正则化机制下多粒度神经网络剪枝方法研究

刘 奇,陈 莹
(江南大学轻工过程先进控制教育部重点实验室,江苏无锡 214122)

摘 要: 目前流行的模型压缩剪枝算法裁减的对象通常是整个卷积核. 一些网络结构中存在特征图维度匹配的硬性要求,如ResNet中的残差结构主干上最后一个卷积层的卷积核个数以及Inception网络中的级联操作前所有分支上最后一个卷积层的卷积核个数都不能改变,这直接限定了剪枝的空间. 本文提出一种正则化机制下的多粒度神经网络剪枝方法,针对维度匹配限制了剪枝空间的问题,设计从粗到细的多粒度剪枝策略,在稀疏化的同时维持了处于维度匹配位置的卷积层中卷积核的数量不变. 并且,本文提出一种自适应L1正则化的稀疏方式,可以使网络在更新参数的同时兼顾到网络结构的变化. 稀疏化后的卷积核不仅有比原卷积核更少的参数和计算量,而且拥有更加优异的结构性质,使网络具有更高的表达能力. 例如,在CIFAR-10上,针对VGG-16,相比基准网络,在计算量压缩了76.73%的情况下,准确率提高了0.19%;针对ResNet-56,在计算量压缩了82.54%的情况下,准确率只下降了0.14%. 在ImageNet上,针对ResNet-50,在计算量压缩了56.95%的情况下,准确率只下降了0.48%. 本文方法优于现有先进的剪枝方法.

关键词: 卷积神经网络;正则化;剪枝;维度匹配;自适应L1正则化
基金项目: 国家自然科学基金(No.62173160)
中图分类号: TP391.41 **文献标识码:** A **文章编号:** 0372-2112(2023)08-2202-11
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210844

Research on Multi-Granularity Neural Network Pruning Method with Regularization Mechanism

LIU Qi, CHEN Ying
(*The Key Laboratory of Advanced Process Control for Light Industry* (*Ministry of Education*), *Jiangnan University*, *Wuxi*, *Jiangsu* 214122, *China*)

Abstract: At present, the object of pruning algorithm is usually the whole convolution kernel. The rigid requirement of feature graph dimension matching in some network structures, e. g. the number of the last convolution kernel on the backbone of residual structure in ResNet and the number of convolution kernel of all branches before concatenation operation in Inception network cannot be changed, directly limits the pruning space. To solve the problem of dimensional matching that limits the pruning space, a multi-granularity pruning strategy from coarse to fine is designed to maintain dimensional matching, which keeps the number of convolution kernels in the convolution layers positioning for dimensional matching unchanged while increasing the sparsity of the neural network. Moreover, an adaptive L1 regularization sparse method is presented, which enables the network update parameters while taking into account the changes in the network structure. The sparse convolution kernel not only has fewer parameters and calculations than the original convolution kernel, but also has more excellent structural properties, which enables the network better ability for feature representation. For VGG-16 on CIFAR-10, the accuracy is increased by 0.19% when the calculation amount is compressed by 76.73% compared with the baseline network; for ResNet-56, the accuracy rate is reduced by only 0.14% when the calculation amount is compressed by 82.54%. For ResNet-50 on ImageNet, when the calculation amount is compressed by 56.95%, the accuracy rate is only reduced by 0.48%. So the proposed method is better than the existing advanced pruning methods.

Key words: convolutional neural network; regularization; prune; dimensionality matching; adaptive L1 regularization
Foundation Item(s): National Natural Science Foundation of China (No.62173160)

收稿日期:2021-07-05;修回日期:2021-12-10;责任编辑:王天慧

1 引言

近年来,深度学习在计算机视觉领域发展迅速,在图片分类、目标检测等领域取得了突出的成就[1,2]. 但是卷积神经网络高计算量和参数量的特点限制了其在各种硬件平台和边缘设备上的部署. 为了解决此类问题,模型压缩应运而生. 目前,深度学习模型压缩方向主要分为模型剪枝[3]、量化[4]、蒸馏[5]以及轻量级网络设计[6].

Dai等[7]使用知识蒸馏的方法在目标检测领域进行模型压缩,将具有良好特征提取能力的大型深度学习模型设定为教师,引导小型网络性能不断逼近该教师网络,从而将大型网络的性能迁移到小网络上. Tai等[8]提出一种低阶张量分解的新算法,用于消除卷积核中的冗余. Dettmers[9]开发并测试8bit近似算法,将32bit的梯度和激活值压缩到8bit,通过GPU集群测试模型和数据的并行化性能,取得了两倍的数据传输加速.

神经网络剪枝分为结构化剪枝和非结构化剪枝. 非结构化剪枝是裁剪网络中卷积核的单个权重. 例如,当权重值趋近0时即置0,可以在网络不牺牲性能的前提下产生稀疏网络. 但是,因为卷积核中非0权重的位置是不规则且随机的,裁剪时要额外记录权重的位置信息,增加处理器计算负担. 而且稀疏网络无法以结构化的方式呈现,使非结构化剪枝的方法无法在通用处理器上实现加速. 非结构化剪枝的方法有很多,Han等[10]提出基于L1范数准则修剪网络权重并微调恢复性能的方法. Liu等[11]提出一种频域动态修剪方案,利用CNN中的空间相关性,在每次迭代中动态修剪频域系数. 结构化剪枝对象是整个过滤器、通道,或者整个层. 因为卷积核的结构没有改变,所以无需专门的硬件或者深度学习库来支持. Lin等[12]通过蜂群算法自动寻找最优网络子结构,再通过重训练网络恢复因裁剪失去的精度. Li等[13]通过L1范数准则对过滤器进行排序,剪掉排名在后的部分,实现了网络的压缩. Liu等[14]通过对BN层的比例系数γ施加L1正则,最后将比例系数γ值较小部分对应的通道移除. Kang等[15]除了考虑BN层γ系数,还考虑了Relu层非线性化的影响,实现了更精准的通道移除. Zhuang等[16]针对BN层的γ系数,提出了一种极正则化的方法,能有区分地将不重要通道的γ系数置0,同时保留重要的通道. Meng等[17]提出一种直接在卷积核内部进行裁剪的算法,该方法可以在保持卷积核数量不变的前提下,使卷积核稀疏化.

结构化剪枝的剪枝对象一般是整个通道或者整个卷积核. 一些网络结构中存在特征图维度匹配的硬性要求,如ResNet中的残差结构主干上最后一个卷积层的卷积核个数以及Inception网络中的级联操作前所有分支上最后一个卷积层的卷积核个数都不能改变,这直接限定了结构化剪枝的空间.

