中国科学: 信息科学 2022 年 第 52 卷 第 3 期: 430–442
SCIENTIA SINICA Informationis
论文

基于特征归因和泰勒展开引导重要度评价的梯度流剪枝

高畅[1,2], 王家祺[1,2], 景丽萍[1,2*], 于剑[1,2]

1. 北京交通大学计算机与信息技术学院, 北京 100044
2. 北京交通大学交通数据分析与挖掘北京市重点实验室, 北京 100044
* 通信作者. E-mail: lpjing@bjtu.edu.cn

收稿日期: 2021–05–19; 修回日期: 2021–07–13; 接受日期: 2021–09–01; 网络出版日期: 2022–03–04

北京市自然科学基金 (批准号: Z180006)、中国科学院光电信息处理重点实验室开放课题基金 (批准号: OEIP-O-202004)、国家科技研发计划资助 (批准号: 2020AAA0106800)、国家自然科学基金项目 (批准号: 61822601, 61773050) 和教育部指导高校科技创新规划项目资助

摘要 卷积神经网络压缩是近年来研究的热点. 本文将模型存在冗余的原因归结为部分卷积核未学到任务相关特征. 为去除这部分冗余, 本文基于剪枝框架, 从卷积核学习任务相关特征的程度和卷积核对损失函数的影响两个角度出发, 提出一种新颖的重要度评价标准. 此评价标准能准确量化卷积核的重要度, 并以此指导卷积核剪枝操作. 此外, 本文还将梯度流策略引入到卷积核剪枝的过程中, 在每次训练迭代中根据重要性和压缩率将卷积核分成两类并对它们分别用不同的更新策略. 对于冗余参数, 此策略将目标函数反传的梯度进行截流, 仅使其权重逐渐衰减直至为零. 本文在 VGGNet 和 ResNet 两种网络框架上对此剪枝算法进行验证. 结果表明: 本算法不仅能够在分类精度、计算量、参数量和任务相关特征的保留程度上优于当前主流剪枝算法, 而且在高压缩率情况下表现优越.

关键词 卷积神经网络, 压缩, 剪枝, 任务相关特征, 梯度流

1 引言

近些年, 随着可用数据量的增长和硬件计算能力的显著增强, 深度学习 [1] 在计算机视觉领域获得了前所未有的发展. 从基本的图像分类任务 [2,3], 到一些更高级的应用, 例如物体识别 [4~6]、目标检测 [7,8]、图像分割 [9,10] 等, 都能表现出极佳的性能. 然而深度模型普遍拥有规模庞大的参数量, 随之而来的就是需要强大算力和巨大的存储空间来支持模型的精准决策, 这些使得模型难以移植到计算能力受限的终端设备 (例如移动嵌入式设备) 中. 模型压缩算法在此背景下应运而生, 如何在保持模型高性能决策的情况下减少模型的占用空间和计算量成为研究热点.

引用格式: 高畅, 王家祺, 景丽萍, 等. 基于特征归因和泰勒展开引导重要度评价的梯度流剪枝. 中国科学: 信息科学, 2022, 52: 430–442, doi: 10.1360/SSI-2021-0172
Gao C, Wang J Q, Jing L P, et al. Gradient flow pruning based on the evaluation of the importance of characteristic attribution and Taylor-guidance (in Chinese). Sci Sin Inform, 2022, 52: 430–442, doi: 10.1360/SSI-2021-0172

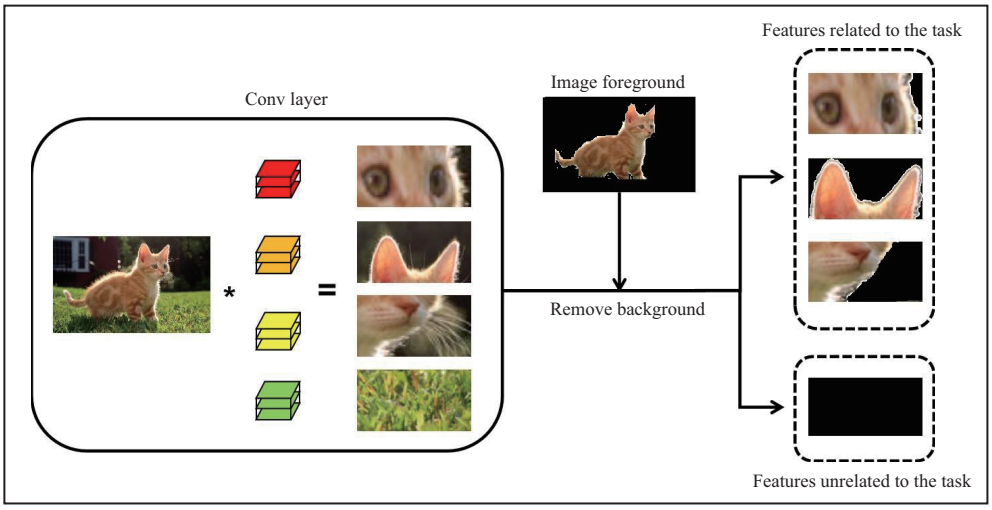

图 1 (网络版彩图) 冗余特征产生过程

Figure 1 (Color online) The process of redundant features generation

剪枝作为一种常见的压缩策略, 由于其原理简单、容易实现、压缩效果显著, 被研究者广泛采纳. 本文采用的压缩策略是卷积核剪枝, 卷积核重要度评价标准的设计是剪枝策略中的重点和难点, 可以直接影响模型中重要卷积核的划分, 从而影响剪枝的最终结果. 故我们深入探究模型存在冗余的原因, 并以此指导设计重要度评价标准. 如图 1 中所示, 对于任一卷积层来说, 卷积过程的输出特征图中可以分为任务相关特征图和任务无关特征图两部分. 任务无关特征图通常只包含与任务无关的特征, 例如背景特征, 这些特征对于模型决策影响甚微, 无关特征图的对应卷积核学到特征通常与任务无关, 因此可以将此类卷积核归为冗余. 另外, 我们考虑到将背景特征全部归为冗余特征可能会丢失部分信息. 虽然背景中不包含目标特征, 但背景特征中包含的语义信息对模型的决策可能仍有贡献. 故为避免此类特征被归为任务无关特征并去除对应的卷积核, 从而影响分类决策, 本文还将剪枝对损失函数的影响纳入评价指标. 最终本研究从两方面对卷积核的重要度进行评价: 卷积核学到任务相关特征的程度和卷积核对于损失函数的影响.

