文章编号:1005-0523(2024)06-0112-09

基于滤波衰减的自知识蒸馏压缩算法

熊李艳,黄佳文,黄晓辉,陈庆森

(华东交通大学信息与软件工程学院,江西 南昌 330013)

摘要:【目的】为了解决模型剪枝后性能损失严重的问题,提出了一种基于滤波衰减和自知识蒸馏的压缩算法。**【方法】**文章通过滤波衰减机制来保留冗余滤波器的信息,进而缩小剪枝前后的模型差异,降低剪枝导致的性能损耗。同时,在剪枝过程中引入一个退火衰减函数,使得滤波器的衰减呈现动态变化,进而能够快速高效地搜索模型的最佳子结构,提高模型的收敛速度。此外,还利用自知识蒸馏技术在预训练模型和压缩模型之间进行知识转移。**【结果】**结果表明,该压缩算法在减少VGG-16模型37.3%FLOPs的条件下,将模型精度提升了0.12个百分点。**【结论】**该方法能够为卷积神经网络提供一种更稳定、更高效的模型压缩方法。

关键词:卷积神经网络;滤波器剪枝;图像分类;知识蒸馏

中图分类号:TP391.4 **文献标志码:**A

本文引用格式:熊李艳,黄佳文,黄晓辉,等. 基于滤波衰减的自知识蒸馏压缩算法[J]. 华东交通大学学报,2024,41(6):112-120.

Self-Knowledge Distillation Compression Algorithm Based on Filter Attenuation

Xiong Liyan, Huang Jiawen, Huang Xiaohui, Chen Qingsen

(School of Information and Software Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract:【Objective】To address the severe performance loss after model pruning,a compression algorithm based on filter attenuation and self-knowledge distillation is proposed. **[Method]** This method utilized a filter attenuation mechanism to preserve the information of redundant filters, thereby minimizing the model difference before and after pruning, and reducing the performance loss caused by pruning. Meanwhile, an annealing attenuation function was introduced during the pruning process to dynamically change the attenuation of filters, enabling a fast and efficient search for the optimal substructure of the model and improving the convergence speed of the model. Additionally, self-knowledge distillation was employed for knowledge transfer between the pre-trained model and the compressed model. **[Result]** The results show that this compression algorithm improves the model accuracy by 0.12 percentage points while reducing the FLOPs of the VGG-16 model by 37.3%. **[Conclusion]** This method provides a more stable and efficient model compression approach for convolutional neural networks.

Key words: convolutional neural networks; filter pruning; image classification; knowledge distillation

Citation format: XIONG L Y, HUANG J W, HUANG X H, et al. Self-knowledge distillation compression

收稿日期:2024-03-15

基金项目:国家自然科学基金项目(62067002,61967006,62062033);江西省交通厅科技项目(2022X0040)

algorithm based on filter attenuation[J]. Journal of East China Jiaotong University，2024，41(6)：112-120.

【研究意义】近些年来，卷积神经网络（comvolutional nearal network，CNN）备受学术界和工业界的关注。事实证明卷积神经网络[1]在广泛的计算机视觉应用中非常有效。然而，将这些计算量庞大的网络直接应用于机器人、自动驾驶汽车和移动式设备等资源受限的环境时，难以在保持高性能的同时具有高效的计算效率。因此，对深度神经网络进行模型压缩具有非常重要的现实意义。

【研究进展】虽然深度神经网络通常具有大量的模型参数和计算量，但部分研究学者表示，删除网络中超过90%的模型参数并不会使模型的性能大幅下降[2]。这说明当前的神经网络具有大量的冗余结构，筛选并有效剔除这些冗余结构可以大大提高模型的性能和效率。因此，许多模型压缩与加速方法相关技术被相继提出。通过这些技术压缩后的小模型具有响应速度快、占用存储小和能源消耗低等优点，从而在各种生活场景中得到广泛应用。学者们一直在探索缩小网络模型规模的方法，采用知识蒸馏[3-5]、轻量化结构设计[6]和模型剪枝[7-8]等技术，作为加速模型推理、降低网络复杂性以及在低资源设备上部署神经网络的重要工具。Aghasi等[9]将剪枝等效为一个凸优化问题，在每个卷积层中求解合适的稀疏集，以减少网络中不必要的连接，同时确保输入和输出不变。Chen等[10]提出Octave卷积，将输入空间维度特征分解为高频和低频两个分量分别进行卷积，利用低频的结构特性减少存储和计算开销。何俊杰[11]提出一种基于特征相关性分析的通道剪枝算法。

