基于剪枝的卷积神经网络压缩算法综述

◆ 王兆丰年　欧中洪　宋美娜
北京邮电大学计算机学院（国家示范性软件学院），北京 100876

摘　要　近年来，随着深度学习技术的飞速发展，卷积神经网络（CNN）变得越来越流行，应用范围也日益广泛。为提高模型准确性，CNN逐渐增加网络的深度和宽度，导致网络结构逐渐膨胀，现有的深度神经网络通常拥有数百万甚至上千万个参数，需要耗费大量的计算和存储资源。在计算能力受限的移动设备（如智能手机或平板电脑）上部署和运行深度神经网络，面临巨大挑战。这亟须一种可在有限计算能力设备上提高运行效率且不影响精度的方法。神经网络压缩是在不影响计算精度的前提下，减小网络规模并提高存储效率的一类有效方法，近些年得到广泛关注和应用。神经网络压缩包括多个分支，本文对基于剪枝的神经网络压缩算法进行全面论述，针对剪枝算法现有的分类法进行更详细的分类展开，并进行系统描述；将调查的文献按分类法进行归纳总结，详细论述每个工作的主要贡献；针对目前剪枝算法的特点和不足，对其未来发展趋势展开讨论。

关键词：深度学习；卷积神经网络；网络压缩；剪枝算法
中图分类号：TP183　　**文献标识码**：A
文章编号：1009-2412(2022)03-0054-09
DOI：10.3969/j.issn.1009-2412.2022.03.008

0　引言

在卷积神经网络运算中，卷积层是运算量最大的层，而且耗费时间。近年来，神经网络参数逐渐增多，大型网络存在数百万甚至上千万个参数，其中部分参数存在冗余。此外，在给定体系结构下，逐层探索每层的最佳滤波器数量，即卷积层宽度也极具挑战。随着移动设备的快速普及，在移动设备上运行卷积神经网络算法具有较好的前景，但移动设备性能目前尚无法跟传统设备相比，卷积神经网络的计算强度阻碍其在移动设备等资源受限设备上的部署。

基于以上原因，为减少计算和降低存储，产业界通常将卷积神经网络进行压缩，即利用数据将已训练好的神经网络进行精简，从而得到轻量化且相对准确的神经网络。现有神经网络模型压缩主要包括5个方向：剪枝、矩阵分解、知识蒸馏、量化和神经网络结构搜索。

本文对基于剪枝算法的卷积神经网络压缩进行全面系统的综述。阐述剪枝算法的国内外研究现状，包括剪枝算法的分类以及各小类的剪枝算法，并对剪枝算法的发展趋势进行展望。

1　剪枝算法及其分类

最早的剪枝算法源于Yann等[1]提出的最优脑损伤（optimal brain damage）算法，其思想是利用基于损失函数的海森（Hessian）矩阵来计算参数权重，然后对权重低的参数进行剪枝。Hassibi等[2]基于最优脑损伤优化算法提出最优脑手术（optimal brain surgeon）算法。二者都基于海森矩阵计算，但最优脑手术算法可以修剪更多的权重参数，因此在同一测试集上具有更好的泛化能力。二者也有相似的缺点，即在每一轮迭代中需要计算并更新所有参数的显著性，复杂度较高。

事实上，复杂度较高是大多数细粒度剪枝算法的常见问题。这里的粒度指神经网络的稀疏结构。

收稿日期：2021-08-23　　修回日期：2022-05-20
项目资助：科技创新2030项目（2020AAA0107500）。
通信作者：欧中洪，副教授，zhonghong.ou@bupt.edu.cn。

细粒度剪枝算法也叫非结构化剪枝算法，其考虑待剪枝层每个滤波器中的每个参数元素，并单独删除每个冗余参数；与之相对应，粗粒度剪枝算法直接考虑删除整个滤波器的结构信息，由于其考虑的是整体的结构化信息，所以粗粒度剪枝算法也叫结构化剪枝算法。

1.1 非结构化剪枝算法

非结构化剪枝算法即判断单个参数是否需要剪枝。最优脑损伤算法和最优脑手术算法都属于非结构化剪枝算法。非结构化剪枝算法对每个参数都考虑筛选，计算量较大；同时，对卷积核中参数的剪枝无法对卷积矩阵计算进行加速优化。因此，非结构化剪枝算法已逐渐被结构化剪枝算法所替代。

1.2 结构化剪枝算法

不同于非结构化剪枝算法，结构化剪枝算法的对象不是单个参数，它可以是一个向量、一个内核，也可以是一个滤波器。滤波器有时指一个卷积核，在一些网络框架中也可以指多个卷积核。滤波器剪枝算法即通过计算直接删除一个或多个卷积核。一次剪掉整个结构化信息是由于网络不同，有的卷积核计算得到的特征图矩阵趋于零矩阵，或存在两个特征图矩阵的数值近似相等，这种情况下删除特征值小和冗余矩阵对整体结果影响不大，但可以很大程度上减少计算量。

