激光与光电子学进展 先进成像

基于深度学习的自适应动态滤波器剪枝方法

褚晶辉，李梦，吕卫*
天津大学电气自动化与信息工程学院，天津 300072

摘要 模型压缩可以有效地促进卷积神经网络在资源受限设备上的部署。作为一个研究热点，滤波器剪枝已经受到了从学术界到工业界的广泛关注。滤波器剪枝的本质是对重要滤波器进行选择和保留。然而，现有的研究主要集中在静态滤波器和局部滤波器的选择上，压缩后的模型仍然存在一定的冗余。基于此，提出了一种自适应动态滤波器剪枝方法，该方法通过引入一个激活权值生成模块来生成每个滤波器的激活值。将模块嵌入各种经典网络中，来动态评估卷积层中所有滤波器的重要性，并自适应地选择能提取更丰富信息的滤波器来重构剪枝后的网络。在CIFAR-10和AUC数据集上使用不同卷积神经网络进行了实验，所提方法在CIFAR-10数据集上与目前几种主流的剪枝方法相比具有更优越的性能。在AUC数据集上进行剪枝前后压缩70%左右计算量的情况下，准确率下降不超过0.3个百分点。在不同网络上的实验证明了该方法在不同模型上的泛化能力。

关键词 机器视觉；深度学习；卷积神经网络；滤波器剪枝；分类模型；模型压缩
中图分类号 TP391.4 **文献标志码** A **DOI**：10.3788/LOP202259.2415003

Adaptive Dynamic Filter Pruning Approach Based on Deep Learning

Chu Jinghui, Li Meng, Lü Wei*
School of Electrical and Information Engineering, Tianjin University, 300072, *China*

Abstract Model compression can significantly improve the deployment of convolutional neural networks on limited-resource devices. Filter pruning has gradually drawn attention from academia and industry as a research hotspot. The essence of filter pruning is the selection and retention of important filters. Existing research has primarily focused on static and interlayer filter selection, which still has redundancy in the compressed model. We propose an adaptive dynamic filter pruning approach in this paper wherein an activation weight generation module is introduced to generate the activation weight of each filter. The importance of filters in global convolutional layers is dynamically evaluated by embedding the activation weight generation module in various classical networks, and the filters that extract richer information are adaptively selected to reconstruct the pruned networks. Experiments are performed on CIFAR-10 and AUC datasets using different convolutional neural networks, among which the proposed method has better performance than several mainstream pruning methods on CIFAR-10 dataset. The accuracy decreased by 0.3 percentage points when the computation was compressed by ~70% before and after pruning on the AUC dataset. Experiments on various networks demonstrate the proposed method's ability to generalize to different models.

Key words machine vision; deep learning; convolutional neural network; filter pruning; classification model; model compression

1 引 言

卷积神经网络(CNNs)是一种常被应用于提取图像特征的模型，如今已成功应用在各种计算机视觉任务，例如分类任务[1-2]、检测任务[3-4]和分割任务[5]。相比于传统人工方法，卷积神经网络具有高准确性的优势。但受限于移动设备和可穿戴设备内存大小和计算能力，现有的卷积神经网络如VGG[6]、ResNet[7]、GoogLeNet[8]、DenseNet[9]等复杂模型的计算量和参数量所带来的高额存储空间和计算资源消耗等问题导致模型不能直接部署在这些设备上。因此，以去除参数、减小计算量、同时保持高精度为目标的模型压缩已成为一个热门的研究领域。近年来出现了很多优秀的模型压缩方法，高晗等[10]将模型压缩方法进行了归类，主

