中国科学: 信息科学 2021 年 第 51 卷 第 1 期: 13–26
SCIENTIA SINICA Informationis
论文

基于特征归因重要度评价的卷积网络剪枝

张彪[1,2], 杨朋波[1,2], 桑基韬[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. 交通数据分析与挖掘北京市重点实验室 (北京交通大学), 北京 100044
* 通信作者. E-mail: jianyu@bjtu.edu.cn

收稿日期: 2020–06–19; 接受日期: 2020–08–05; 网络出版日期: 2020–12–29

国家重点研发计划 (批准号: 2017YFC1703506) 和国家自然科学基金重点项目 (批准号: 61632004, 61832002, 61672518) 资助

摘要 近几年, 深度模型在诸多任务中取得了巨大成功, 但是深度模型需要大量的存储和计算资源实现精确决策, 研究者为了将深度模型应用到资源受限的终端设备中, 设计了模型压缩的优化策略来降低模型占存和计算量. 本文基于剪枝压缩框架, 从卷积核重要度评价的角度提出了两种模型剪枝算法. (1) 由于每个卷积核都可以学习到其独有特征信息, 因此本文提出了一种归因评价机制用于评价卷积核所学特征与因果特征的相关度, 将模型中与因果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, 称此算法为归因剪枝. (2) 第 2 种剪枝算法基于迭代优化剪枝框架, 采用卷积通道和梯度中正相关特征评价相应卷积核重要度, 以便于提高剪枝冗余卷积核的精准度, 称为 Taylor-guided 剪枝算法. 本文在 VGGNet 和 ResNet 两种网络架构上进行实验验证, 结果表明: 归因剪枝算法可以极大地保留原模型的归因特征; 并且两种剪枝算法能够取得比当前主流剪枝算法更优异的压缩效果.

关键词 深度学习, 网络剪枝, 归因, 压缩, Taylor 展开

1 引言

卷积神经网络 [1] 在物体识别 [2~4]、目标检测 [5,6]、图像分割 [7,8] 等计算机视觉任务中取得了前所未有的成功. 然而深度学习 [9] 模型的弊端也较为普遍, 比如拥有庞大参数量的模型在训练或决策过程中, 需要大量的电力和算力支撑网络模型的实现, 这使得深度模型在终端部署中难以达到理想状态. 为了降低网络模型的空间占用率以及提升网络模型的运行速度, 网络模型的压缩算法应运而生, 它为深度网络模型在工业应用中的推广做出了重要贡献.

近几年在网络模型压缩问题上的主要研究范畴分为 5 类 [10], 包括参数剪枝 [11,12]、参数量化 [13]、低秩分解 [14]、知识蒸馏 [15] 和紧凑卷积核的设计 [16,17]. 在以上算法中, 参数剪枝算法由于计算方便

引用格式: 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, doi: 10.1360/SSI-2020-0186
Zhang B, Yang P B, Sang J T, et al. Convolution network pruning based on the evaluation of the importance of characteristic attributions (in Chinese). Sci Sin Inform, 2021, 51: 13–26, doi: 10.1360/SSI-2020-0186

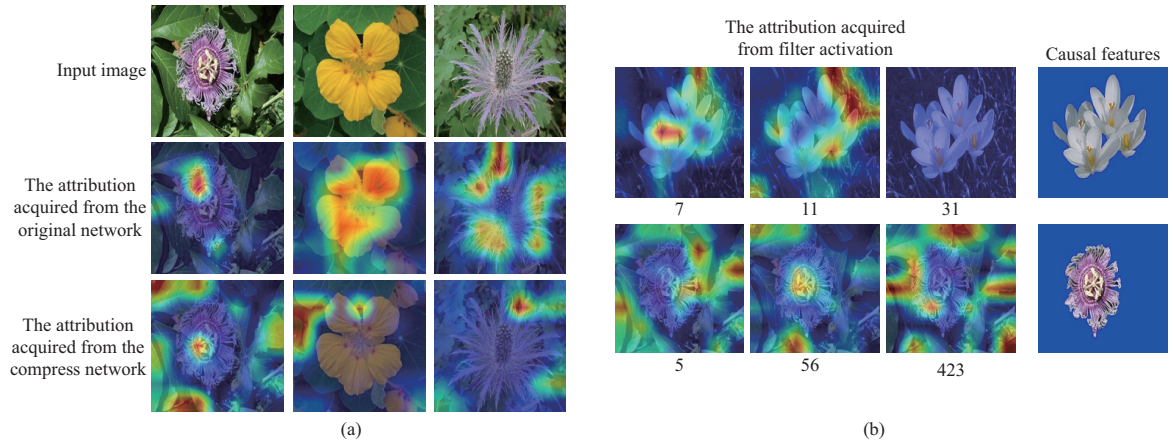

图 1　(网络版彩图) (a) 模型的归因特征; (b) 卷积核的归因特征

Figure 1　(Color online) (a) Attributional characteristics of the model; (b) attribution characteristics of the filter

且压缩效果显著, 在近几年被越来越多的研究人员所采纳. 参数剪枝算法直接将网络模型中间参数或者卷积核去除, 从而减少模型参数量以达到模型压缩、加速的目的. 参数剪枝在实现方式上可以将其分为非结构化参数剪枝 [12] 和结构化参数剪枝 [11]. 其中非结构化参数剪枝将模型中独立的参数值进行去除, 具体采用权重置零的方式, 以便于获得高稀疏度的网络模型, 而卷积核的数量不变, 要加速此种模型需要特殊的硬件和软件支持. 结构化参数剪枝是将整个卷积核以及相应的连接进行去除, 然后通过简单的调优操作使得模型达到最优状态, 如此的处理方式降低了压缩模型对硬件的要求, 在实际操作过程中更易实现.

本文采用结构化剪枝框架对网络模型进行压缩, 从而达到参数和浮点运算次数 (floating point of operations, FLOPs) 减少的目的. 卷积核的重要度评价是结构化剪枝算法的难点和重点; 卷积核的重要度评价结果将决定哪些卷积核需要进行剪枝, 不同卷积核的剪枝会对压缩模型的决策产生不同的影响. 在模型剪枝过程中, 现存的网络模型压缩算法往往只将模型的压缩比、运算速度和精度作为压缩模型评价标准. 本文希望压缩模型不仅能够保证目标任务的识别精度, 同时压缩模型在决策过程中还能够保留原始模型的归因 [18] 特征. 针对特定任务, 本文将图像特征分为与目标物体相关的因果特征和与目标物体无关的背景特征; 同时将模型学到的与决策相关的特征称为归因特征; 我们希望模型在决策时的归因特征与因果特征有较大的交并比 (intersection over union, IOU).

