物联网中模型剪枝技术：现状、方法和展望

赵军辉[1]，李怀城[1]，王东明[2]，李佳珉[2]，周一青[3]，束锋[4]
（1. 北京交通大学电子信息工程学院，北京 100044；2. 东南大学信息科学与工程学院，江苏 南京 211189；
3. 中国科学院计算技术研究所，北京 100190；4. 海南大学信息与通信工程学院，海南 海口 570228）

摘　要：在物联网（IoT, Internet of things）技术迅速发展的背景下，IoT 设备受到计算能力、存储空间、通信带宽以及电池寿命的限制，在运行复杂的人工智能（AI, artificial intelligence）算法中，特别是深度学习模型中面临着挑战。模型剪枝技术通过减少神经网络中的冗余参数，在不损伤 AI 模型性能的前提下可以有效地降低计算和存储需求。该技术适合用于优化部署在物联网设备上的 AI 模型。首先，回顾了当前流行的结构化剪枝和非结构化剪枝两种典型的模型剪枝技术，两种剪枝技术分别适用于不同的应用场景。之后，详细分析了这些方法在 IoT 环境下的多样化应用。最后，结合最新研究成果，详细探讨了当前模型剪枝的局限性，并对物联网中模型剪枝方法未来的发展方向进行了展望。

关键词：物联网；资源限制；模型剪枝；人工智能；深度学习
中图分类号：TN915.08
文献标志码：A
doi：10.11959/j.issn.2096−3750.2024.00448

Model pruning techniques in the Internet of things: state of the art, methods and perspectives

ZHAO Junhui[1], LI Huaicheng[1], WANG Dongming[2], LI Jiamin[2], ZHOU Yiqing[3], SHU Feng[4]
1. School of Electronic Information Engineering, Beijing Jiaotong University, Beijing 100044, China
2. School of Information Science and Engineering, Southeast University, Nanjing 211189, China
3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
4. School of Information and Communication Engineering, Hainan University, Haikou 570228, China

Abstract: In the context of the rapid development of Internet of things (IoT) technology, IoT devices faced challenges in running complex artificial intelligence (AI) algorithms, especially deep learning models, due to the limitations of computing power, storage space, communication bandwidth, and battery life. Model pruning technology could effectively reduce computation and storage requirements by reducing redundant parameters in neural networks without impairing the performance of AI models. This technique was extremely suitable for optimising AI models deployed on IoT devices. Firstly, two typical model pruning techniques-structured pruning and unstructured pruning, which were currently popular and suitable for different application scenarios, were reviewed. Secondly, the diverse applications of these methods in IoT environments were analysed in detail. Finally, the limitations of the current model pruning were discussed in detail in the light of the latest research results, and the future development direction of model pruning methods in IoT was outlooked.

Key words: IoT, resource constraints, model pruning, AI, deep learning

收稿日期：2024−10−15；**修回日期：**2024−12−10
通信作者：赵军辉，junhuizhao@hotmail.com
基金项目：国家自然科学基金资助项目（No. U2001213）；国家重点研发计划（No. 2020YFB1807204）
Foundation Items: The National Natural Science Foundation of China (No. U2001213), The National Key Research and Development Program of China (No. 2020YFB1807204)

0 引言

物联网（IoT, Internet of things）是由大量相互连接的物理设备组成的网络，这些设备能够通过互联网进行数据收集和信息交换，并根据环境变化做出响应[1]。随着全球数字化进程的推进，IoT 已经被广泛应用于智能家居[2]、智慧城市[3]和工业自动化[4]等多个领域。IoT 设备的数量迅速增长，全球算力需求和数据总量呈现高速增长的趋势，仅 2022 年全球总数据量就达到了 81 ZB[5]。人工智能（AI, artificial intelligence），特别是深度学习（DL, deep learning）技术，正在 IoT 设备中得到越来越广泛的应用。深度学习凭借其强大的功能，为 IoT 设备提供了更高的智能水平，使其能够进行复杂的数据分析和模式识别[6]。然而，深度学习在 IoT 中的应用面临着由庞大参数量和复杂计算结构带来的极高计算资源消耗的挑战。

近年来，人工智能生成内容（AIGC, artificial intelligence generated content）在计算机科学及相关领域受到了广泛关注，各种 AIGC 产品不断涌现。AIGC 基于大型语言模型（LLM, large language model）实现，这些模型在多种任务中表现出色[7]，但其卓越性能伴随着巨大的尺寸和计算需求。以 GPT-175B[8]为例，该模型包含 1 750 亿个参数，使用半精度格式存储至少需要 320 GB 的内存空间。要运行此模型，至少需要 5 个 A100 GPU，每个 GPU 拥有 80 GB 内存。然而，这样的硬件要求是 IoT 设备无法满足的，因为它们通常受限于计算能力、存储空间和能源供应。另一方面，随着边缘计算、联邦学习等技术的发展，学习范式已经从云端下沉到边缘环境[9]。边缘环境中，不断增长的数据数量大、种类多，IoT 设备难以承受逐渐增长的通信与计算的算力消耗。为了跟踪环境变化并提供可靠的服务，AI 模型必须不断改进以捕捉短期趋势，这需要大量的计算资源，例如在线训练[10]，在线训练会增加计算资源和时间资源的消耗。此外，Google[11]以相同参数量为基准，对比了稠密小模型与稀疏大模型的性能。实验结果表明，在参数量相同的情况下，稀疏大模型拥有更好的效果。所以，有必要基于大模型进行稀疏处理，以得到效果更好的模型。

为了解决 IoT 设备算力不足的问题，模型压缩技术被引入来降低模型的计算和存储需求。其中，模型剪枝是一种备受关注的技术，本文主要关注 IoT 中模型剪枝相关技术的现状、方法和展望。模型剪枝通过删除模型中的冗余参数，有效缩小模型规模，显著减少计算和存储需求，同时保持模型性能不变。大量冗余参数对模型的性能影响很小甚至没有影响，因此，直接修剪这些冗余参数后，模型的性能下降最小。同时，修剪可以使模型更易存储[12]、提升内存效率[13]、提升计算效率[14]。对于资源有限的 IoT 设备，模型剪枝能够在保持精度的同时大幅降低能耗[15]，延长电池寿命，满足 IoT 应用的关键需求。模型剪枝的引入在很大程度上改善了 IoT 设备的资源利用效率，这是因为 IoT 设备通常需要在有限的能耗和硬件条件下处理复杂的智能任务[16]。随着 IoT 应用场景的扩大和设备数量的增长，剪枝技术不仅提高了设备的处理效率，还减轻了网络传输和数据处理的负载。这为开发复杂的智能应用程序提供了可能性，同时保障了系统的实时性和响应速度。另外，模型剪枝还支持 AI 模型在低带宽、低延迟的边缘环境中独立运行，使得分布式系统体系结构变得更加可行。

尽管模型剪枝在深度学习领域已经取得了许多进展[17]，但在 IoT 中的具体应用和发展仍然是一个新兴的研究领域，同时也面临着许多挑战。例如，如何在保证模型性能的同时最大化地进行参数修剪，以及如何在不同的应用场景中动态调整剪枝策略，以适应变化的环境需求等。因此，本文通过系统的综述，详细分析 IoT 中模型剪枝的现有研究，讨论未来可能的研究方向，为相关研究提供参考和指导。本文的目标是支持 IoT 领域的研究人员有效地了解与应用模型剪枝技术，以实现更智能、更高效的 IoT 系统。