由于一次可考虑并删除整个结构化信息，而不用对单个参数进行逐一计算，结构化剪枝算法的复杂度远低于非结构化剪枝算法，所以结构化剪枝算法已成为剪枝算法的研究趋势。近年来通过算法优化，在不损失精度的前提下，结构化剪枝算法也可以实现与非结构化剪枝算法相似的稀疏率。

1.3 剪枝算法分类

结构化剪枝算法与非结构化剪枝算法的区别在于是否会一次性删除整个节点或滤波器。非结构化剪枝算法考虑每个滤波器的每个元素，删除滤波器中元素为 0 的参数信息；而结构化剪枝算法直接考虑删除整个滤波器的结构化信息。剪枝算法的分类[3]如图 1 所示，将结构化剪枝算法分为向量级剪枝算法、内核级剪枝算法、组级剪枝算法和滤波器剪枝算法。向量级剪枝算法将卷积核中的向量作为修剪的结构单位进行剪枝；内核级剪枝算法对滤波器中的二维卷积核进行剪枝；组级剪枝算法根据滤波器上的相同稀疏模式剪枝算法，当多个滤波器拥有相同的稀疏模式时，卷积滤波器可表示为一个细化的稠密矩阵，利用组级剪枝算法，卷积可以通过稠密矩阵乘法实现；滤波器剪枝算法对卷积滤波器或信道进行剪枝。

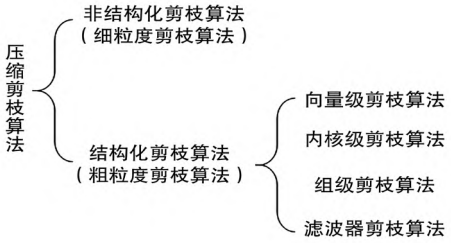

图 1 剪枝算法的分类

结构的稀疏性会影响硬件加速的效率和预测精度。结构化剪枝算法带来更多稀疏规则，同时对结构化信息的剪枝可以跳过卷积计算，该算法更适合硬件加速。但一次性对结构化信息的剪枝使得新模型难以保持与原模型相同的精度，这需要算法层面的优化。

结构化剪枝算法也存在不足。一是在滤波器剪枝算法中，滤波器之间的关系在很大程度上被忽略，但滤波器有时会以协同方式运行进行准确预测，单独考虑每个滤波器的显著性剪枝无法达到最佳剪枝效果；二是普通剪枝算法需要设置裁剪阈值等超参数，需要手动多次设置以达到较好的剪枝效果，无法实现完全自动的学习模式。

2 国内外研究进展

2.1 非结构化剪枝算法的研究进展

表 1 展示的是非结构化剪枝算法代表性工作及其主要贡献。

Yann 等[1]提出的最优脑损伤算法是最早的非结构化剪枝算法，也是最早的剪枝算法。之后的非结构化剪枝算法大多是在最优脑损伤和最优脑手术算法[2]的思想上加以改进的。

Han 等[4]提出一种简单的非结构化剪枝算法，直接根据参数的权值来评估神经元的显著性，决定是否剪枝。Srinivas 等[5]提出直接构造权重的显著性矩阵并进行排序，选择不显著的冗余节点进行剪枝。Hu 等[6]提出将每个滤波器对应的输出特征映射的非零比例作为滤波器的显著性指标。Guo 等[7]提出一种动态的非结构化剪枝算法，其通过恢复错误修剪的参数，来提高网络准确率。

表 1　非结构化剪枝算法代表性工作

分类	研究人员及其文献	主要贡献
非结构化剪枝算法	Yann 等 [1]	提出最优脑损伤算法
	Hassibi 等 [2]	提出最优脑手术算法
	Han 等 [4]	提出基于模型参数值大小修剪的剪枝算法
	Srinivas 等 [5]	提出构建权重的显著性矩阵并排序的剪枝算法
	Hu 等 [6]	提出基于输出特征图的非零比例剪枝算法
	Guo 等 [7]	提出一种动态非结构化剪枝算法
	Ye 等 [8]	提出基于贪心选择子网络的剪枝算法
	Frankle 等 [9]	提出彩票假说
	Malach 等 [10]	优化彩票假说

