 6(a)和(b)可以看出,ResNet 20 模型整体呈现网络层由浅至深 WD 值下降的趋势,而最后一层为全连接层,由于该层权重数量远小于深层卷积层,因此 WD 变化相对剧烈。对比实验结果,自适应压缩相较于固定参数压缩获得了更低的 WD 值,说明其与原始全精度模型权重分布情况的匹配度更高,更易于保证压缩后模型精度不受影响,然而,由于自适应压缩方法的不稳定位宽造成了较大的层间信息波动,因此导致部分层的 WD 值较大,但自适应压缩方法与固定位宽压缩方法相比,整体 WD 值下降了约 0.85,且模型精度提升了 0.1%。

以 ResNet 20 模型在 CIFAR-10 数据集上的实验为例,通过对稀疏化和量化的联合压缩以及层级自适应压缩方法在模型性能方面的影响和分析表明,SQS 方法在获得高压缩率的同时可使模型处于较低的精度损失水平,而且通过对不同网络层参数冗余特点的自适应匹配,进一步降低了压缩前后模型参数的分布差异,从而提高了模型的精度还原能力,有利于减低微调成本,相较于单一且使用人为设置固定参数的传统压缩方法具有明显的优越性。



(a) 稀疏化
(a) Sparsity



(b) 量化
(b) Quantization

图 6　稀疏化和量化方法的层级压缩敏感性

Fig. 6　Layer-wise compression sensitivity of sparse and quantified methods

3.4　实际遥感图像处理应用评估

为进一步验证本文所提出的 SQS 联合压缩方法在实际应用中的有效性,采用由高分 1 号和高分 2 号卫星实际在轨拍摄的可见光遥感图像数据,对 SQS 在 VGG、ResNet 系列和 MobileNet 系列模型上的压缩性能进行实验验证。实验中所采用数据为自建数据集,包含船舶、陆地、海洋和云雾 4 类样本,总计 52 414 张全色可见光遥感图像。SQS 压缩方法配置与 3.1 节中一致,实验结果如表 4 所示。

表 4　SQS 在可见光遥感图像数据集上的压缩结果

Table 4　Compression results of SQS on remote sensing optical image dataset

模型	*Acc.* ↓/%	S_p/%	*Ave.* /bits	*Comp. ratio*
VGG16	0.3	93.3	3.2	150.1×
ResNet 18	−0.4	94.9	3.1	202.7×
ResNet 20	0.9	97.1	4.6	241.8×
ResNet 56	−0.5	96.5	3.6	253.7×
ResNet 110	−1.9	96.8	3.5	284.2×
MobileNet V1	1.2	97.2	4.0	283.1×
MobileNet V2	−1.2	88.4	3.4	80.3×

由表 4 中实验结果可知,对于 3 类测试模型,SQS 方法最高可获得 284.2 倍的压缩率,而整体精度损失最大仅为 1.2%,在模型参数规模和精度之间取得了良好的平衡。由此可见,本文所提出的 SQS 方法在实际的遥感目标分类任务中,仍能表现出较为优异的模型压缩性能和精度保持能力,充分表明了该方法在存储资源容量受限的遥感图像在轨处理领域具有较大的应用潜力和实际价值。

4　结　　论

本文提出了一种面向深度神经网络的自适应联合压缩方法,利用稀疏化和量化的协同作用对模型连接结构和参数位宽的冗余性进行深度压缩,以大幅提升模型压缩率,降低模型规模对存储资源的占用,同时通过可学习的压缩参数对模型稀疏化阈值和量化位宽进行层级的自适应优化,从而实现细粒度的精准压缩,在获得高压缩比的同时降低模型精度损失。本文在主流神经网络 VGG、ResNet 和 MobileNet 开展的实验结果表明,模型精度损失在小于 2.5% 的情况下,压缩率分别达到了 143.0×、151.6×和 19.7×,存储资源消耗降低最高为 148.5×,充分说明了 SQS 方法在平衡模型精度和压缩率方面的优越性。在未来的研究中,将在降低模型参数存储规模的研究基础上,进一步探索模型推理过程中参数传输对实际性能的影响,例如,通过对访存密集的中间特征图进行压缩,以及利用高效的参数编解码方法,降低参数传输频率,提高有效带宽占用比率,从而减少实际推理中参数传输对计算时间、能耗和数据吞吐量的不良影响。

参考文献

[1] 刘钊, 孙洁娣, 温江涛. 基于多层面压缩深度神经网络的轴承故障诊断[J]. 电子测量与仪器学报, 2022, 36(7): 189-198.
LIU ZH, SUN J D, WEN J T. Bearing fault diagnosis method based on multi-dimension compressed deep neural network[J]. Journal of Electronic Measurement and Instrumentation, 2022, 36(7): 189-198.