【关键问题】通道剪枝方法作为最常用的压缩方法之一，旨在识别和消除神经网络中的冗余连接。目前大多数通道剪枝方法有两种处理策略：一是直接移除冗余滤波器然后进行微调；二是将冗余滤波器的权重置为零，继续进行训练。然而，这两种策略都存在一定的缺陷。冗余滤波器含有一定的信息量，直接将这部分信息进行移除是不明智的。因为在剪枝过程中，信息的损失是不可逆的，冗余滤波器被移除时会导致剪枝前后模型的性能差异。如何缩小这种差异是保证压缩后模型性能的关键。

【创新特色】为了解决上述问题，提出了一种新颖的剪枝方法，该方法是一种基于滤波衰减的自知识蒸馏压缩算法（filter decay and self-knowledge distillation，FD-SKD）。在训练过程中，它并不直接消除冗余滤波器或者将其权重置为零，而是以逐步衰减的方式保留冗余滤波器中有价值的信息，最终将模型压缩到预定的目标大小，并且通过与自知识蒸馏方法联合，可以最大限度地利用预训练模型的基础信息。图1展示了传统的剪枝方法与FD-SKD的不同之处。传统剪枝通常直接将模型中的冗余滤波器删除或者置为零，而FD-SKD则在剪枝过程中保留了冗余滤波器的部分信息，减小了剪枝前后的模型差异，这使得剪枝后的模型具有与原模型更相近的性能。

1　问题建模

一个深度CNN网络可以用参数表示为

$$\boldsymbol{W}=\left\{\boldsymbol{W}^l=\left[\boldsymbol{w}_1^l,\boldsymbol{w}_2^l,\cdots,\boldsymbol{w}_{C_{\text{out}}^l}^l\right]\in R^{C_{\text{out}}^l\times C_{\text{in}}^l\times K^l\times K^l}\right\} \tag{1}$$

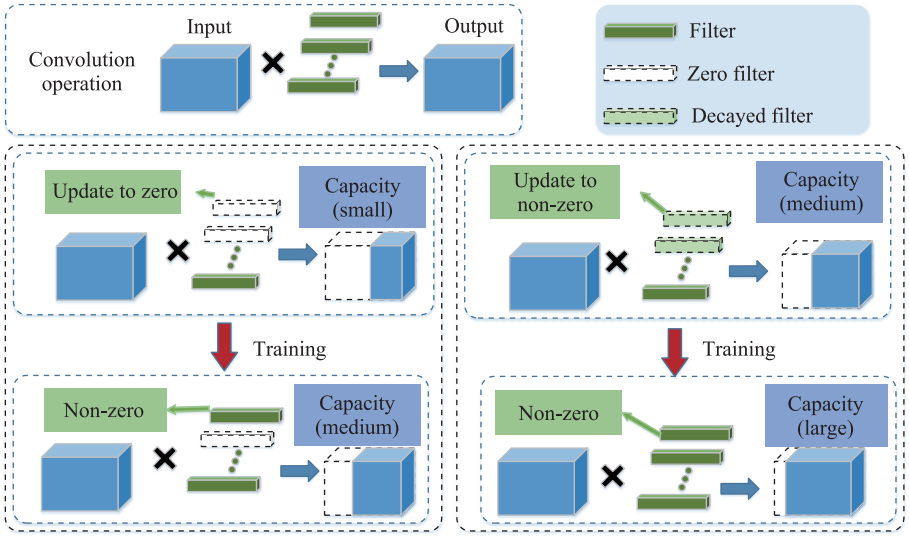
式中：$\boldsymbol{W}^l$ 为第 l 层的连接权重矩阵；L 为网络的总层数，$1\leqslant l\leqslant L$；C_{out}^l 和 C_{in}^l 分别为第 l 个卷积层的输出通道数和输入通道数；K^l 为第 l 个卷积层的卷积核大小。假定第 l 层的输入和输出分别为 I^l 和 O^l，且 $I^l\in C_{\text{in}}^l\times H_{\text{in}}^l\times W_{\text{in}}^l$ 和 $O^l\in C_{\text{out}}^l\times H_{\text{out}}^l\times W_{\text{out}}^l$，$H_*^l$ 和 W_*^l 分别为中间特征图的高和宽。那么第 l 层的卷积操作可以被写为