在评价网络模型决策判据时, 利用 Grad-CAM [19] (gradients class activation mapping) 计算 VGG-16 模型的归因特征, 如图 1(a) 所示, 它展示了压缩模型 (采用文献 [20] 中的 Taylor 剪枝算法进行压缩) 与原始模型在识别图片时归因特征图的对比结果. 从图中可以看出原始网络模型识别相应的图片时, 根据具有明显判别能力的因果特征进行决策; 而压缩模型的分类判据弱化为背景特征, 这表明该剪枝算法导致压缩模型的归因特征较大地偏离了原模型的归因特征. 导致归因特征偏离的原因可能是在剪枝过程中对决策有重要贡献的卷积核被移除. 为了避免这种情况发生, 我们需要在剪枝优化的过程中考虑卷积核对决策的贡献度这一指标.

评价卷积核在决策过程中的贡献度时, 采用 Grad-CAM 获得相应卷积核学习到的图像特征. 图 1(b) 展示了激活 VGG-16 网络最后一层卷积核的归因特征, 归因图下方数字代表卷积核编号. 根据卷积核归因图可以发现网络中的每个卷积核均学习到了自己所独有的特征信息; 剪枝去除相应卷积核, 该卷积核可以拟合的特征也会随之从网络中被删除, 所以不同的卷积核剪枝策略将对网络模型决策时的归

因特征产生不同的影响. 为了最大程度地保留原模型在决策时的归因特征, 剪枝算法需要保留在决策过程中具有因果特征的卷积核. 本文设计卷积核的归因评价机制, 以寻找能够拟合因果特征的卷积核. 在实现时本文通过剪枝去除与因果特征相关度低的卷积核, 以达到模型压缩的目的, 同时也能够最大力度地保留原始模型的归因特征. 本文将上述模型剪枝算法称为归因剪枝.

Springenberg 等 [21] 证明了与任务正相关的卷积特征更能准确表达卷积核的真实含义. 本文基于 Molchanov 等 [20,22] 提出的优化剪枝框架, 即根据特征图的 Taylor 一阶展开式计算相应卷积核的重要度, 提出第 2 种剪枝算法. 该算法在具体实现过程中综合正相关特征表达卷积核真实含义和 Taylor 一阶展开式评价卷积核重要度这两点, 它只采取正相关特征信息进行卷积核的评价, 这样的评价方式将会使得卷积核重要度评价结果更为精确. 基于上述对优化剪枝算法进行改进的结论, 本文将其命名为 Taylor-guided 剪枝算法.

本文在评价压缩模型的过程中, 采取了常用的模型压缩评价标准: 模型识别精度 (top-1)、模型的参数量以及 FLOPs, 同时对比并分析了归因剪枝算法所压缩模型和原模型的归因特征. 将本文所提出的算法在 VGGNet [23] 和 ResNet [4] 模型上进行实验验证, 采用的数据集为 flower-102 [24] 和 cifar-10 [25].

本文的贡献主要有两个方面:

(1) 利用归因机制, 计算出每个卷积核在整个数据集上的平均归因重要度, 根据归因重要度对模型进行剪枝处理, 将此算法称为归因剪枝算法; 该算法在剪枝过程中能够有效地保留原模型中的归因特征. 据我们了解, 本文的研究是第一次把归因机制引入到剪枝过程中的工作.

(2) 由于 ReLU 激活函数的存在, 模型特征图在前向传递过程中总是以正相关特征进行传播, 因此本文提出只采用正相关信息评价卷积核重要度的计算方式, 此时对卷积核的重要度评价结果更为精准, 我们将其称为 Taylor-guided 剪枝算法.

2 相关工作

LeCun 等 [26] 和 Hassibi 等 [27] 提出最优脑损伤压缩算法, 该算法采用网络中参数的二阶偏导值评价参数的重要度, 其中文献 [26] 是模型压缩技术的开山之作. 剪枝作为压缩的主流算法之一, 分为非结构化剪枝和结构化剪枝, 其中非结构化剪枝主流文章有: Han 等 [12,28] 迭代地设置阈值以去除小于阈值的参数; Guo 等 [29] 提出 connection splicing 算法, 为了防止在迭代剪枝过程中, 前期不正确的剪枝方式而影响压缩模型的精度, 从而动态地对网络进行剪枝. 而结构化剪枝的研究在近几年才得以发展, 从对卷积核的重要度评价方式上可以将结构化剪枝算法分为两种, 其中一种是根据卷积核输出特征图得到卷积核的重要度, 称为数据驱动的剪枝算法; 另一种直接根据卷积核参数值给出卷积核的重要度, 称为独立于数据的剪枝算法 [30].

数据驱动的剪枝算法有: Hu 等 [31] 提出了 APoZ 算法, 用于计算卷积核输出特征图 0 值的数量, 此算法认为输出特征图的 0 值越多其重要性越低, 然后将重要度低的特征图所对应的卷积核进行剪枝. Lin 等 [32] 提出动态剪枝算法, 在实现过程中可以根据实际剪枝情况, 对网络做细致的调节. Lin 等 [33] 提出 HRank 算法, 该算法采用特征图的秩表达相应卷积核的重要度. Wang 等 [34] 对正向传播中的特征图进行子空间聚类, 同时根据聚类结果和已经设置的条件去除相应的卷积核. Lin 等 [35] 使用生成对抗的思想提取面向特定数据集且性能最优的子网络. Gao 等 [36] 提出特征增强/抑制算法, 用于对运行时的显著性卷积核进行放大并跳过不重要的卷积核. Huang 等 [37] 提出稀疏结构化选择 (sparse structure selection, SSS) 算法, 在相应的网络框架中引入稀疏正则化, 从而将压缩转换为稀疏正则化的

优化问题. Molchanov 等 [20,22] 结合了卷积核的正向激活值以及反向梯度值, 用两值之积共同评价卷积核的重要性. 文献 [22] 基于文献 [20] 的思想, 将剪枝算法推广到 ResNet 和 DenseNet 含跳跃连接的网络模型中.