收稿日期：2021-09-06；**修回日期**：2021-10-01；**录用日期**：2021-10-27
通信作者：*luwei@tju.edu.cn

要可以分为网络剪枝[11-12]、知识蒸馏[13]、模型量化[14]和轻量化设计[15-16]等4类。

目前的压缩与加速方法多是为图片分类任务的卷积神经网络模型设计的，然而在实际应用中，还有大量其他模型应用于人工智能领域。在上述类别中，网络剪枝通过去除冗余权值或冗余的滤波器来进行模型压缩，是为了开发更小、更高效的神经网络，大量实验结果证明了其在模型压缩方面的广泛适用性。作为网络剪枝的一个分支，滤波器剪枝通过去除冗余滤波器来显著减小模型的大小，已成为当前模型压缩方法的一个热门方向，近年来也引起了大量相关人员的关注。例如，He等[11]引入了一种迭代两步算法，该算法使用套索算法（LASSO）回归来去除冗余滤波器，基于最小二乘复原算法来有效修剪网络层，主要针对训练好的模型进行剪枝，避免了重新训练的复杂过程，同时参考张量分解中重建特征图的优化方法，不考虑单个参数的重要性，直接最小化输出特征图的重建误差，逐层进行剪枝操作。Lin等[12]提出了一种结构化剪枝方法，该方法通过生成对抗学习（GAL）对滤波器和其他结构进行端到端的联合剪枝，用来解决现有剪枝方法依赖分层的多阶段优化和针对特定网络结构进行剪枝的问题。Yu等[17]定义最后一个与Softmax层相连的隐藏层为final response layer（FRL），通过特征选择器来确定各个特征的重要性得分，之后进行反向传播得到整个网络各层的得分，再根据剪裁比率进行剪枝，其剪枝的原则是FRL输出的重建误差最小。Lin等[18]提出了一种通过观察特征映射的秩（HRank）来确定滤波器的相对重要性的滤波器剪枝方法，该方法从数学的角度判定滤波器的相对重要性，可以解释为在同层网络中滤波器的秩越高说明经过此通道的特征信息含量越丰富，但对网络中每层剪枝率的设定没有给出相应解释。Zhang等[19]将剪枝问题视为具有组合约束条件的非凸优化问题，并利用交替方向乘法器（ADMM）将其分解为两个子问题，可分别用随机梯度下降（SGD）法和解析法求解。Li等[20]提出了一种衡量神经网络中滤波器对输出精度影响的方法，并且修剪对输出精度影响很小的滤波器。Zhuang等[21]引入额外的识别感知损失，辅助选择真正有助于识别的滤波器，联合重建误差共同优化。

虽然现存的滤波器剪枝方法可以显著减小模型的大小，但仍然存在一些挑战。首先是如何在不同的层中分配保留的滤波器的数量。目前的剪枝方法[18,20]按照事先预设的总剪枝率，固定比例地去剪枝每一层网络，而不具体考虑网络不同层滤波器重要性的不同程度。这样只局限在单层里按固定比例选择较重要的滤波器，而不是从全局比较滤波器的重要性，因此限制了剪枝的效果和网络性能。其次是如何动态适应地修剪网络。Gao等[22]提出了一种基于动态开闭滤波器的剪枝方法：当一开始滤波器就被判定为不重要滤波器而被关闭时，在后续训练过程中很难被开启，对实验最终结果也会有一定影响；当上层网络的滤波器被判定为不活跃而被删掉时，会影响下层网络滤波器的活跃性判定。上述剪枝方法在剪枝时忽略了被修剪滤波器对后续滤波器的影响，可能会得到非最优的剪枝方案。

为此，本文在网络中引入激活权值生成模块（AWGM），该模块利用前一层输入的特征映射预测当前层滤波器的激活值。滤波器可以从其输入的特征映射中提取各种相似特征与特异特征，因此可以采用前一层的特征映射来预测当前层滤波器的重要程度。首先使用AWGM来获得所有滤波器的激活值。其次计算滤波器的激活权值并对激活权值进行排序，结合总体剪枝比设置阈值。最后，通过去除权值小于阈值的滤波器来对网络进行剪枝。该方法能有效地修剪掉每层网络中的冗余滤波器，大大减小了模型的参数量和计算量，同时保持了模型的精度。

2 滤波器剪枝方法

2.1 滤波器剪枝方法分类

大量的理论和实验表明，在原有网络层的基础上对模型进行剪裁可以在维持高精度的基础上大幅度削减运算量和参数量。卷积层通常是空间稀疏的，也就是说，它的激活输出可能只包含很小的有效区域。一般来说，传统的滤波器剪枝过程包括3个阶段：预训练原始模型；根据原始模型信息计算每个滤波器的重要性，并继承该模型保留的滤波器的参数；微调训练修剪过的模型以恢复丢失的性能。