1 模型剪枝概述

本节首先介绍了模型剪枝的概念和基础的分类，并简单地对比了模型剪枝与其他模型压缩算法的差别，最后对模型剪枝的优缺点进行了分析。

1.1 模型剪枝的介绍

模型剪枝是一种用于压缩 AI 模型的技术，其目标是通过删除冗余或对模型性能贡献较小的参数（如权重、神经元或通道），在保证模型性能的同时，减小模型的计算资源需求，模型剪枝样例如

图1所示。剪枝过程可以在不同的粒度级别上进行，例如逐个权重、逐层、逐通道或整个滤波器。剪枝可以是一次性的，即在训练后进行，也可以是逐步的，即在训练过程中交替进行剪枝和微调。一次性剪枝的优点是效率高，而逐步剪枝可以更好地保证模型性能。

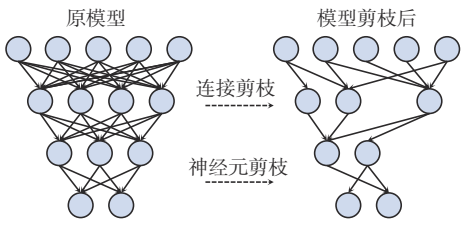

图1　模型剪枝样例

模型剪枝技术已经在各种类型的神经网络中得到了应用，包括全连接层、卷积神经网络（CNN, convolution neural network）和循环神经网络（RNN, recurrent neural network）等。尤其在计算机视觉任务中，如图像分类[18]、目标检测[19]和语义分割[20]等，剪枝已经成为一种常用的模型压缩方法，并取得了显著的成果。根据剪枝标准和策略，模型剪枝主要分为两大类，即结构化剪枝和非结构化剪枝，结构化剪枝和非结构化剪枝如图2所示。

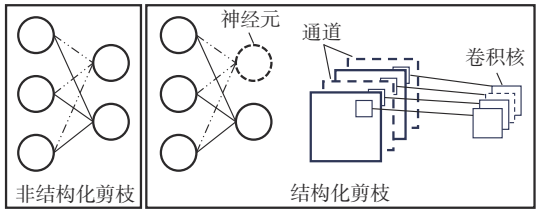

图2　结构化剪枝和非结构化剪枝

1.1.1　结构化剪枝

在网络层级或更大结构上进行剪枝的技术。与逐个剪掉单独的参数（如权重）不同，结构化剪枝通常会以更大粒度的方式操作，例如剪掉整个滤波器、神经元、通道或层。由于剪掉的是整个神经元或滤波器，结构化剪枝生成的模型通常能够很好地与现有硬件加速器兼容。这是因为剪枝后的模型仍然保持较规则的结构。另一方面，由于剪掉的是较大结构单元，剪枝后的模型比较容易集成到现有的深度学习框架中，减少了部署的复杂性。

1.1.2　非结构化剪枝

逐个删除模型中的个别权重。这种方式更加灵活，可以在神经网络的任何位置剪掉权重，而无须考虑模型结构。通常，这种剪枝方式通过对权重的大小进行排序，剪掉绝对值较小的权重。逐个权重进行剪枝，可以更精细地选择要剪掉的参数，实现更高的剪枝密度，从而显著减少计算和存储需求。

1.2　模型剪枝与其他模型压缩技术对比及优缺点

1.2.1　模型剪枝与其他模型压缩技术对比

除模型剪枝以外，常见的模型压缩技术包括量化（quantization）[21]、低秩近似（low rank approximation）[22]、知识蒸馏（knowledge distillation）[23]等技术。模型剪枝与其他模型压缩技术对比见表1。模型剪枝与其他模型压缩方法最大的差异在于，模型剪枝针对模型特定的节点进行删除判断。

1.2.2　模型剪枝的优点

一方面，模型剪枝技术显著地降低了模型的计算和存储需求，使得AI模型能够在资源受限的设备上运行，尤其是IoT系统中的端侧设备。剪枝有效地减少了模型参数数量，从而减小了模型占用的存储空间。这对于存储资源有限的IoT设备极为重要。剪枝后的模型可以大幅地降低能耗，这对电池寿命长和能效要求高的IoT设备尤其重要。此外，剪枝还可以加快模型的响应速度，提高运行效率，适合在实时性要求高的应用中使用。

表1　　模型剪枝与其他模型压缩技术对比

模型压缩技术	机制	相似点	差异点
模型剪枝	通过一定规则删除冗余和不重要的参数来减少模型的复杂性和大小	模型剪枝与其他模型压缩方法在本质上都是减少模型的参数量与计算复杂度，从而缩小模型规模，提升其在资源受限环境中的可部署性。这些方法通过精简模型结构或去除冗余参数，力求在减少计算资源消耗的同时，尽可能维持模型的原有性能	/
量化	将高位宽的权重通过均匀映射等方式映射为低位宽的权重，常见的低位宽为16位、8位、4位		量化需要将权重的值降低位宽，模型本身结构不会变化；模型剪枝会改变网络结构
低秩近似	通过对模型权重矩阵进行分解来减少参数		低秩分解着重于矩阵的计算与简化，模型剪枝更注重网络结构本身
知识蒸馏	让小模型学习大模型的输出来提升性能，通常适用于需要完全重构小模型的场景		知识蒸馏需要训练一个新的小模型，模型剪枝直接基于原模型进行操作

另一方面，剪枝技术可以增强模型的可移植性，使其适应不同平台和环境，提高在多种设备上的部署灵活性。例如，Qi等[24]提出了一种端-边-云模式下典型的模型剪枝应用架构。在IoT应用中，可以在云服务器端设计一个通用的大型神经网络模型，然后由用户在边缘节点下载模型，针对特定任务进行剪枝，用于解决不同的问题，端-边-云模式的模型剪枝应用架构如图3所示。这样做避免了反复设计模型的需求，节省了算力资源。

1.2.3　模型剪枝的缺点

尽管模型剪枝具有诸多优点，但也存在一些不足之处。一个主要缺点是，在大多数情况下，原模型需要针对不同的任务以及应用场景进行不同程度的剪枝，且剪枝后的模型需要进行重新训练，以恢复或优化其性能。这些操作可能需要反复进行，这无疑增加了模型训练的时间。

另外，剪枝可能导致模型性能降低，特别是在过度剪枝的情况下，因此，剪枝过程需要非常谨慎，以平衡模型的精度与剪枝率。剪枝的有效性依赖于准确识别模型中不重要的参数，这需要复杂地计算和调优。模型剪枝的自动化程度较低，通常需要专家的深度参与，并且剪枝策略难以一概而论地适用于所有网络架构。剪枝后模型的结构可能变得不规则，如非均匀分布的参数，这不利于某些硬件的加速优化。

2　常见的模型剪枝算法

2.1　结构化剪枝

相比于非结构化剪枝，结构化剪枝通过生成对硬件友好的模型，提供了实际加速的好处[25]。按照不同的标准，结构化剪枝可以大致分为权重剪枝（weight-based pruning）[26]、激活值剪枝（activation-based pruning）[27-29]、正则化剪枝（regularization pruning）[30-32]、神经结构搜索（NAS, neural architecture search）剪枝[33-38]和动态剪枝（dynamic pruning）[27, 39]。在本节后续的小节中会对这些算法进行详细介绍。

2.1.1　权重剪枝

权重剪枝主要通过一定的评估规则，确定模型中哪些滤波器或者通道更重要，进而进行保留。权重剪枝不需要依赖输入数据，所需要的算力资源更少，实际中应用较多。权重剪枝分为两个方向，分别是范数过滤与相关性过滤。