卷积核学到的特征与输出特征图中的特征有密切关系 [11], 特征图中包含了卷积核学到的语义信息 [12,13], 因此可以通过量化输出特征图中的任务相关特征来评价卷积核特征学习程度. 图 2(a) 展示了模型压缩前后的归因特征图, 压缩过程中移除学到任务相关特征的卷积核会破坏输出特征图中的任务相关特征. 图 2(b) 展示了 VGG-16 末层多个卷积核的归因特征图, 从图中可发现每个卷积核学到的特征差别很大, 为区别拟合出任务相关特征的卷积核, 从而最大程度地保留任务相关特征, 首先要量化特征图中的任务相关特征. 任务相关特征是目标上对模型决策有重要贡献的特征, 因此本文用模型决策的归因特征 [14] 和目标特征的交集 [15] 来拟合任务相关特征, 归因特征即对模型决策有重要贡献的特征. 本文将量化任务相关特征的结果, 称为特征因子, 最终可以用它来衡量卷积核任务相关特征的学习程度.

特征因子去除了背景特征, 只根据任务相关特征判定卷积核的重要程度. 但背景特征中可能会包含部分对决策有贡献的信息, 去除此类信息会影响模型决策. 损失函数的值与模型决策关系密切, 故可利用卷积核对损失函数的影响表示卷积核对模型决策的影响. 本文引入由泰勒一阶展开式推导得出的损失因子 [16], 它通过拟合去除卷积核前后损失函数的变化, 来衡量卷积核对损失函数的影响. 最终,

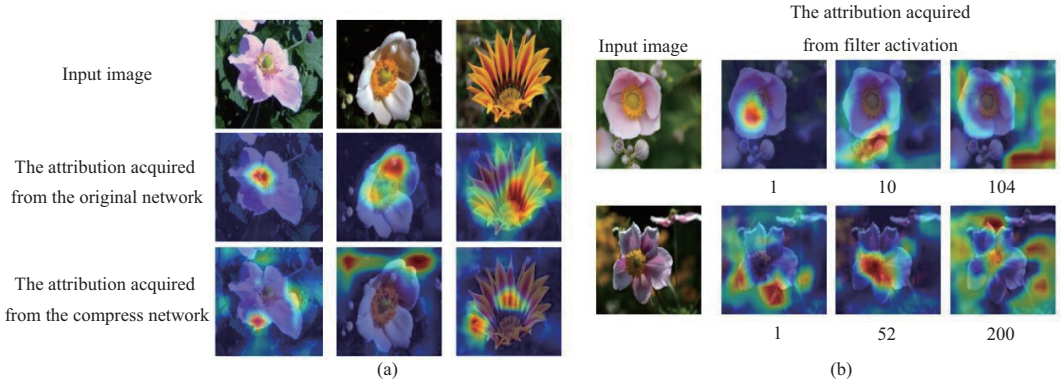

图 2 (网络版彩图) (a) 模型的归因特征; (b) 卷积核的归因特征

Figure 2 (Color online) (a) Attributional characteristics of the model; (b) attribution characteristics of the filter

本文结合特征因子和损失因子, 提出一种新颖的多度量角度的重要性评价标准.

除卷积核重要度评价标准以外, 剪枝过程对模型性能也有重要影响. 当前剪枝工作大多是直接去除冗余卷积核, 再通过微调过程降低精度的下降程度. 这样面临两方面问题. (1) 剪枝后的模型训练困难, 并且很难预测微调后的精度 [17], 例如, 卷积核剪枝模型易陷入一个坏的局部极值, 甚至产生一个不可接受的精度下降. (2) 微调过程繁杂, 简化微调步骤会进一步造成精度下降. 故本文将梯度流剪枝策略 [17] 首次引入到卷积核剪枝的过程中. 对于冗余卷积核, 梯度流策略让其在训练过程中权重逐渐归零. 最终, 策略将权重归零的卷积核直接去除, 且不会对模型精度造成任何影响, 因此不需要微调过程.

本文基于 VGGNet [3] 和 ResNet [6] 网络架构对算法进行验证, 并与几种常用的压缩策略进行性能对比, 使用的数据集包括 flower-102 [18], cifar-10 [19] 和 ImageNet [20]. 在评价性能上, 本文不仅采用常见的模型压缩评价指标: 模型分类精度、模型参数量、浮点数计算量和前向推理加速比, 而且考虑了任务相关特征的保留程度.

本文的贡献主要有以下两个方面:

(1) 本文深入探究模型冗余产生的原因 —— 部分卷积核未学到任务相关特征, 并从卷积核学到任务相关特征的程度和卷积核对损失函数的影响两方面入手, 提出基于特征归因和泰勒展开引导的卷积核重要度评价标准.

(2) 本文首次将梯度流策略引入到卷积核剪枝的工作中. 算法将剪枝和训练过程融为一体, 不重要的卷积核在训练过程中逐渐归零, 降低了剪枝过程中的精度下降, 并且去除了繁杂的微调过程.

2 相关工作

卷积神经网络的压缩策略主要分为 5 类 [21]. 一些工作致力于分解或拟合参数张量 [22]; 量化和二值化技术 [23] 通过使用占位较少的数据类型近似表示 32 位浮点数以压缩模型; 知识蒸馏 [24] 将知识从一个大的教师网络迁移到较小的学生网络; 还有部分工作直接设计冗余参数较少的紧致模型 [25,26] 以达到压缩目的. 剪枝作为本文研究重点, 将在本节详细介绍.