独立于数据的剪枝算法有: Li 等 [11] 提出用卷积核 L1 范数表达其重要度, 通过剪枝去除 L1 范数较小的卷积核以及对应的连接达到压缩的目的. 类似地, He 等 [38] 采用卷积核的 L2 范数表达其重要度. Liu 等 [39] 利用 Batch Normalization 层的缩放因子评价卷积核的重要度. Zhao 等 [40] 首先在训练阶段对 Batch Normalization 层的缩放因子施加稀疏约束, 并进一步采用缩放因子对卷积核进行重要度评价. He 等 [30] 提出使用卷积核几何中位数评价其重要度的算法, 该算法旨在找到每层中卷积核所表达的共同信息, 以便剪枝去除冗余信息. Zhuo 等 [41] 对网络模型中的卷积核进行谱聚类操作, 并从中找出不重要的卷积核得以剪枝.

本文的归因剪枝算法不同于已存在的所有剪枝算法, 它引入像素级标签去引导网络模型的剪枝, 以保留模型归因特征. 文献 [20] 中剪枝算法未考虑网络在前向激活过程中过滤负相关特征这一特性, 而是采用激活之前的特征图评价卷积核的重要度. 不同于该算法, 本文中 Taylor-guided 剪枝算法只采用正相关特征进行卷积核的重要度评价.

3 基于卷积核重要度评价的卷积网络剪枝

卷积核重要度的评价作为模型剪枝过程中的重点工作, 本节将详细介绍本文所涉及的两种剪枝算法, 它们从不同角度评价卷积核的重要度. 3.2 小节总结了结构化剪枝过程的通用计算过程, 然后 3.3 和 3.4 小节分别从卷积核归因和对优化剪枝算法改进的角度, 详细介绍本文提出的归因剪枝算法和 Taylor-guided 剪枝算法, 同时通过数学公式对两种剪枝算法的计算过程进行详细说明.

3.1 符号说明

假定一个已经训练收敛的模型拥有 M 个卷积层, 用 $W_i = \{w_1^i, w_2^i, w_3^i, \ldots, w_{n_i}^i\} \in \mathbb{R}^{n_i \times n_{i-1} \times k_i \times k_i}$ 表示第 i 层卷积参数, 其中 $w_j^i \in \mathbb{R}^{n_{i-1} \times k_i \times k_i}$ 表示第 i 层中第 j 个卷积核的参数, n_i 表示第 i 层的卷积核数量, k_i 表示卷积核的大小. 整个模型的卷积核数用 N 表示, $N = \sum_{i=1}^{M} n_i$. 用 $O_i = \{o_1^i, o_2^i, o_3^i, \ldots, o_{n_i}^i\} \in \mathbb{R}^{n_i \times h_i \times g_i}$ 表示第 i 层卷积结果, 第 i 层中第 j 个卷积核的输出为 o_j^i, h_i 和 g_i 表示特征图的高和宽. 令 D 为数据集, $L(D, o_j^i)$ 表示在训练数据集 D 上且模型含有卷积核 w_j^i 时的损失.

3.2 网络剪枝

网络压缩要求模型性能的损失在可接受范围内, 尽最大力度压缩模型. 目前常用描述网络性能的指标为模型识别精度, 这里设 P 为模型的精度, 利用 $P_{\text{ori}} - P_{\text{com}} \leqslant \varepsilon$ 限定剪枝停止边界, 其中 ε 是可设定的边界值, P_{ori} 和 P_{com} 分别表示原始网络模型的精度和压缩后网络模型的精度.

$$\begin{aligned} &\min_{\delta} \ L(\delta_j^i \cdot w_j^i) \\ &\text{s.t.} \sum_{i=1}^{M} \sum_{j=1}^{n_i} \delta_j^i = \beta N. \end{aligned} \tag{1}$$

模型的剪枝过程可以转化为式 (1) 表示的优化过程, $L(\delta_j^i \cdot w_j^i)$ 表示压缩模型的损失. 式 (1) 中的 δ_j^i 与网络权重参数 w_j^i 一一对应, $\delta_i = \{\delta_1^i, \delta_2^i, \delta_3^i, \ldots, \delta_{n_i}^i\}$, $\delta_j^i \in \{0, 1\}$, δ_j^i 表示相应的卷积核剪枝与

否. $\beta=\frac{\sum_{i=1}^{M}\sum_{j=1}^{n_i}\delta_j^i}{N}$ 表示要保留卷积核的比例, $0<\beta<1$, β 的设定根据式 (2) 得到.

$$\min\ \beta \quad \text{s.t. } P_{\text{ori}}-P_{\text{com}}\leqslant\varepsilon, \tag{2}$$

其中, ε 表示所要求的精度误差限. 式 (2) 表示模型在压缩过程中, 当精度损失大于 ε 时, 压缩将停止; 精度的损失在 ε 范围内时, 保留最少数量的卷积核数. 由于模型至少需要一定数量的卷积核完整地拟合整个数据集的特征, 所以在实际计算中为了避免模型被过度压缩, 需设置 β 的最小取值 $\beta_{\min}$.

上述过程中的重点计算参数为 δ, 本文将其称为卷积核掩码. 卷积核剪枝便是从卷积层 W 中移除一系列 w, 本文中每个卷积核的剪枝与否由 δ 表示; 参数 δ 的值需要根据卷积核重要度进行设定, 重要度低的卷积核对应的 δ 置为 0, 意味着相应卷积核将被去除, 反之, 置为 1. 本文在 3.3 和 3.4 小节提供两种计算 δ 的算法.

3.3 归因剪枝

我们在模型归因剪枝过程中, 采用归因评价机制计算卷积核所对应的归因特征, 并根据该归因特征计算相应卷积核的归因重要度, 合理地给出 δ 值, 用于表示相对应的卷积核剪枝与否. 在给定预训练模型和训练数据集的情况下, 归因剪枝算法在计算卷积核的重要性时采用如下计算步骤.

$$\alpha_{i,j}^{c}=\frac{1}{h\times g}\sum_{q=1}^{h}\sum_{r=1}^{g}\left(\frac{\partial c}{\partial o_j^i}\right)_{q,r}. \tag{3}$$

式 (3) 中首先计算类别 c 的置信度对第 i 层第 j 个卷积核输出特征图的梯度, 这里 h 和 g 分别表示输出特征图的高和宽; 然后对梯度图采用平均池化操作以得到第 i 层中第 j 卷积核对类别 c 的贡献程度 $\alpha_{i,j}^{c}$, 称之为归因因子.