Zhuang等[16]的方法为结构化粗粒度剪枝,维度匹配问题限制了剪枝空间. Meng等[17]的方法直接使用细粒度的剪枝,没有使神经网络在通道上得到充分的裁剪,在稀疏网络时,使用的L1正则化方法没有使各个网络层间实现均匀裁剪,影响了网络表达能力.

针对上述问题,本文提出一种正则化机制下的多粒度的剪枝方案. 以Zhuang等[16]的方法为基础,融入了改进的Meng等[17]的方法,在BN层上对网络进行一次基于通道的粗剪枝,针对维度匹配限制了剪枝空间的问题,将剩余卷积核再进行一次基于自适应L1正则化的细剪枝,在稀疏化的同时使处于维度匹配位置的卷积层中卷积核的数量保持不变[17]. 由于自适应L1正则化的有效性,稀疏化后的卷积层不仅有比原卷积层更少的参数和计算量,而且拥有更加优异的结构性质,使网络在低计算量低参数量的前提下具有更高的表达能力. 另外,本文训练网络无须微调,采用了边训练边裁剪的方式,让网络有足够时间调整结构和参数,在弥补网络因裁剪带来的精度损失的同时,大大减少了训练时间.

本文主要贡献为:(1)融合了粗细两种剪枝方法,发挥出两种剪枝方法的优势,并避免了各自存在的缺陷;(2)提出自适应L1正则化的细剪枝方法,使网络被裁剪得更加均匀,拥有更加优异的结构性质,具有更高的表达能力;(3)采取边训练边剪枝的方式,无须微调操作,大大节省GPU训练时间.

2 网络剪枝总体方案

在神经网络稀疏化的任务中,通过向损失函数添加待稀疏对象的L1范数,可以在实现稀疏化的同时提取到更重要的特征[15~17]. 本文的多粒度剪枝用到的两种稀疏方法都基于L1正则的演变. 在基于通道的粗剪枝过程中,参照Zhuang等[16]的方法,在对BN层中缩放因子γ施加L1正则项的同时引入了极正则项. 该极正则化项能在通道稀疏化的过程中,撤销对重要通道的稀疏,具体过程见第2.1节. 在基于卷积核的细剪枝过程中,受到上述极正则项的启发,本文提出一种自适应L1正则化方法,可以使稀疏化后的卷积层不仅有比原卷积层更少的参数和计算量,而且拥有更加优异的结构性质,使网络在低计算量低参数量的前提下具有更高的表达能力,具体过程见第2.2节.

图1为网络剪枝总体方案图. 网络在训练的过程中,同时进行粗粒度和细粒度的参数正则化稀疏,一旦参数小于一定的阈值,则将与参数相关的通道及卷积核置0. 并在训练结束后,将参数置0的结构进行剪枝,

从而得到压缩后的网络. 为了方便理解,后文将对粗剪枝和细剪枝过程分别进行描述.

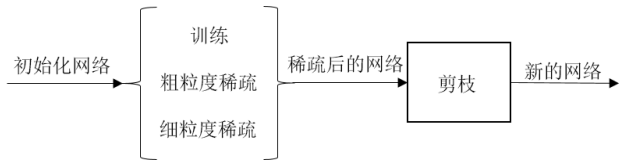


图1 网络剪枝总体方案

2.1 基于极正则化的粗剪枝策略

基于通道的剪枝是一种流行的模型压缩结构化剪枝方式. 其中,关键步骤是选取并判别通道的重要性. Liu 等[14]通过对 BN 层的比例系数 γ 施加 L1 正则,最后将比例系数 γ 值较小部分对应的通道移除. Zhuang 等[16]针对 BN 层的 γ 系数,提出了一种极正则化的方法. 该极正则化方法能更好地判别通道重要性,在网络的训练过程中,能有区分地将不重要通道的 γ 系数向 0 值更新迭代,同时保留重要的通道.

2.1.1 BN层

在模型剪枝的工作中,使用稀疏化尺度因子进行通道选择是一种流行的方式. 此方法十分灵活,且适用于任何卷积神经网络[16]. 在神经网络中,特征图经过卷积后会经过批标准处理. BN 层对特征图的处理过程可以表示为

$$z_{\text{out}}=\gamma\cdot\frac{z_{\text{in}}-\bar{z}}{\sqrt{\sigma^2+\varepsilon}}+b \tag{1}$$

其中,z_{in} 是输入数据,$\bar{z}$ 和 σ^2 是输入数据的平均值和方差,ε 是值趋近于 0 的正常数,避免分母为 0,γ 和 b 则为 BN 层中的缩放因子和偏置. 在压缩网络时,给每个通道的 γ 缩放因子施加稀疏处理,可以使重要性较低的通道的缩放因子 γ 趋近于 0,裁剪掉这些不重要的通道可以大大压缩模型[16].

2.1.2 极正则化

以 n 个特征图的缩放因子举例. 令向量 $\gamma=(\gamma_1,\gamma_2,\cdots,\gamma_i,\cdots,\gamma_n)$,其中 γ_i 代表第 i 个特征图的缩放因子,令 $\bar{\gamma}$ 为 n 个特征图缩放因子的平均值为

$$\bar{\gamma}=\frac{1}{n}\cdot\sum_{i=1}^{n}\gamma_i \tag{2}$$

定义极正则化公式为

$$R(\gamma)=t\|\gamma\|_1-\|\gamma-\bar{\gamma}\|_1=\sum_{i=1}^{n}t|\gamma_i|-|\gamma_i-\bar{\gamma}| \tag{3}$$

最小化极正则项时,L1 正则项 $\|\gamma\|_1$ 会使所有的缩放因子 γ 向 0 稀疏,对重要的特征层和不重要的特征层没有足够的区分力度. 引入正则项-$\|\gamma-\bar{\gamma}\|_1$ 的目的是使 γ_i 尽可能地远离平均值[16]. 在训练网络时,重要的特征图缩放因子会有逐渐大于平均值的趋势,从而得以保留,而不重要的特征图缩放因子则会有小于平均值且不断趋向 0 的趋势,从而会被裁剪. 超参数 t 控制正则项惩罚力度,t 值越大;正则项惩罚力度越大,网络稀疏程度越大.