$$O_i^l=f(I^l,W_i^l),\ 1\leqslant i\leqslant C_{\text{out}}^l \tag{2}$$

式中：f 为卷积运算。假定 $F=\{F^1,F^2,\cdots,F^L\}$ 为整个网络中滤波器的集合，其中 F^l 是第 l 层的滤波器集合，也就是 $F^l=\{\boldsymbol{w}_1^l,\boldsymbol{w}_2^l,\cdots,\boldsymbol{w}_{C_{\text{out}}^l}^l\}$，$\boldsymbol{w}_i^l$ 为第 l 层中第 i 个滤波器权重矩阵。把 F 分为两个子集：保留滤波器子集 K 与被剪枝滤波器子集 P，得到

$$K\bigcup P=F,\quad K\bigcap P=\varnothing \tag{3}$$

剪枝的目标就是在既定的稀疏约束下最小化损失函数，给定一个数据集 $D=\left\{\left(x_n,y_n\right)\right\}_{n=1}^N$，其中 x_n 表示第 n 个输入数据，y_n 是与其相关的输出。最终



(a) Previous pruning (b) FD-SKD

图1 传统剪枝对比FD-SKD

Fig. 1 Comparison between previous pruning and FD-SKD

的约束优化问题就能够被表述为

$$\min_K L(K,D)=\min_K \frac{1}{N}\sum_{n=1}^{N}L\left(K;(x_n,y_n)\right),\quad \text{s.t.:}\ \frac{|K|}{|F|}=1-p \tag{4}$$

式中：$L(\bullet)$为损失函数；$|\bullet|$为滤波器集合的基数；p为模型的剪枝率。

2 滤波衰减

2.1 滤波器筛选

假设在第l个卷积层的剪枝率为p_l，那么需要在第l层选择$p_lC_{\text{out}}^l$个滤波器进行剪枝。对于冗余滤波器的选择，有很多不同的标准可以衡量滤波器的重要性。例如，p-范数，几何中位数(geometric median，GM)。p-范数的计算式为

$$\left\|w_i^l\right\|_p=\sqrt[p]{\sum_{n=1}^{C_{\text{in}}^l}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\left|w_i^l(n,k_1,k_2)\right|^p} \tag{5}$$

对于几何中位数，其计算的表达式为

$$\left\|w_i^l\right\|_{\text{GM}}=\sum_{j=1}^{C_{\text{out}}^l}\left\|w_i^l-w_j^l\right\|_2 \tag{6}$$

式中：$\|\bullet\|$是欧几里得距离。根据这类筛选标准，可以通过计算获得滤波器的重要性分数$\boldsymbol{G}^l=\{g_i^l,i\in[1,C_{\text{out}}^l]\}$，其中$\boldsymbol{G}^l$是第$l$层滤波器的分数向量，$g_i^l$是第$l$层中第$i$个滤波器的分数。然后将这些分数从低到高排名，选择其中分数最低的前$p_lC_{\text{out}}^l$滤波器进行剪枝。具体而言，在第l层中，被剪枝的滤波器集合可以表示为

$$P^l=\left\{\hat{F}_1^l,\hat{F}_2^l,\cdots,\hat{F}_{p^lC_{\text{out}}^l}^l\right\} \tag{7}$$

式中：$\hat{F}_i^l$是第l层中分数排名第i小的滤波器。

2.2 滤波器衰减

大多数传统的滤波器剪枝方法会直接移除这些不重要的滤波器，或者将它们置为0，然后微调剪枝后的模型以适应新的结构并恢复性能。通常会为每个卷积层设置一个掩码本用于指示滤波器是否被剪枝，假设第l个卷积层的掩码为B^l，其表达式为

$$B_i^l=\begin{cases}0, & \text{if}\ \ \hat{F}_i^l\in P^l,\\ 1, & \text{if}\ \ \text{其它},\end{cases}\quad 1\leqslant i\leqslant C_{\text{out}}^l \tag{8}$$

当B_i^l等于0时，意味着第l层中的第i个滤波器应该被剪枝，否则应该保留。在这种情况下，式(2)可以改写为

$$O_i^l=f(I^l,W_i^l)*B_i^l,\quad 1\leqslant i\leqslant C_{\text{out}}^l \tag{9}$$