现有的滤波器剪枝策略可分为静态剪枝[18]和动态剪枝[22]。静态剪枝方法旨在去除每一层网络中固定数量或比例的非重要滤波器，从而完成剪枝任务。传统的静态剪枝方法训练过程非常复杂，并且未考虑到被去除的滤波器对网络中其他滤波器的影响，所以往往会对网络的自适应能力产生负面影响。动态剪枝方法自适应地从网络中选择要剪枝的滤波器。动态剪枝方法虽然对网络的适应性较强，但这种方法只是将不重要的通道或权值重置为零，而没有完全去除它们，因此模型的大小保持不变。综上所述，进行模型剪枝不仅要对通道进行修剪来大幅缩小模型大小，而且要准确选择网络层中每层保留的滤波器数量，否则不仅无法有效减小卷积神经网络中的计算量，而且也会对模型准确率造成不利影响。Ma等[23]指出，卷积层中的神经元专门识别不同的特征，并且一个神经元的重要性很大程度上取决于输入。两张不同图像输入网络，会引起卷积层中不同滤波器的响应值不同，因此根据输入来选择较重要的滤波器并将其保留是很有必要的。

2.2 动态自适应滤波器剪枝方法

所提方法的自适应性主要体现在根据网络输入的训练数据来对滤波器重要性进行判断，通过AWGM，每一层的输出可以选择对当前层重要的滤波器进行特

征增强与抑制。在网络训练和反向传播的过程中，滤波器可以通过观察卷积操作的输入和输出特征来进行学习和自我调整，并且由于AWGM是以模块形式对网络中每个卷积层进行操作的，可以适用于所有具有卷积层的分类网络。同时根据激活权值对滤波器进行排序，通过全局排序实现在全局选取重要滤波器。所提方法也可以广泛应用于现有的主流卷积神经网络架构中，针对不同网络设计了不同且有效的组合方案。

2.2.1　激活权值生成模块

所提滤波器剪枝方法的结构如图1所示。左侧为原始图像，原始输入分别进入原始网络和AWGM模块。AWGM根据输入的特征图自适应地生成每个滤波器的激活值，从而进行特征的增强和抑制。首先，在CIFAR-10数据集[24]上训练原始网络和AWGM模块以获得最佳性能。然后，AWGM根据已训练好的模型中所有滤波器的激活值，生成对应的激活权值。其处理过程如下：每个滤波器的整体激活权值等于此滤波器的每个激活值与均值之差的绝对值之和，整体激活权值用于对滤波器进行重要性排序；最后根据原网络中预计保留的神经元的比例计算出修剪阈值，将整体激活权值小于阈值的滤波器删除，保留的滤波器从原始网络中恢复对应通道的参数，并对修剪后的网络进行再一次训练以恢复网络损失的性能。