模型剪枝 [11,15,17,27~30] 试图删除不重要的参数, 并且保证模型精度不显著下降. 剪枝算法分为非结构化剪枝 [17,28,29] 和结构化剪枝 [11,15,27,30]. 其中非结构化剪枝的方法有: Han 等 [28,29] 迭代设置阈值并且对于小于阈值的参数进行剪枝; Ding 等 [17] 提出一个新颖的优化策略 GSM (global sparse

momentum SGD), 该策略将更新规则分成两部分, 并自动分配各层最佳压缩率, 只需要一个全局压缩率就可完成压缩. 非结构化剪枝可以利用极少的参数拟合原模型, 但必须依靠特定的软硬件平台才能减少计算量.

结构化剪枝去除模型中的结构 (例如神经元、卷积核 [15,27] 和信道 [11]). Li 等 [11] 从一个可解释的视角研究卷积神经网络的压缩问题. 文章从理论角度揭示了输入特征图与二维卷积核之间的关系, 提出一种基于核稀疏度和熵的衡量指标并在特征不可知的情况下来指导压缩. 然后基于这个新的度量进一步进行核聚类, 实现卷积神经网络的高精度信道压缩. Ding 等 [27] 为解决剪枝后精度下降的问题, 提出了一个新颖的优化方法: 中心随机梯度下降, 可以在训练过程中把多个卷积核集中到参数空间中的一个点. 最终当训练完成时, 删除相同的卷积核可以不产生任何的精度损失, 因此不需要微调过程. 另外有些工作把归因机制引入到了剪枝工作中. Yeom 等 [30] 基于神经网络的可解释性, 提出一种新的剪枝准则: 利用可解释人工智能概念得到的相关度, 自动找到最相关权重或卷积核. Zhang 等 [15] 把归因机制引入到卷积核剪枝的工作, 提出一种归因评价机制用于评价卷积核所学特征与目标物体的相关度, 然后将模型中与目标物体特征相关度较低的卷积核进行裁剪, 以实现模型压缩的目的, 同时也能够保留原模型的归因特征. 结构化剪枝主要面临了两个难题: (1) 准确评估结构的重要度; (2) 降低剪枝操作中的精度损失. 本文针对这两个问题, 提出我们的解决方案.

3 卷积核重要度评价

评价卷积核在网络中的重要度是模型剪枝的核心工作, 本文分析卷积核的重要度由任务相关特征的学习程度和它本身对模型损失函数的影响来共同决定. 故本节将详细介绍本文提出的基于特征归因和泰勒展开引导的卷积核重要度评价标准以及计算过程. 3.1 和 3.2 小节分别介绍了如何量化卷积核的任务相关特征学习程度和卷积核对损失函数的影响; 3.3 小节叙述了完整的卷积核重要度定义.

一般来说, 设原模型卷积层的个数为 L 个. 利用 $W_i = \{w_1^i, w_2^i, w_3^i, \ldots, w_{n_i}^i\}$, 将输入张量 $O^{i-1} \in \mathbb{R}^{n_{i-1}\times h_{i-1}\times w_{i-1}}$ 转换成输出张量 $O^i \in \mathbb{R}^{n_i\times h_i\times w_i}$. 在这里, W_i 表示第 i 层的参数, $w_j^i \in \mathbb{R}^{n_{i-1}\times k_i\times k_i}$ 表示第 i 层第 j 个卷积核的参数, 模型卷积核的总个数为 N, n_i 表示第 i 层的卷积核个数, k_i 表示第 i 层的卷积核大小. $O^i = \{o_1^i, o_2^i, o_3^i, \ldots, o_{n_i}^i\}$ 表示第 i 层的输出结果, 第 i 层的第 j 个卷积核的输出结果表示为 o_j^i, h_i 和 w_i 分别表示第 i 层输出特征图的高和宽.

3.1 卷积核对任务相关特征的学习程度

本文在量化卷积核的特征学习程度过程中, 采用归因方法 XGrad-CAM [31] 计算卷积核对应输出特征图中的归因特征, 再根据像素级标签, 过滤背景中的归因特征, 得到任务相关特征, 并根据其计算卷积核的特征学习程度. 特征图是由卷积核激活产成的, 特征图中的特征直接关系到卷积核学到的特征, 故可以通过计算特征图中包含任务相关特征的程度, 得到对应卷积核学到任务相关特征的程度. 卷积核的任务相关特征学习程度计算步骤如下:

$$\alpha_{i,j}^c = \frac{\sum_{q=1}^{h}\sum_{r=1}^{g}\left(\frac{\partial s_c}{\partial o_j^i}\cdot o_j^i\right)_{q,r}}{\sum_{q=1}^{h}\sum_{r=1}^{g}(o_j^i)_{q,r}}, \tag{1}$$

其中, s_c 表示 c 类的置信度, $\alpha_{i,j}^c$ 表示第 i 层第 j 个卷积核对类别 c 的贡献权重, 称为归因因子, 归因因子可通过第 i 层第 j 个卷积核的输出值与类别 c 的置信度对输出值的导数相乘得出, h 和 g 分别表示特征图的高和宽, 归因因子用来评估各层特征图对于决策的重要程度.

利用归因因子对各卷积核的输出特征图加权, 就可以得到第 i 层的第 j 个卷积核学到的对决策有贡献的特征, 也就是归因特征, 计算过程如下:

$$\mathrm{XCAM}_{i,j} = \mathrm{ReLU}\left(\alpha_{i,j}^{c} \cdot o_j^i\right). \tag{2}$$

除此之外还要去除目标区域之外的归因特征, 从而确保卷积核学到的是任务相关特征, 而非背景特征. 卷积核任务相关特征学习程度可以利用下式求解:

$$\mathrm{DC}_{i,j} = \sum_g \sum_h \left(\mathrm{XCAM}_{i,j}\right)_{g,h} \cdot \mathrm{label}_{g,h}, \tag{3}$$

这里 label 为样本的像素级标签, 背景区域为 0, 目标区域为 1, 需要通过下采样使标签和特征图尺寸相同, 然后与归因特征图进行逐点相乘, 并逐点累加, 得到基于单个样本的第 i 层第 j 个卷积核的任务相关特征学习程度, 即特征因子 $\mathrm{DC}_{i,j}$.