对于范数过滤，计算滤波器的规范值作为度量，对整个网络结构中的每个权重计算其对应的重要性，判断是否保留。一般的范数表达式为

$$\left\| F_i^l \right\|_p = \sqrt[p]{\sum_{n=1}^{N_l} \sum_{k_1=1}^{K_l} \sum_{k_2=1}^{K_l} \left| F_i^l(n, k_1, k_2) \right|^p} \tag{1}$$

其中，N_l为第l层输入通道的大小；K_l为滤波器的大小；$i \in N_l$为第l层的第i个滤波器；p为范数的阶数，两种常见的范数阶数为L1-范数（曼哈顿范数）和L2-范数（欧几里得范数）。Li等[14]提出了一种针对CNN的权重剪枝方法，该方法通过计算每个滤波器的L1-范数来确定各自的重要性，对重要性低的滤波器进行剪枝。该研究进一步探究了基于L2-范数的剪枝效果，发现两种范数的效果相当，L1-范数略优。He等[40]基于L2-范数过滤提出了一种软剪枝（SFP, soft filter purning）策略。与其他剪枝方式不同的是，SFP在滤波器确定被剪掉的信息后，仍将其保留并参与后续的训练，直到下一次训练轮次结束才将其剪掉。

对于相关性过滤，与范数过滤不同，利用同一层滤波器之间的关系来发现冗余的滤波器，进而进行剪枝。He等[26]发现基于范数滤波计算出的范数标准差太小，且很多滤波器有相似的重要性，难

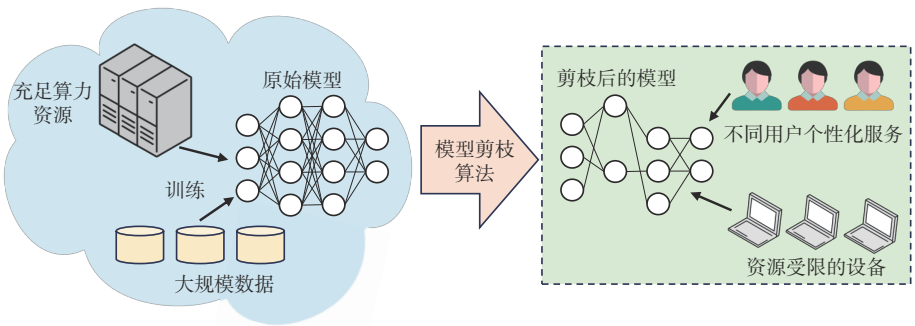

图3　端-边-云模式的模型剪枝应用架构

以取舍。He等[26]提出了一种基于几何中位数（GM, geometric median）的滤波器评价指标用于计算各个滤波器的重要性。通过计算GM，对每个滤波器进行判断，若滤波器的值接近GM，则表示信息是冗余的，可以被剪掉。Yvinec等[41]提出了一种无数据的结构化剪枝方法，利用一种新的标量深度神经网络（DNN, deep neural network）权重分布密度的自适应序列，增加由其权重向量表示的相同神经元的数量，并根据冗杂神经元的相对相似性合并冗余的神经元，达到网络修剪的目的。

2.1.2　激活值剪枝

基于激活值的剪枝方法利用激活映射来进行模型剪枝，其主要关注神经网络中神经元或通道的激活值，剪除那些对最终输出影响较小的部分。首先，通过前向传播统计每个神经元的激活值，通常会统计多个输入样本的激活值，以获得稳定的均值和方差等统计量。然后，根据这些统计量设定剪枝标准，如选择激活值较小或方差较小的神经元或通道进行剪除，因为这些部分对模型最终输出的影响相对较小。依据设定的标准，移除选定的神经元或通道是常见的做法，其中一种策略是直接将这些部分的参数置0，或者从模型结构中实际删除。剪枝完成后，通常需要对模型进行微调，即在剪枝后的模型上继续训练一段时间，以恢复因剪枝造成的性能下降。为了评估当前层滤波器的重要性，可以通过当前层与相邻层的激活值来做剪枝决策。

Hu[29]发现在一些大型的网络中，部分结构无论收到什么输入，很大一部分神经元的输出几乎为0。Hu[29]认为0激活的神经元是冗余的，可以在不影响网络整体准确性的情况下删除这些神经元。通过循环去除这些0激活的神经元并进行重复训练这一过程，最终在不损失模型精度的前提下达到高参数压缩比。Lin等[28]认为对于CNN每一层的激活图所包含的信息量可以通过其本身的秩来评估，而激活图秩的大小对应滤波器的重要性。首先，需要计算由单个滤波器得到的激活图的平均秩。然后，将所有激活图的平均秩进行排序，通过设定阈值来决定哪些激活图对应的滤波器需要保留。Gao等[27]利用前一层激活图的平均池化结果作为滤波器的重要性评估依据。

2.1.3　正则化剪枝

在神经网络中，可以使用正则化剪枝来实现结构化的稀疏性，从而削减不必要的通道和滤波器等组件，提高网络的效率。正则化剪枝可以通过添加不同的稀疏性正则器来学习结构化稀疏网络。稀疏性正则化通过惩罚不必要的参数，使某些参数变为0，从而达到剪枝的效果。对于包含批量归一化（BN, batch normalization）层的网络，可以将稀疏性正则化应用到BN的参数上。稀疏性正则化也可以直接应用于滤波器上，其中，Group Lasso是一种常用的正则化方法，可以结构性地使滤波器稀疏化。Torsten等[32]将一般Group Lasso定义为以下凸优化问题的解。

$$\min_{\beta \in \mathbb{R}^p}\left(\left\| y-\sum_{g=1}^{G}\boldsymbol{X}_g\boldsymbol{\beta}_g\right\|_2^2+\lambda\sum_{g=1}^{G}\sqrt{n_g}\left\|\boldsymbol{\beta}_g\right\|_2\right) \tag{2}$$

其中，y为目标值，激活矩阵被分成G组，构成矩阵$\boldsymbol{X}_g$，$\boldsymbol{\beta}_g$为系统向量，n_g为第g组的大小，λ为调优参数。在对滤波器剪枝的情况下，第一项可以看作激活映射的重构误差，即损失值，第二项可以写成一种范数表达。Liu等[30]为模型中的每个BN层引入了一个缩放因子γ作为评估中间激活图不同通道重要性的指标，通过裁剪重要性低的通道来实现模型剪枝。缩放因子γ对应于每个通道的权重，激活图的不同通道根据γ进行加权。权重越低，代表该通道的重要性越低，越需要被剪枝。在训练过程中，通过对γ添加L1正则化来实现其稀疏性。Zhuang等[31]认为L1正则化会将所有的缩放因子推向0，即L1正则化器会导致网络过于简化，缺乏神经元之间的区分。更合理的修剪方法是只抑制比例因子为0的神经元，同时保留更重要的神经元。Zhuang等[31]提出一种极化正则化（polarization regularizer）的神经元级别结构剪枝。在原有的L1项上再加一个惩罚项，使缩放因子尽可能地与它们的平均值分开，避免过度剪枝。

2.1.4　神经结构搜索剪枝

NAS是一种自动化方法，用于设计神经网络架构，其目标是找到性能最优的神经网络结构。基于NAS的剪枝算法，首先对模型进行初步的架构搜索，然后对找到的架构进行剪枝，以减少参数量和计算量。常见的NAS剪枝算法包括进化算法、强化学习（RL, reinforcement learning）[42-43]和梯度优化。

基于进化算法的剪枝模拟进化过程，通过选择、变异和交叉来迭代生成更优的架构。Liu等[38]