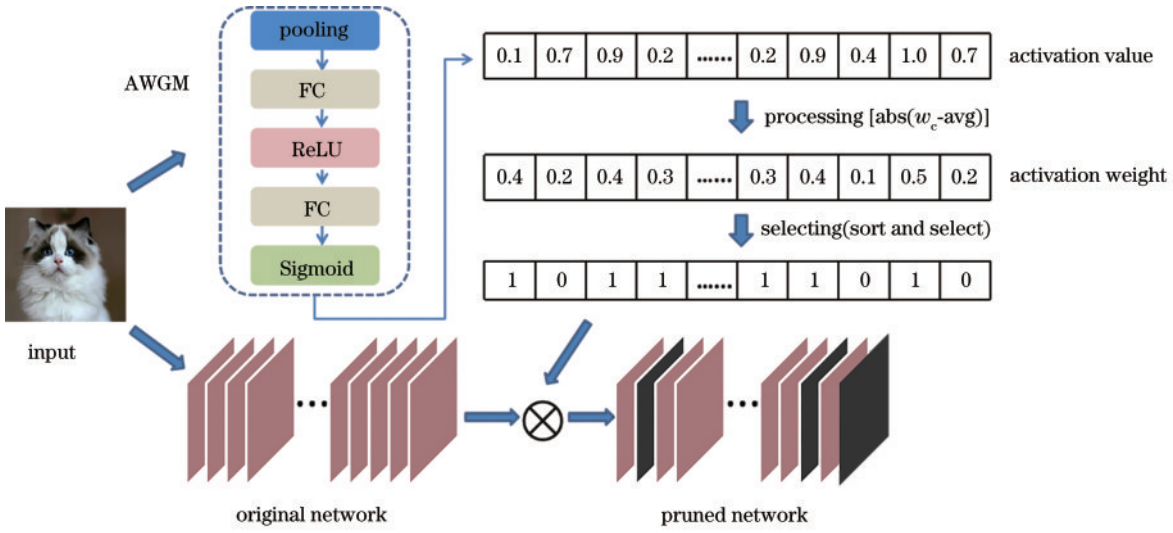

图1　滤波器剪枝方法的结构

Fig. 1　Structure of filter pruning approach

针对VGG网络而言，如图1所示，由原始网络conv层中得到的输出结果可以表达为

$$x_l = f(x_{l-1})。\tag{1}$$

为了生成不同滤波器的激活值，将AWGM模块应用到每个卷积层，从而生成层内滤波器的激活值：

$$w_c = \sigma\left[W_2\delta(W_1 z_{l-1})\right],\tag{2}$$

$$z_{l-1} = \frac{1}{h\times w}\sum_{i=1}^{h}\sum_{j=1}^{w}x_{l-1}(i,j),\tag{3}$$

式中：δ和σ分别表示激活函数ReLU[25]和Sigmoid[26]；W_1和W_2分别表示模块中第1个和第2个全连接层的权值；w_c表示神经网络中第l层滤波器的激活值；z_{l-1}表示通过全局平均池化操作在其空间维度$h\times w$中压缩的结果。

最后通过将w_c和x_l相乘来生成加权特征：

$$x_l^w = w_c x_l。\tag{4}$$

加权后的特征x_l^w既是当前层网络的输出，同时也是下一层网络的输入，之后的网络可以根据经过增强和抑制后的输入特征图来判定剪枝之后的网络所带来的影响，并且根据结果自适应调整当前层网络的重要信息分布。

2.2.2　基于AWGM的剪枝方法

AWGM可以根据输入特征映射预测每个输出滤波器的重要性，生成的对应激活值表示此滤波器对输入特征图的响应程度。值得注意的是，Gao等[22]指出，经过Sigmoid函数处理的激活值分布在0～1之间，激活值越偏离中心点表明滤波器对输入特征反应越强烈，而在剪枝过程中应当选择对输入响应强烈的滤波器进行保留。

计算滤波器激活值与均值之差可以判断单个滤波器对不同输入响应的强烈程度。本实验组采用激活值与均值之差的绝对值作为当前网络层中样本滤波器的激活权值，然后将所有训练样本的激活权值相加作为整体激活权值，并利用整体激活权值对所有滤波器进行排序。设置整体剪枝比例，根据排序结果得到阈值，将整体激活权值小于设定阈值的滤波器删除，并由此可以计算出每一层的剪枝率。剪枝过程如图1所示，图中将整体激活权值小于设定阈值的需要剪枝的通道置为0，保留的通道置为1，在实际操作过程中，根据计算出的每层网络的剪枝率对通道数进行相应修改，之后从原始网络中恢复需要

保留的滤波器对应的通道的网络权重，最后进行精度恢复训练。

此外，AWGM模块也可以适应不同的网络结构，本实验组针对不同网络设计了不同且有效的组合方案。AWGM与ResNet-56、GoogLeNet、DenseNet-40的组合如图2所示。