[2] 高晗, 田育龙, 许封元, 等. 深度学习模型压缩与加速综述[J]. 软件学报, 2021, 32(1): 68-92.
GAO H, TIAN Y L, XU F Y, et al. Survey of deep learning model compression and acceleration[J]. Journal

of Software, 2021, 32(1): 68-92.

[3] GHIMIRE D, KIL D, KIM S H J E. A survey on efficient convolutional neural networks and hardware acceleration[J]. Electronics, 2022, 11(6): 945.

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

[5] 彭继慎, 孙礼鑫, 王凯, 等. 基于模型压缩的 ED-YOLO 电力巡检无人机避障目标检测算法[J]. 仪器仪表学报, 2021, 42(10): 161-170.
PENG J SH, SUN L X, WANG K, et al. ED-YOLO power inspection UAV obstacle avoidance target detection algorithm based on model compression[J]. Chinese Journal of Scientific Instrument, 2021, 42(10): 161-170.

[6] 张政馗, 庞为光, 谢文静, 等. 面向实时应用的深度学习研究综述[J]. 软件学报, 2020, 31(9): 2654-2677.
ZANG ZH K, PANG W G, XIE W J, et al. Deep learning for real-time applications: A survey[J]. Journal of Software, 2020, 31(9): 2654-2677.

[7] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural network[C]. Proceedings of the 28th International Conference on Neural Information Processing Systems, 2015,1: 1135-1143.

[8] JACOB B, KLIGYS S, CHEN B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 2704-2713.

[9] HAN S, MAO H, DALLY W. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[C]. International Conference on Learning Representation, 2016.

[10] YANG H, GUI S, ZHU Y, et al. Automatic neural network compression by sparsity-quantization joint learning: A constrained optimization-based approach[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 2178-2188.

[11] GONZALEZ-CARABARIN L, HUIJBEN I A, VEELING B, et al. Dynamic probabilistic pruning: A general framework for hardware-constrained pruning at different granularities[J]. IEEE Transactions on Neural Networks Learning Systems, 2022, DOI: 10.1109/TNNLS.2022.3176809.

[12] FAN Y, PANG W, LU S J A I. HFPQ: Deep neural network compression by hardware-friendly pruning-quantization[J]. Applied Intelligence, 2021, 51(10): 7016-7028.

[13] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. Computer Science, 2014, DOI:10.48550/arXiv.1409.1556.

[14] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[15] SANDLER M, HOWARD A, ZHU M, et al. MobilenetV2: Inverted residuals and linear bottlenecks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4510-4520.

[16] LIN M, JI R, WANG Y, et al. Hrank: Filter pruning using high-rank feature map[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 1529-1538.

[17] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[18] KUMAR A, SHAIKH A M, LI Y, et al. Pruning filters with L1-norm and capped L1-norm for CNN compression[J]. Applied Intelligence, 2021, 51(2): 1152-1160.

[19] 魏钰轩, 陈莹. 基于自适应层信息熵的卷积神经网络压缩[J]. 电子学报, 2022, 50(10): 2398-2408.
WEI Y X, CHEN Y. Convolutional neural network compression based on adaptive layer entropy[J]. Acta

Electronica Sinica, 2022, 50(10): 2398-2408.

[20] LIU Z, LI J, SHEN Z, et al. Learning efficient convolutional networks through network slimming[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 2736-2744.

[21] XIAO X, WANG Z, RAJASEKARAN S. Autoprune: Automatic network pruning by regularizing auxiliary parameters[C]. Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019: 13699-13709.

[22] KUSUPATI A, RAMANUJAN V, SOMANI R, et al. Soft threshold weight reparameterization for learnable sparsity[C]. International Conference on Machine Learning, 2020: 5544-5555.

[23] DONG Z, YAO Z, GHOLAMI A, et al. Hawq: Hessian aware quantization of neural networks with mixed-precision[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 293-302.