Ye 等 [8] 提出基于贪心选择的算法以寻求良好的子网络，期望通过贪心方法从空网络开始，在原始网络中将权重高的神经元加入选择后的网络中，最终得到一个比原始网络更小的子网络而不显著降低准确性。该算法与传统的从大型网络中对冗余神经元进行剪枝的思路相反，其从空网络开始添加重要神经元。该算法的缺点是选择子网络后要逐渐微调，以检验选择的子网络是否性能较好。

Frankle 等 [9] 在 2018 年提出一种叫作彩票假说的思想。在一个大型网络内部，如果子网络及其初始化使得训练特别有效，它们一起被称为“中奖彩票”。一个随机初始化、含有“中奖彩票”的密集神经网络在训练最多相同迭代次数后可达到原始网络的测试精度。彩票假说理论可概括为：如果某个参数在神经网络模型中很重要，那么它在开始训练之前就很重要。该研究做了一系列实验来验证假说的正确性，实验结果表明，在 MNIST 和 CIFAR-10 数据集中上找到的“中奖彩票”，其全连接层和卷积前馈架构的大小都只有原始网络的 10% ～ 20%。

2020 年，Malach 等 [10] 首次从数学理论层面证明上述彩票假说。他们认为彩票假说是一个随机初始化的神经网络，并为此提出一个更有利的假说：在探索一个有界分布或具有有界权重的目标网络时，可以探索一个充分过度参数化的随机权值神经网络，其包含一个与目标网络精度大致相同的子网络，而无须进一步训练。该理论为彩票假说的一个增强版本。

虽然早期使用非结构化剪枝算法可以减轻参数冗余的情况，但非结构化剪枝算法还存在以下缺点：①非结构化剪枝算法因为考虑单个神经元对网络的影响，计算量较大。②简单使用非结构化剪枝算法并不能加速稀疏矩阵计算，因为矩阵大小并未改变。③依赖软件和硬件的非结构化剪枝算法无法在所有深度学习框架中使用。因此，近些年的研究倾向于使用结构化剪枝算法。

2.2　结构化剪枝算法的研究进展

如前所述，结构化剪枝算法倾向于对整个滤波器信息进行剪枝。表 2 展示的是除滤波器剪枝算法外其他结构化剪枝算法的代表性工作及其主要贡献。现有结构化剪枝算法的重点在滤波器剪枝算法，其他算法的研究较少。

表 2　结构化剪枝算法（除滤波器剪枝算法）代表性工作

分类		研究人员及其文献	主要贡献
结构化剪枝算法	向量级剪枝算法	Park 等 [11]	提出一种高效的通用稀疏与密集矩阵乘法算法
	内核级剪枝算法	Anwar 等 [12]	提出一种内核跨步剪枝算法
	组级剪枝算法	Wen 等 [13]	提出基于 group lasso（组级最小绝对值收敛和选择器）的损失函数
		Zhou 等 [14]	提出在目标函数中加入结构稀疏化的限制
		Lebedev 等 [15]	提出分组脑损伤算法

向量级剪枝算法比非结构化剪枝算法占用更少的存储空间，因为向量级剪枝算法只需更少的索引来指引剪枝参数。值得一提的是 Lebedev 等 [15] 提出的分组脑损伤算法。该算法在传统深度模型的损失函数上增加结构化稀疏项，利用随机梯度下降学习结构化稀疏损失函数，对小于指定阈值的滤波器进行剪枝。由于组级剪枝算法可利用基本线性代数程序库，其在稀疏级上也能实现线性加速，该算法在 AlexNet 模型中所有卷积层上实现 3.2 倍的加速。

表 3 展示的是滤波器剪枝算法的代表性工作及其主要贡献。滤波器剪枝算法修剪的是滤波器或信道，使深度神经网络更薄、更轻。当一层滤波器被修剪后，与其相连的下一层输入信道也会被相应修剪。