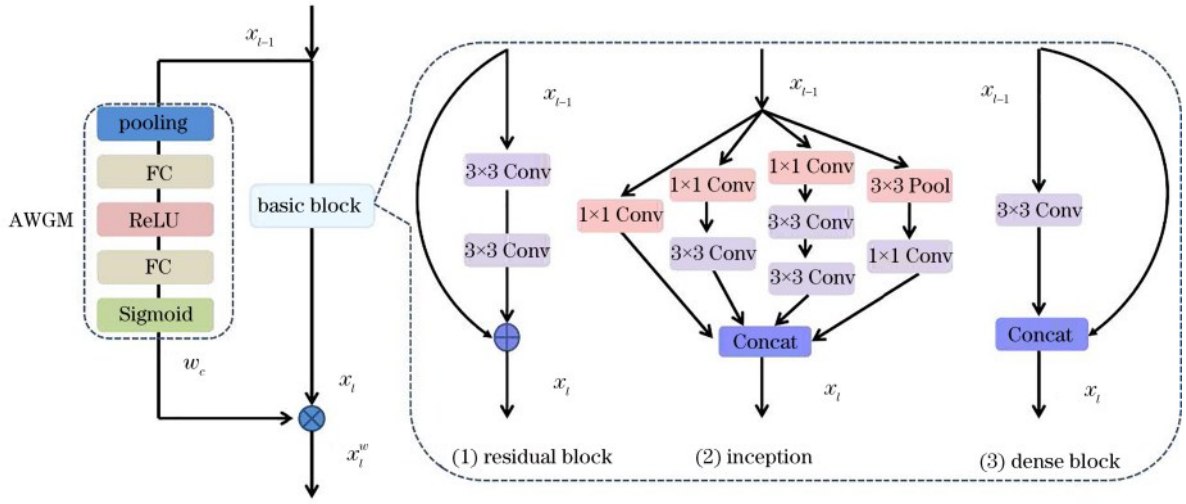

图2　AWGM和各种网络中基本块的组合

Fig. 2　Combination of AWGM and basic blocks in various networks

3　实验结果分析

在CIFAR-10数据集[24]上比较了VGG-16[6]、ResNet-56[7]、GoogLeNet[8]和DenseNet-40[9]等4种经典卷积神经网络架构下基于不同剪枝方法的实验结果，并且展示了在驾驶员行为检测AUC数据集[27]上对VGG-16[6]、VGG-19[6]、ResNet-56[7]等3个模型上剪枝前后的实验结果。使用准确率(Accuracy)来评价网络的性能，使用浮点运算(FLOPs)和参数量(Parameters)来评价网络的复杂性。

3.1　实验平台及数据集

实验使用NVIDIA 1080Ti显卡，操作系统为Linux-Ubuntu 16.04，深度学习框架为Pytorch1.3-gpu。在进行剪枝效果对比实验时，选用CIFAR-10数据集[24]和AUC驾驶行为数据集[27]进行训练和测试。

CIFAR-10数据集[24]包含10个类别，分别为飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船和卡车，每类图像均有6千张，其中没有任何的重叠情况，也不会在同一张照片中出现两类事物。数据集共有6万张图像，分布如表1所示，其中5万张图像用于训练，1万张图像用于测试，且图像大小均为32×32。

Abouelnaga等[27]于2017年创建了AUC驾驶行为数据集，这是一个公开的具有10类驾驶动作的数据集，包括17308张图像，其中12977张图像用于训练而4331张图像用于测试，图像大小为224×224。该数据集收集了来自7个国家的31名参与者[埃及(24)、德国(2)、美国(1)、加拿大(1)、乌干达(1)、巴勒斯坦(1)和摩洛哥(1)]在4种不同车辆环境中的驾驶图像。AUC数据集包含的10种驾驶行为分别为正常驾驶、喝水、右手接打手机、左手接打手机、右手发送短信、左手发送短信、操作收音机、整理仪表、向后座接东西、和乘客对话这些动作。每类驾驶行为的训练和测试图片数量如表2所示。

表1　CIFAR-10数据集

Table 1　CIFAR-10 dataset

Training	Test
50000	10000

表2　AUC驾驶行为数据集

Table 2　AUC driving behavior dataset

Category	Training	Test
Drive Safe	2764	922
Drinking	1209	403
Talk Right	917	306
Talk Left	1020	341
Text Right	1480	494
Text Left	975	326
Adjust Radio	915	305
Hair & Makeup	901	301
Reach Behind	869	290
Talk to Passenger	1927	643