2.1.3 粗剪枝过程

模型进行稀疏化训练时,给神经网络每个 BN 层的 γ 参数施加极正则化,训练过程中缩放因子 γ 的值逐渐趋于稀疏[16]. 图 2 为神经网络第 l 层特征图共 n_l 个通道的剪枝示意图. 引入全局阈值 δ_1,训练过程中某时刻第 i 层特征图稀疏缩放因子 γ 值若小于阈值 δ_1,则直接将该通道移除,即移除与该通道连接的所有输入输出卷积核[16],从而得到稀疏后的网络. 图中黄色的 γ 值小于阈值 δ_1,故对应的黄色通道图被剪掉,蓝色的 γ 值大于阈值 δ_1,对应的蓝色通道图被保留,一次粗剪枝过后,共保留 n_l' 个通道.

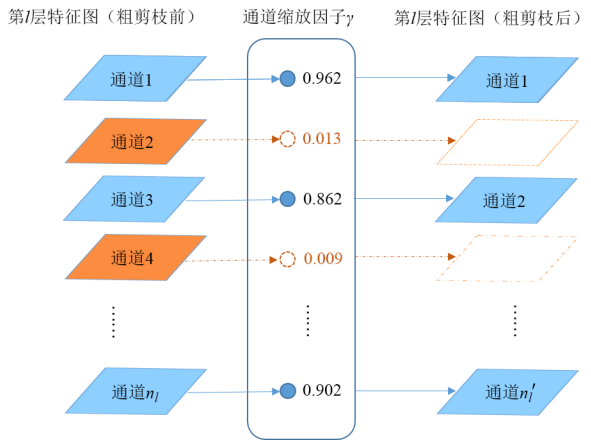


图2 粗剪枝示意图

2.2 基于自适应L1正则化的细剪枝策略

通道剪枝减去的是整个过滤器或者整个卷积核,是基于网络维度匹配规则[18]的. 本文对网络进行一次基于通道的粗剪枝后,针对维度匹配限制了剪枝空间的问题,给剩余卷积核加上一层掩膜结构,用掩膜稀疏的方法再进行一次细剪枝,可以使处于维度匹配位置的卷积层在稀疏化的同时卷积核数量不变. 文献[16]中提出的极正则化方法仅仅适用于基于缩放因子 γ 的通道剪枝,在掩膜稀疏任务上没有取得良好效果(具体见第 3.5 节). 受到极正则化的启发,本文提出一种自适应 L1 正则化的掩膜稀疏方法,可以使稀疏化后的卷积层不仅有比原卷积层更少的参数和计算量,而且拥有更加优异的结构性质,使网络在低计算量低参数量的前提下具有更高的表达能力.

2.2.1 过滤器掩膜

假设神经网络第个 l 卷积层的权重 $W^l\in R^{N\times C\times K\times K}$,$N$ 代表该层卷积核的个数,C 代表通道数,K 代表卷积核

尺寸. 本文将每个维度为$C\times K\times K$的卷积核从高度和宽度方向展开,得到$K\times K$个维度为$C\times 1\times 1$的小卷积核,给每个小卷积核乘上初始值为1的掩膜$M^{l}_{n,i,j}$,其中n为第个l卷积层中N个卷积核的索引,i和j分别为第n个卷积核的高度索引和宽度索引.

2.2.2　基于掩膜的细剪枝

本文对掩膜$M^{l}_{n,i,j}$进行稀疏化,以至进一步地裁剪网络. 当某个$M^{l}_{n,i,j}$值趋于0时,对应的$R^{C\times 1\times 1}$小卷积核可以被移除. 基于L1正则化的掩膜稀疏方法得到了广泛使用并取得了有效成果[14,19],但L1正则化在训练过程中会对每个掩膜实施同等的稀疏力度,往往会出现一部分卷积层被裁剪得多,另外一部分被裁剪得少的情况,没有考虑网络的结构和卷积核的个数对网络表达能力的影响. 文献[12,20]提出,除了网络参数,卷积层卷积核的结构和个数也会影响网络的精确度,对每个卷积层过多的裁剪会导致网络精确度骤降. 为了改善这种情况,需要在训练网络时监督网络每个卷积层卷积核的数目变化,一旦某层卷积核数量低于一定比例,衰减L1正则稀疏因子,并且卷积核数量越少,稀疏因子应该衰减得越剧烈. 据此,本文提出一种自适应L1正则化方法. 掩膜的自适应L1正则化公式为

$$\mathrm{g}(M)=\sum_{l=1}^{L}\rho_l \mathrm{g}(M^{l})=\sum_{l=1}^{L}\rho_l\left(\sum_{n=1}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K}|M^{l}_{n,i,j}|\right) \tag{4}$$

$$\rho_l=\begin{cases}1 & \dfrac{n_{l1}}{n_{l0}}\geqslant\delta_3\\ \dfrac{n_{l1}}{n_{l0}} & \dfrac{n_{l1}}{n_{l0}}<\delta_3\end{cases} \tag{5}$$

其中,$\mathrm{g}(M)$为网络所有掩膜的L1范数,L为网络总层数,ρ_l为第l个卷积层掩膜的缩放因子,n_{l0}为第l个卷积层剪枝前小卷积核个数,n_{l1}为第l个卷积层经过粗细剪枝后小卷积核个数,δ_3为设定的阈值($0<\delta_3<100\%$).

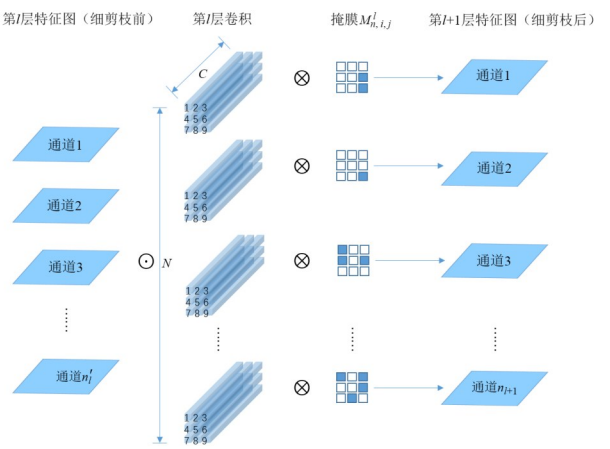
图3为细剪枝过程示意图. 以第l层特征图中共n'_l个特征图与N个3×3的卷积核卷积得到n_{l+1}个特征图为例. 图3(a)为稀疏化的掩膜示意图. 在剪枝过程中,设定全局阈值δ_2,当掩膜某元素的值小于阈值时,其对应的小卷积核的权重将被置0且不再更新,并在训练结束去移除. 图中白色掩膜代表的是待移除的部分,蓝色掩膜代表被保留的部分. 图3(b)描述了细剪枝以及剪枝过后的前向传播过程. 由于在大卷积核中裁剪掉小的卷积核破坏了大卷积核的原本结构,本文采取求和顺序不一样的卷积操作:每个小卷积核和输入通道进行独立卷积,得到对应的小特征图,再将这些小特征图相加得到最终的通道特征图. 标准卷积过程可描述为

$$X^{l+1}_{n,h,w}=\sum_{c}^{C}\sum_{i}^{K}\sum_{j}^{K}W^{l}_{n,c,i,j}\times X^{l}_{n,h-i+\frac{K+1}{2},w-j+\frac{K+1}{2}} \tag{6}$$