结合归因因子和相应卷积核的输出特征图, 便可以得到第 i 层第 j 个卷积核所学特征在原图中的归因区域, 即

$$\text{CAM}_{i,j}=\text{ReLU}(\alpha_{i,j}^{c}\cdot o_j^i). \tag{4}$$

$\text{CAM}_{i,j}$ 可视化结果为图 2 中的特征归因图. 因为我们只需获得卷积核归因重要度排序结果, 所以归因特征和因果特征的 IOU 计算可以转换为

$$\text{valuate}=\frac{1}{\text{num}}\sum_{D}\sum_{g}\sum_{h}\left(\text{CAM}_{i,j}\right)_{g,h}\cdot\text{label}_{g,h}, \tag{5}$$

其中 $\text{label}_{g,h}=\text{sign}(\text{ReLU}(\text{label}))$, label 为图像的像素级标签. 首先对 label 进行下采样, 使其能够与相应特征图进行点乘计算, 然后使用 sign 函数和 ReLU 函数将其转换为只含 $0,1$ 值的二值矩阵 $\text{label}_{g,h}$, 其中因果特征区域为 1, 背景特征区域为 0. num 为数据集中图片的数量. 式 (5) 首先将第 i 层第 j 个卷积核的归因区域图 $\text{CAM}_{i,j}$ 和图像对应的 $\text{label}_{g,h}$ 进行点乘, 并累加点乘结果作为卷积核在单张图像上的归因重要度, 最后针对整个数据集去计算每一个卷积核的平均归因评价结果, 即式 (5) 中的 valuate. 此算法对于式 (1) 中的 δ 的取值可以表述为

$$\delta=\begin{cases}1, & \text{valuate}\geqslant T,\\ 0, & \text{otherwise},\end{cases} \tag{6}$$

其中 T 为阈值, $T=\text{valuate}_{\tau}$ 表示在对 valuate 由小到大排序后取第 τ 个值作为阈值, τ 表示每次迭代过程中需去除的卷积核数量, 当所求卷积核的归因评价值大于或等于阈值 T 时, 则该卷积核所对应的掩码 δ 置为 1, 反之置为 0. 具体计算步骤如图 2 所示.

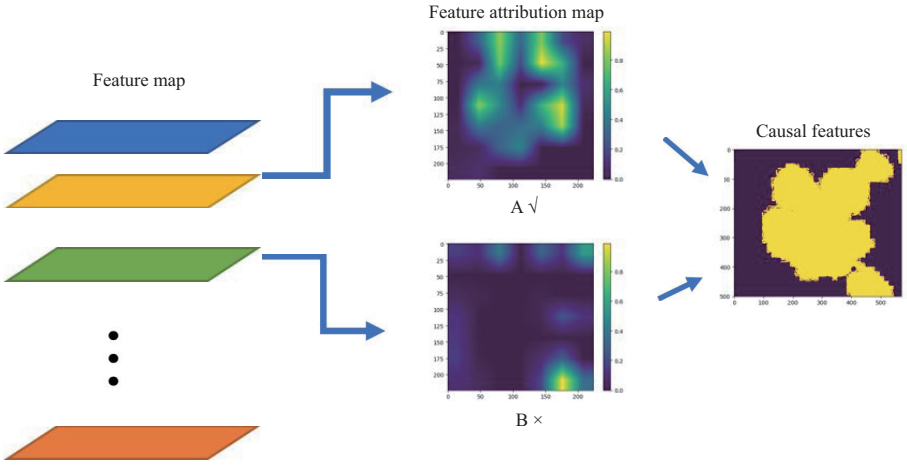

图 2　(网络版彩图) 归因剪枝算法图示

Figure 2　(Color online) Illustration of attribution pruning method

图 2 为归因剪枝算法示意图, 该算法首先计算每一个卷积核对应的特征归因图, 根据卷积核所归因的特征并结合像素级标签, 决定相应卷积核的 δ 值, 用 δ 表达该卷积核的剪枝与否. 如图中特征归因图所示, A 的归因特征与因果特征的 IOU 值较高, 而 B 与因果特征的 IOU 值较低, 于是将剪枝去除 B 对应的卷积核. 图中的 "$\surd$" 表示该卷积核保留, "$\times$" 意味着该卷积核将被去除.

本小节从归因的角度计算卷积核重要度, 因为在剪枝过程中只保留归因特征与因果特征相关性高的卷积核, 于是该剪枝算法既可以增加模型的压缩比例, 提高模型的识别精度, 同时也可以最大程度保留原模型的归因特征.

3.4　Taylor-guided 剪枝

值 δ 需要根据相应卷积核重要度进行设定, 于是可以将 δ 值的设置转换为卷积核重要度的计算. 根据消融实验的设计原理, 卷积核的重要度可以由式 (7) 计算得到, 即卷积核重要度等于单独的去除该卷积核后的损失与完整网络的损失之差. 若被删除的卷积核在网络模型中有着重要的作用, 则得到的损失之差越大, 反之越小.

$$|\triangle L(o_j^i)| = |L(D, o_j^i = 0) - L(D, o_j^i)|, \tag{7}$$

其中右侧的第 1 项为移除卷积核 w_j^i 后的网络损失, 第 2 项为完整网络的损失. 对 $L(D, o_j^i = 0)$ 使用一阶 Taylor 展开如

$$L(D, o_j^i = 0) = L(D, o_j^i) - \frac{\partial c}{\partial o_j^i} o_j^i + R_1(o_j^i = 0), \tag{8}$$

其中 $R_1(o_j^i = 0)$ 表示为

$$R_1(o_j^i = 0) = \frac{\partial^2 L}{\partial ({o_j^i}^2 = \xi)} \frac{{o_j^i}^2}{2}, \tag{9}$$

其为高阶函数, 该项可以忽略. 经过化简可以得到卷积核 w_j^i 的重要度评价值, 即

$$|\triangle L(o_j^i)| = \left| \frac{\partial c}{\partial o_j^i} o_j^i \right|. \tag{10}$$

