DOI：10.11830/ISSN.1000-5013.202011013

采用可替代滤波器的卷积神经网络模型剪枝方法

周密[1,2]，张维纬[1,2]，陶英杰[1,2]，余浩然[1,2]

（1. 华侨大学 工学院，福建 泉州 362021；
2. 华侨大学 工业智能化与系统福建省高校工程研究中心，福建 泉州 362021）

摘要： 将卷积神经网络模型中某一层的所有滤波器抽象到一个欧几里德空间，对其中能被其他滤波器共同表示的滤波器剪枝，降低滤波器冗余，避免精度损失. 使用强化学习进行边训练边剪枝，经过微调恢复神经网络模型性能. 结果表明：剪枝并微调后的神经网络模型精度损失较小，参数量与浮点计算量显著减少.

关键词： 剪枝方法；神经网络模型；滤波器；深度学习；强化学习；边缘智能

中图分类号： TP 183；TP 391.41 **文献标志码：** A **文章编号：** 1000-5013(2022)02-0245-07

Pruning Method of Convolutional Neural Network Using Replaceable Filter

ZHOU Mi[1,2], ZHANG Weiwei[1,2],
TAO Yingjie[1,2], YU Haoran[1,2]

(1. College of Engineering, Huaqiao University, Quanzhou 362021, China;
2. Industrial Intelligence and System Fujian University Engineering Research Center, Huaqiao University, Quanzhou 362021, China)

Abstract: All filters in a certain layer of the convolutional neural network are abstracted into an Euclidean space, Pruning the filter that can be jointly represented by other filters, reducing the redundancy of filter and avoiding the loss of accuracy. Using reinforcement learning to prune while training, the neural network model performance is restored through fine-tuning. The results show that, after pruning and fine-tuning, the loss of neural network model accuracy is smaller, the calculated amount of parameters and floating-point are significantly reduced.

Keywords: pruning method; neural network model; filter; deep learning; reinforcement learning; edge intelligence

近年来，深度学习和神经网络的发展使神经网络模型的效果越来越好，但随着无人驾驶与智能移动设备等相关领域的不断发展与创新，对计算能力较弱的边缘设备上的深度神经网络模型的要求越来越高. 当神经网络模型部署在移动设备上时，由于深度神经网络的特性，参数量及浮点计算量都是极其庞大的. 例如，当使用 107 层 YOLO v3 网络结构检测分辨率为 416 px ×416 px 的图像时，将产生 240 MB

收稿日期： 2020-11-04

通信作者： 张维纬(1982-)，男，副教授，博士，主要从事大数据、物联网及边缘智能的研究. E-mail：178483968@qq.com.

基金项目： 国家自然科学基金面上资助项目(61976098)；福建省泉州市科技计划项目(2020C067)；华侨大学研究生科研创新能力培育计划项目(18014084015)

的参数量及多达650亿次的乘法累加计算量.然而,支持百亿次计算量的边缘设备的成本是相对较高的,例如,在NVIDIA Tesla T4上运行YOLOv3网络结构,每秒可实时检测40帧图像,该设备市场售价近3万元人民币,远超出普遍的经济承受能力,而现有的神经网络模型在低成本设备上很难兼顾神经网络模型的精确性与计算速度.因此,对于移动边缘设备来说,应在保持精度不降低或者略微降低的情况下,尽可能保证低的存储空间及少的浮点计算量[1].

将神经网络模型剪枝部署于移动边缘设备具有优势[1],在非结构化修剪方式方面,Guo等[2]提出一种动态的参数剪枝算法以逼近神经网络模型压缩的理论极限.Carreira-Perpinán等[3]通过两步(学习和压缩)交替优化的剪枝方法将原参数向约束表示的可行集投影,自动找到每层的最优稀疏比.Ding等[4]通过一阶泰勒展开式判断剪枝对最终输出造成的影响排序,网络中的权重根据排序进行分类.对造成影响较大的权重,采用常规的随机梯度下降(SGD)更新,对其他的权重则进行权重衰减,从而实现动态网络剪枝.非结构化的权重修剪方式能够有效降低神经网络模型的大小,但是剪枝完成以后的神经网络模型需要特定的硬件支持,故其泛化性较差.在结构化修剪中,He等[5]提出CP(channel-pruning)算法,通过添加范数约束权重$L1$进行LASSO(least absolute shrinkage and selection operator)回归,选择合适的稀疏通道对训练好的神经网络模型进行剪枝.Huang等[6]使用数据驱动的方式移除神经网络模型中多余的滤波器.You等[7]将通过纯残差方式连接的GBN分配到同一组,对全局滤波器重要性进行排序,并对排名较后的进行分组剪枝.He等[8]通过计算几何中心的方式对处于几何中心的冗余滤波器进行剪枝.