提出了一种基于元学习的神经网络通道剪枝方法，用于深层神经网络的自动通道修剪。该方法首先训练一个PruningNet，它可以为任何目标网络生成剪枝结构的权重参数，然后采用进化搜索算法在约束条件下搜索最优结构。Lin等[34]提出了一种基于人工蜂群（ABC, artificial bee colony）算法的通道剪枝新方法。为了应对深层网络中庞大的剪枝结构组合问题，该方法将保留的通道限制在特定的空间，从而显著减少了剪枝结构的组合数量。最终，该方法将最优剪枝结构的搜索转化为一个优化问题，并集成ABC算法进行自动求解。

基于RL的剪枝，通过训练一个RL代理来生成网络架构，并根据性能反馈进行优化。He等[33]利用深度确定性策略梯度算法（DDPG, deep deterministic policy gradient）[44]来提供模型压缩策略，基于DDPG的模型剪枝架构如图4所示。

网络以层为单位，对当前层的结构嵌入得到当前层的状态，将状态传入DDPG智能体，通过智能体的预测得到当前层的稀疏比。在当前层按照稀疏比压缩后，再对下一层进行相同的处理，直到所有层压缩完毕。基于每秒浮点运算次数（FLOPs, floating point operations per second）与压缩后模型的准确率构成DDPG的奖励函数R为

$$R_{\mathrm{FLOPs}} = -\mathrm{Error}\cdot \ln(\mathrm{FLOPs}) \tag{3}$$

$$R_{\mathrm{Param}} = -\mathrm{Error}\cdot \ln(\#\mathrm{Param}) \tag{4}$$

其中，Error为模型误差，#Param为模型参数量。除了单智能体RL方法，Alwani等[35]提出了一种多智能体RL架构，为网络中的每个通道分配一个代理，并使用轻策略梯度方法来学习保留或删除哪些神经元或通道。网络中的每个智能体只需要学习一个参数是否需要剪掉，从而训练速度更快。

基于梯度优化的剪枝，通过设计一个连续的搜索空间并使用梯度下降方法来优化架构参数。它通过计算和更新网络参数的梯度来决定哪些连接或神经元是重要的，进而进行剪枝。Ning等[37]认为，基于离散搜索的分层剪枝效率较低，并提出了一种可微稀疏分配（DSA, differentiable sparsity allocation）的端到端剪枝方法。该方法利用一种新的可微分剪枝过程，通过基于梯度的优化来确定分层剪枝比。DSA在连续空间中分配稀疏性，比基于离散求值和搜索的方法更为有效。Li等[36]提出了可微分超剪枝（DHP, differentiable hyper pruning）进行网络自动剪枝。该方法通过在网络中引入超网络来生成每一层的权重参数，然后使用可微的剪枝策略对超网络进行训练，以实现网络的剪枝。

2.1.5　动态剪枝

动态剪枝是在神经网络推理过程中，根据当前输入数据的特性，动态决定剪枝策略的一种方法。与静态剪枝不同，动态剪枝不在训练后固定移除神经元或权重，而是依据输入数据自适应地调整网络结构。在训练过程中，动态剪枝旨在通过调整剪枝掩模，保持模型的表现能力。这种方法也被称为软剪枝，以便在需要时恢复不当的剪枝决策。在推理过程中，动态指网络根据不同输入样本进行的自适应修剪[27]。

Gao等[27]提出了一种称为特征增强与抑制（FBS, feature boosting and suppression）的方法，用于动态调整CNN输出通道的强化或抑制。FBS在现有卷积层中引入了小型辅助连接，这使得重要信息可以自由流动，同时直接跳过不重要通道的计算。与永久性删除通道的剪枝方法不同，该方法保留了完整的网络结构，并通过动态跳过不重要的

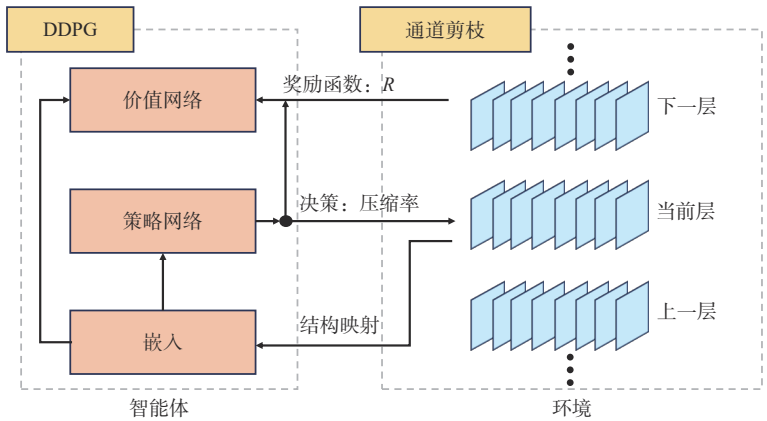

图4　基于DDPG的模型剪枝架构

输入和输出通道来提高卷积操作的速度。Lin等[39]提出了一种动态剪枝与反馈（DPF, dynamic model pruning with feedback），使用一种新的显著标准，在整个训练过程中识别网络中的重要权重，以提取出候选掩码。该算法不仅生成修剪后的稀疏模型，还联合生成一个密集模型，并用于在训练过程中纠正剪枝错误。

2.2　非结构化剪枝

非结构化剪枝删除以单个权重为基础，如神经元之间的层间连接，而不是像结构化剪枝那样删除整个神经元或滤波器。

Guo等[45]提出，由于隐藏神经元之间复杂的相互联系，一旦网络剪枝开始，参数的重要性可能会发生巨大变化，且修剪之后的连接没有机会恢复，不恰当的修剪可能会导致严重的准确性损失。另一方面，为了保证高压缩率与高准确率，剪枝的过程需要多次交替剪枝和再训练的迭代，这是个非常耗时的过程[46]。文献[45]提出了一种动态网络手术（DNS, dynamic network surgery）方法，通过持续的网络维护来切断冗余连接。该方法涉及两个关键操作，即修剪和拼接。修剪操作是为了压缩网络模型，但过度修剪或不正确的修剪会带来准确性损失。为了弥补这一损失，在网络剪枝的过程中加入拼接操作，从而在任何时候都可以恢复重要的连接，带参数冗余模型的动态网络手术模型如图5所示。

非结构化剪枝也常被用于LLM。Frantar和Alistarh[47]提出了一种名为SparseGPT的一次性剪枝方法。该方法通过将模型剪枝问题转化为一系列大规模稀疏回归实例来简化处理。然后，这些实例通过一个新开发的近似稀疏回归求解器来解决，该求解器的高效性使其能够在单个图形处理器（GPU, graphics processing unit）上处理1 750亿规模的GPT（generative pre-trained transformer）模型。此外，SparseGPT在精度方面足够精确，可以在无须进行任何微调的情况下，将剪枝后的模型精度损失降至可忽略不计的程度。SparseGPT虽然不需要额外训练，但其仍需进行一个计算密集型的权重更新过程[48]。Sun等[48]提出了一种通过权重和激活的剪枝方法。该方法结合权重大小和输入激活的重要性来决定需要剪除的权重。其核心是引入了一种新的修剪度量方式，通过每个权重的大小和相应输入激活的范数乘积来进行评估，并借助一小组校准数据进行估计。在每个线性层输出项中对局部权重进行比较，并去除较低优先级的权重。该方法在保留剩余权重完整性的前提下，成功地将LLM修剪为高度稀疏的形式。