3.2 卷积核对损失函数的影响

一般来说, 虽然背景特征和任务相关特征之间不存在交集, 但背景特征仍可能提供部分对模型决策有价值的信息. 故只考虑卷积核是否学到任务相关特征会丢失背景区域中的有效信息, 从而影响最终决策. 本文利用损失函数与模型决策的密切关系, 通过卷积核对损失函数的影响计算卷积核对模型决策的影响. W 为原模型参数, 设 W' 为剪枝后模型的参数, $|L(x,y,W) - L(x,y,W')|$ 为剪枝前后损失函数的变化, 剪枝过程旨在找到令 $|L(x,y,W) - L(x,y,W')|$ 最小的 W'. 本文使用泰勒展开式拟合剪枝前后损失函数的变化, 以此度量卷积核对损失函数的影响. 此度量标准在深层网络中仍然可行且计算量小, 仅需卷积核的权重和反传梯度就可计算.

剪掉的卷积核对损失函数的影响要尽可能小, 从而保证模型决策基本保持不变. 令第 i 层第 j 个卷积核 w_j^i 的参数为零, 此时的模型代表剪掉第 i 层第 j 个卷积核的模型. 本小节在式 (4) 中利用泰勒展开式, 将卷积核去除前后的损失函数联系起来, 通过二者的差值评估卷积核对于损失函数的影响, 差值大, 则影响大, 即卷积核更重要, 反之越冗余.

$$L\left(x,y,W_{w_j^i \leftarrow 0}\right) = L\left(x,y,W\right) - \frac{\partial L\left(x,y,W\right)}{\partial w_j^i}\left(0 - w_j^i\right) + o\left({w_j^i}^2\right). \tag{4}$$

高阶项计算繁杂且数值一般较小, 故忽略高阶项, 式 (4) 转化为

$$\left|L\left(x,y,W_{w_j^i \leftarrow 0}\right) - L\left(x,y,W\right)\right| = \left|\frac{\partial L\left(x,y,W\right)}{\partial w_j^i} w_j^i\right| = \mathrm{LC}_{i,j}. \tag{5}$$

式 (5) 拟合了去掉第 i 层的第 j 个卷积核损失函数的变化, 以此得到卷积核对损失函数的重要度 $\mathrm{LC}_{i,j}$.

3.3 定义卷积核重要度

卷积核的重要度取决于两部分: 任务相关特征的学习程度和卷积核对损失函数的影响, 为了控制二者的平衡, 本文还引入了超参数 β. 基于整个数据集的卷积核重要度的计算如下:

$$\mathrm{valuate}_{i,j} = \frac{1}{\mathrm{num}} \sum_D \left(\mathrm{LC}_{i,j} + \beta \times \mathrm{DC}_{i,j}\right), \tag{6}$$

通过上述算法, 就可以得到卷积核的重要度, $\mathrm{valuate}_{i,j}$ 表示第 i 层第 j 个卷积核的重要度, 用 D 代表整个数据集, num 为数据集样本个数. 卷积核的重要度随着模型的深度不同而变化, 为避免对某层压

缩过度而造成瓶颈, 本文采用 L_2 范数对基于整个数据集上的重要度进行层级正则化, 最终卷积核重要度为

$$\text{valuate}_{i,j} = \frac{\text{valuate}_{i,j}}{\sqrt{\sum_{j=1}^{n_i} \left(\text{valuate}_{i,j}\right)^2}}. \tag{7}$$

4 基于重要度的梯度流剪枝

本节主要介绍梯度流剪枝策略如何在重要度评价的指导下对模型进行剪枝. 4.1 小节叙述了梯度流剪枝策略; 4.2 小节阐述了完整的算法框架.

4.1 梯度流剪枝策略

为克服引言中总结的传统剪枝方法存在的诸多问题, 本文首次将梯度流策略用在卷积核剪枝上, 通过直接改变目标函数反传的梯度流来偏离训练方向, 从而达到高压缩率并同时维持精度. 直观地说, 本文用梯度来更新重要卷积核以最小化目标函数, 同时对冗余卷积核进行惩罚, 使其逐渐接近于零.

给定全局压缩率 C, 用 $Q = N \times C$ 记录模型中重要的卷积核个数. 每轮训练开始时, 根据上一轮计算的重要度, 卷积核被划分成两部分. 接下来每次迭代, 都对模型输入一小批样本, 用普通的链式法则计算梯度, 对 Q 个重要卷积核执行梯度更新和权重衰减, 对其余部分只执行权重衰减, 例如 L_2 正则化. L 表示目标函数, β 表示动量系数, η 表示权重衰减系数, α 表示学习率, 普通的动量随机梯度下降更新过程如下:

$$\begin{aligned} Z^{(k+1)} &\leftarrow \beta Z^{(k)} + \eta W^{(k)} + \frac{\partial L}{\partial W^{(k)}}, \\ W^{(k+1)} &\leftarrow W^{(k)} - \alpha Z^{(k+1)}. \end{aligned} \tag{8}$$

本文使用动量随机梯度下降进行优化, 从而加速了冗余卷积核接近零的速度. 为了截断反传给冗余卷积核的梯度流, 还引入了一个掩码列表 $B = \{b_1, b_2, b_3, \ldots, b_L\}$, 其中 $b_i \in [0,1]^{n_i}$, 每个元素对应模型第 i 层中的一个卷积核, 卷积核重要置 1, 否则置 0. B 的计算过程如下:

$$b_{i,j} = \begin{cases} 1, & \text{valuate}_{i,j} \geqslant Q_{\text{th}}\left(\text{valuate}\right), \\ 0, & \text{otherwise}. \end{cases} \tag{9}$$

Q_{th} 表示元素中第 Q 大的值, 对于重要度小于阈值的卷积核掩码置零, 意味着不再有梯度反传给它们, 引入掩码之后更新规则转变为

$$\begin{aligned} Z^{(k+1)} &\leftarrow \beta Z^{(k)} + \eta W^{(k)} + B \times \frac{\partial L}{\partial W^{(k)}}, \\ W^{(k+1)} &\leftarrow W^{(k)} - \alpha Z^{(k+1)}, \end{aligned} \tag{10}$$

对每个卷积核来说, B 的计算是激活选择, 若全为 1, 则相当于普通的动量随机梯度下降.