3.2　CIFAR-10数据集参数设定及剪枝实验结果

实验中所采用的优化算法为SGD算法。在实验中，初始学习率和总体迭代次数分别设置为0.1和200，每次训练的批次大小为128。在4种网络结构上分别与He等[11]的方法、GAL[12]、HRank[18]、Li等[20]的

方法、Zhao 等[28]的方法进行了比较。

表 3 展示了与 Li 等[20]的方法和 HRank[18]在 VGG-16 网络模型上的性能比较。HRank[18]在不同的剪枝率下进行了实验，因此为了更好地进行比较，本实验组在实验中设置了两个剪枝率(0.75 和 0.85)。从表 3 可以看出，所提方法在 FLOPs 上较原始网络 VGG-16 减少了 56% 和 70%，在参数量上仅占原始网络的 14% 和 10% 的情况下在准确率上分别只降低了 0.31 个百分点和 0.82 个百分点。与 Li 等[20]的方法相比，所提方法能够在 FLOPs 和参数量分别降低 70 MB 和 3.3 MB 的情况下准确率依旧提升了 0.25 个百分点。与 HRank[18]相比，所提方法在两种剪枝比下都达到了较高的准确率并且具有更少的 FLOPs 和参数量。

表 3 所提方法与其他方法在 VGG-16 上的性能比较

Table 3 Performance comparison between proposed method and other methods on VGG16

Method	Accuracy /%	FLOPs /MB	Parameters /MB
VGG-16[6]	93.96	313.73	14.98
Li[20]	93.40	206.00	5.40
HRank[18]	93.43	145.61	2.51
Proposed method(0.75)	93.65	135.79	2.11
HRank[18]	92.34	108.61	2.64
Proposed method(0.85)	93.14	94.35	1.53

表 4 展示了在 ResNet-56 上的结果。与 Li 等[20]的方法、GAL[12]和 HRank[18]的比较验证了所提方法的有效性。所提方法在 ResNet-56 上达到 90.84% 准确率的基础上显著减少了 FLOPs 和参数量。与其他方法相比，所提方法在最小的 FLOPs 和参数量的情况下达到了最高的准确率。特别是，与 He 等[11]的方法相比，所提方法只使用了其一半的 FLOPs，仍然提高了 0.04 个百分点的准确率。

表 4 所提方法与其他方法在 ResNet-56 上的性能比较

Table 4 Performance comparison between proposed method and other methods on ResNet-56

Method	Accuracy /%	FLOPs /MB	Parameters /MB
ResNet-56	93.26	125.49	0.85
He[11]	90.80	62.00	
GAL[12]	90.36	49.99	0.29
HRank[18]	90.72	32.52	0.27
Proposed method	90.84	29.83	0.18

表 5 展示了所提方法与 GAL[12]、HRank[18]和 Li 等[20]的方法在 GoogLeNet 上的比较结果。与原始网络相比，所提方法减少了 59% 的 FLOPs 和 52% 的参数量，但仅降低了 0.21 个百分点的准确率。与 GAL[12]和 Li[20]相比，所提方法在最小的 FLOPs 和参数量的情况下达到了最高的准确率。与 HRank[18]相比，所提方法在减少 0.07 MB FLOPs 的基础上，将准确率提高了 0.31 个百分点，整体模型仅增加了 0.2 MB 的参数量。

表 5 所提方法与其他方法在 GoogLeNet 上的性能比较

Table 5 Performance comparison between proposed method and other methods on GoogLeNet

Method	Accuracy /%	FLOPs /MB	Parameters /MB
GoogLeNet	95.05	1.52	6.15
GAL[12]	93.93	0.94	3.12
Li[20]	94.54	1.02	3.51
HRank[18]	94.53	0.69	2.74
Proposed method	94.84	0.62	2.94

表 6 总结了 DenseNet-40 上的实验结果。与原始网络相比，所提方法在 FLOPs 和参数量分别减少 64% 和 73% 的情况下依旧保持 93.1% 的高准确率。所提方法与 Zhao 等[28]的方法相比达到了相近的准确率，但减少了 35.9% 的 FLOPs 和 33.3% 的参数量。与 GAL[12]和 HRank[18]相比，Zhao 等[28]的方法 FLOPs 分别减少了 21.9% 和 9.2%，参数分别减少了 37.8% 和 41.7%，但只损失了 0.5 个百分点的准确率。