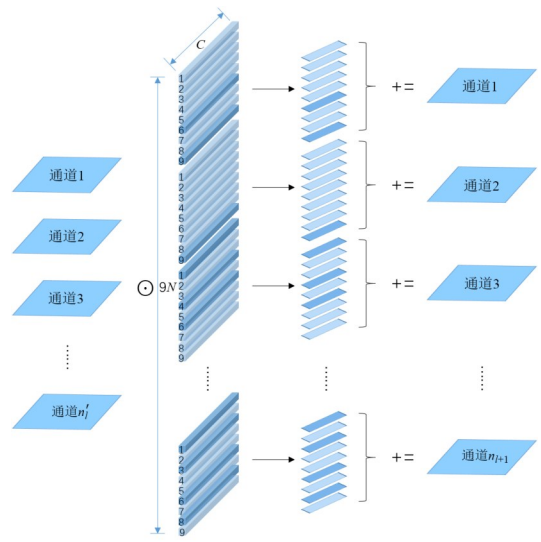
细剪枝卷积过程为

$$X^{l+1}_{n,h,w}=\sum_{i}^{K}\sum_{i}^{K}\left(\sum_{c}^{C}W^{l}_{n,c,i,j}\times X^{l}_{n,h-i+\frac{K+1}{2},w-j+\frac{K+1}{2}}\right) \tag{7}$$

其中,$X^{l+1}_{n,h,w}$为第$l+1$层特征图上的一点,细剪枝卷积仅仅改变了原始卷积方法的卷积顺序,没有产生额外的计算量. 图中,浅蓝色的卷积核是待移除的部分,深蓝色卷积核则是被保留部分.



(a) 稀疏化后的掩膜示意图



(b) 细剪枝及剪枝后前向传播过程图

图3　细剪枝过程示意图

2.3　网络损失函数

经过粗剪枝和细剪枝后网络损失函数为

$$\mathrm{loss}_{\mathrm{pruned}}=\mathrm{loss}_{\mathrm{orig}}+\alpha R(\gamma)+\beta \mathrm{g}(\mathrm{M}) \tag{8}$$

其中,$\mathrm{loss}_{\mathrm{pruned}}$为网络总损失函数,$\mathrm{loss}_{\mathrm{orig}}$为原网络交叉熵损失函数,$\alpha$为极正则化惩罚系数,$\beta$为自适应L1正则项惩罚系数.

3 实验与分析

3.1 数据集和模型选取

本文运用基于自适应L1正则化的粗细剪枝方法，在3个数据集（CIFAR-10，CIFAR-100，ImageNet）和4个主流网络（VGG-16，ResNet-50，ResNet-56，ResNet-110，GoogLenet）中进行了实验.

CIFAR-10数据集由10个类的60 000个32×32彩色图像组成，每个类有6 000个图像. 有50 000个训练图像和10 000个测试图像. CIFAR-100数据集[21]有100个类，每个类包含600个图像即500个训练图像和100个测试图像. ImageNet[22]数据集有1000个类，训练集里有128万张图片，测试集里有5万张图片.

对于VGG-16，本文对每次卷积输出的特征图做一次粗剪枝，再对每个卷积层做一次细剪枝. 全连接层的输入则取决于最后一层特征图的维度数量.

对于ResNet-50中的残差结构，为了维持维度匹配，对主干网络上第一个卷积层只采取粗剪枝，对主干网络上第二个卷积层采取粗细剪枝，对第三个卷积层不予剪枝处理.

在ResNet-56和ResNet-110网络中，对于残差结构，为了维持维度匹配，仅仅对主干网络上第一个卷积层输出的特征图采取粗剪枝，对主干上所有卷积层都采取细剪枝.

对于GoogLenet中的Inception结构，如图4所示，第1和第4分支上的1×1卷积，本文不做剪枝处理. 在第2和第3分支中，对2.a和3.a只做粗剪枝处理，对2.b和3.c只做细剪枝处理，对3.b则做粗细剪枝处理.

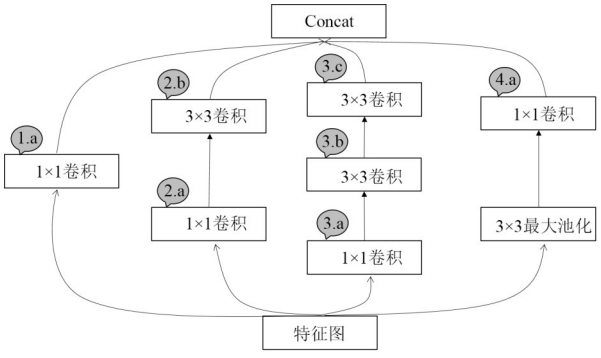


图4 Inception结构剪枝示意图

3.2 实验设置

实验环境为Pytorch-1.7.0框架，Python3.6.9，NVIDIA GeForce RTX 3090 24GB显卡.

模型训练方式与文献[16]相似. 对于CIFAR-10和CIFAR100，文献[16]训练200个epoch后再微调200个epoch. 对于ImageNet，文献[16]训练了120个epoch后再微调了128个epoch，本文在训练模型后精度已经收敛，故不需要微调过程，节省了大量训练时间.

对于CIFAR-10和CIFAR100，本文训练epoch次数设置为300，初始学习率设置为0.1，并在epoch次数为150和225时分别衰减10倍. 对于ImageNet，本文训练epoch次数为160，初始学习率设置为0.1，并在epoch次数为40，80，120时分别衰减10倍. 为了控制网络超参数的数量，本文经验地设定粗剪枝裁剪阈值以及超参数值t分别为δ_1=0.1，t=1.5，细剪枝的裁剪阈值δ_2=0.1，L1自适应正则化中δ_3=10%. 本文可调超参数为α和β，用于调整网络压缩率和精确度的平衡. 其中，α控制粗剪枝的力度，β控制细剪枝的力度. 根据经验，α的值控制在1×10^{-5}到5×10^{-5}之间，β值控制在1×10^{-6}~5×10^{-5}可使网络达到理想的压缩率和精确度平衡.