上述大部分剪枝策略都是基于启发式规则,此类方法着重于滤波器重要性的排序与修剪,但可能导致次优修剪,并且需要专家不断调节参数.确定性策略梯度算法中试错与奖惩的结合能够很好地检索剪枝策略[9].He等[10]通过使用深度确定性策略梯度(DDPG)算法得到各层剪枝率的最优组合,提出一种自动化剪枝AMC(automl for model compression and acceleration on mobile devices)算法.对神经网络模型所有层采用同一剪枝率的剪枝方法[8]可能会导精度下降,本文结合滤波器可替代的特性与强化学习,提出自动化可替代滤波器剪枝方法.

1 自动化可替代滤波器的剪枝方法

1.1 总体框架

手动剪枝方法依赖手动调节参数,对神经网络模型每一层进行相同压缩率的剪枝方法可能会降低神经网络模型的精度.受AMC算法的启发[10],利用强化学习的方式进行自动化剪枝,不需要专家手动调节参数,不仅避免耗时,还能获得组合剪枝策略.策略针对神经网络模型不同层实施不同的压缩率,使整体上达到目标压缩率.相较于随机性策略的强化学习方法,DDPG算法的效率更高,能够通过更少的数据获得连续且确定的动作输出.同时,DDPG算法中离线策略能够很好地避免确定性策略对环境探索存在的不足.在自动化剪枝过程中,引入可替代滤波器的搜索,搜索神经网络模型每一层中能够被同一层中其他滤波器表示的可替代滤波器.通过对可替代滤波器的剪枝,能最大程度保留原有神经网络模型的精度.

算法完整框架,如图1所示.算法由DDPG算法、获取剪枝的目标滤波器和滤波器的剪枝操作3部

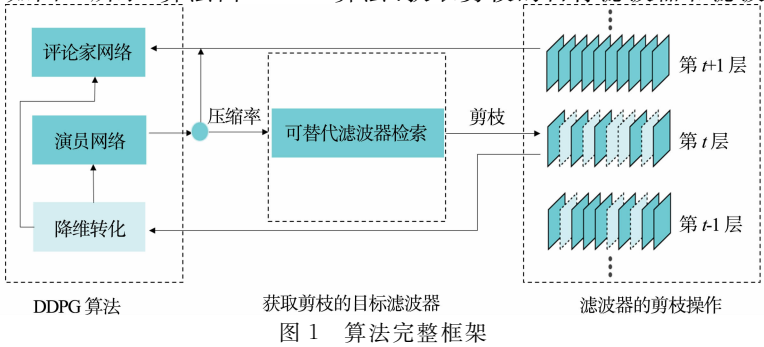

图1 算法完整框架

Fig. 1 Complete framework of algorithm

分组成. 在算法运行过程中，DDPG 算法在获得当前神经网络模型层的属性特征以后，经过降维转化，将原本的离散变量转化为连续的变量，并传导给演员网络与评论家网络，由演员网络输出确定性的神经网络模型的每一层剪枝压缩率，指导可替代滤波器进行剪枝. 将可替代滤波器的参数值归零，从而达到剪枝效果，在执行完神经网络模型所有层的剪枝后，算法在验证集上根据浮点计算量进行准确性奖励，并将结果返回给评论家网络，同时，将当前的剪枝策略保存到经验采样池中，然后，执行下一轮的网络剪枝，直到所有剪枝周期结束.