表 6 所提方法与其他方法在 DenseNet-40 上的性能比较

Table 6 Performance comparison between proposed method and other methods on DenseNet-40

Method	Accuracy /%	FLOPs /MB	Parameters /MB
DenseNet-40	94.81	282	1.04
Zhao[28]	93.16	156	0.42
GAL[12]	93.53	128.11	0.45
HRank[18]	93.68	110.15	0.48
Proposed method	93.1	100	0.28

3.3 AUC 数据集参数设定及剪枝实验结果

实验中所采用的优化算法为 SGD 算法，初始学习率和总体迭代次数都分别设置为 0.1 和 200，每次训练的批次大小为 32。

表 7 展示了在 VGG-16、VGG-19 和 ResNet-56 上的实验结果。在 VGG-16 上，所提方法在 FLOPs 较原始网络减少 69%，参数量降低了 65% 的情况下，只降低了 0.14 个百分点的准确率。在 VGG-19 上，所提方法能够在 FLOPs 和参数量分别降低 14.96 MB 和 17.64 MB 的情况下仅降低 0.05 个百分点的准确率。在 ResNet-56 上，所提方法在 FLOPs 由 6.2 MB 减少为 2.36 MB，参数量由原始的 0.85 MB 压缩为 0.37 MB 的情况下仅降低 0.26 个百分点的准确率。

表 7 在 VGG-16、VGG-19 和 ResNet-56 上的实验结果

Table 7 Experimental results on VGG16, VGG-19, and ResNet-56

Method	Accuracy /%	FLOPs /MB	Parameters /MB
VGG-16	95.01	15.41	40.43
Proposed method	94.87	4.72	14.18
VGG-19	95.08	19.58	45.74
Proposed method	95.03	4.62	28.10
ResNet-56	93.47	6.2	0.85
Proposed method	93.21	2.36	0.37

4 结 论

提出了一种自适应动态的滤波器剪枝方法，该方法通过一个激活权值生成模块来生成每个卷积层中滤波器的激活值并且计算出每个滤波器的激活权值，通过求和得出网络中每个滤波器的整体激活权值，最后通过全局排序来选择重要的滤波器。使用所提方法在 CIFAR-10 数据集上对 VGG-16、ResNet-56、GoogLeNet 与 DenseNet-40 和在 AUC 数据集上对 VGG-16、VGG-19 与 ResNet-56 进行剪枝的实验结果表明，该方法在保持准确率、减少 FLOPs 和参数量等方面具有良好的效果。未来的工作将继续从动态平衡 FLOPs 和参数量的角度来进一步优化所提方法。

参 考 文 献

[1] 王朝晖，康欢，陈多芳，等. 轻量化深度网络辅助于无透镜计算显微图像的细胞分类[J]. 中国激光，2022，49(5)：0507204.
Wang Z H, Kang H, Chen D F, et al. Lightweight deep learning network assisted cell classification using lensless computational microscopic imaging data[J]. Chinese Journal of Lasers, 2022, 49(5): 0507204.

[2] Li X X, Yu L Y, Yang X C, et al. ReMarNet: conjoint relation and margin learning for small-sample image classification[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 31(4): 1569-1579.

[3] Luo X F, Hu H F. Selected and refined local attention module for object detection[J]. Electronics Letters, 2020, 56(14): 712-714.

[4] 刘荻，张焱，赵琰，等. 基于特征重聚焦网络的多尺度近岸舰船检测[J]. 光学学报，2021，41(22)：2215001.
Liu D, Zhang Y, Zhao Y, et al. Multi-scale inshore ship detection based on feature re-focusing network[J]. Acta Optica Sinica, 2021, 41(22): 2215001.

[5] 国强，彭龙. 基于三维卷积神经网络与超像素分割的高光谱分类[J]. 光学学报，2021，41(22)：2210001.
Guo Q, Peng L. Hyperspectral classification based on 3D convolutional neural network and super pixel segmentation [J]. Acta Optica Sinica, 2021, 41(22): 2210001.