本文的方法命名为MGP-X. 其中，X代表模型计算量压缩到X%左右时的实验结果. 文中有3个模型压缩评价指标. 精确度（Accuracy）代表模型预测能力，参数量（Parameters）代表模型的大小，计算量（FLOPs，即浮点运算量）表示模型推理一张图片所需的计算量，用来衡量网络前向推理的速度，PR（Pruning Rate）代表压缩率.

3.3 CIFAR-10实验结果

VGG-16. CIFAR-10在经过不同的剪枝方法后得到的指标结果如表1所示. 包括基准指标、SSS[23]、GAL[24]、HRank[18]、POLAR[16]、AMAS[25]、LEGR[26]和SWP[17]. 由表1可看到，MGP可以在实现高压缩率的基础上依然实现高的精确度. 和LEGR相比，虽然计算量压缩率低约一个百分点，但是精确度却提高了1.7%. 和SSS相比，可以在精确度提高1.1%的同时，压缩率提高34.4%；与基准相比，可以在压缩率达到76.73%时，实现更高的准确率. 实验MGP-75超参数设置为：$\alpha=1\times10^{-5}$，$\beta=1.6\times10^{-5}$.

ResNet-56. 由表2可看出，MGP在基准的计算量上压缩76.7%后，准确率还可以提升0.4个百分点. 相比于Hinge，在准确率相同的情况下，MGP可以使网络多压缩26.18%. 相比SWP，虽然参数量压缩量少了0.01M，但计算量压缩率却提升了7%. 相比PFEC+KESI，可以在网络多压缩14.68%的情况下，精确度高出0.35%. 实验MGP-75超参数设置为：$\alpha=1\times10^{-5}$，$\beta=1.5\times10^{-5}$，实验MGP-80超参数设置为：$\alpha=1\times10^{-5}$，$\beta=2.3\times10^{-5}$.

ResNet-110. 由表3可知，MGP在计算量压缩率达到79.64%时，相比基准，准确率还提升了0.64%. 相比与其他方法，我们可以在计算量压缩率远远领先的基础上保持更高的准确率，与LRF[26]相比，我们在计算量压缩率领先约10%的基础上还能保持更高（0.19%）的精确度. 与GAL-0.1相比，MGP可以在精确率高0.55%的情况下，压缩率高60.94%. 据我们所知，此为该网络

表1　VGG-16在CIFAR-10上结果

模型	准确率	计算量(PR)	参数量(PR)
Vgg-16	93.96%	627.48M(0.00%)	14.95M(0.00%)
SSS[23](2018)	93.02%	366.26M(42.30%)	3.93M(73.70%)
GAL-0.1[24](2019)	93.42%	343.78M(45.20%)	2.67M(82.20%)
POLAR[16](2020)	93.92%	284.04M(54.00%)	—
HRank[18](2020)	92.34%	217.22M(65.40%)	2.64M(82.30%)
SWP[17](2020)	93.65%	180.93M(71.16%)	1.08M(92.70%)
AMAS-0.005[25](2021)	93.22%	169.62M(72.97%)	4.18M(72.04%)
MGP-75	94.15%	145.98M(76.73%)	0.9M(93.98%)
LEGR[26](2020)	92.40%	140.60M(77.59%)	—

表2　ResNet-56在CIFAR-10上结果

模型	准确率	计算量(PR)	参数量(PR)
ResNet-56	93.26%	250.98M(0.00%)	0.85M(0.00%)
SSS[23](2018)	93.39%	178.7M(28.80%)	0.59M(30.60%)
HRank[18](2020)	93.52%	177.44M(29.30%)	0.71M(16.80%)
GAL-0.6[24](2019)	93.38%	156.60M(37.60%)	0.75M(11.80%)
Hinge[27](2020)	93.69%	125.44M(50.00%)	0.44M(51.27%)
MCH[28](2021)	93.23%	125.44M(50.00%)	—
LEGR[26](2020)	93.70%	117.8M(53.10%)	—
PFEC+KESI[29](2020)[l]	93.34%	96.63M(61.50%)	0.28M(67.06%)
POLAR[16](2020)	92.63%	72.93M(70.94%)	—
LRF-60[30](2021)	93.19%	65.50M(73.90%)	0.22M(74.10%)
SWP[17](2020)	92.98%	61.36M(75.55%)	0.19M(77.64%)
MGP−75	93.69%	59.79M(76.18%)	0.28M(67.06%)
AMAS-0.2[25](2021)	91.70%	54.56M(78.28%)	0.20M(76.47%)
MGP-80	93.12%	43.81M(82.54%)	0.20M(76.47%)

在模型剪枝方法中的最佳表现. 实验MGP-75超参数设置为: α=1×10^{-5}, β=3×10^{-6}, 实验MGP-80超参数设置为α=1×10^{-5}, β=7×10^{-6}.

GoogLeNet. 由表4可知, 相比基准, MGP可以在保持准确率基本不变的情况下, 实现72%左右的计算量压缩率, 而且明显优于其他方法. 相比AMAS, MGP可以在压缩率高1.32%的前提下, 精确度高0.49%. 相比Hrank, MGP可以在精确度高0.5%的基础上, 压缩率高出17.4%. 实验MGP-70超参数设置为α=5×10^{-5}, β=1.5×10^{-5}.