表 3　滤波器剪枝算法代表性工作

分类		研究人员及其文献	主要贡献
经典的“三步走”算法	静态剪枝算法	Hu 等 [6]	提出判断神经元重要性的标准
		Jin 等 [16]	提出新的相位迭代剪枝算法
		Molchanov 等 [17]	提出基于全局搜索显著性过滤的剪枝算法
		Molchanov 等 [18]	对 Molchanov 等 [17] 提出的算法改进，可以在任意网络层上进行一致性扩展
		He 等 [19]	提出两步迭代剪枝算法
		Luo 等 [20]	提出使用下层映射指导当前层剪枝算法
		Luo 等 [21]	提出基于激活响应熵值的剪枝标准
		Yang 等 [22]	提出减少计算能耗的剪枝算法
		Ding 等 [23]	提出近似 Oracle 滤波器剪枝算法
		Lemaire 等 [24]	提出稀疏学习框架剪枝算法
		Chin 等 [25]	提出层补偿剪枝算法
		Li 等 [26]	提出偏序剪枝算法
		Wang 等 [27]	提出基于正则化的剪枝算法
		Wang 等 [28]	提出基于 Kronecker因子特征基的重新参数化算法
		Guo 等 [29]	提出基于分类损失和特征显著性的信道剪枝算法
	动态剪枝算法	Lin 等 [30]	提出生成对抗学习方法剪枝算法
		Ding 等 [31]	提出基于动量随机梯度下降的优化剪枝算法
		Gao 等 [32]	提出基于特征增强和抑制的剪枝算法
		Zhang 等 [33]	提出统一、系统的卷积神经网络结构权重剪枝框架
		He 等 [34]	提出元滤波器剪枝算法
		Wang 等 [35]	提出对抗样本监测剪枝算法
		Lin 等 [36]	提出结构化稀疏正则化滤波器剪枝算法
		Liu 等 [37]	提出自动压缩框架
		Tang 等 [38]	提出考虑所有实例的特征动态修剪信道剪枝算法
其他算法	元学习	Liu 等 [39]	提出基于元学习的信道剪枝算法
	Play and Prune 框架	Singh 等 [40]	提出滤波器剪枝算法的最小–最大框架 play and prune（PP）
	向心随机梯度下降	Ding 等 [41]	提出基于向心随机梯度下降的剪枝算法
	可转移的架构搜索	Dong 等 [42]	提出基于可转移的架构搜索的剪枝算法
	基于变分贝叶斯的信道剪枝算法	Zhao 等 [43]	提出基于变分贝叶斯的剪枝算法
	自动修剪器	Luo 等 [44]	提出自动修剪器（AutoPruner）

续表

分类		研究人员及其文献	主要贡献
其他算法	基于反馈信号	Lin 等 [45]	提出基于反馈信号的动态剪枝算法
	训练前初始化修剪	Lee 等 [46]	提出对指定网络在训练前先初始化修剪的剪枝算法
	简单的随机剪枝算法	Mittal 等 [47]	提出基于简单的随机修剪策略的剪枝算法
	条件自动信道剪枝算法	Liu 等 [48]	提出条件自动信道剪枝（CACP）算法
	从头开始剪枝算法	Liu 等 [49]	重新思考剪枝的价值
		Crowley 等 [50]	对简化网络和剪枝网络进行系统比较
		Wang 等 [51]	提出随机初始化剪枝算法

如表 3 所示，滤波器剪枝算法可进一步细分为经典的“三步走”算法和其他算法。经典的“三步走”算法指的是预训练大型网络、剪枝并进行微调，再迭代多次直至得到预期性能的网络。具体过程如图 2 所示。

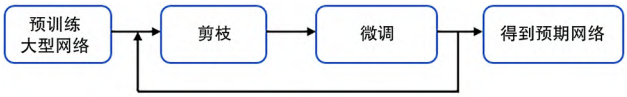

图 2　经典的“三步走”算法流程

经典的“三步走”算法也可进一步细分为静态剪枝算法和动态剪枝算法。静态剪枝算法的研究主要集中在“三步走”中的剪枝步骤，根据权重选择修剪的参数。实验证明，权重剪枝算法可在不损失精度的情况下实现良好的模型剪枝率，从而降低大规模深度神经网络的计算和存储需求。Hu 等 [6] 提出计算每个滤波器对应输出特征图的非零比例的算法，并将其作为判断滤波器剪枝算法的标准，如果一个滤波器总是输出相同值，则意味着该滤波器包含的信息较少，应该进行修剪。Molchanov 等 [17] 在 2016 年提出基于全局搜索滤波器显著性的算法，其基于泰勒展开准则对目标函数进行展开，判断使目标函数变化最小的滤波器为显著滤波器，并对需要剪枝的滤波器用零值代替。2019 年，Molchanov 等 [18] 改进了上述算法，该算法可在任何网络层上进行一致性扩展，而无须对每层进行敏感性分析。Molchanov 等 [17-18] 采用经典的“三步走”算法，在第二步“剪枝”中判断滤波器的显著性，并对显著性低的滤波器进行剪枝。Luo 等 [20] 提出 ThiNet 剪枝算法，其通过卷积计算建立当前层的滤波器与下一层的滤波器输入通道的一对一关系，并利用这一关系来探索下一层卷积核输入通道的显著性，而非直接考虑当前层滤波器参数的显著性。Ding 等 [23] 提出近似 Oracle 滤波器剪枝算法。该算法是一种多路径训练滤波器的框架，以二分搜索方式搜索滤波器，并对模型进行微调。它先采用随机屏蔽滤波器方式尝试剪枝，然后计算并累计下一层输出的变化，以此来搜索显著性最低的滤波器。