1.2　可替代滤波器的检索

神经网络模型中同一层的某些滤波器所包含的信息是能够被同一层中其他滤波器表示[8]，即被表示的滤波器为一组可替代滤波器. 可替代滤波器的检索以卷积层的剪枝为例，$\boldsymbol{F}_{i,j}$ 表示第 i 层的第 j 个滤波器，$\boldsymbol{F}_{i,j}$ 可以表示为 $\mathbf{R}^{N_i\times K\times K}$，其中，$K$ 为滤波器的尺寸，N_i 表示第 i 层输入通道数. 因此，神经网络模型 $W(i)$ 的第 i 层表示为 $\{\boldsymbol{F}_{i,j},1\leqslant j\leqslant N_{i+1}\}$，神经网络模型表示为 $\{W(i)\in\mathbf{R}^{N_{i+1}\times N_i\times K\times K},1\leqslant i\leqslant L\}$；第 i 层权重压缩后表示为 $N_i\times N'_{i+1}\times K\times K$，其中，$N_{i+1}$ 表示第 i 层输出通道数；N'_{i+1} 表示剪枝后第 i 层输出通道数；稀疏率表示为 N'_{i+1}/N_{i+1}.

在可替代滤波器搜索策略上，通过几何中位数的方式进行搜索[11]，如果该滤波器可以被几何中心点周围其余滤波器表示，即可判定其为可替代滤波器. 由于几何中位数是欧几里得空间中数据中心性的经典鲁棒性估计[11]，因此，可以使用几何中位数的方法获取第 i 层网络中所有滤波器的信息. 同时，对处于几何中心的滤波器进行选择，神经网络模型第 i 层中某一个滤波器到其他滤波器的距离之和为

$$\boldsymbol{x}^{\mathrm{GM}}=\arg\min_{\boldsymbol{x}\in\mathbf{R}^{N_i\times K\times K}}\sum_{j'\in[1,N_{i+1}]}\|\boldsymbol{x}-\boldsymbol{F}_{i,j'}\|_2. \tag{1}$$

在寻找可替代滤波器的过程中，如果一个滤波器到神经网络模型某一层中所有其他滤波器的整个距离最短，就可以认为它是处于最靠近中心的滤波器. 通过求解该滤波器与其他滤波器的最小欧氏距离，就能够找到其中最接近几何中心的滤波器，即认为选定的滤波器 $\boldsymbol{F}_{i,j^*}$ 含有的信息能够被同一层中其他的滤波器含有的信息表示，所以修剪这些滤波器对于网络的性能几乎没有影响. 由于计算几何中心十分耗时，且在滤波器的分布中可能存在没有任何一个滤波器处于几何中心的情形. 所以，距离最短问题可以转化为所有候选层中的滤波器距离其他滤波器的距离和最小问题，即

$$\boldsymbol{F}_{i,j^*}=\arg\min_{\boldsymbol{x}}\sum_{j\in[1,N_{i+1}]}\|\boldsymbol{x}-\boldsymbol{F}_{i,j}\|_2,\qquad \boldsymbol{x}\in\{\boldsymbol{F}_{(i,1),\cdots,(i,n)},\boldsymbol{F}_{i,N_{i+1}}\}. \tag{2}$$

在实际检索中，计算获得的几何中心并不能保证滤波器在该层中一定存在[8]，即如果在该层的几何中心处不存在滤波器，那么距离其他滤波器距离和最小的滤波器可以认定为该层的几何中心滤波器.

在实际剪枝过程中，根据层次的不同，执行不同数量的滤波器剪枝，即通过不同层的压缩率与该层的滤波器数量计算执行剪枝的滤波器数量. 在滤波器距离的排序中，将所有的滤波器进行几何中心距离计算，获得距离排序. 按照需要剪枝的滤波器数量进行剪枝. 该选定的滤波器为

$$\begin{aligned}\boldsymbol{F}_{i,j^*}&=\arg\min_{\boldsymbol{x}}\sum_{j'\in[1,N_{i+1}]}\|\boldsymbol{x}-\boldsymbol{F}_{i,j'}\|_2=\\&\arg\min_{\boldsymbol{x}}\sum_{j\in[1,N_{i+1}],\boldsymbol{F}_{i,j}\neq\boldsymbol{x}}\|\boldsymbol{x}-\boldsymbol{F}_{i,j}\|_2+[\|\boldsymbol{x}-\boldsymbol{F}_{i,j}\|_2]_{\boldsymbol{F}_{i,j}}=\boldsymbol{x}.\end{aligned} \tag{3}$$

剔除中心滤波器后，迭代计算，剔除选定距离的滤波器，直到满足该层剪枝数量的滤波器被全部找到为止. 因此，所有选定的滤波器 $\boldsymbol{F}_{i,j^*}$ 和该层剩余的滤波器共享绝大多数的基本信息，也即该滤波器所包含的信息可以被其他滤波器代替表示. 为了避免删除滤波器导致网络模型的不规则，将冗余滤波器的权重参数设置为零，而不是直接将滤波器删除，剪枝以后的模型经过微调能够很快恢复到原始性能.