3　物联网中模型剪枝的应用

3.1　模型剪枝在微控单元中的应用

微控单元（MCU, micro control unit）是低功耗、资源受限的设备，通常基于系统级芯片（SoC, system on chip），只有几KB到几MB的内存，而AI模型的大小通常为几十MB甚至更大，难以直接存储和运行。通过去除模型中冗余或不重要的部分，模型剪枝可以在保证模型性能的前提下显著地降低模型的计算复杂度和存储需求，使其能够在MCU上高效运行。鉴于嵌入式设备性能有限，需要在部署前通过修剪足够的权重来压缩模型。必须在不显著影响准确性的前提下，对模型进行深度压缩，并确保剪枝后的模型可以高效运行[49]。

在MCU上，结构化剪枝方法表现出硬件友好性。Widmann等[50]在树莓派上使用结构化剪枝对类似LeNet-5的CNN模型进行优化，以分类Fashion-MNIST数据集。结果表明，当模型参数被结构化剪枝去除75%时，推理能耗减少了59.06%，准确率仅下降了3.01%。Liberis和Lane[51]提出了一种用于CNN的可微分结构化剪枝方法，该方法整合了模型的MCU特定的资源使用情况和参数重要性反馈，以获得高度压缩但准确的模型。在基准图像和音频分类任务的评估中，该方法有效地提高了

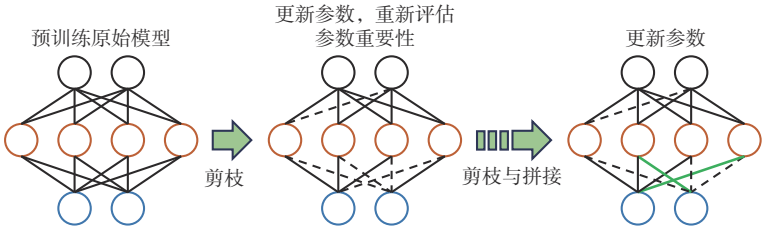

图5　带参数冗余模型的动态网络手术模型

MCU关键资源的使用率，且几乎没有额外开销。由MCU驱动的智能传感器被大量部署在边缘节点，也被用于音频相关任务。Mohaimenuzzaman等[52]对经过结构化剪枝压缩的模型进行了研究，并应用于MCU上的音频分类任务，其性能与二值化网络XNOR-Net[53]进行了实验对比。其中，二值化网络是通过将模型权重与激活值简化为二进制形式的模型轻量化技术。实验结果表明，在处理更为复杂的任务时，剪枝优化后的轻量化模型能够更好地拟合数据，展现出了更好的性能。

对于MCU上的非结构化剪枝，Han等[49]提出了具有修剪单元选择、执行前修剪优化、运行时加速和执行后低成本存储的DTMM。其中，DTMM是用于MCU部署和执行机器学习模型的库。在低功耗设备上，如MCU，直接部署机器学习模型能够提升数据保护，减少带宽使用，并支持设备上的数据处理。Widmann等[50]通过改进的非结构化剪枝来降低MCU的能耗，并实现了剪枝后的模型精度与能耗的平衡。这些实例展示了模型剪枝技术对于MCU部署机器学习模型的有效性。

3.2　模型剪枝在视觉任务中的应用

视觉任务是IoT应用中最重要的任务之一，涵盖了图像分类、目标检测、语义分割等众多领域。其目标是从图像或视频数据中提取有意义的信息，并应用于各种场景[54]。然而，现有的AI模型，如VGG16[55]、YOLO[56]、Visual Transformer[57]等，虽然在公开数据集如CIFAR-10[58]、ImageNet[59]、PASCAL VOC[60]上取得了显著的成果，但其庞大的参数量和计算复杂度使其难以直接部署到资源受限的IoT设备上。许多IoT应用需要实时响应，如自动驾驶的感知任务、无人机控制等，这对模型的推理速度提出了很高的要求。

结构化剪枝和非结构化剪枝都可以在一定程度上加速模型的推理，但由于非结构化剪枝的稀疏性且不规则，其硬件加速效果有限。在低时延场景中，研究者更关注结构化剪枝技术。Junior等[61]提出了一种基于滤波器剪枝的修剪算法，随机消除滤波器，减少给定DNN模型的内存占用，可以在用户指定的内存占用范围内找到给定DNN的修剪版本。修剪后的DNN能部署在内存仅有1 GB的树莓派设备上，用于检测城市河流洪水事件。Zhao等[62]提出了一种基于神经网络滤波器的细化剪枝量化和权重的策略。该方法将神经网络中的滤波器细化为条状滤波器条带，接着利用一个评估指标来细化滤光片的部分重要性，剪掉不重要的滤光片条，最后再将剩余的滤波器重新组织。在Android 11软件运行环境与ARMx86硬件运行环境中，利用手机摄像头或系统文件获取图像数据，分别基于ResNet[63]与VGG16进行了模型压缩实验。实验结果表明，该方法能将ResNet的参数量和计算量分别减少到原来的1/4和1/5，VGG16的参数量和计算量分别减少到原来的1/14和1/3。蒋文翔等[64]提出了一种利用BN缩放因子对通道进行加权修剪的方法，用于YOLOv3进行目标检测工作。修剪后的YOLOv3可以部署在硬件配置较低的机器环境中，且其检测准确率与原模型相当。Liu等[65]提出了一种多级滤波器修剪算法，通过输出特征图的图像熵来判断每个滤波器的重要性。在CIFAR-10数据集上，该算法可以减少VGG16模型的54.5%浮点运算和31.9%的图形内存。Wang等[66]利用特征图之间的量化相似性来指导滤波器修剪过程，去除计算过程中的冗余信息。在边缘设备上，该模型剪枝方法可以将MobileNetV2[67]模型的推理速度提高1.53倍。

不同的剪枝方法可以相互结合，实现协同效应。Zhao等[68]提出了一种基于分层信息熵的融合剪枝算法，该方法首先通过亲和传播聚类识别相似滤波器并去除冗余部分，然后使用信息熵分层和BN层缩放因子进一步剪枝通道，最后通过微调训练恢复精度，实现了在不损失网络精度的前提下减小网络模型大小。剪枝后的模型在嵌入式设备上单次运行仅需252.84 ms。

另外，一些先进的数学方法也可以用于支持和优化模型剪枝过程。Dai等[69]提出了一种改进的基于朴素贝叶斯推理的通道参数修剪方法，以获得更高准确性的稀疏模型。在配备GPU处理器Mali-T880的手机Samsung Edge上进行了测试实验，该方法用于AlexNet[70]与VGG16模型，在精度、参数压缩和浮点运算减少方面都达到了不错的效果。Wu等[71]提出了一种基于Taylor的网络剪枝方案，将预先训练的DNN部署到检测机器人中。即在VGG16和ResNet18两种网络上进行普遍表面缺陷检测的试验。结果表明，该文献所提的方法在不降低损伤检测性能的情况下显著提高了资源效率。同

样是基于Taylor扩展，Qi等[24]将Taylor准则引入模型剪枝，用于评估CNN中每个通道的重要性，重新分配剩余的通道，移除重要性低的通道。在CIFAR-10数据集上进行了VGG模型的试验评估，该方法能够显著地减少FLOPs和模型参数量，同时保持高准确性。

3.3　模型剪枝在分布式架构中的应用

IoT应用日益增多，数据规模不断扩大，传统的集中式架构逐渐难以满足需求。分布式架构应运而生，它将计算和存储资源分散到各个节点上，通过协同工作来处理数据和执行任务。这种架构具备更高的可扩展性、容错性和数据处理能力，成为IoT发展的重要趋势。分布式架构虽然优势明显，但面临着节点间通信成本高的挑战，尤其是在模型训练过程中，需要频繁交换参数。而大型模型会进一步加剧这一问题，成为制约分布式架构性能的关键因素。模型剪枝通过压缩模型大小，有效地降低了节点间传输的数据量，从而缓解通信瓶颈，提高分布式训练效率。