静态剪枝算法较容易实现，但忽略了输入图像对参数显著性的影响；动态剪枝算法可根据不同的输入动态确定修剪路径来实现剪枝。Gao 等 [32] 提出了动态剪枝算法，其生成特征图的显著性高度依赖于输入图像。在训练过程中，模型会动态预测下一层卷积信道的显著性，并跳过显著性差的信道，而不像部分静态剪枝算法一样永久删除该信道。这使得不同输入图像可以根据自身特征动态跳过不同的信道。实验结果表明，这种算法在 VGG-16 模型上计算量减少 80%，在 VGG-18 模型上计算量减少 50%。He 等 [35] 提出元滤波器剪枝算法，与多数滤波器剪枝算法忽略滤波器以协同方式进行工作不同，该算法考虑滤波器间的关系，还为滤波器剪枝构建一个元剪枝框架，以便在滤波器分布发生变化时自适应选择合适的剪枝方式。

同时，动态剪枝算法也存在一些问题。一些动态剪枝算法是通过强化学习的方式实现的，这种方式会带来额外的计算成本；另一些动态计算决策使用的是开 / 关剪枝决策或执行路径选择，这种方式是不可微调的，导致梯度下降算法不能使用。如果单纯让剪枝决策依赖于输入图像而不加改进，当对输入图像进行变形时，如马赛克、遮挡等，就可能会对剪枝决策造成影响，导致结果准确率不稳定。

与经典的“三步走”算法对应的其他算法可通过某种方式无须进行迭代微调，或不再预训练大型网络，是一种对“三步走”算法的改进。普通剪枝

算法通常需要手动或基于某些经验标准来设置每层压缩比，然后迭代选择需要剪枝的信道，耗时且需要人工频繁干预。Liu 等[39]提出的元学习剪枝算法先使用随机结构抽样方法训练剪枝网络，并使用元学习预测被剪枝网络的准确性。该算法可在无人为干预的情况下搜索不同约束下的剪枝网络，且搜索时不需要微调。Mittal 等[47]通过实验展现不同结果，其从网络中随机修剪 25% ～ 50% 的滤波器，获得与当时最先进的剪枝算法相同的性能。他们认为剪枝网络的性能不是来自于特定标准的选择，而是深度神经网络的固有可塑性。Liu 等[48]为解决想要得到多个压缩率结果就需重复多次相同实验的问题，提出一种条件自动信道剪枝（CACP）算法，该算法通过一个信道剪枝过程就能生成不同压缩率下的压缩模型。

综上所述，剪枝算法从最早期的最优脑损伤算法发展至今，逐步趋于结构化，考虑问题也更加全面。现如今基于剪枝的神经网络压缩算法可从不同切入点进行研究，其中剪枝的每一步骤均有优化空间。剪枝后的网络模型更小，运行速度更快，所需存储空间更少，更适合在计算能力受限的移动设备上部署。基于剪枝的压缩算法在未来还有很大发展空间。

3 发展趋势和研究展望

剪枝算法当前的研究热点是结构化剪枝算法，而结构化剪枝算法的研究热点是滤波器剪枝算法。本节首先介绍两种新的剪枝算法，其次讨论剪枝算法存在的问题，最后探讨未来的研究方向。

3.1 从头剪枝算法

滤波器剪枝算法的流程是经典的“三步走”算法，即预训练、剪枝和微调。剪枝阶段按照不同标准减掉多余信道，保留重要信道并保持精度。Liu 等[49]认为，在经典的“三步走”算法上进行改进的算法具有两个相同之处：①初始时均需训练一个大型超参数化网络，这是“三步走”的重要环节，因为预训练可为剪枝提供具有更强表达能力和优化能力的高性能模型，可在不影响精度的前提下，在预训练模型中安全删除冗余参数；②剪枝结构及其相关权重被认为是获得最终有效模型的关键，所以在剪枝后只会进行多次微调，而不会重新从头开始训练模型。

Liu 等[49]对多个网络体系结构的剪枝算法进行评估，提出一些新的看法：①对于具有预定义目标的网络体系结构的结构化剪枝算法，直接训练随机初始化的小目标模型可以实现与经典的“三步走”算法获得的模型相同甚至更好的性能，在此情况下，无须从大型模型开始训练，可以直接从头开始训练目标模型；②对于每一层网络修剪不同信道数的结构化剪枝算法，从头开始训练模型可以实现与微调模型相同甚至更好的性能，说明剪枝算法更重要的是寻找一个合适的体系结构，而不仅仅是保留重要权重，即便这个合适的体系结构也需要通过训练大型模型来获得。