根据式 (10) 可以发现, 卷积核的重要度可以由正向激活值和反向梯度之积表示. 在模型的计算过程中每层特征图中正值越大表示相应特征与目标结果越相关, 而负值意味着负相关. 网络模型的卷积过程可以表示为 $O_i = f(I_{i-1}, W_i)$, 其中 I_{i-1} 表示输入到第 i 层的特征图; 而网络模型的第 i 层在执行完卷积操作后, 最终输入到 $i+1$ 层的值为 $I_i = \mathrm{ReLU}(O_i)$. 从 I_i 的计算方式可以看出, 隐藏层输出特征图只包含正相关信息, 负值表达的特征成为冗余信息在计算过程中被过滤除去. 于是在评价网络内部卷积核重要度时, 为了保证重要度评价结果的精准性, 我们改进为只利用正相关特征对卷积核进行评价. 于是本小节将一个卷积核的重要度最终设置为

$$|\triangle L(o_j^i)| = \left| \mathrm{ReLU}\left(\frac{\partial c}{\partial o_j^i}\right) \cdot \mathrm{ReLU}(o_j^i) \right|. \tag{11}$$

此算法对于式 (1) 中 δ 的取值可以表述为

$$\delta = \begin{cases} 1, & |\triangle L(o_j^i)| \geqslant T, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

其中 T 为重要度阈值. 首先对卷积核重要度从小到大进行排序, 并设置 $T = |\triangle L(o)|_\tau$, 表示取排序后的第 τ 个值为阈值; 当卷积核重要度大于该阈值时, δ 取 1, 表示保留此卷积核; 反之设置为 0.

本节只采用正相关特征去评价卷积核在决策过程中的重要度, 这样的评价结果更为精准, 这对同时采用正相关特征和负相关特征进行评价的方式起到改善作用.

3.5 算法框架

将归因剪枝算法或者 Taylor-guided 剪枝算法作为卷积核评价算法, 在计算过程中由于每层卷积核的重要度均值随着模型深度的变化而变化, 于是本文采用 L2 范数对重要度进行层级正则化. 设 va_j^i 为第 i 层第 j 个卷积核通过上述两个算法之一求得的重要度, 则正则化公式为

$$va_j^i = \frac{va_j^i}{\sqrt{\sum_j (va_j^i)^2}}. \tag{13}$$

利用给定的数据集和模型按照算法 1 进行剪枝压缩.

Algorithm 1 Pruning algorithm for convolutional neural networks

Input: Datasets D, convergent MODEL, accuracy reduction boundary ε, the number of filters pruned τ during iterative pruning, and the minimum proportion of filters retained $\beta_{\min}$.

$\varphi = 1$ represents the number of filters currently retained/the number of original filters;
while $P_{\mathrm{ori}} - P_{\mathrm{com}} \leqslant \varepsilon$ and $\varphi \geqslant \beta_{\min}$ **do**
  Dataset D is input into MODEL and computed forward;
  Evaluate the importance of each filter in MODEL by attribution or Taylor-guided pruning method;
  The importance is regularized by L2 norm;
  Sort from small to large by regularization result and set $T = \mathrm{valuate}_\tau$ or $T = |\triangle L(o)|_\tau$;
  Update δ according to T;
  Use δ to prune corresponding filters, and finetune MODEL;
  $\varphi = \varphi - \frac{\tau}{N}$ and calculate the accuracy of compression model P_{com}.
end while

Output: The compressed and finetuned MODEL.

表 1 VGG-16 基于 flower-102 数据集的剪枝结果

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

Model	Top-1 (%)	FLOPs (PR (%))	Parameters (PR (%))
VGG-16	76.86	1.56×10^{10} (0.0)	1.35×10^{8} (0.0)
Attribution (low compression ratio)	76.62	3.86×10^{9} (75.26)	4.42×10^{7} (67.26)
Taylor-guided (low compression ratio)	75.76	4.40×10^{9} (71.79)	1.09×10^{8} (19.26)
L1 [11]	74.23	2.03×10^{9} (86.99)	4.20×10^{7} (68.89)
Taylor [20]	71.00	1.07×10^{9} (93.14)	2.66×10^{7} (80.30)
Taylor-guided (high compression ratio)	72.36	1.11×10^{9} (92.88)	3.36×10^{7} (75.11)
Attribution (high compression ratio)	74.90	5.55×10^{8} (96.44)	2.34×10^{7} (83.04)

4 实验

为了证明本文所提出算法的有效性, 本文在 flower-102 数据集上对两种剪枝算法进行实验验证, 另外采用 cifar-10 数据集单独验证 Taylor-guided 剪枝算法的压缩有效性. 我们将剪枝算法应用于不同的网络构架中, 包括 VGGNet, ResNet. 4.1 小节给出了本文实验中具体超参数的设置准则; 4.2 小节对具体实验结果进行解释说明, 除此之外, 还展示了模型在压缩前后, cifar-10 中每一个类别的 top-1 变化情况.

4.1 实验说明

本文在实验中为了保证剪枝后的模型仍然可以正常工作, 需要根据具体任务限定 ε, 即限定模型压缩后精度损失的范围; 同时为了防止模型被过度压缩, 本文还将设置卷积核最小保留比 $\beta_{\min}$ 的取值, 在剪枝过程中 β 不能低于此值以保证模型的稳定性 [20].

评价标准. 为了评价压缩后模型的工作状态, 本文采用常用的评价标准 —— 参数量 (parameters) 和 FLOPs 及其压缩比率 (pruning rate, PR), 计算模型的参数量以及计算量. 另外我们还用 top-1 精度评估被压缩模型与原始模型之间的差距, 同时还对比了原模型和归因剪枝压缩模型的归因特征.

配置. 本文利用 PyTorch [42] 框架实现文中所涉及的实验. 在原始模型训练以及压缩模型微调阶段, 采用随机梯度下降 (stochastic gradient descent, SGD) 算法进行网络的调优; 在归因剪枝算法以及 Taylor-guided 剪枝算法实验中将网络的学习率设置为 0.01, 动量为 0.9; 在训练阶段以及压缩模型微调阶段, 图片均以 64 张为一个批次. 在每一次剪枝完成后, 采用 10 次迭代的方式对压缩模型调优. 本文所有实验在 2 块 NIVIDIA TITAN Xp GPUs 上完成.

4.2 结果和分析

4.2.1 基于 flower-102 的实验结果

本小节在 flower-102 数据集上分别对 VGG 和 ResNet 两种网络模型进行压缩验证, 并结合几种常用压缩算法的实验结果, 对比它们所生成模型性能的优劣.