本小节通过改变训练过程中梯度的流向, 使冗余卷积核逐渐衰减直至归零, 后续只需在训练结束后直接去除归零的卷积核, 此剪枝操作不会造成任何精度损失, 故省去微调过程并且相较于直接减去卷积核, 逐步衰减更能减少精度下降程度.

4.2 算法框架

本小节将提出的重要度评价标准和梯度流剪枝策略结合成一个完整的剪枝过程, 具体细节见算法 1.

Algorithm 1 Pruning algorithm for convolutional neural networks

Input: Dataset D, convergent MODEL, compression ratio C, iteration epochs.

Output: The compressed CMODEL;

```
for i = 1 to epochs do
    for batch in D do
        Input batch into MODEL and compute forward;
        Evaluate the importance of every convolutional filter in MODEL on the batch by attribution and Taylor-guided pruning method;
        Perform weight decay to penalize unimportant parameters and both the objective function related gradients and weight decay to update others;
    end for
    The importance on the D is regularized by L2 norm;
    According to the importance and compression ratio C, the convolution kernel is divided into two categories: the important and the redundant;
end for
The convolutional filter with weight close to zero is deleted.
```

5 实验与结果

本文利用 Pytorch 框架实现文中涉及的实验, 并且在 flower-102, cifar-10 和 ImageNet 数据集上对提出的算法进行有效性实验验证. 我们还将剪枝算法应用于不同的模型框架, 包括 VGGNet 和 ResNet. 5.1 小节给出了实验细则; 5.2 小节具体展示了算法验证及对比结果; 5.3 小节展示了模型分类精度随压缩率的变化情况.

5.1 实验说明

本文为保证剪枝后的模型仍能正常工作, 即保持轻量和准确之间的平衡, 使用超参数 C: 去除的卷积核数量和原模型卷积核总数量的比值, 来控制模型的压缩程度.

数据集. 实验所使用的 flower-102 数据集由 102 类产自英国的花卉组成, 每类由 40–258 张照片组成. 除图像以及类别标签之外, 数据集还包含每张图片的像素级标签. cifar-10 数据集由来自 10 个类的小型图片组成, 其中训练集包括 50000 张, 测试集包括 10000 张. ImageNet 的子集 ILSVRC 2012 由来自 1000 个不同类的图片组成, 其中训练集包括 128167 张图片, 验证集包括 50000 张图片, 测试集包括 100000 张图片.

评价标准. 为评估压缩算法的有效性, 我们不仅在模型分类精度 (top-1)、参数量、浮点数计算次数, 以及前向推理加速比这些常用指标上对模型压缩前后进行对比, 还对比了任务相关特征的保留程度.

配置. 实验中所有的训练阶段以及微调阶段, 均采用随机梯度下降算法进行网络优化. 对于数据集 ImageNet, 使用官方提供的预训练模型, 在微调阶段, 学习率为 0.005, 动量为 0.98, 图片以 128 张为一个批次送入模型, 微调轮数为 50, 每经过 10 轮训练衰减为之前的 0.1, 权重衰减系数为 0.0001. 对于其他数据集, 在训练阶段, 训练轮数为 500, 网络的学习率为 0.1, 每经过 100 轮训练学习率衰减为之前的 0.1, 动量为 0.9, 权重衰减系数为 0.0001; 在微调阶段, 学习率为 0.005, 动量为 0.98, 图片均以 32 张为一个批次送入模型. 本文所有实验均在 2 块 NIVIDIA TITAN Xp GPUs 上完成.

表 1 VGG-16 基于 flower-102 的剪枝结果

Table 1 Pruning results of VGG-16 on flower-102

Model	Top-1 (%)	FLOPs (PR (%))	Parameters (PR (%))	Speedup (time)
VGG-16	91.18	1.56×10^{10} (0.0)	1.35×10^{8} (0.0)	1x
Our method	91.37	1.71×10^{9} (89.04)	1.06×10^{6} (99.21)	4.22x
Attribution [15]	88.43	3.18×10^{9} (79.62)	2.10×10^{7} (84.44)	3.81x
GSM [17]	90.49	–	1.35×10^{7} (90.00)	–
LRP [29]	87.67	4.50×10^{9} (71.15)	3.75×10^{8} (72.22)	3.65x
Taylor [16]	87.55	3.03×10^{9} (80.58)	3.48×10^{7} (74.19)	4.12x

表 2 VGG-16 基于 cifar-10 的剪枝结果

Table 2 Pruning results of VGG-16 on cifar-10

Model	Top-1 (%)	FLOPs (PR (%))	Parameters (PR (%))	Speedup (time)
VGG-16	90.65	1.56×10^{10} (0.0)	1.35×10^{8} (0.0)	1x
Our method	89.01	1.53×10^{9} (90.19)	1.83×10^{6} (98.64)	4.52x
Attribution [15]	87.79	3.11×10^{9} (80.06)	1.92×10^{7} (85.73)	4.01x
GSM [17]	88.45	–	1.35×10^{7} (90.00)	–
LRP [29]	88.15	5.02×10^{9} (67.82)	4.51×10^{7} (66.59)	3.24x
Taylor [16]	87.50	2.28×10^{9} (85.38)	2.14×10^{7} (84.11)	4.04x

5.2 算法性能验证及对比

本小节在 VGG 和 ResNet 两种网络架构上对我们的算法进行了验证, 并且在分类精度、压缩率、前向推理加速比和任务相关特征等指标上对比了几种高性能的压缩算法.