然而，与初始模型相比，这会导致剪枝后的模型损失大部分容量。同时，如果丢弃了大部分预训练信息，会导致模型性能明显下降。为了解决这个问题，使用滤波衰减策略对滤波器进行剪枝。具体而言，滤波衰减策略将滤波器的权重重置为原本的一定比例，而不是将其直接删除。因此，引入一个超参数d_{r}，称为衰减率，用于控制冗余滤波器中保留的权重比例。然后式(8)可以表示为

$$\hat{B}_i^l=\begin{cases}d_{\text{r}}, & \text{if}\ \ \hat{F}_i^l\in P^l,\\ 1, & \text{if}\ \ \text{其它},\end{cases}\quad 1\leqslant i\leqslant C_{\text{out}}^l \tag{10}$$

式中：d_r 的取值范围为[0, 1]。图2展示了 d_r 对训练过程的影响：当 d_r 设定较大时（例如 $d_r=0.8$），模型收敛较慢，需要训练很长时间才能达到预定的压缩目标；当 d_r 设定较小时（例如 $d_r=0.1$），虽然模型的收敛速度很快，但会导致剪枝模型与初始模型之间的结构差异过大从而损坏模型性能。当大量的滤波器被剪枝时，这个问题变得更加严重。在这种情况下，很难找到一个能够平衡收敛速度和模型性能的阈值。

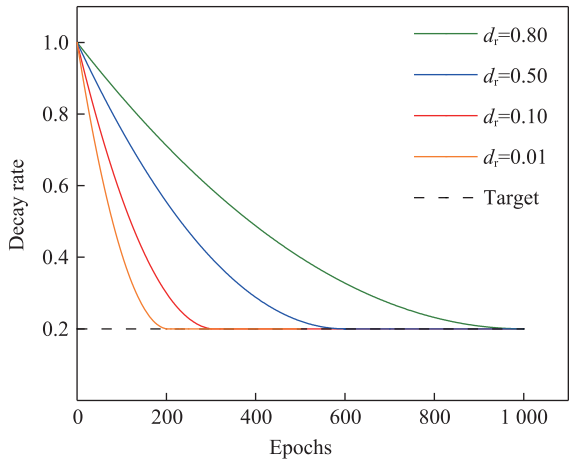


图2　衰减率对训练周期的影响
Fig. 2　Effect of decay rate on Epochs

2.3　退火衰减

为了解决上述问题，提出了一种退火衰减策略，使得衰减率随着训练过程而动态变化。具体而言，引入一个退火函数 $T(n)$，衰减率通过 $T(n)$ 可以表示为

$$d_r(n)=\frac{1}{1+e^{-T(n)}},\quad 1\leqslant n\leqslant N \tag{11}$$

式中：n 和 N 分别表示当前训练时期和总训练时期；退火函数的温度指数 T 初始化为一个较高的值 $T=T_0$。在训练到第 n 个时期时，温度指数 T 将退火到 $T(n)=\xi(n)T_0$，其中 $\xi(n)$ 表示温度退火方案，如线性退火或Sigmoid退火方案。该算法选用线性退火方案 $\xi(n)=1-(2\times n)/N$。与此同时，衰减率随着训练过程逐渐减小并趋近于零。然后，式(10)可以表示为

$$\hat{B}_i^l(n)=\begin{cases}\dfrac{1}{1+e^{-T(n)}}, & \hat{F}_i^l\in P^l(n)\\ 1, & \text{其它}\end{cases},\quad 1\leqslant i\leqslant C_{\text{out}}^l, 1\leqslant n\leqslant N \tag{12}$$

在训练过程开始时，d_r 设置在一个较高水平，保留了大部分被修剪滤波器的信息。随着训练的进行，冗余的滤波器包含的信息越来越少。因此通过降低 d_r，能够使网络的收敛速度开始加快，以便迅速达到预期的压缩目标。修剪模型后，通过微调以使模型适应新的网络结构，并持续迭代这一过程。当冗余滤波器的权重降为零并停止变化时，可以安全地移除这些冗余滤波器，获得压缩后的模型结构，然后对模型进行再训练使其适应新的结构，得到最终的压缩模型。实验中剪枝的训练次数占总训练次数的50%。卷积层的滤波衰减过程如图3所示。