表3　ResNet-110在CIFAR-10上结果

模型	准确率	计算量(PR)	参数量(PR)
ResNet-110	93.50%	505.78M(0.00%)	1.72M(0.00%)
GAL-0.1[24](2019)	93.59%	411.4M(18.70%)	1.65M(4.07%)
NISP[31](2018)	93.32%	286.7M(43.30%)	—
GAL-0.5[24](2019)	92.74%	260.40M(48.50%)	0.95M(44.80%)
FPGM[32](2019)	93.74%	242.00M(52.20%)	—
AMAS-0.05[25](2021)	93.99%	218.12M(56.87%)	0.70M(59.30%)
HRank[18](2020)	93.36%	211.4M(58.20%)	0.70M(59.30%)
LRF-50[30](2021)	94.34%	189.16M(62.60%)	0.63M(63.50%)
ABC-60%[14](2020)	93.58%	179.74M(64.50%)	0.56M(67.40%)
MGP-75	94.53%	134.36M(73.44%)	0.59M(65.70%)
SWP[17](2020)	93.67%	123.8M(75.52%)	0.52M(69.77%)
MGP-80	94.14%	102.96M(79.64%)	0.43M(75.00%)

表4 GoogLeNet在CIFAR-10上结果

模型	准确率	计算量(PR)	参数量(PR)
GoogLeNet	95.05%	3048.62M(0.00%)	6.15M(0.00%)
L1[15](2016)	94.54%	2040M(33.08%)	3.51M(42.93)
GAL-0.05[24](2019)	94.53%	1880M(38.33%)	3.12M(49.27%)
HRank[18](2020)	94.53%	1380M(54.73%)	2.74M(55.45%)
ABC-30%[14](2020)	94.84%	1026.38M(66.33%)	2.46M(60.00%)
AMAS-0.05[25](2021)	94.54%	889.96M(70.81%)	2.09M(66.02%)
MGP-70	95.03%	849.62M(72.13%)	1.94M(68.45%)

3.4 CIFAR-100实验结果

VGG-16. 由表5可以看到,MGP可以在计算量压缩率为65.42%时,准确率相比基线提高了0.5%. 相比于POLAR,我们可以在维持准确率基本不变的基础上实现更高(22%)的计算量压缩率. 相比于SWP,可以在参数量压缩率相当的前提下,计算量压缩率提升11.5%和准确率提升1.6%.实验MGP-65超参数设置为α=1×10^{-5},β=2×10^{-5},实验MGP-75超参数设置为α=1×10^{-5},β=3×10^{-5}.

ResNet-56. 由表6可看出,MGP可以在计算量压缩率为60%左右,仍然保持和基准相当的准确率. 相比于PF-A,本文方法可以在计算量压缩率远超54.35%的前提上,准确率还提高0.8%. 相比POLAR,MGP可以在准确率相同的前提下,压缩率提升29.88%,可见细剪枝的高效性. 实验MGP-55超参数设置为α=1×10^{-5}, β=1.2×10^{-5}, 实验MGP-65超参数设置为α=1×10^{-5},β=2×10^{-5}.

表5 VGG-16在CIFAR-100上结果

模型	准确率	计算量(PR)	参数量(PR)
VGG-16	73.83%	627.48M(0.00%)	14.95M(0.00%)
NS[16](2017)	74.20%	389.04M(38.00%)	—
COP[33](2019)	71.77%	357.66M(43.00%)	4.01M(73.20%)
POLAR[16](2020)	74.25%	357.06M(43.10%)	—
SWP[17](2020)	71.58%	232.2M(62.99%)	1.80M(89.95%)
MGP-65	74.34%	216.99M(65.42%)	2.18M(85.42%)
MGP-75	73.17%	159.8M(74.53%)	1.67M(88.83%)

表6 ResNet-56在CIFAR-100上结果

模型	准确率	计算量(PR)	参数量(PR)
ResNet-56	71.92%	250.98M(0.00%)	0.85M(0.00%)
PF-A[34](2017)	70.42%	224.88M(10.40%)	0.77M(9.40%)
POLAR[16](2020)	72.46%	188M(25.09%)	—
PF-B[34](2017)	69.95%	181.7M(27.60%)	0.73M(13.70%)
SWP[17](2020)	71.17%	121.9M(51.43%)	0.67M(21.17%)
MGP-55	72.46%	113.02M(54.97%)	0.62M(27.06%)
MGP-65	71.71%	88.48M(64.75%)	0.53M(37.65%)

3.5 ImageNet实验结果

ResNet-50. 由表7可知,MGP在网络压缩率达到56.95%的情况下,精度只下降了0.48%.相比POLAR,本文在多压缩240M计算量的前提下,准确度上升了0.04%. 相比LRF-50,本文在多压缩420M计算量的前提下,准确度只下降了0.11%. 相比ABC-70%,本文在多压缩60M的前提下,准确度仍高出2.15%. 实验MGP-55超参数设置为α=1×10^{-5},β=6×10^{-5}.

3.6 消融实验

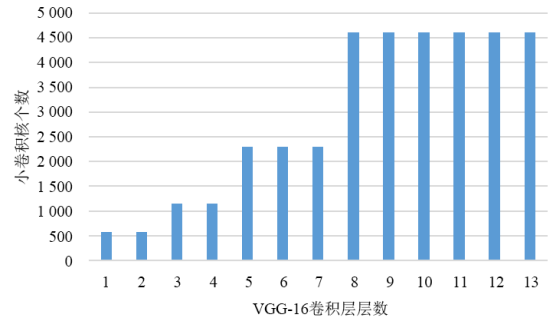
3.6.1 L1正则化和自适应L1正则化比较

经过粗细剪枝后,网络在保持高精确度的同时可以达到较高的压缩率. 图5显示了VGG-16网络在经过相同的粗剪枝后,再通过2种不同正则化方法的细剪枝,剪枝前和剪枝后的小卷积核个数变化. 在图5(a)和(b)的对比中,可以观察到,相比前半部分卷积层,网络的高压缩率主要体现在后半部分的卷积层上. 如图5(b),在VGG16

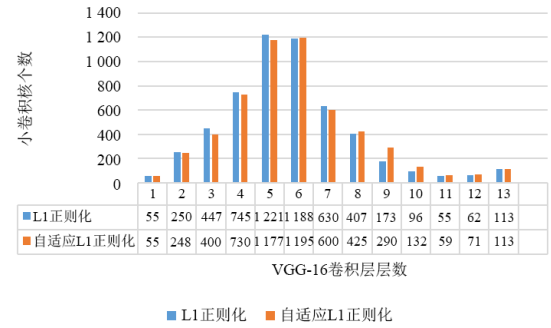
表7　ResNet-10在ImageNet上结果

模型	准确率	计算量(PR)	参数量(PR)
ResNet-50	76.15%	8.2B(0.00%)	25.50M(0.00%)
GAL-0.5[24](2019)	71.95%	4.66B(43.17%)	21.20M(16.86%)
SSS[23](2018)	71.82%	4.66B(43.17%)	15.60M(38.82%)
HRank[18](2020)	74.98%	4.60B(43.90%)	16.15B(36.67%)
LRF-50[30](2021)	75.78%	3.95B(51.80%)	12.98M(49.10%)
POLAR[16](2020)	75.63%	3.77B(54.00%)	—
MCH[28](2021)	75.60%	3.61B(56.00%)	—
ABC-70%[14](2020)	73.52%	3.59B(56.22%)	11.24M(55.92%)
MGP-55	75.67%	3.53B(56.95%)	13.42M(47.37%)

网络的粗细剪枝过程中,和L1正则化细剪枝相比,引入自适应正则项,利用控制变量法,保持其他参数量不变,略微增大L1正则化惩罚系数β,既可以使前半部分卷积层得到进一步裁剪,又可以保留更多后半部分的卷积核. 这可以使网络在保持相同水平甚至更高的裁剪力度下,得到相同甚至更高的精确度,精度对比见第3.6.2节.