VGG-16. 本部分实验中, β 设为 1.0, C 设为 0.3, 我们提出的算法与其他剪枝算法基于数据集 flower-102 的模型性能对比结果见表 1. 从表中可以得出本文算法在分类精度 Top-1 上, 明显优于 Attribution 算法、LRP (layer-wise relevance propagation) 算法和 Taylor 算法, 也略优于同样使用梯度流剪枝的 GSM 算法. 从计算加速来看, GSM 算法属于非结构化剪枝, 需要特殊的硬件和软件平台才能起到加速的效果, 在正常软硬件上没有加速效果, 故实验中没有统计 GSM 算法的计算加速率. 此外相较于其他 3 个对比试验, 本算法压缩的浮点数计算量也明显更大 (89.04% vs. 79.62%, 71.15%, 80.58%) 且前向推理时间较短. 对于全连接层, 我们用类似 ResNet 结构的处理, 用全局平均池化层代替全连接层, 极大程度地降低了全连接层的参数冗余. 故在参数量上, 本文算法远远少于其他剪枝算法. 表 2 展示了基于 cifar-10 数据集上的结果, 可以明显看出本文算法在高压缩率下仍然要优于其他对比算法. 综合以上 4 个指标在两个数据集上的结果, 我们提出的算法可以应用于 VGGNet 并且性能优越, 从而证明了此算法可以用于串行卷积模型的压缩工作.

此外, 实验还对比了原 VGG-16 模型和各剪枝算法得到的压缩模型的归因特征图, 如图 3 所示. 从图中可以明显发现本文提出的算法所得模型对在目标区域上的归因特征即任务相关特征保留程度强于 Taylor 和 GSM. 这也证明了我们提出的算法可以更好地保留任务相关特征.

ResNet-18. 本部分基于 ResNet-18 模型进行实验验证. 实验中, β 设置为 1.0, C 为 0.3, 与其他剪枝算法的模型性能对比结果见表 3. 对于跳跃连接结构剪枝造成的前后特征图尺寸不匹配问题, 我们用不含信息的张量 (值全为零) 补全. 从表中可以得到, 在分类精度、计算量、参数量, 以及前向推

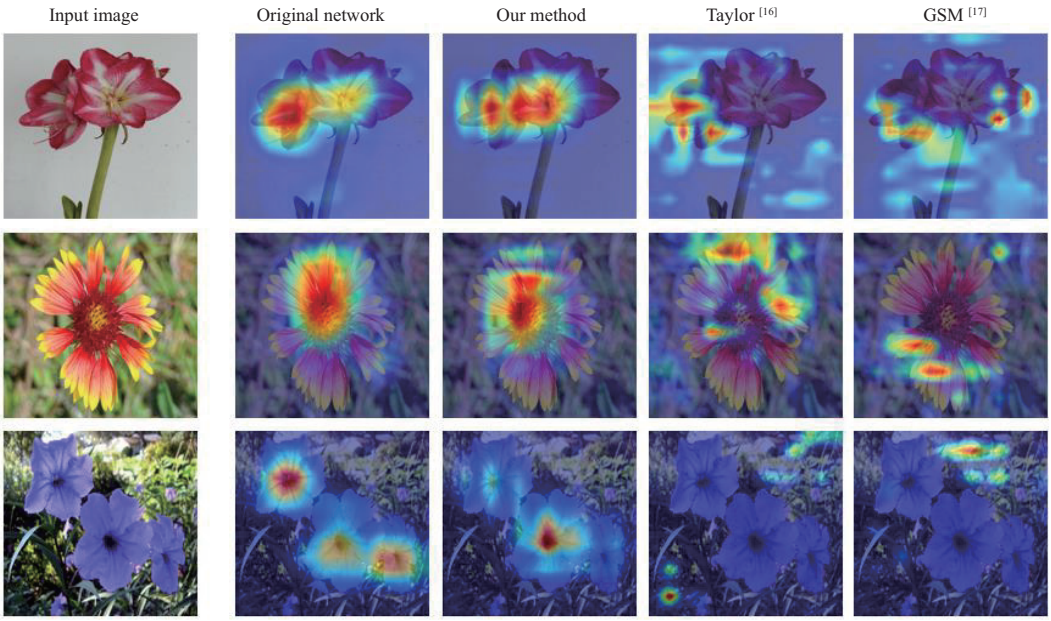

图 3 (网络版彩图) 原模型与压缩模型获得的归因特征可视化效果

Figure 3 (Color online) Illustration of attribution features between the original model and compressed model

表 3 ResNet-18 基于 flower-102 的剪枝结果

Table 3 Pruning results of ResNet-18 on flower-102

Model	Top-1 (%)	FLOPs (PR (%))	Parameters (PR (%))	Speedup (time)
ResNet-18	78.82	1.88×10^9 (0.0)	1.19×10^7 (0.0)	1x
Our method	79.18	1.71×10^8 (90.09)	8.66×10^5 (92.72)	4.36x
Attribution [15]	77.16	5.32×10^8 (71.70)	3.17×10^6 (73.36)	3.86x
GSM [17]	78.62	–	1.19×10^6 (90.00)	–
LRP [29]	77.82	1.09×10^9 (42.02)	4.68×10^6 (60.71)	2.99x
Taylor [16]	76.96	4.18×10^8 (77.77)	3.12×10^6 (73.78)	4.03x

理时间 4 个指标上, 我们所提算法都要优于其他剪枝方法. 此实验证明本文提出的算法可以在含有跳跃连接的模型中应用并且表现优越.