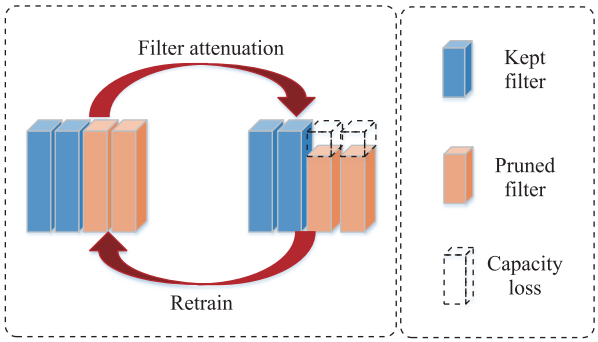


图3　滤波器的衰减过程
Fig. 3　Process of filter attenuation

2.4　自知识蒸馏

自知识蒸馏是一种独特的蒸馏方式，与传统的知识蒸馏方法截然不同。它的特点在于模型同时扮演教师网络和学生网络，并进行自我学习。这种方法的优势在于不需要模型过于庞大和性能过好，而是通过自身的预训练知识进行学习并更新参数。本文中介绍的自知识蒸馏方法采用了滤波衰减和知识蒸馏相结合的网络架构，不需要依赖庞大的教师网络，而是以并行方式相互学习并更新参数，从而提升了网络的准确度。本节中提出了一种滤波衰减与自知识蒸馏相结合的方法，这种方法充分利用了滤波器的预训练信息。具体来说，将预训练模型用作教师模型，学生模型在剪枝之前被设置与预训练模型相同。在剪枝过程中，学生模型将通过滤波器衰减方法减小到压缩目标，同时教师模型将通过其输出的软标签对学生模型的输出结果进行修正。FD-SKD整体的网络模型结构如图4所示。

在知识蒸馏算法中，假设输入样本为 x 及对应的真实标签为 $y\in R^{1\times N}$，经过教师模型T和学生模

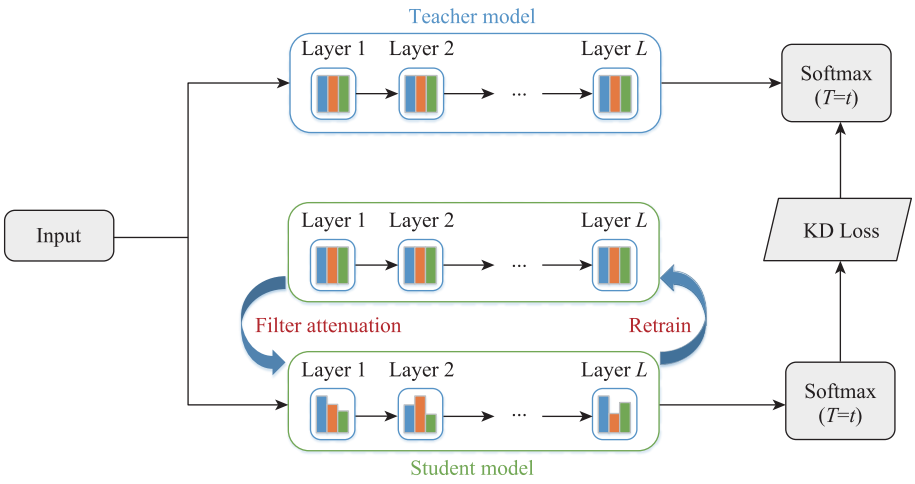


图4 FD-SKD模型结构

Fig. 4 Structure of the FD-SKD model

型S后，可以得到如式(13)与式(14)所示的类别预测概率 $p_i(x,t)$ 和 $q_i(x,t)$ 分别为

$$p_i(x,t)=\frac{\exp\left(T_i(x)/t\right)}{\sum_{j=1}^{N}\exp\left(T_j(x)/t\right)},\quad i=1,2,\cdots,N \tag{13}$$

$$q_i(x,t)=\frac{\exp\left(S_i(x)/t\right)}{\sum_{j=1}^{N}\exp\left(S_j(x)/t\right)},\quad i=1,2,\cdots,N \tag{14}$$

2.5 损失函数

自知识蒸馏涉及两种损失：传统的交叉熵损失和知识蒸馏损失。知识蒸馏损失使用KL散度(Kullback-Leibler divergence)来衡量教师模型和学生模型预测之间的差异。因此，总损失函数可以表示为

$$L_{\text{all}}=\lambda L_{\text{cla}}+(1-\lambda)\underbrace{\text{KL}(q^{\text{t}},q^{\text{s}})}_{T=t} \tag{15}$$

式中：L_{cla} 为用于训练的交叉熵分类损失；q^{s} 和 q^{t} 分别为学生模型和教师模型输出的软标签；KL(•)为KL散度；参数 λ 是为了在训练过程中平衡这两种损失的比例。