联邦学习（FL, federated learning）是一种分布式学习范式，它使大型IoT设备能够协作训练共享模型，同时保护本地数据的隐私[72]。Xu等[72]设计了一种具有个性化模型修剪和自适应通信的高效DFL框架，利用模型剪枝技术设计了一种个性化的剪枝比率确定方法。对于个性化剪枝，FL中每个设备根据通信和隐私要求定制修剪比率。在响应邻居的模型请求之前，各设备根据确定的剪枝比例对本地模型进行剪枝。其中，剪枝的关键操作是通过评估它们对最终精度的重要性来从模型中去除冗余权值。

在分布式IoT环境中，FL的协作效率通常受到通信资源和计算资源限制的影响。Xu等[73]认为IoT设备通常具有有限的计算资源和较差的网络连接，这使得遵循FL模式训练DNN不可行或非常缓慢。Xu等[73]提出了一种新的高效FL框架，包括3个阶段，即结构化剪枝、权重量化和选择性更新，以降低计算、存储和通信的成本，从而加速FL训练过程。在性能远低于正常GPU的设备上进行试验评估，同时将链路带宽设置为1 Mbit/s，实验结果表明，将该方法用于AlexNet与VGG16模型可以有效地控制训练开销，同时保证学习性能。Du等[74]针对去中心化FL会受到系统的通信资源约束的问题，提出了一种通信网络拓扑修剪方法，通过在确保收敛的同时修剪低数据速率的不良链路来降低通信开销。Prakash等[75]考虑到底层DNN非常庞大，无法将其直接部署到资源受限的计算和内存受限的IoT设备上，中央服务器和客户端之间频繁交换模型更新可能会导致通信瓶颈。Prakash等[75]同时利用基于权重的剪枝与量化来对DNN进行压缩，通过减少FL的计算、内存和网络占用空间，低端IoT设备能够有效参与FL过程。

FL也未能幸免于一些新兴的攻击，如成员推理攻击[76]。Shen等[76]提出了一种轻量级防御机制，针对IoT中的联邦学习能够防止局部模型和全局模型的成员推理攻击，同时保证模型的高效性。该机制在每一轮联邦学习过程中为局部模型添加特定设计的修剪扰动。参数筛选器选择那些对模型测试准确性影响较小，但对成员推理攻击贡献较大的模型参数。然后，使用噪声发生器找到可以在保证较高模型精度的同时降低攻击精度的剪枝噪声，从而保护每个参与者的成员隐私。

4　挑战与展望

根据前文对剪枝算法的介绍与模型剪枝在IoT中的应用实例，模型剪枝技术已得到了广泛的探索与认可。尽管模型剪枝在IoT中的应用展现出了巨大的潜力，但在实际应用中仍面临着一系列挑战。

4.1　结构化剪枝与非结构化剪枝存在的不足

在IoT设备上，模型剪枝在结构化和非结构化两种形式中各自存在一些缺陷。结构化剪枝尽管可以简化模型架构，但会以固定的模式（例如，剪掉特定的神经网络通道、卷积核或整个神经元）来减少模型的尺寸，即缺少灵活性，可能导致模型中重要的特征或通道被删除，从而对模型的整体性能和精度产生较大影响。而非结构化剪枝虽然提供了更细粒度的参数减少，但不规则的稀疏性限制了在资源受限的IoT设备上的硬件加速效果，增加了算法实现的复杂性和计算开销。未来的发展需要专门为IoT场景设计混合剪枝策略，以同时保障模型性能和优化计算效率。此外，开发支持稀疏运算的IoT友好硬件、自适应剪枝算法以及能够灵活适应不同场景的自适应剪枝策略，将显著提升IoT设备的智能处理能力，促进剪枝技术在实际应用中更广泛地实施和创新。

4.2 如何开发无须再训练且计算资源消耗少的剪枝方法

当前，大多数剪枝方法依赖于重新训练过程，这会带来额外的计算开销和时间成本。对于资源受限的IoT设备，如何开发无须训练的剪枝方法成为亟待解决的问题。这类方法可以在不需要重新训练或经过少量训练的情况下，直接对模型进行剪枝，极大地减少了实施成本和复杂度。这将助力IoT设备在面对动态变化的任务需求时，快速适应新的计算约束。

4.3 如何开发分布式环境下的模型剪枝算法

目前，模型剪枝主要应用于单一设备，以提升性能和节省资源。然而，随着计算资源的不断分布和扩展，未来这种技术的应用范围将不再局限于单一设备，有可能形成多设备协同的处理方式，从而提高整体系统的效率与效能。这种新的处理方式不仅可以充分利用现有的硬件资源，还可以在多设备之间均衡负载，互相补足。更为重要的是，多设备协同处理在提升系统整体效率与效能方面具有巨大的潜力。例如，当一个设备的计算能力达到极限时，可以即时将部分计算任务分配到其他空闲设备上，从而避免处理复杂任务时可能遇到的性能瓶颈。协同处理也可以使数据的传输和处理更加高效快捷，因为不同设备间可以进行并行计算，减少了等待时间和延迟。通过多设备的协同工作，整个系统将更智能、更高效地运行，不仅实现了节能降耗，还可以应对更复杂、更高负荷的计算任务，为未来各类高科技应用场景提供坚实的技术支撑。

4.4 模型剪枝的局限性

模型剪枝虽然可以有效地减少IoT设备的计算负担，但其在性能和通用性上仍存在一些局限性。剪枝操作可能导致模型精度下降，特别是在处理复杂任务时。此外，不同的剪枝算法通常需要针对特定的模型和数据集进行调整和优化，这增加了实际应用的难度。未来的研究需要开发更加通用且具有鲁棒性的剪枝方法，以确保在不同应用场景下都能获得良好的剪枝效果和模型性能。为了进一步提升模型在IoT设备上的运行效率，模型剪枝需要与其他模型压缩技术，如量化、知识蒸馏、低秩近似等，结合起来使用。多种压缩技术的协同作用可以从不同层面减少模型的计算和存储开销，从而达到更优的压缩效果。例如，将剪枝技术与量化技术相结合，可以在减少模型参数数量的同时大幅降低模型的位宽。这要求研究人员深入理解不同压缩技术之间的互补性，并开发出统一的框架来实现多技术联合优化。

5 结束语

模型剪枝作为一种高效的模型压缩技术，在IoT中的应用呈现出广阔的前景。通过对常见剪枝算法与IoT中的剪枝算法应用的深入探讨，可以发现这些方法各具特色，能够针对不同的应用场景和资源约束条件提供定制化的解决方案。模型剪枝在IoT设备中显著地提升了计算效率和资源利用率，使得低功耗、低存储、高性能的智能应用成为可能。

然而，随着IoT技术以及LLMs的不断发展，剪枝算法面临着诸多的挑战，例如，如何在保证模型精度的同时进一步提升剪枝率，如何在动态环境中实现自适应剪枝，LLM如何在IoT设备上高效地部署等。因此，未来的研究工作需结合IoT特有的需求，持续优化和创新剪枝技术，以应对日益复杂的应用场景和不断增长的数据量。模型剪枝将在推动智能IoT设备普及和提升网络整体性能方面发挥关键作用，是IoT领域值得深入研究和探索的重要方向。

参考文献：

[1] 张琦, 杨浩, QUEK T, 等. 物联网的核心本质: 数据联网[J]. 物联网学报, 2017, 1(3): 10-16.
ZHANG Q, YANG H, QUEK T, et al. Kernel of Internet of things: Internet of data[J]. Chinese Journal on Internet of Things, 2017, 1(3): 10-16.