ResNet-50. 本部分使用 ResNet-50 模型基于大规模数据集 ImageNet 进行实验验证. 实验中, β 设置为 1.0, C 为 0.4, 与其他剪枝算法的模型性能对比结果见表 4. 从表中可以得到, 在分类精度、计算量、参数量, 以及前向推理时间 4 个指标上, 我们所提算法都要优于其他剪枝方法. 通过在不同数据集上的结果对比, 可以发现本文算法在大规模数据集上的表现稍逊于小数据集. 我们分析主要原因有两点: (1) 类别多的数据集任务相关特征的区分比较困难. (2) 梯度流剪枝通过增加重要度评估次数来降低重要卷积核误删的风险, 但大规模训练集图片多, 导致每次迭代时间较长且次数较少, 不利于梯度流策略发挥作用. 此实验证明本文在大规模数据集虽表现不如小数据集, 但仍能保持较优的性能.

5.3 压缩率分析

本小节在 VGGNet 和 ResNet 两种网络架构上比较了归因梯度流算法与其他对比算法在不同压

表 4 ResNet-50 基于 ImageNet 的剪枝结果

Table 4 Pruning results of ResNet-50 on ImageNet

Model	Top-1 (%)	FLOPs (PR (%))	Parameters (PR (%))	Speedup (time)
ResNet-50	75.72	3.44×10^9 (0.0)	2.56×10^7 (0.0)	1x
Our method	74.51	6.45×10^8 (81.23)	4.29×10^6 (83.21)	3.67x
Attribution [15]	72.47	8.51×10^8 (75.26)	8.29×10^6 (67.62)	3.19x
GSM [17]	74.30	–	6.39×10^6 (80.00)	–
LRP [29]	73.82	1.44×10^9 (58.28)	1.09×10^7 (57.25)	1.98x
Taylor [16]	72.23	1.08×10^9 (68.47)	9.36×10^6 (63.43)	2.34x

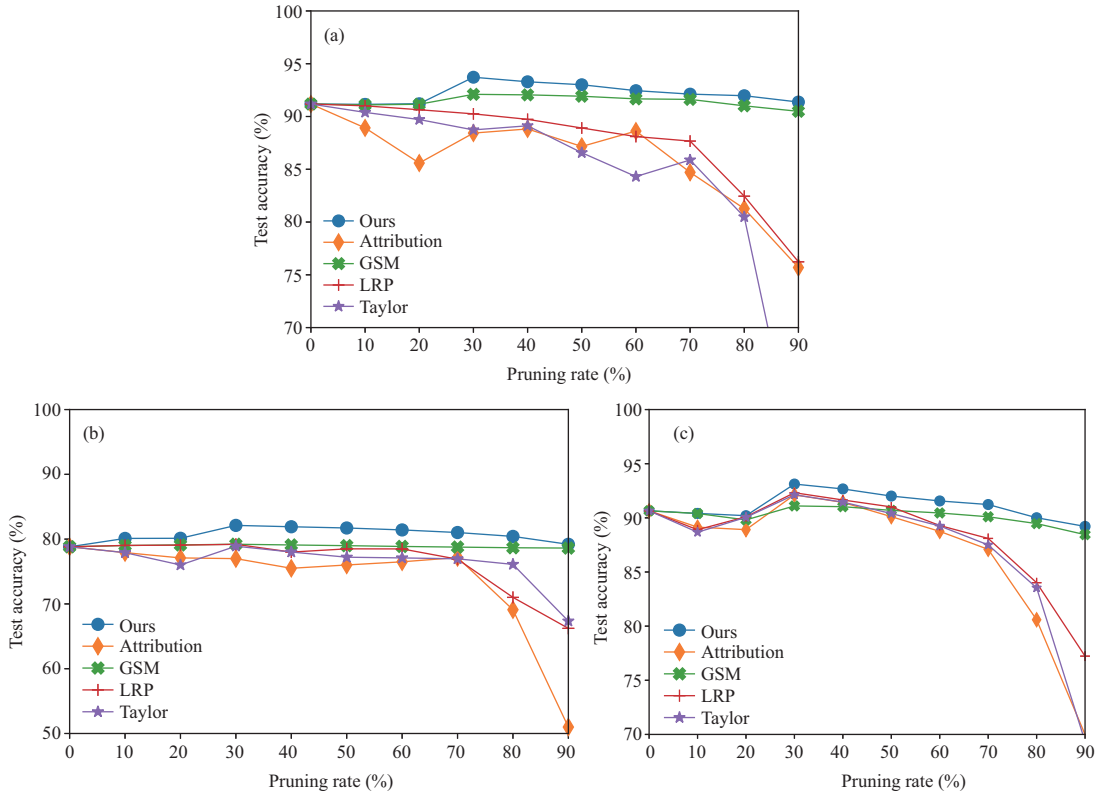

图 4 (网络版彩图) 压缩率变化下各方法测试精度对比

Figure 4 (Color online) Comparison of test accuracy in different methods as pruning rate changes. (a) VGG-16 on flower-102; (b) ResNet-18 on flower-102; (c) VGG-16 on cifar-10

缩率下, 分类精度的变化情况, 结果见图 4.

本部分实验中, β 设为 1.0, 图 4(a) 和 (b) 使用的是 flower-102 数据集. 从图 4(a) 中可以明显看出相较于 Attribution 算法、LRP 算法和 Taylor 算法, 压缩率的上升对本文算法的分类精度影响不大, 并且在高压缩率的情况下, 本文算法的分类精度远高于以上 3 种算法. 图 4(b) 展示了基于 ResNet 架构的结果, 结果表明本文算法在压缩率逐渐提高的情况下, 分类精度变化曲线平缓且有较高值.

此外, 本小节还基于 cifar-10 数据集对 VGG-16 的精度进行了随压缩率变化的监测, 结果如图 4(c) 所示. 从图中, 可以看出随着压缩率变化, 本文算法所得模型的精度没有明显的变化, 而且在高压缩率

下仍可以保持较高精度, 这一点和在 flower-102 数据集上一致.

以上结果验证了使冗余卷积核逐渐衰减直至为零要优于直接去掉, 从而表明了梯度流剪枝策略的优越性. 此外, 压缩率处于 0.3 时, 大部分算法精度有明显地上升, 这个现象可能是因为去除冗余卷积核导致输出结果中任务无关特征减少, 从而提高了模型分类精度.