[6] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[EB/OL]. (2014-09-04)[2021-04-04]. https://arxiv.org/abs/1409.1556.

[7] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 770-778.

[8] Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions[C]//2015 IEEE Conference on Computer Vision and Pattern Recognition, June 7-12, 2015, Boston, MA. New York: IEEE Press, 2015.

[9] Huang G, Liu Z, van der Maaten L, et al. Densely connected convolutional networks[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 2261-2269.

[10] 高晗，田育龙，许封元，等. 深度学习模型压缩与加速综述[J]. 软件学报，2021，32(1)：68-92.
Gao H, Tian Y L, Xu F Y, et al. Survey of deep learning model compression and acceleration[J]. Journal of Software, 2021, 32(1): 68-92.

[11] He Y H, Zhang X Y, Sun J. Channel pruning for accelerating very deep neural networks[C]//2017 IEEE International Conference on Computer Vision, October 22-29, 2017, Venice, Italy. New York: IEEE Press, 2017: 1398-1406.

[12] 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), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 2785-2794.

[13] Xu X X, Zou Q, Lin X, et al. Integral knowledge distillation for multi-person pose estimation[J]. IEEE Signal Processing Letters, 2020, 27: 436-440.

[14] Song H, Mao H Z, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding[EB/OL]. (2015-10-01)[2021-04-05]. https://arxiv.org/abs/1510.00149.

[15] Zhang J T. Seesaw-Net: convolution neural network with uneven group convolution[EB/OL]. (2019-05-09)[2021-04-05]. https://arxiv.org/abs/1905.03672v5.

[16] Sun K, Li M J, Liu D, et al. IGCV3: interleaved low-rank group convolutions for efficient deep neural networks [EB/OL]. (2018-06-01)[2021-04-05]. https://arxiv.org/abs/1806.00178.

[17] 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, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 9194-9203.

[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), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 1526-1535.

[19] Zhang T Y, Ye S K, Zhang K Q, et al. A systematic DNN weight pruning framework using alternating direction method of multipliers[M]//Ferrari V, Hebert

M, Sminchisescu C, et al. Computer vision-ECCV 2018. Lecture notes in computer science. Cham: Springer, 2018, 11212: 191-207.

[20] Li H, Kadav A, Durdanovic I, et al. Pruning filters for efficient convnets[EB/OL]. (2016-08-31) [2021-04-05]. https://arxiv.org/abs/1608.08710v1.

[21] Zhuang B H, Shen C H, Tan M K, et al. Towards effective low-bitwidth convolutional neural networks[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 7920-7928.

[22] Gao X T, Zhao Y R, Dudziak L, et al. Dynamic channel pruning: feature boosting and suppression[EB/OL]. (2018-10-12)[2021-04-05]. https://arxiv.org/abs/1810.05331v2.

[23] Ma X, Guo J D, Tang S H, et al. Learning connected attentions for convolutional neural networks[C]//2021 IEEE International Conference on Multimedia and Expo, July 5-9, 2021, Shenzhen, China. New York: IEEE Press, 2021.

[24] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[J]. Handbook of Systemic Autoimmune Diseases. 2009, 1(4): 1-60.

[25] Nair V, Hinton G E. Rectified linear units improve restricted Boltzmann machines[C]//Proceedings of the 27th International Conference on Machine Learning, June 21-24, 2010, Haifa, Israel. Madison: Omnipress, 2010: 807-814.

[26] Han J, Moraga C. The influence of the sigmoid function parameters on the speed of backpropagation learning[M]// Mira J, Sandoval F. From natural to artificial neural computation. Lecture notes in computer science. Heidelberg: Springer, 1995, 930: 195-201.

[27] Abouelnaga Y, Eraqi H M, Moustafa M N. Real-time distracted driver posture classification [EB/OL]. (2017-06-28)[2021-04-06]. https://arxiv.org/abs/1706.09498v2.

[28] Zhao C L, Ni B B, Zhang J, et al. Variational convolutional neural network pruning[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 2775-2784.