[2] ALZOUBI A. Machine learning for intelligent energy consumption in smart homes[J]. International Journal of Computations, Information and Manufacturing (IJCIM), 2022, 2(1): 62-75.

[3] HERATH H M K K M B, MITTAL M. Adoption of artificial intelligence in smart cities: a comprehensive review[J]. International Journal of Information Management Data Insights, 2022, 2(1): 100076.

[4] RIBEIRO J, RUI L M, ECKHARDT T, et al. Robotic process automation and artificial intelligence in industry 4.0-a literature review[J]. Procedia Computer Science, 2021, 181: 51-58.

[5] 中国信息通信研究院. 中国算力发展指数白皮书 [R]. 2023.
CAICT. China arithmetic development index white paper[R]. 2023.

[6] 殷浩然, 苗世洪, 韩佶, 等. 基于三维卷积神经网络的配电物联网异常辨识方法[J]. 电力系统自动化, 2022, 46(1): 42-50.
YIN H R, MIAO S H, HAN J, et al. Anomaly identification method for distribution Internet of things based on three-dimensional convolutional neural network[J]. Automation of Electric Power Systems, 2022, 46(1): 42-50.

[7] CHANG Y P, WANG X, WANG J D, et al. A survey on evaluation of large language models[J]. ACM Transactions on Intelligent Systems and Technology, 2024, 15(3): 1-45.

[8] BROWN T B. Language models are few-shot learners[J]. arXiv preprint, 2020, arXiv:14165.

[9] CHEN S Z, TAO Y M, YU D X, et al. Privacy-preserving collaborative learning for multiarmed bandits in IoT[J]. IEEE Internet of Things Journal, 2021, 8(5): 3276-3286.

[10] TIAN H S, YU M C, WANG W. Continuum: a platform for cost-aware, low-latency continual learning[C]//Proceedings of the ACM Symposium on Cloud Computing. New York: ACM, 2018: 26-40.

[11] ZHU M, GUPTA S. To prune, or not to prune: exploring the efficacy of pruning for model compression[J]. arXiv preprint, 2017, arXiv:01878.

[12] ARDAKANI A, JI Z Y, SMITHSON S C, et al. Learning recurrent binary/ternary weights[J]. arXiv preprint, 2018, arXiv: 1809.11086.

[13] YANG T J, CHEN Y H, SZE V. Designing energy-efficient convolutional neural networks using energy-aware pruning[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2017: 6071-6079.

[14] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[J]. arXiv preprint, 2016, arXiv: 1608.08710.

[15] MOHANTY L, KUMAR A, MEHTA V, et al. Pruning techniques for artificial intelligence networks: a deeper look at their engineering design and bias: the first review of its kind[J]. Multimedia Tools and Applications, 2024: 1-75.

[16] LI Q P, ZHAO J H, GONG Y, et al. Energy-efficient computation offloading and resource allocation in fog computing for Internet of everything[J]. China Communications, 2019, 16(3): 32-41.

[17] LI Z, MENG L. A survey of model pruning for deep neural network[C]//Proceedings of the International Symposium on Advanced Technologies and Applications in the Internet of Things, 2022: 25-34.

[18] HAN S, MAO H Z, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv preprint, 2015, arXiv: 1510.00149.

[19] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 779-788.

[20] MINAEE S, BOYKOV Y, PORIKLI F, et al. Image segmentation using deep learning: a survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(7): 3523-3542.

[21] XIAO G, LIN J, SEZNEC M, et al. Smoothquant: accurate and efficient post-training quantization for large language models[C]//Proceedings of the 40th International Conference on Machine Learning. New York: ACM, 2023, 202: 38087-38099.

[22] GUO Y Y, WANG G Z, KANKANHALLI M. PELA: learning parameter-efficient models with low-rank approximation[C]//Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2024: 15699-15709.

[23] BEYER L, ZHAI X H, ROYER A, et al. Knowledge distillation: a good teacher is patient and consistent[C]//Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022: 10915-10924.

[24] QI C, SHEN S B, LI R P, et al. An efficient pruning scheme of deep neural networks for Internet of things applications[J]. EURASIP Journal on Advances in Signal Processing, 2021, 2021(1): 31.

[25] HE Y, XIAO L G. Structured pruning for deep convolutional neural networks: a survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(5): 2900-2919.

[26] HE Y, LIU P, WANG Z W, et al. Filter pruning via geometric median for deep convolutional neural networks acceleration[C]//Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2019: 4335-4344.

[27] GAO X T, ZHAO Y R, DUDZIAK Ł, et al. Dynamic channel pruning: feature boosting and suppression[J]. arXiv preprint, 2018, arXiv: 05331.

[28] LIN M B, JI R R, WANG Y, et al. HRank: filter pruning using high-rank feature map[C]//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 1526-1535.

[29] HU H. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures[J]. arXiv preprint, 2016, arXiv: 03250.

[30] LIU Z, LI J G, SHEN Z Q, et al. Learning efficient convolutional networks through network slimming[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2017: 2755-2763.

[31] Zhuang T, ZHANG Z X, HUANG Y H, et al. Neuron-level structured pruning using polarization regularizer[J]. Advances in neural information processing systems, 2020, 33: 9865-9877.

[32] TORSTEN H, DAN A, TAL B N, et al. Sparsity in deep learning: pruning and growth for efficient inference and training in neural networks[J]. Journal of Machine Learning Research, 2021, 22(241): 1-124.

[33] HE Y H, LIN J, LIU Z J, et al. AMC: AutoML for model compression and acceleration on mobile devices[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 815-832.

[34] LIN M, JI R, ZHANG Y, et al. Channel pruning via automatic

structure search[J]. arXiv preprint, 2020, arXiv: 200108565.

[35] ALWANI M, WANG Y, MADHAVAN V. DECORE: deep compression with reinforcement learning[C]//Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022: 12339-12349.

[36] LI Y, GU S, ZHANG K, et al. DHP: differentiable meta pruning via hypernetworks[C]//Proceedings of the Computer Vision-ECCV 2020: 16th European Conference. New York: ACM, 2020, 608-624.

[37] NING X F, ZHAO T C, LI W S, et al. DSA: more efficient budgeted pruning via differentiable sparsity allocation[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020: 592-607.

[38] LIU Z C, MU H Y, ZHANG X Y, et al. MetaPruning: meta learning for automatic neural network channel pruning[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2019: 3295-3304.

[39] LIN T, STICH S U, BARBA L, et al. Dynamic model pruning with feedback[J]. arXiv preprint, 2020, arXiv: 2006.07253.

[40] HE Y, KANG G, DONG X, et al. Soft filter pruning for accelerating deep convolutional neural networks[J]. arXiv preprint, 2018, arXiv: 06866.

[41] YVINEC E, DAPOGNY A, CORD M, et al. Red: looking for redundancies for data-free structured compression of deep neural networks[J]. arXiv preprint, 2021, arXiv: 2105.14797.

[42] NIE Y W, ZHAO J H, LIU J, et al. Energy-efficient UAV trajectory design for backscatter communication: a deep reinforcement learning approach[J]. China Communications, 2020, 17(10): 129-141.

[43] NIE Y W, ZHAO J H, GAO F F, et al. Semi-distributed resource management in UAV-aided MEC systems: a multi-agent federated reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2021, 70(12): 13162-13173.