1.3　强化学习

对于模型所有层(敏感层与非敏感层)均进行相同压缩率的剪枝操作[8]，模型每一层的滤波器剪枝数量为目标压缩率与该层的所有滤波器数量之积. 此方式可能导致参数重要性高的敏感层的剪枝率过高，导致不必要的精度损失. 以 DDPG 算法作为代理连续控制压缩比，对不同神经网络模型层次以不同压缩率进行自动化剪枝. DDPG 算法由评论家当前网络、评论家目标网络，演员当前网络、演员目标网络 4 个神经网络构成. 环境状态 S_t 的 10 个属性特征分别为 $n,c,h,w,\mathrm{stride},k,\mathrm{FLOPs}[t],\mathrm{Re}_{\mathrm{all}},\mathrm{Rest}$,

a_{t-1}. DDPG 代理能够很好地将所有的卷积层区分开,卷积核大小为 $n\times c\times k\times k$,输入值为 $c\times h\times w$;FLOPs[t]为第 t 层的浮点计算次数;$\mathrm{Re}_{\mathrm{all}}$为前面所有已剪枝层减少的浮点计算量(FLOPs)的总数;Rest 为余下各层中剩余的 FLOPs 总数. 在这些属性特征传递给代理之前,所有的属性特征都会被压缩到[0,1],以避免因为数值大小对模型训练产生的影响.

由于模型的剪枝对于模型的稀疏程度十分敏感,随着离散动作数量的急剧增加,动作空间的搜索难度也急剧加大,导致很难做到高效且快速的检索. 所以,通过选择 $a\in(0,1]$的连续动作空间进行更精确的压缩. 在强化学习指导剪枝的过程中,代理从滤波器剪枝的环境中获得其所处的第 t 层的环境状态 S_t,获得当前特征向量 $\boldsymbol{\varphi}(S)$. 将 S_t 状态下的动作 A_t 作为当前层的压缩率,将目标压缩率下需要剪枝的滤波器的总数与已完成剪枝的滤波器数量的差值作为剩余需要剪枝的滤波器的数量,通过 DDPG 算法反馈的压缩率执行当前层剪枝过程. 完成当前层的剪枝后,得到当前层剪枝完成以后的短期奖励 R 及第 t 层剪枝完成以后的新环境状态 S_{t+1}.

将 $\boldsymbol{\varphi}(S)$,A_t,R, S_{t+1}对应的特征向量 $\boldsymbol{\varphi}(S_{t+1})$及判断是否为终止状态的 is_end 存放到经验回放池中,其中,is_end 用来评估是否处于剪枝模型的最后一层. 对于模型的下一层,代理会接收上一层剪枝完成后所产生的新环境状态 S_{t+1}. 执行与第 t 层网络模型相同的自动化剪枝流程,直到滤波器的剪枝数量达到目标剪枝数量. 在判断完成最后一层的剪枝后,当 is_end 值为真时,模型会在验证集上评估剪枝完成的模型的准确性奖励 R_{FLOPs}并返回给代理,同时,将此次的剪枝策略及奖励 R_{FLOPs}存放到字典中.

在 DDPG 经验回放池中采集 m 个样本$\{\boldsymbol{\varphi}(S_j),A_j,R_j,\boldsymbol{\varphi}(S'_j),\mathrm{is_end}_j\}$,其中,$j=1,2,3\cdots m$. 由评论家当前网络计算获得当前目标 Q 值 y_j,即

$$y_j=\begin{cases}R_j-b, & \mathrm{is\_end}_j\ \text{为真},\\ \mathrm{R_j-b}+\gamma\mathrm{Q}'(\boldsymbol{\varphi}(S'_j),\pi_{\theta'}(\boldsymbol{\varphi}(S'_j)),\omega'), & \mathrm{is\_end}_j\ \text{为假}.\end{cases}\tag{4}$$

式(4)中:$\mathrm{Q}'(\boldsymbol{\varphi}(S'_j),\pi_{\theta'}(\boldsymbol{\varphi}(S'_j)),\omega')$通过评论家目标网络获得;$\pi_{\theta'}(\boldsymbol{\varphi}(S'_j)$则通过演员目标网络获得;$\gamma$ 值设置为 1,以避免短期奖励的优先级过高;b 为基线奖励,在代理更新期间通过减去 b 减少梯度估计的方差,梯度估计的值为此前奖励的指数移动平均值.