VGG-16. 本部分实验中, 设置 $\varepsilon = 5\%$, 同时令 $\beta_{\min}$=10%. 表 1 对比了几种剪枝算法的压缩性能, 包括 Li 等 [11] 提出的 L1 范数剪枝算法和 Molchanov 等 [20] 提出的 Taylor 剪枝算法. 相比于 L1 范数剪枝算法和 Taylor 剪枝算法, 从表中可以看出本文提出的归因剪枝算法和 Taylor-guided 剪枝算法在高压缩比的设置下仍然可以达到较优的 top-1 值. 尤其在对比 L1 范数剪枝算法和归因剪枝算法

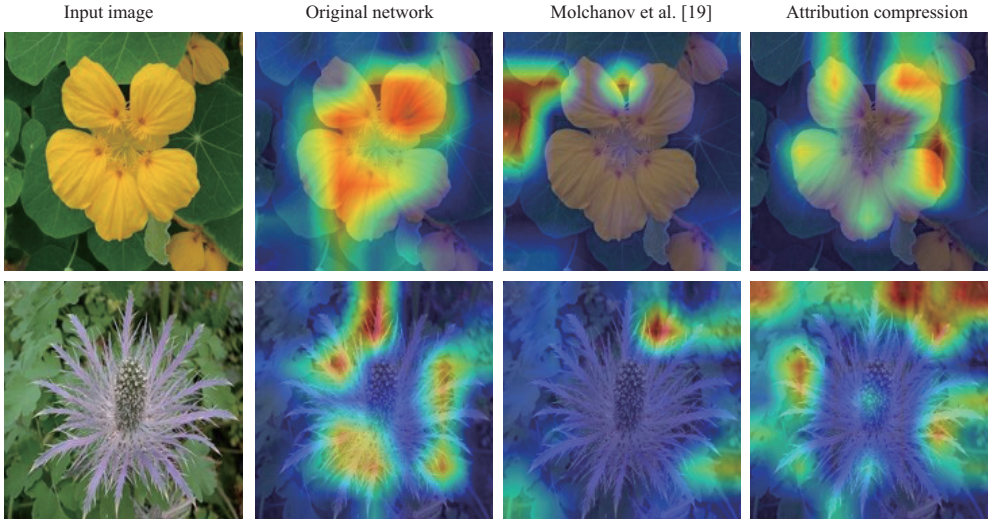

图 3 (网络版彩图) 原模型和压缩模型决策时的归因特征可视化对比图

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

表 2 ResNet-18/ResNet-50 基于 flower-102 数据集的剪枝结果

Table 2 Pruning results of ResNet-18/ResNet-50 on flower-102

Model	Top-1 (%)	FLOPs (PR (%))	Parameters (PR (%))
ResNet-18/ResNet-50	75.39/85.68	1.88×10^9 (0.0)/6.59×10^9 (0.0)	1.19×10^7 (0.0)/4.02×10^7 (0.0)
Taylor [20]	70.62/81.67	7.06×10^8 (62.45)/2.27×10^9 (65.55)	2.46×10^6 (79.33)/1.29×10^7 (67.91)
Taylor-guided	73.86/82.96	7.51×10^8 (60.05)/2.30×10^9 (65.10)	2.07×10^6 (82.61)/8.73×10^6 (78.28)
Attribution	74.53/82.95	6.03×10^8 (67.93)/2.11×10^9 (67.98)	2.58×10^6 (78.32)/9.48×10^6 (76.42)

时, 我们可以发现在压缩模型达到相近的 top-1 时, 归因剪枝算法可以获得更低 FLOPs 和参数量的压缩模型 (96.44% vs. 86.99%, 83.04% vs. 68.89%). 此外对比归因剪枝算法和 Taylor 剪枝算法, 我们可以看出归因剪枝算法对模型 FLOPs 和参数量的压缩比 (96.44% vs. 93.14%, 83.04% vs. 80.30%) 要高于 Taylor 剪枝算法的压缩比. 采用归因剪枝算法和 Taylor-guided 剪枝算法对模型处理时, 在 FLOPs 和参数量的减少量分别为 75.26%, 71.79% 和 67.26%, 19.26% 时, 模型的 top-1 值可以达到与原始模型相近水平. 以上结果充分地说明归因剪枝算法和 Taylor-guided 剪枝算法可以应用于 VGGNet 且表现突出, 证明了两种剪枝算法在平铺式卷积模型压缩中的优异表现.

本部分分别利用归因剪枝和 Molchanov 等 [20] 提出的 Taylor 剪枝算法将 VGG-16 网络卷积核数量压缩到原模型数量的 20%, 并对比两压缩模型的归因特征, 结果如图 3 所示. 从图中对比可以发现, 采用归因剪枝算法压缩的模型其归因特征保留程度要优于 Taylor 剪枝算法压缩的模型. 这也证明了本文的归因剪枝算法在保留原模型归因特征时的优越性.

ResNet-18. 本部分采用基于 ResNet-18 结构的变体 (详细网络结构见附录 A) 进行验证. 实验中, 设置 $\beta_{\min} = 33\%$, 且令 $\varepsilon = 5\%$; 实验结果如表 2 所示. 通过 Taylor-guided 剪枝算法与 Molchanov 等 [20] 提出的 Taylor 剪枝算法结果的对比可以发现, Taylor-guided 剪枝算法生成的网络模型占存更小 (减少的参数量分别为 82.61% vs. 79.33%), 同时它的 top-1 要高于后者的 top-1 (73.86% vs. 70.62%). 另外利用归因剪枝算法进行模型剪枝时, 其 top-1 损失为 0.86%, 与 Taylor 剪枝算法相比, 其 FLOPs

表 3 基于 cifar-10 的 VGG 模型压缩结果

Table 3 Pruning results of VGGNet on cifar-10

Model	Top-1 (%)	FLOPs (PR (%))	Parameters (PR (%))
VGG-16	93.96	1.56×10^{10} (0.0)	1.35×10^{8} (0.0)
L1 [11]	93.40	2.06×10^{8} (34.39)	5.04×10^{6} (65.71)
SSS [37]	93.02	1.83×10^{8} (41.72)	3.95×10^{6} (73.13)
Zhao et al. [40]	93.18	1.90×10^{8} (39.49)	3.92×10^{6} (73.33)
Taylor [20]	93.20	1.28×10^{8} (59.24)	4.20×10^{6} (71.43)
Taylor-guided	93.21	6.50×10^{7} (79.30)	2.05×10^{6} (86.05)

压缩率更高 (97.93% vs. 62.45%). 根据此实验可以证明本文提出的算法在 ResNet 网络架构压缩过程中, 能够以较大压缩率压缩模型同时可以保证模型拥有较高的 top-1 值. 此实验表明本文算法可以在含有跳跃连接的架构中成功运用.