除精度高以外，从头开始训练预定义目标模型与其他剪枝算法过程相比，还具有以下优势：①由于模型较小，该算法可以使用较少的 GPU 内存来训练模型，且可能比训练原始的大模型更快；②无须逐层进行微调，无须针对不同的网络体系结构进行定制；③避免调整剪枝过程中涉及的其他超参数。

3.2 三维剪枝算法

Wang 等[52]认为现在的滤波器剪枝算法、层级剪枝算法等，都只是从深度、宽度或分辨率中的一个维度修剪网络，会导致这个维度的规模过分减少。而如果可以从这 3 个维度全面修剪，就不会导致某一个维度过度缺损。所以 Wang 等将待剪枝模型的精度和深度 / 宽度 / 分辨率之间的关系转换为一个多项式回归，并最大化此多项式来获得 3 个维度的最优值。同时，通过迭代剪枝和微调的方法更快地收集数据。实验结果表明，其他剪枝算法会在提升运行速度时造成一定的精度损失，三维剪枝算法在 ResNet-32 架构、CIFAR-10 数据集上比基准精度高 0.09%，在 ResNet-101 架构、TinyImageNet 数据集上比基准的精度高 0.44%。

3.3 剪枝算法存在的问题

当前剪枝算法还存在一些问题，主要表现如下。

（1）剪枝算法的评估标准不直接。检验剪枝算法性能好坏常采用间接度量，如使用 FLOPS（每秒执行的浮点运算次数）来评估网络复杂度，但浮点运算数并不能真正反映实际的推理速度，即单位时间内网络能处理的图像数量。此外，影响推理速度的另一个重要指标——内存访问的优劣，也不能由浮点运算数来表示[26]。

（2）基于各维度显著性剪枝的算法存在一些通用问题。上文介绍的研究多计算参数的显著性，但参数的显著性可能随时间变化，同一信道对不同图

像也会产生不同映射。卷积层的神经元被设计用来识别不同图像中的特征映射，神经元的相对显著性会受输入图像的影响[32]，滤波器剪枝算法中的动态剪枝算法即考虑到这一点而加以改进。

3.4 未来的研究方向

剪枝算法具有广阔的发展前景，未来的发展方向主要分为两种：一是研究剪枝算法现有的挑战性问题，如参数的显著性非静态，可能会根据输入数据而改变；二是提出性能更好的算法，如利用机器学习方法取代人为调整超参数。总体而言，研究剪枝算法的切入点是在如何提高模型压缩性能的前提下，防止模型精度下降过大。

对于滤波器剪枝算法，切入点可选择研究在经典的“三步走”算法中，如何使用复杂度更低的算法准确计算滤波器的显著性。该切入点不仅适用于滤波器剪枝算法，也适用于其他结构化剪枝算法和非结构化剪枝算法，即利用算法计算神经元、卷积核中向量或卷积核的显著性。

如果跳出“三步走”算法的固有思维，滤波器剪枝算法的切入点可放在如何消除迭代剪枝或如何避免进行大型网络预训练上。该方向已有相关研究，但数量较少，属于当前较新的研究方向。消除迭代剪枝指研究能否对某卷积层只需剪枝一遍即能得到期望的性能，而无须连续迭代微调结果，从头剪枝算法即为该方向的研究。

另一个有前景的研究方向是研究不依赖人为设计超参数的算法，目前该方向的相关研究较少，但是非常重要的一个方向。人为调整超参数或人为根据不同卷积层结构调整参数是一项耗时耗力的工作。如果不依赖人工，如采用模拟退火算法或强化学习自动探索适合每一卷积层剪枝的超参数，将会大幅降低模型压缩成本。

参考文献

[1] YANN L C, DENKER J S, SOLLA S A. Optimal brain damage[J]. Advances in neural information processing systems, 1990, 2(279): 598–605.

[2] HASSIBI B, STORK D G. Second order derivatives for network pruning: optimal brain surgeon[J]. Advances in neural information processing systems, 1993, 5: 164–171.

[3] MAO H, HAN S, POOL J, et al. Exploring the regularity of sparse structure in convolutional neural networks[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1705.08922.

[4] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural network[J]. Advances in neural information processing systems, 2015, 28: 1135–1143.

[5] SRINIVAS S, BABU R V. Data-free parameter pruning for deep neural networks[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1507.06149.

[6] HU H, PENG R, TAI Y W, et al. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1607.03250.

[7] GUO Y, YAO A, CHEN Y. Dynamic network surgery for efficient DNNs[J]. Advances in neural information processing systems, 2016, 30: 1387–1395.

[8] YE M, GONG C, NIE L, et al. Good subnetworks provably exist: pruning via greedy forward selection[C]//Proceedings of the 37th International Conference on Machine Learning. Princeton: IMLS, 2020: 10820–10830.