每一次演员当前网络选择的动作 A 会增加一定的噪声 N,并且噪声会在每轮剪枝完后呈指数衰减. 最终和环境交互的动作 A 为

$$A=\pi_\theta(S)+N.\tag{5}$$

DDPG 算法通过均方差损失函数、梯度反响传播的方式更新评论家当前网络的所有参数,并定期将其复制到评论家目标网络. 即

$$L(w)=\frac{1}{n}\sum_{j=1}^{m}(y_j-Q(\boldsymbol{\varphi}(S_j),A_j,\omega))^2.\tag{6}$$

式(6)中:$\boldsymbol{\varphi}(S_j)$为当前 S_j 状态获得的特征向量;$Q(\boldsymbol{\varphi}(S_j),A_j,\omega)$由评论家当前网络获得.

损失函数通过梯度反响传播的方式更新演员当前网络的所有参数 θ,即

$$J(\theta)=-\frac{1}{m}\sum_{j=1}^{m}Q(S_j,A_j,\theta).\tag{7}$$

式(7)中:$Q(S_j,A_j,\theta)$由评论家当前网络计算所得.

在不损失模型精度情况下,为了准确找出压缩情况,算法通过总结文献[10]的研究经验,将奖励函数设置为

$$R_{\mathrm{FLOPs}}=-\mathrm{Error}\cdot\ln(\mathrm{FLOPs}).\tag{8}$$

式(8)中:ln(FLOPs)以 e 为底的对数,且与 Error 成反比,该奖励函数对误差敏感,能更好支持模型剪枝中对 FLOPs 的减少.

提出的自动化可替代滤波器剪枝算法,如算法 1 所示.

算法 1

Input:training data:X.

Initialize:model parameter $\mathrm{W}=\{\mathrm{W}_{(\mathrm{i})},0\leqslant\mathrm{i}\leqslant\mathrm{L}\}$ Action,$\mathrm{N_i}$

for episode=1;episode$\leqslant$episode$_{\max}$;episode + + do

Update the model parameter W based on X
for i = 1;i ⩽ L; i + + do
Initialize the reduced model size so far
$W_{all}=\sum_L W_L$
$W_{rest}=\sum_{L=i+1} W_L$
$W_{duty}=\alpha * W_{all}\text{-}Action_{max} * W_{rest}-W_{reduced}$
$Action_i=\max(Action_t, W_{duty}/W_i)$
$W_{reduced}=W_{reduced}+Action_i * W_i$
Find $Action_i * N_{i+1}$ filters that satisfy Equation 2
Zeroize selected filters
If i==L
Update the Reward
end if
end for
end for
Obtain the compact model W * from W
Output: The compact model and its parameters W *

2　实验结果与分析

2.1　实验环境

搭建 Pytorch 环境进行训练，系统为 Ubuntu18.04，电脑硬件配置为 Intel(R)Core(TM)i7-6700 CPU@3.4 GHz，NVIDIA GeForce TITAN Xp；软件配置为 CUDAToolkitv10.0，Anaconda3 和 Pycharm 2019.

2.2　数据集与网络模型

算法在公开的 CIFAR-10 数据集的 VGGNet-16 和 ResNet-56[12] 两种模型及公开的 ImageNet[13] 数据集的 Mobilenet-V1 和 Mobilenet-V2 两种模型上进行实验. CIFAR-10 数据集包含 10 类，每一类包含 6 000 张图片，共 60 000 张图片，每张图片均是分辨率为 32 px×32 px 的三通道 RGB(三原色)图片，其中：50 000 张为训练集图片，10 000 张为测试集图片. ImageNet 数据集作为大型数据集，训练集包含分辨率为 224 px×224 px 的 1 000 类图片，每类图片包含 1 300 张图片. 测试集包含 50 000 张图片.