ResNet-50. 本部分采用 ResNet-50 模型进行验证. 实验中, 设置 $\beta_{\min} = 33\%$, $\varepsilon = 5\%$; 实验结果如表 2 所示. 相比于 Taylor 剪枝算法, 归因剪枝与 Taylor-guided 剪枝在保持较高的 top-1 (82.95 vs. 82.96 vs. 81.67) 时, 可以取得更高的参数压缩比 (参数压缩比: 76.42% vs. 78.28% vs. 67.91%). 根据 ResNet-50 的压缩结果, 我们可以发现归因剪枝和 Taylor-guided 剪枝在较深的模型中同样可以表现出较优的压缩性能.

4.2.2 基于 cifar-10 的实验结果

本部分基于 cifar-10 数据集对 VGG 网络框架进行压缩, 以验证 Taylor-guided 剪枝算法在模型 (小尺度图像作为训练集) 压缩中的有效性. VGG-16 网络基于 cifar-10 的剪枝结果如表 3 所示, 此时 $\beta_{\min} = 10\%$, 令 $\varepsilon = 2\%$. 从表中可以看出 Taylor-guided 剪枝算法在 FLOPs 减少 79.30%、参数压缩 86.05% 的情况下, 仍然可以保持着较高的 top-1 值 (93.21%). 当压缩模型 top-1 接近 93.2% 时, 我们可以发现通过 Taylor-guided 剪枝压缩的模型, 其 FLOPs 和参数量 (79.30%, 86.05%) 的压缩率要优于 L1 剪枝算法 (34.39%, 65.71%) 和 SSS 剪枝算法 (41.72%, 73.13%) 的压缩率; 此外相对于 Zhao 等所提算法和 Taylor 剪枝算法, Taylor-guided 剪枝在高压缩比的情况下 (FLOPs 压缩率对比 79.30% vs. 39.49% vs. 41.72%, 参数量压缩率 86.05% vs. 73.33% vs. 73.13%), 仍然能够获得较高的 top-1 值 (93.21%). 这也表明 Taylor-guided 剪枝在压缩通过小尺度图片训练的网络时, 也可以表现出优越的性能.

Taylor-guided 剪枝算法在完成模型剪枝后, 此时压缩模型的 top-1 为 93.21%, FLOPs 和参数量的减少率分别为 79.30% 和 86.05%. 为了了解剪枝过程对不同类别的影响情况, 我们对 cifar-10 中每一个类别的 top-1 进行展示, 结果如表 4 所示. 从表中可以发现模型压缩过程对每个类别的影响是不同的, 我们可以看到 Deer 类在压缩之后模型的精度有所提高, 而 Car 类别的精度保持不变, 其余类别的识别精度均有不同程度的降低. 以上结果进一步地表明网络中不同卷积核对每个类的贡献程度有较大的差异, 删除一个卷积核可能只对某一类别图片有影响, 即该卷积核可能只包含该类别所独有的特征.

5 总结

本文为了避免在压缩过程中丢弃与任务相关的归因特征, 提出了一种归因剪枝算法; 该算法利用卷积核学习到的特征信息去评价卷积核归因重要度, 然后剪枝去除所学归因特征与因果特征 IOU 较

表 4 Taylor-guided 剪枝与原始模型在 cifar-10 中各类别的精度对比

Table 4 Precision comparison of Taylor-guided pruning and original model in cifar-10

Class	Class of original model top-1 (%)	Class of compress model top-1 (%)
Plane	92.86	89.29
Car	94.00	94.00
Bird	87.34	79.75
Cat	82.19	80.82
Deer	85.45	89.09
Dog	84.75	79.66
Frog	92.86	87.50
Horse	98.44	93.75
Ship	94.83	93.60
Truck	93.59	92.31

低的卷积核, 从而达到模型压缩的目的且最大程度保留了原模型的归因特征. 本文通过对现有压缩算法和归因剪枝算法进行实验对比以及归因可视化分析, 证明了归因剪枝算法在保留原模型归因特征和精度上的有效性. 另外基于特征图 Taylor 一阶展开式去评估相应卷积核重要度的计算方式, 本文只采用与目标任务正相关的通道和梯度特征去评价卷积核重要性, 该算法与主流压缩算法相比, 可以获得高压缩比和高 top-1 的压缩模型, 我们称之为 Taylor-guided 剪枝算法.

我们通过实验将本文提出的算法与目前较为流行的剪枝算法进行比较, 证明了本文提出的算法在降低模型复杂度以及模型容量方面具有优异的表现. 尽管我们的归因剪枝算法可以降低压缩网络归因特征的偏离程度, 但是在细粒度特征的保持上还有较大的提升空间. 未来我们将对此算法进行优化; 并对模型中具体类别特征进行分析, 以便于提取用于特定类别的小尺度网络模型.

参考文献

1 LeCun Y. Generalization and network design strategies. Connectionism Perspective, 1989, 19: 143–155
2 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
3 Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1–9
4 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778
5 Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014. 580–587
6 Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 91–99
7 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 3431–3440
8 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
9 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444
10 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
11 Li H, Kadav A, Durdanovic I, et al. Pruning filters for efficient convnets. 2016. ArXiv: 1608.08710
12 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
13 Chen W, Wilson J, Tyree S, et al. Compressing neural networks with the hashing trick. In: Proceedings of International