[44] ZHAO J H, HE L, ZHANG D Y, et al. A TP-DDPG algorithm based on cache assistance for task offloading in urban rail transit[J]. IEEE Transactions on Vehicular Technology, 2023, 72(8): 10671-10681.

[45] GUO Y, YAO A, CHEN Y. Dynamic network surgery for efficient dnns[J]. arXiv preprint, 2023, arXiv: 1608.04493.

[46] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[J]. Advances in Neural Information Processing Systems, 2015: 1135-1143.

[47] FRANTAR E, ALISTARH D. SparseGPT: massive language models can be accurately pruned in one-shot[J]. arXiv preprint, 2023, arXiv: 2301.00774.

[48] SUN M, LIU Z, BAIR A, et al. A simple and effective pruning approach for large language models[J]. arXiv preprint, 2023, arXiv: 11695.

[49] HAN L X, XIAO Z, LI Z J. DTMM: deploying TinyML models on extremely weak IoT devices with pruning[C]//Proceedings of the IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2024: 1999-2008.

[50] WIDMANN T, MERKLE F, NOCKER M, et al. Pruning for power: optimizing energy efficiency in IoT with neural network pruning[M]//Communications in Computer and Information Science. Cham: Springer Nature Switzerland, 2023: 251-263.

[51] LIBERIS E, LANE N D. Differentiable neural network pruning to enable smart applications on microcontrollers[C]//Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. New York: ACM, 2022, 6(4): 1-19.

[52] MOHAIMENUZZAMAN M, BERGMEIR C, MEYER B. Pruning vs XNOR-Net: a comprehensive study of deep learning for audio classification on edge-devices[J]. IEEE Access, 2022, 10: 6696-6707.

[53] RASTEGARI M, ORDONEZ V, REDMON J, et al. XNOR-Net: ImageNet classification using binary convolutional neural networks[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016: 525-542.

[54] SHI J Y, ZHAO J H, WANG D M, et al. Lane detection by variational auto-encoder with normalizing flow for autonomous driving[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(12): 21757-21768.

[55] YANG H Y, NI J G, GAO J Y, et al. A novel method for peanut variety identification and classification by improved VGG16[J]. Scientific Reports, 2021, 11(1): 15756.

[56] SHAFIEE M J, CHYWL B, LI F, et al. Fast YOLO: a fast you only look once system for real-time embedded object detection in video[J]. arXiv preprint, 2017, arXiv:05943.

[57] ZHANG Q, YANG Y-B. Rest: an efficient transformer for visual recognition[J]. Advances in Neural Information Processing Systems, 2021, 34: 15475-85.

[58] KRIZHEVSKY A. Learning multiple layers of features from tiny images[J]. Handbook of Systemic Autoimmune Diseases, 2009, 1(4): 1-60.

[59] DENG J, DONG W, SOCHER R, et al. ImageNet: a large-scale hierarchical image database[C]//Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Piscataway: IEEE Press, 2009, 248-255.

[60] EVERINGHAM M, ALI ESLAMI S M, VAN GOOL L, et al. The pascal visual object classes challenge: a retrospective[J]. International Journal of Computer Vision, 2015, 111(1): 98-136.

[61] Junior F F, NONATO L G, RANIERI C M, et al. Memory-based pruning of deep neural networks for IoT devices applied to flood detection[J]. Sensors, 2021, 21(22): 7506.

[62] ZHAO M, TONG X D, WU W X, et al. A novel deep-learning model compression based on filter-stripe group pruning and its IoT application[J]. Sensors, 2022, 22(15): 5623.

[63] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscat-

away: IEEE Press, 2016: 770-778.
[64] 蒋文翔, 张传臣. 一种基于剪枝YOLO的传感器识别方法[J]. 中国科技信息, 2019(22): 71-73.
JIANG W X, ZHANG C C. A sensor identification method based on pruning YOLO[J]. China Science and Technology Information, 2019(22): 71-73.
[65] LIU X G, WU L S, DAI C, et al. Compressing CNNs using multi-level filter pruning for the edge nodes of multimedia Internet of things[J]. IEEE Internet of Things Journal, 2021, 8(14): 11041-11051.
[66] WANG Z D, LIU X X, HUANG L, et al. QSFM: model pruning based on quantified similarity between feature maps for AI on edge[J]. IEEE Internet of Things Journal, 2022, 9(23): 24506-24515.
[67] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]//Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 4510-4520.
[68] ZHAO M, HU M, LI M, et al. A novel fusion pruning algorithm based on information entropy stratification and IoT application[J]. Electronics, 2022, 11(8): 1212.
[69] DAI C, LIU X G, CHENG H Q, et al. Compressing deep model with pruning and tucker decomposition for smart embedded systems[J]. IEEE Internet of Things Journal, 2022, 9(16): 14490-14500.
[70] ALOM M Z, TAHA T, YAKOPCIC C, et al. The history began from AlexNet: a comprehensive survey on deep learning approaches[J]. arXiv preprint, 2018, arXiv: 01164.
[71] WU R T, SINGLA A, JAHANSHAHI M R, et al. Pruning deep convolutional neural networks for efficient edge computing in condition assessment of infrastructures[J]. Computer-Aided Civil and Infrastructure Engineering, 2019, 34(9): 774-789.
[72] XU Y, XIAO M J, WU J, et al. Enhancing decentralized federated learning with model pruning and adaptive communication[J]. IEEE Transactions on Industrial Informatics, 2024, PP(99): 1-15.
[73] XU W Y, FANG W W, DING Y, et al. Accelerating federated learning for IoT in big data analytics with pruning, quantization and selective updating[J]. IEEE Access, 2021, 9: 38457-38466.
[74] DU M X, ZHENG H F, GAO M, et al. Adaptive decentralized federated learning in resource-constrained IoT networks[J]. IEEE Internet of Things Journal, 2024, 11(6): 10739-10753.
[75] PRAKASH P, DING J H, CHEN R, et al. IoT device friendly and communication-efficient federated learning via joint model pruning and quantization[J]. IEEE Internet of Things Journal, 2022, 9(15): 13638-13650.
[76] SHEN M, MENG J, XU K, et al. MemDefense: defending against membership inference attacks in IoT-based federated learning via pruning perturbations[J]. IEEE Transactions on Big Data, 2024, PP(99): 1-13.

[作者简介]

赵军辉（1973– ），男，博士，北京交通大学电子信息工程学院教授、博士生导师，主要研究方向为无线与移动通信及相关应用、5G移动通信技术、高速铁路通信、车载通信网络、无线定位和认知无线电。

李怀城（1998– ），男，北京交通大学电子信息工程学院博士生，主要研究方向为绿色通信、模型压缩、图像处理。

王东明（1977– ），男，博士，东南大学信息科学与工程学院教授、博士生导师，主要研究方向为无蜂窝大规模分布式MIMO基础理论与技术、6G无线传输关键技术研究、毫米波分布式MIMO理论与技术、5G物理层协议栈。

李佳珉（1983– ），男，博士，东南大学信息科学与工程学院教授、博士生导师，主要研究方向为6G无蜂窝智能无线接入网、海量终端高可靠低时延通信、通感一体化、6G极致连接、未来移动通信综合试验平台的研发及试验验证。

周一青（1975– ），女，博士，中国科学院计算技术研究所无线通信技术研究中心研究员副主任、博士生导师，主要研究方向为宽带无线通信技术、通信与计算融合、异构网络、协同传输、绿色无线电等。

束锋（1973– ），男，博士，海南大学信息与通信工程学院教授、博士生导师，主要研究方向为宽智能无线通信、信息安全、大规模MIMO测向与定位。