[9] FRANKLE J, CARBIN M. The lottery ticket hypothesis: finding sparse, trainable neural networks[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1803.03635.

[10] MALACH E, YEHUDAI G, SHALEV-SCHWARTZ S, et al. Proving the lottery ticket hypothesis: pruning is all you need[C]. Princeton: IMLS, 2020: 6682–6691.

[11] PARK J, LI S, WEN W, et al. Faster CNNs with direct sparse convolutions and guided pruning[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1608.01409.

[12] ANWAR S, HWANG K, SUNG W. Structured pruning of deep convolutional neural networks[J]. ACM journal on emerging technologies in computing systems, 2017, 13(3): 1–18.

[13] WEN W, WU C, WANG Y, et al. Learning structured sparsity in deep neural networks[J]. Advances in neural information processing systems, 2016, 30: 2082–2090.

[14] ZHOU H, ALVAREZ J M, PORIKLI F. Less is more: towards compact CNNs[C]//European Conference on Computer Vision. Berlin: Springer, 2016: 662–677.

[15] LEBEDEV V, LEMPITSKY V. Fast ConvNets using group-wise brain damage[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016: 2554–2564.

[16] JIN X, YUAN X, FENG J, et al. Training skinny deep neural networks with iterative hard thresholding methods[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1607.05423.

[17] MOLCHANOV P, TYREE S, KARRAS T, et al. Pruning convolutional neural networks for resource efficient inference[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1611.06440.

[18] MOLCHANOV P, MALLYA A, TYREE S, et al. Importance estimation for neural network pruning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Lang Beach: IEEE Computer Society, 2019: 11264–11272.

[19] HE Y, ZHANG X, SUN J. Channel pruning for accelerating very deep neural networks[C]//Proceedings of the IEEE International Conference on Computer Vision (ICCV). Venice: IEEE Computer Society, 2017: 1389–1397.

[20] LUO J H, WU J, LIN W. Thinet: a filter level pruning method for deep neural network compression[C]//Proceedings of the IEEE International Conference on Computer Vision (ICCV). Los Alamitos: IEEE Computer Society, 2017: 5068–5076.

[21] LUO J H, WU J. An entropy-based pruning method for CNN compression [EB/OL]. [2021–08–20]. https://arxiv.org/abs/1706.05791.

[22] YANG T J, CHEN Y H, SZE V. Designing energy-efficient convolutional neural networks using energy-aware pruning[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017: 5687–5695.

[23] DING X, DING G, GUO Y, et al. Approximated Oracle filter pruning for destructive CNN width optimization[C]//International Conference on Machine Learning. Princeton: IMLS, 2019: 1607–1616.

[24] LEMAIRE C, ACHKAR A, JODOIN P M. Structured pruning of neural networks with budget-aware regularization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos: IEEE, 2019: 9108–9116.

[25] CHIN T W, ZHANG C, MARCULESCU D. Layer-compensated pruning for resource-constrained convolutional neural networks[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1810.00518.

[26] LI X, ZHOU Y, PAN Z, et al. Partial order pruning: for best speed/accuracy trade-off in neural architecture search[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach: IEEE, 2019: 9145–9153.

[27] WANG H, ZHANG Q, WANG Y, et al. Structured pruning for efficient convnets via incremental regularization[C]//Proceedings of the International Joint Conference on Neural Networks (IJCNN). Budapest: IEEE, 2019: 1–8.

[28] WANG C, GROSSE R, FIDLER S, et al. Eigendamage: Structured pruning in the Kronecker-factored eigenbasis[C]//International Conference on Machine Learning. Princeton: IMLS, 2019: 6566–6575.

[29] GUO J, OUYANG W, XU D. Channel pruning guided by classification loss and feature importance[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2020.

[30] LIN S, JI R, YAN C, et al. Towards optimal structured CNN pruning via generative adversarial learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach: IEEE, 2019: 2790–2799.

[31] DING X, DING G, ZHOU X, et al. Global sparse momentum SGD for pruning very deep neural networks[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc. 2019: 6382–6394.

[32] GAO X, ZHAO Y, DUDZIAK Ł, et al. Dynamic channel pruning: feature boosting and suppression[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1810.05331.

[33] ZHANG T, ZHANG K, YE S, et al. StructADMM: a systematic, high-efficiency framework of structured weight pruning for DNNs[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1807.11091.

[34] HE Y, LIU P, ZHU L, et al. Filter pruning by switching to neighboring CNNs with good attributes [EB/OL]. [2022–05–20]. http://arxiv.org/abs/1904.03961v2.