6 总结

本文分析模型存在冗余是由于部分卷积核未学到任务相关特征, 故为去除此类卷积核以达到压缩目的, 从两个角度评价卷积核的重要性: 卷积核学到任务相关特征的程度和卷积核对损失函数的影响, 并将两方面影响量化结合为卷积核重要度的衡量因子. 此外, 我们将非结构化剪枝中的梯度流策略迁移到卷积核剪枝的工作中, 对冗余卷积核采取权重逐渐衰减直至为零的策略. 与直接去除卷积核相比, 由于剪枝操作不当对模型精度造成的不可逆损失显著降低.

本文通过实验对比了我们的算法与目前较为流行的剪枝算法, 实验证明本文算法在分类精度、计算量、参数量、前向推理加速比, 以及保留任务相关特征方面具有优异的表现. 尽管该算法考虑了保留任务相关特征, 但只考虑了特征的强度, 缺少特征丰富性的度量, 无法区分学到重复任务相关特征的卷积核. 未来我们将针对此问题进行优化, 以便更准确精细地剪枝.

参考文献

1 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444

2 Huang G, Liu Z, van der Maaten L, et al. Densely connected convolutional networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

3 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556

4 Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012. 1097–1105

5 Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1–9

6 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778

7 Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014. 580–587

8 Ren S, He K, Girshick R, et al. Faster R-CNN: towards realtime object detection with region proposal networks. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 91–99

9 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015. 3431–3440

10 Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell, 2018, 40: 834–848

11 Li Y C, Lin S H, Zhang B C, et al. Exploiting kernel sparsity and entropy for interpretable CNN compression. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019

12 Wei X S, Luo J H, Wu J, et al. Selective convolutional descriptor aggregation for fine-grained image retrieval. IEEE Trans Image Process, 2017, 26: 2868–2881

13 Wei X S, Zhang C L, Wu J, et al. Unsupervised object discovery and co-localization by deep descriptor transformation. Pattern Recogn, 2019, 88: 113–126

14 Schulz K, Sixt L, Tombari F, et al. Restricting the flow: information bottlenecks for attribution. 2020. ArXiv:2001.00396

15 Zhang B, Yang P B, Sang J T, et al. Convolution network pruning based on the evaluation of the importance of

characteristic attributions. Sci Sin Inform, 2021, 51: 13–26 [张彪, 杨朋波, 桑基韬, 等. 基于特征归因重要度评价的卷积网络剪枝. 中国科学: 信息科学, 2021, 51: 13–26]
16 Molchanov P, Tyree S, Karras T, et al. Pruning convolutional neural networks for resource efficient inference. 2016. ArXiv:1611.06440
17 Ding X H, Zhou X X, Guo Y C, et al. Global sparse momentum SGD for pruning very deep neural networks. In: Proceedings of Advances in Neural Information Processing Systems. 2019. 6382–6394
18 Nilsback M E, Zisserman A. Automated flower classification over a large number of classes. In: Proceedings of the 6th Indian Conference on Computer Vision, Graphics & Image Processing, 2008. 722–729
19 Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. University of Toronto Technical Report, 2009
20 Deng J, Dong W, Socher R, et al. Imagenet: a large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255
21 Ji R R, Lin S H, Chao F, et al. A review of deep neural network compression and acceleration. J Comput Res Dev, 2018, 55: 1871–1888
22 Denton E L, Zaremba W, Bruna J, et al. Exploiting linear structure within convolutional networks for efficient evaluation. In: Proceedings of Advances in Neural Information Processing Systems, 2014. 1269–1277
23 Courbariaux M, Hubara I, Soudry D, et al. Binarized neural networks: training deep neural networks with weights and activations constrained to +l or −1. 2016. arXiv:1602.02830
24 Buciluǎ C, Caruana R, Niculescu-Mizil A. Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006. 535–541
25 Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. 2016. ArXiv:1602.07360
26 Howard A G, Zhu M L, Chen B, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications. 2017. ArXiv:1704.04861
27 Ding X H, Ding G G, Guo Y C, et al. Centripetal SGD for pruning very deep convolutional networks with complicated structure. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 4943–4953
28 Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 1135–1143
29 Han S, Mao H, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. 2015. ArXiv:1510.00149
30 Yeom S K, Seegerer P, Lapuschkin S, et al. Pruning by explaining: a novel criterion for deep neural network pruning. Pattern Recogn, 2021, 115: 107899
31 Fu R G, Hu Q Y, Dong X H, et al. Axiom-based grad-CAM: towards accurate visualization and explanation of CNNs. 2020. ArXiv:2008.02312

Gradient flow pruning based on the evaluation of the importance of characteristic attribution and Taylor-guidance

Chang GAO[1,2], Jiaqi WANG[1,2], Liping JING[1,2*] & Jian YU[1,2]

1. *School of Computer and Information Technology, Beijing Jiaotong University, Beijing* 100044*, China*;
2. *Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing* 100044*, China*
* Corresponding author. E-mail: lpjing@bjtu.edu.cn

Abstract Compressing convolutional neural networks (CNNs) have received ever-increasing research focus. In this paper, we attribute the redundancy of the model to the fact that some filters have not learned features related to the task. In order to remove this part of redundancy, based on the pruning framework, we propose a novel evaluation criterion of the importance from two aspects: the degree of features related to the task learned by filter and the influence of removing filter on the loss function. The proposed evaluation criteria are used to quantify the importance of filters and to guide the pruning of filters. In addition, the gradient flow strategy is introduced into filter pruning. In each training iteration, filters are divided into two categories according to importance and compression ratio that will be updated using different rules. For the redundant filters, we perform the update with no gradients derived from the objective function but only the ordinary weight decay to penalize their values. We comprehensively evaluate the classification accuracy, compression, speedup and retention degree of features related to the task of the proposed method on VGGNet and ResNet. Our method demonstrates superior performance gains over previous ones and superior in the case of high compression ratio.

Keywords convolutional neural networks, compressing, pruning, features related to task, gradient flow