2.3　对比算法

对比算法有 CP 算法[5] 与 AMC 算法[10]. 实验效果衡量指标主要有算法准确率、参数量及浮点计算量. 算法准确率用来衡量模型分类结果的好坏. 参数量用来衡量神经网络所占计算机内存的大小，参数量的压缩比例是指剪枝后的参数量与未剪枝的参数量的减少比例. 浮点计算量指的是浮点运算次数，用来衡量网络模型复杂程度，浮点计算量的减少说明网络在实际运算中可以取得加速效果，减少量越大，说明加速效果越明显.

2.4　CIFAR-10 数据集实验结果

将提出的算法运用到不含直连的 VGGNet-16 与含有直连的 ResNet-56 神经网络模型上，VGGNet-16 模型，ResNet-56 模型在 CIFAR-10 数据集上的剪枝比较，如表 1 所示. 表 1 中：η 为压缩率；ε 为精度，$\Delta\varepsilon$ 为精度变化.

由表 1 可知：VGGNet-16 神经网络模型的压缩率为 20%时，相较于 CP 算法与 AMC 算法，文中算法性能降低得最小，精度仅降低了 1.1%；ResNet-56 神经网络模型压缩率为 50%时，相较于 CP 算法与 AMC 算法，文中算法性能降低得较小，精度仅降低了 0.7%.

对于 FLOPs 约束下的常用模型的剪枝，文中算法通过强化学习执行不同剪枝率，保证学习过程产生最优剪枝组合策略，能够很好地避免敏感层的过度剪枝. 对比根据经验手动调参方法与启发式方法，处于几何中心的冗余滤波器的剪枝能获得更好的剪枝效果，能够很好地保留对模型精度影响大的滤波

器.对比引入静态正则项进行自动化剪枝的AMC算法,文中算法通过对可替代滤波器进行冗余滤波器剪枝,能够很好地保留包含较多有用信息的滤波器,从而获得更低的精度损失.

表1 VGGNet-16模型,ResNet-56模型在CIFAR-10数据集上的剪枝算法比较

Tab.1 Comparison of pruning algorithms using VGG-16 model and ResNet-56 model On CIFAR-10 dataset

剪枝算法	VGGNet-16模型			ResNet-56模型		
	η/%	ε/%	$\Delta\varepsilon$/%	η/%	ε/%	$\Delta\varepsilon$/%
CP算法	20	68.8	−1.7	50	91.8	−1.0
AMC算法	20	69.2	−1.4	50	91.9	−0.9
文中算法	20	69.6	−1.1	50	92.1	−0.7

2.5 ImageNet数据集实验结果

将提出的算法运用到ImageNet数据集上,在保持相同压缩率下,文中算法在Mobilenet-V1模型和Mobilenet-V2模型上的剪枝效果比较,如表2所示.表2中:−表示未执行剪枝操作;ε(Top-1)为排名第一的类别与实际相符的精度.

由表2可知:在保持相同压缩率条件下,文中算法保留的模型能够获得更好的精度保留;当FLOPs为41×10^6次时,相比于基准算法,文中算法在Mobilenet-V1模型上的精确度高出7.5%,文中算法在Mobilenet-V2模型上的精确度高出4.7%.

表2 文中算法在Mobilenet-V1模型和Mobilenet-V2模型上的剪枝效果比较

Tab.2 Comparison of pruning effect of proposed algorithm on Mobilenet-V1 model and Mobilenet-V2 model

模型	基准算法		文中算法	
	ε(Top-1)/%	FLOPs/$\times10^6$	ε(Top-1)/%	FLOPs/$\times10^6$
Mobilenet-V1	70.6	569	—	—
	68.4	325	70.4	325
	63.7	149	66.3	150
	50.6	41	58.1	41
Mobilenet-V2	74.7	585	—	—
	72.0	313	72.9	309
	67.2	140	68.6	139
	54.6	43	59.3	41

文中算法与AMC算法的Top-1准确性比较,如表3所示.由表3可知:与AMC算法相比,文中算法能够更好地保留模型中重要的滤波器.同时,对于保留相同数量的滤波器,能够保证更小的精度损失,而获得更好的结果.具体表现为在保持基本相同的FLOPs减少量的基础上,文中算法在精度损失上更小,剪枝完成后的Mobilenet-V1模型精度提高了0.1%,Mobilenet-V2模型精度提高了0.9%.