[35] WANG Y, ZHANG X, HU X, et al. Dynamic network pruning with interpretable layerwise channel selection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2020.

[36] LIN S, JI R, LI Y, et al. Toward compact convnets via structure-sparsity regularized filter pruning[J]. IEEE transactions on neural networks and learning systems, 2019, 31(2): 574–588.

[37] LIU N, MA X, XU Z, et al. Autocompress: an automatic DNN structured pruning framework for ultra-high compression rates[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2020.

[38] TANG Y, WANG Y, XU Y, et al. Manifold regularized dynamic network pruning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville: IEEE Computer Society, 2021: 5018–5028.

[39] LIU Z, MU H, ZHANG X, et al. Metapruning: Meta learning for automatic neural network channel pruning[C]//IEEE/CVF International Conference on Computer Vision(ICCV), Seoul: IEEE Computer Society, 2019: 3296–3305.

[40] SINGH P, VERMA V K, RAI P, et al. Play and prune: adaptive filter pruning for deep model compression[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1905.04446.

[41] DING X, DING G, GUO Y, et al. Centripetal SGD for pruning very deep convolutional networks with complicated structure[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach: IEEE Computer Society, 2019: 4938–4948.

[42] DONG X, YANG Y. Network pruning via transformable architecture search[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc. 2019: 760–771.

[43] ZHAO C, NI B, ZHANG J, et al. Variational convolutional neural network pruning[C]//Proceedings of the IEEE/CVF Conference on

Computer Vision and Pattern Recognition (CVPR). Long Beach: IEEE, 2019: 2780–2789.

[44] LUO J H, WU J. AutoPruner: an end-to-end trainable filter pruning method for efficient deep model inference[J]. Pattern recognition, 2020, 107: 107461.

[45] LIN T, STICH S U, BARBA L, et al. Dynamic model pruning with feedback[EB/OL]. [2021–08–20]. https://arxiv.org/abs/2006.07253.

[46] LEE N, AJANTHAN T, TORR P H S. SNIP: single-shot network pruning based on connection sensitivity[EB/OL]. [2019–02–23]. https://arxiv.org/abs/1810.02340.

[47] MITTAL D, BHARDWAJ S, KHAPRA M M, et al. Recovering from random pruning: on the plasticity of deep convolutional neural networks[C]//2018 IEEE Winter Conference on Applications of Computer Vision (WACV). Lake Tahoe: IEEE, 2018: 848–857.

[48] LIU Y, GUO Y, GUO J, et al. Conditional automated channel pruning for deep neural networks[J]. IEEE signal processing letters, 2021, 28: 1275–1279.

[49] LIU Z, SUN M, ZHOU T, et al. Rethinking the value of network pruning[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1810.05270.

[50] CROWLEY E J, TURNER J, STORKEY A, et al. A closer look at structured pruning for neural network compression[EB/OL]. [2021–08–20]. https://arxiv.org/abs/1810.04622.

[51] WANG Y, ZHANG X, XIE L, et al. Pruning from scratch[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2020.

[52] WANG W, CHEN M, ZHAO S, et al. Accelerate CNNs from three dimensions: a comprehensive pruning framework[C]//International Conference on Machine Learning. Princeton: IMLS, 2021: 10717–10726.

A Comprehensive Survey of Pruning-Based Techniques for Convolutional Neural Networks Compression

WANG Zhaofengnian, OU Zhonghong, SONG Meina

School of Computer Science (National Pilot Software School), Beijing University of Posts and Telecommunications, Beijing 100876

Abstract: In recent years, with the rapid development of deep learning technology, convolutional neural network (CNN) has become more and more popular, and its application range has become more and more extensive. In order to improve the accuracy of the model, CNN gradually increased the depth and width of the network, but it led to the gradual expansion of the network structure. Existing deep neural networks usually have tens of millions of parameters, which require a lot of computing and storage capabilities. In addition, it is also a big challenge to deploy deep neural networks on mobile devices with limited computing power (such as smartphones or tablets). In these cases, a method is needed to improve the efficiency without affecting the accuracy.

Neural network compression is an effective method to reduce the network size and improve storage efficiency without affecting the accuracy. This kind of method has been widely concerned and applied in recent years. The neural network compression includes many branches. This paper conducts a comprehensive survey on the pruning network algorithm of neural network compression. In particular, this paper provides more detailed sub-categories to the existing taxonomy of the pruning algorithm to make it more systematic. And it classifies the investigated researches, and then clearly presents the main features of each research. Finally, it considers the future development trend of the pruning algorithm. According to the characteristics of the pruning algorithm and the existing shortcomings of the algorithm, this paper explains the entry point for further work.

Keywords: deep learning; convolutional neural network (CNN); network compression; pruning network