Conference on Machine Learning, 2015. 2285–2294
14 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
15 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
16 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
17 Howard A G, Zhu M, Chen B, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications. 2017. ArXiv: 1704.04861
18 Schulz K, Sixt L, Tombari F, et al. Restricting the flow: information bottlenecks for attribution. 2020. ArXiv: 2001.00396
19 Selvaraju R R, Cogswell M, Das A, et al. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 618–626
20 Molchanov P, Tyree S, Karras T, et al. Pruning convolutional neural networks for resource efficient inference. 2016. ArXiv: 1611.06440
21 Springenberg J T, Dosovitskiy A, Brox T, et al. Striving for simplicity: the all convolutional net. 2014. ArXiv: 1412.6806
22 Molchanov P, Mallya A, Tyree S, et al. Importance estimation for neural network pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 11264–11272
23 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv: 1409.1556
24 Nilsback M E, Zisserman A. Automated flower classification over a large number of classes. In: Proceedings of 2008 6th Indian Conference on Computer Vision, Graphics & Image Processing, 2008. 722–729
25 Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. In: Handbook of Systemic Autoimmune Diseases. Technical Report, 2009, 1(4)
26 LeCun Y, Denker J S, Solla S A. Optimal brain damage. In: Proceedings of Advances in Neural Information Processing Systems, 1990. 598–605
27 Hassibi B, Stork D G. Second order derivatives for network pruning: optimal brain surgeon. In: Proceedings of Advances in Neural Information Processing Systems, 1993. 164–171
28 Han S, Mao H, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. 2015. ArXiv: 1510.00149
29 Guo Y, Yao A, Chen Y. Dynamic network surgery for efficient DNNs. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 1379–1387
30 He Y, Liu P, Wang Z, et al. Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 4340–4349
31 Hu H, Peng R, Tai Y W, et al. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures. 2016. ArXiv: 1607.03250
32 Lin S, Ji R, Li Y, et al. Accelerating convolutional networks via global & dynamic filter pruning. In: Proceedings of 27th International Joint Conference on Artificial Intelligence, 2018. 2425–2432
33 Lin M, Ji R, Wang Y, et al. HRank: filter pruning using high-rank feature map. 2020. ArXiv: 2002.10179
34 Wang D, Zhou L, Zhang X, et al. Exploring linear relationship in feature map subspace for convnets compression. 2018. ArXiv: 1803.05729
35 Lin S, Ji R, Yan C, et al. Towards optimal structured cnn pruning via generative adversarial learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 2790–2799
36 Gao X, Zhao Y, Dudziak L, et al. Dynamic channel pruning: feature boosting and suppression. 2018. ArXiv: 1810.05331
37 Huang Z, Wang N. Data-driven sparse structure selection for deep neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 304–320
38 He Y, Kang G, Dong X, et al. Soft filter pruning for accelerating deep convolutional neural networks. 2018. ArXiv: 1808.06866
39 Liu Z, Li J, Shen Z, et al. Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 2736–2744
40 Zhao C, Ni B, Zhang J, et al. Variational convolutional neural network pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 2780–2789
41 Zhuo H, Qian X, Fu Y, et al. Scsp: spectral clustering filter pruning with soft self-adaption manners. 2018. ArXiv: 1806.05320
42 Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. In: Proceedings of Conference and Workshop on Neural Information Processing Systems, 2017

附录 A

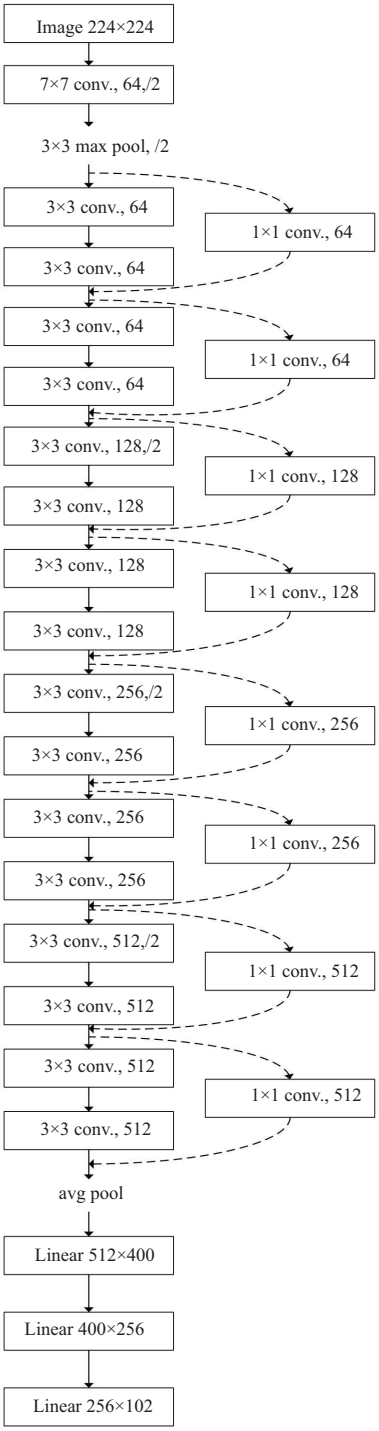

图 A1 ResNet18 结构图示

Figure A1 The ResNet18 architecture

Convolution network pruning based on the evaluation of the importance of characteristic attributions

Biao ZHANG[1,2], Pengbo YANG[1,2], Jitao SANG[1,2] & Jian YU[1,2*]

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

Abstract Although deep learning models have recently achieved remarkable performance in many tasks, they require massive memory footprint and computing power to achieve efficient inference. The researchers propose a number of compression methods to compress the capacity and computation of the model so that deep learning can be deployed to resource-constrained mobile terminals. Based on the pruning framework, two pruning methods are proposed from the perspective of filter importance evaluation. (1) As every filter can learn unique features, we propose an attribution mechanism to evaluate the correlation between the features learned by a filter and the causal features. We prune the filter with low correlation so as to compress the model and retain the attribution characteristics of the original model; the process is called attribution pruning. (2) The second pruning method uses positive correlation features in a channel and gradient to evaluate the importance of the filter, which is based on an iterative optimization pruning framework. This method, which is called Taylor-guided pruning, can improve the accuracy of pruning redundant filters. We implement two pruning methods in VGGNet and ResNet. Extensive experiments demonstrate that attribution pruning can greatly retain the attribution characteristics of the original model. Moreover, the two pruning methods can achieve better compression than current mainstream pruning methods.

Keywords deep learning, network pruning, attribution, compression, Taylor expansion

Biao ZHANG was born in 1995. He is an M.S. candidate at Beijing Jiaotong University. His main research interests include deep learning and machine learning.

Pengbo YANG was born in 1993. He is a Ph.D. candidate at Beijing Jiaotong University. His main research interests include deep learning and interpretable machine learning.

Jitao SANG was born in 1985. He received his Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences. Currently, he is a professor and Ph.D. supervisor in Beijing Jiaotong University. His main research interests include social media analysis, multimedia retrieval, and interpretable machine learning.

Jian YU was born in 1969. He received his Ph.D. degree from the Department of Mathematics, Peking University, Beijing, in 2000. Currently, he is a professor and Ph.D. supervisor in Beijing Jiaotong University. His main research interests include machine learning, data mining, and image segmentation.