(a) VGG16网络每个卷积层小卷积核个数



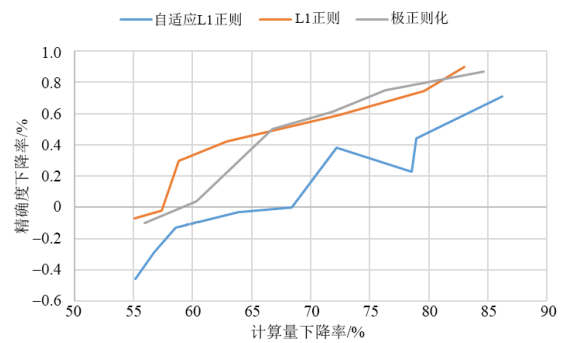
	1	2	3	4	5	6	7	8	9	10	11	12	13
L1正则化	55	250	447	745	1 221	1 188	630	407	173	96	55	62	113
自适应L1正则化	55	248	400	730	1 177	1 195	600	425	290	132	59	71	113

(b) VGG16网络两种正则方法剪枝后卷积核个数对比

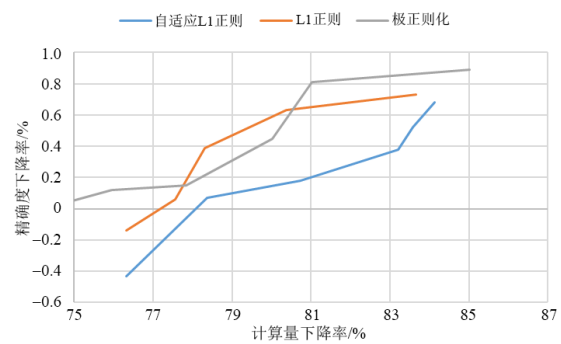
图5　VGG-16剪枝前后卷积核个数的变化

3.6.2　L1、自适应L1正则化和极正则化对比

图6为CIFAR10在VGG16和ResNet56网络上经过极正则化粗剪枝后再以3种不同正则化方式进行细剪枝的实验结果. 随着网络裁剪力度的加大,网络计算量和精确度都呈现下降趋势. 但在相同的计算量下降率的情况下,在细剪枝时运用自适应的L1正则化方法的精确度下降量明显比L1正则化和极正则化方法要少,这是因为自适应L1正则化在裁剪网络冗余参数量的同时还考虑到网络结构和个数的重要性. 当某一卷积层的卷积核个数过于稀少时,本文会减少其稀疏力度,而把网络稀疏的重心转移到其他卷积层上. 在细剪枝阶段,极正则化和L1正则化相比,实验结果没有明显的区分区,是因为极正则方法仅适用于粗剪枝时的稀疏化.



(a) CIFAR10在VGG16上的实验结果



(b) CIFAR10在ResNet56上的实验结果

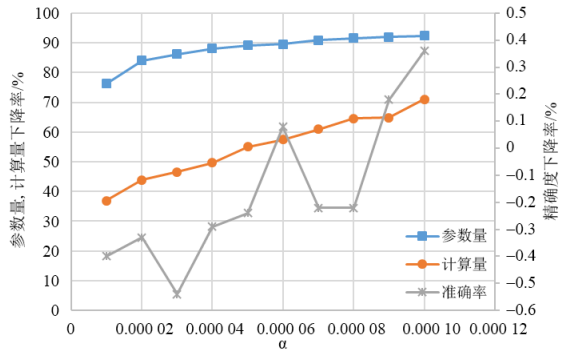
图6　三种正则化方法的对比

3.6.3　超参数α,β分析

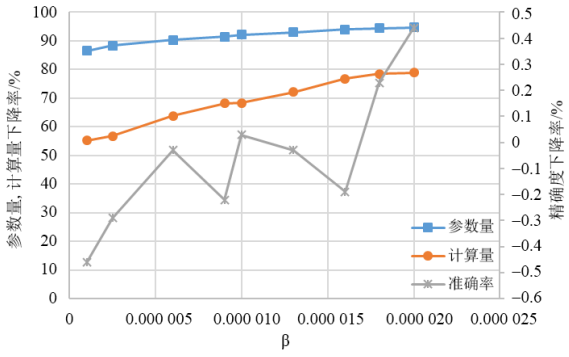
图7为超参数α,β对实验结果的影响. 图7(a)为细剪枝惩罚因子β=0时,粗剪枝惩罚因子α从1×10^{-5}到

1×10^{-4}逐渐增大的过程中，VGG-16网络在CIFAR-10数据集剪枝后参数量、计算量以及精确度下降率的变化情况．当α变大时，网络参数量，计算量以及精确度都呈现递减趋势．当网络计算量下降超过65%左右后，网络精确度下降的趋势明显变快．

图7(b)为粗剪枝惩罚因子$\alpha=1\times10^{-5}$时，细剪枝惩罚因子β从1×10^{-6}到2×10^{-5}逐渐增大过程中，VGG-16网络剪枝后参数量、计算量以及精确度下降率的变化情况．当β变大时，网络参数量、计算量以及精确度都呈现递减趋势．由图可看出，当网络计算量下降低于75%左右前，网络精确度相比基准都是提升状态．当网络计算量下降超过75%左右后，网络精确度下降的趋势才明显变快．由实验结果可见，网络的压缩比例随着超参数α,β的增大而增大，且粗细剪枝方法与直接进行粗剪枝方法相比，能够在保持精度不降的前提下获得更大的剪枝余量．



(a) α对VGG-16网络剪枝的影响



(b) β对VGG-16网络剪枝的影响

图7 α和β对实验结果的影响

4 结论

本文提出一种正则化机制下多粒度的剪枝方案．在BN层上对网络进行一次基于极正则化的粗剪枝，针对维度匹配限制了剪枝空间的问题，同时再对剩余卷积核再进行一次基于自适应L1正则化的细剪枝，在稀疏化的同时使处于维度匹配位置的卷积层中卷积核的数量保持不变．由于自适应L1正则化的有效性，稀疏化后的卷积层不仅有比原卷积层更少的参数和计算量，而且拥有更加优异的结构性质，使网络在低计算量、低参数量的前提下具有更高的表达能力．另外，本文训练网络无须微调，采用了边训练边裁剪的方式，大大减少了训练时间．与近年来相关文献对比，本文的粗细剪枝方法在多个网络的剪枝任务中取得了更好的结果，同时此方法依旧适用于其他神经网络结构的剪枝，具有较高的推广价值．

参考文献

[1] ZHU F D, ZHU L C, YANG Y. Sim-real joint reinforcement transfer for 3D indoor navigation[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 11380-11389.