[24] HUANG C, LIU P, FANG L. MXQN: Mixed quantization for reducing bit-width of weights and activations in deep convolutional neural networks[J]. Applied Intelligence, 2021, 51(7): 4561-4574.

[25] WANG Z, XIAO H, LU J, et al. Generalizable mixed-precision quantization via attribution rank preservation[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021: 5291-5300.

[26] PARK J H, KIM K M, LEE S. Quantized sparse training: A unified trainable framework for joint pruning and quantization in DNNs[J]. ACM Transactions on Embedded Computing Systems, 2022, 21(5): 1-22.

[27] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images[R]. Technical Report, 2009, 1(1): 1-60.

[28] SUN Q, CAO S, CHEN Z. Filter pruning via automatic pruning rate search[C]. Proceedings of the Asian Conference on Computer Vision, 2022: 4293-4309.

[29] GONG R, LIU X, JIANG S, et al. Differentiable soft quantization: Bridging full-precision and low-bit neural networks[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 4852-4861.

[30] RAZANI R, MORIN G, SARI E, et al. Adaptive Binary-Ternary quantization[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: 4613-4618.

[31] LI Y, DONG X, WANG W. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks[J]. ArXiv Preprint, 2019: ArXiv: 1909.13144.

[32] ZHU F, GONG R, YU F, et al. Towards unified int8 training for convolutional neural network[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 1969-1979.

作者简介

姚博文,2015 年和 2018 年于哈尔滨理工大学获得学士和硕士学位,现为哈尔滨工业大学博士生,主要研究方向为领域定制计算方法、软硬件协同的高能效计算方法、星载计算硬件系统。

E-mail: bwyao@hit.edu.cn

Yao Bowen received his B. Sc. degree and M. Sc. degree both from Harbin University of Science and Technology in 2015 and 2018, respectively. He is currently a Ph. D. candidate at Harbin Institute of Technology. His main research interests include domain-special computing, energy-efficient computing with software and hardware co-design, and the on-board computing hardware system.

彭喜元,分别在 1984 年、1987 年和 1992 年于哈尔滨工业大学获得学士学位、硕士学位和博士学位,现为哈尔滨工业大学教授,主要研究方向为自动测试和高级故障诊断技术。

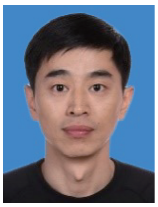
E-mail: pxy@hit.edu.cn

Peng Xiyuan received his B. Sc., M. Sc. and Ph. D. degrees all from Harbin Institute of Technology in 1984, 1987 and 1992, respectively. He is currently a professor and a Ph. D. advisor at Harbin Institute of Technology. His main research interests include automatic test and advanced fault diagnostics technology.

于希明,2016 年于哈尔滨工业大学获得学士学位,现为哈尔滨工业大学博士研究生,主要研究方向为遥感图像处理、语义分割、深度学习模型计算加速等。

E-mail: yuximing@ hit. edu. cn

Yu Ximing received his B. Sc. degree from Harbin Institute of Technology in 2016. He is currently a Ph. D. candidate at Harbin Institute of Technology. His main research interests include remote sensing image processing, semantic segmentation and computing acceleration for deep learning model, etc.



刘连胜,分别在 2006 年、2008 年和 2017 年于哈尔滨工业大学获得学士学位、硕士学位和博士学位,现为哈尔滨工业大学副教授,主要研究方向为信息物理系统、基于 FPGA 的高能效计算技术、故障预测与健康管理等。

E-mail: lianshengliu@ hit. edu. cn

Liu Liansheng received his B. Sc., M. Sc. and Ph. D. degrees all from Harbin Institute of Technology in 2006, 2008 and 2017, respectively. He is currently an associate professor and a Ph. D. advisor at Harbin Institute of Technology. His main research interests include cyber physical system, FPGA-based energy-efficient computing technology, fault prognostics and health management, etc.

彭宇(通信作者),2004 年于哈尔滨工业大学获得博士学位,现为哈尔滨工业大学教授、博士生导师,主要研究方向为虚拟仪器和自动测试、故障预测与健康管理以及可重构计算等。

E-mail: pengyu@ hit. edu. cn

Peng Yu (Corresponding author) received his Ph. D. from Harbin Institute of Technology in 2004. He is currently a professor and a Ph. D. advisor at Harbin Institute of Technology. His main research interests include virtual instruments and automatic test technologies, prognostics and system health management, and reconfigurable computing, etc.