表3 文中算法与AMC算法的Top-1准确性比较

Tab.3 Accuracy comparison of proposed algorithm and AMC algorithms

剪枝算法	FLOPs/$\times10^6$	ε(Top-1)/%	剪枝算法	FLOPs/$\times10^6$	ε(Top-1)/%
0.75×MobileNet-V1	325	68.4	0.75×MobileNet-V2	220	69.8
AMC算法	285	70.5	AMC算法	220	70.8
文中算法	280	70.6	文中算法	219	71.7

3 结束语

针对卷积神经网络庞大的浮点计算量,提出对可替代滤波器进行自动化剪枝的方法.在保证精度基本不降低的情况下,文中算法大幅度了降低浮点计算量,保证了对不同结构的神经网络模型的良好适应性.对于要求极严苛的超小型边缘设备的小存储量,未来的工作将在保证精度降低不大的前提下进行模型的进一步压缩与加速.

参考文献:

[1] ZHOU Zhi,CHEN Xu,LI En,*et al*. Edge intelligence: Paving the last mile of artificial intelligence with edge compu-

ting[J]. Proceedings of the IEEE,2019,107(8):1738-1762. DOI:10.1109/JPROC.2019.2918951.

[2] GUO Yiwen,YAO Anbang,CHEN Yurong. Dynamic network surgery for efficient dnns[C]// Advances in Neural Information Processing Systems. Montreal:[s.n.],2016:1379-1387.

[3] CARREIRA-PERPINÁN M A,IDELBAYE V Y. "Learning-Compression" algorithms for neural net pruning[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake:IEEE Press,2018:8532-8541.

[4] DING Xiaohan,ZHOU Xiangxin,GUO Yuchen,*et al*. Global sparse momentum SGD for pruning very deep neural networks[C]// Advances in Neural Information Processing Systems. Vancouver:[s.n.],2019:6382-6394.

[5] HE Yihui,ZHANG Xiangyu,SUN Jian. Channel pruning for accelerating very deep neural networks[C]// Proceedings of the IEEE International Conference on Computer Vision. Venice:IEEE Press,2017:1389-1397.

[6] HUANG Qiangui,ZHOU K,YOU Suya,*et al*. Learning to prune filters in convolutional neural networks[C]// Winter Conference on Applications of Computer Vision. Lake Tahoe:IEEE Press,2018:709-718.

[7] YOU Zhonghui,YAN Kun,YE Jinmian,*et al*. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks[C]// Conference on Neural Information Processing Systems. Vancouver:[s.n.],2019:2133-2144.

[8] HE Yang,LIU Ping,WANG Ziwei,*et al*. Filter pruning via geometric median for deep convolutional neural networks acceleration[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Long Beach:IEEE Press,2019:4340-4349. DOI:10.1109/CVPR.2019.00447.

[9] SILVER D,LEVER G,HEESS N,*et al*. Deterministic policy gradient algorithms[C]// International Conferenceon Machine Learning. Cambridge:MIT Press,2014:387-395.

[10] HE Yihui,LIN Ji,LIU Zhijian,*et al*. Amc: Automl for model compression and acceleration on mobile devices[C]// Proceedings of the European Conference on Computer Vision. Munich:Springer,2018:784-800.

[11] FLETCHER P T,VENKATASUBRAMANIAN S,JOSHI S. Robust statistics on Riemannian manifolds via the geometric median[C]// 2008 IEEE Conference on Computer Vision and Pattern Recognition. Anchrage:IEEE Press,2008:1-8. DOI:10.1109/CVPR.2008.4587747.

[12] HE Kaiming,ZHANG Xiangyu,REN Shaoqing,*et al*. Deep residual learning for image recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas:IEEE Press,2016:770-778. DOI:10.1109/CVPR.2016.90.

[13] RUSSAKOVSKY O,DENG Jia,SU Hao,*et al*. Imagenet large scale visual recognition challenge[J]. International Journal of Computer Vision,2015,115(3):211-252. DOI:10.1007/s11263-015-0816-y.

（**责任编辑**：陈志贤　　**英文审校**：吴逢铁）