[2] 权宇, 李志欣, 张灿龙, 等. 融合深度扩张网络和轻量化网络的目标检测模型[J]. 电子学报, 2020, 48(2): 390-397.
QUAN Y, LI Z X, ZHANG C L, et al. Fusing deep dilated convolutions network and light-weight network for object detection[J]. Acta Electronica Sinica, 2020, 48(2): 390-397. (in Chinese)

[3] DING X H, DING G G, GUO Y C, et al. Centripetal SGD for pruning very deep convolutional networks with complicated structure[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 4938-4948.

[4] 饶川, 陈靓影, 徐如意, 等. 一种基于动态量化编码的深度神经网络压缩方法[J]. 自动化学报, 2019, 45(10): 1960-1968.
RAO C, CHEN J Y, XU R Y, et al. A dynamic quantization coding based deep neural network compression method[J]. Acta Automatica Sinica, 2019, 45(10): 1960-1968. (in Chinese)

[5] CHO J H, HARIHARAN B. On the efficacy of knowledge distillation[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2020: 4793-4801.

[6] HOWARD A G, ZHU M, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17) [2021-07-05]. https://arxiv.org/abs/1704.04861.

[7] DAI X, JIANG Z R, WU Z, et al. General instance distillation for object detection[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2021: 7838-7847.

[8] TAI C, XIAO T, ZHANG Y, et al. Convolutional neural networks with low-rank regularization[EB/OL]. (2015-11-19)[2021-07-05]. https://arxiv.org/abs/1511.06067.

[9] DETTMERS T. 8-bit approximations for parallelism in deep learning[EB/OL]. (2025-11-14) [2021-07-05]. https://arxiv.org/abs/1511.04561.

[10] 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, 1: 1135-1143.

[11] LIU Z H, XU J Z, PENG X L, et al. Frequency-domain dynamic pruning for convolutional neural networks[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. New York: ACM, 2018: 1051-1061.

[12] LIN M, JI R, ZHANG Y, et al. Channel pruning via automatic structure search[EB/OL]. (2021-01-23) [2021-07-05]. https://arxiv.org/abs/2001.08565.

[13] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[EB/OL]. (2013-08-31) [2021-07-05]. https://arxiv.org/abs/1608.08710.

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

[15] KANG M, HAN B. Operation-aware soft channel pruning using differentiable masks[C]//Proceedings of the 37th International Conference on Machine Learning. New York: ACM, 2020: 5122-5131.

[16] ZHUANG T, ZHANG Z X, HUANG Y H, et al. Neuron-level structured pruning using polarization regularizer[C]// Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: ACM, 2020: 9865-9877.

[17] MENG F, CHENG H, LI K, et al. Pruning filter in filter [EB/OL].(2020-09-30)[2021-07-05]. https://arxiv.org/abs/2009.14410.

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

[19] HE Y H, ZHANG X Y, SUN J. Channel pruning for accelerating very deep neural networks[C]//2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2017: 1398-1406.

[20] LIU Z, SUN M, ZHOU T, et al. Rethinking the value of network pruning[EB/OL]. (2018-10-11) [2021-07-05]. https://arxiv.org/abs/1810.05270.

[21] XU B, WANG N, CHEN T, et al. Empirical evaluation of rectified activations in convolutional network[EB/OL]. (2015-05-05)[2021-07-05]. https://arxiv.org/abs/1505.00853.

[22] DENG J, DONG W, SOCHER R, et al. ImageNet: A large-scale hierarchical image database[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2009: 248-255.

[23] HUANG Z H, WANG N Y. Data-driven sparse structure selection for deep neural networks[C]//Computer Vision - ECCV 2018. Cham: Springer International Publishing, 2018: 317-334.

[24] LIN S H, JI R R, YAN C Q, et al. Towards optimal structured CNN pruning via generative adversarial learning [C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 2785-2794.

[25] WEI Y X, CHEN Y. Structured network pruning via adversarial multi-indicator architecture selection[J]. Circuits, Systems, and Signal Processing, 2021, 40(8): 4127-4143.

[26] CHIN T W, DING R Z, ZHANG C, et al. Towards efficient model compression via learned global ranking[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 1515-1525.

[27] LI Y W, GU S H, MAYER C, et al. Group sparsity: The hinge between filter pruning and decomposition for network compression[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 8015-8024.

[28] Gao S, Huang F, Huang H. Model compression via hyper-structure network[C]//International Conference on Learning Representations, IEEE, 2021: 1-17.

[29] LE D H ", VO T N, THOAI N. Paying more attention to snapshots of Iterative pruning: Improving model compression via ensemble distillation[EB/OL]. (2020-06-20) [2021-07-05]. https://arxiv.org/abs/2006.11487.

[30] JOO D, YI E, BAEK S, et al. Linearly replaceable filters for deep network channel pruning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(9): 8021-8029.

[31] YU R C, LI A, CHEN C F, et al. NISP: pruning networks using neuron importance score propagation[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern

Recognition. Piscataway: IEEE, 2018: 9194-9203.
[32] HE Y, LIU P, WANG Z W, et al. Filter pruning via geometric Median for deep convolutional neural networks acceleration[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 4335-4344.
[33] WANG W, FU C, GUO J, et al. COP: Customized deep model compression via regularized correlation-based filter-level pruning[EB/OL]. (2019-06-25) [2021-07-05]. https://arxiv.org/abs/1906.10337.
[34] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[EB/OL]. (2016-08-31) [2021-07-05]. https://arxiv.org/abs/1608.08710.

作者简介

刘 奇 男,1996年3月生,湖北荆州人. 江南大学硕士研究生. 主要研究方向为图像处理、模型压缩.

陈 莹(通讯作者) 女,1976年11月生,浙江丽水人. 江南大学教授、博士生导师. 主要研究方向为图像处理、信息融合、模式识别.
E-mail: chenying@jiangnan.edu.cn