3 实验结果与分析

3.1 数据集介绍与处理

在CIFAR-10数据集上对提出的FD-SKD算法进行测试以及验证。CIFAR-10是一个由Alex Krizhevsky和Ilya Sutskever整理，Geoffrey Hinton指导完成的数据集，其中包含了10个不同类别的彩色图像。每个类别包含了6 000张尺寸为32×32的RGB彩色图片，这些类别分别是飞机、鸟、汽车、猫、狗、鹿、蛙、船、马和卡车。CIFAR-10数据集的特点是其中包含了真实世界中的物体，这些物体的比例和特征各不相同，给识别任务带来了更大的挑战。

CIFAR-10数据集被分为了50 000张用于训练的图片和10 000张用于测试的图片。另外，数据集分为训练批次5个和测试批次1个，每个批次的图片数量均为10 000张。测试批次在每个分类中随机选择1 000张图像，而训练批次则随机选择图像，1个训练批次内包含的单个类别数量为5 000。为了更好的提取特征，进行数据增强，使用了标准的数据预处理方法，包括随机裁剪、随机水平翻转和将图片填充至40×40的尺寸等操作。

3.2 实验参数

文中所有实验在PyTorch框架下进行，通过NVIDIA 3080Ti GPU进行模型训练。实验中的模型参数设置如表1所示。

3.3 评价指标

模型压缩有3个主要的评估指标，分别是参数数量、计算量和准确度。参数数量反映了网络的规模大小，因为当前的神经网络通常含有大量参数，这些参数需要占用存储空间。卷积层参数数量的计算公式为

$$Params=\left(2*C_{\text{in}}*K^2+1\right)*C_{\text{out}} \tag{16}$$

式中：C_{in} 和 C_{out} 分别为输入通道数和输出通道数；K 为卷积核的大小。计算量表示模型的复杂度，通常使用浮点运算次数(floating point operations per second，FLOPs)作为衡量标准。卷积层的浮点运算

次数计算公式为

$$FLOPs = H * W * \left(2 * C_{\text{in}} * K^2 - 1\right) * C_{\text{out}} \quad (17)$$

式中：H 和 W 分别为特征图的高和宽。准确度表示网络的预测能力，通常是在数据集上的预测精度。一般使用Top-1准确率来衡量，即预测成功的样本数量占总样本数量的比例，其计算公式为

$$Acc = \frac{T_{\text{p}}}{T_{\text{p}} + F_{\text{p}}} \quad (18)$$

式中：T_{p} 和 F_{p} 分别表示预测成功和预测失败的样本数量。一个出色的剪枝算法应当在保证高准确度的同时有效降低参数数量和提高计算速度。但是，在应用场景中，如何平衡准确度、参数量和计算速度是一个难题。当前的趋势是，由于很多工作人员需要在受限设备上进行实时计算，计算速度的重要性日益凸显。

表1　模型参数设置

Tab.1　Parameter settings of the model

Parameter	Value
Epoch	250
Batch size	128
Learning rate	0.01
Pruning rate	0.3
Momentum	0.9
Optimizer	SGD
Weight decay	1e-4
Dropout	0.5
Activation function	ReLU
Initial temperature	10
Distillation temperature	5
Balance value of loss	0.9

3.4　模型验证与结果分析

为了更好的验证FD-SKD的有效性，实验中仅使用滤波衰减的方法用FD表示，使用滤波衰减与自知识蒸馏结合的方法用FD-SKD表示。

3.4.1　在ResNet上的实验结果

对于ResNet，在CIFAR-10数据集上进行了深度为20、32、56、110的FD-SKD算法测试，实验结果如表2所示。

FD-SKD方法在实验结果中被证实是有效的。例如，PFEC在ResNet-56上达到了93.06%的准确率，但是FD-SKD在减少了一半以上的FLOPs的同时，达到了93.62%的准确率。与采用不同的滤波器选择标准在不同层的软剪枝方法LFPC相比，FD-SKD在ResNet-110上的表现仍然更好，准确率进一步提高了1.16%。在采用相同的滤波器选择标准 L_2 范数的情况下，FD-SKD在所有实验中都优于软剪枝方法SFP。例如，SFP对具有1.22 M参数的ResNet-110进行加速时，准确率仅为92.97%，而FD-SKD的准确率是94.15%。FD-SKD的有效性在于其可以保留冗余滤波器中的部分信息，不同于传统方法中直接进行丢弃。

表2　ResNet的实验结果

Tab.2　Experimental results of ResNet

Depth	Method	Params/M	FLOPs/M	Acc/%
20	PFS[12]	—	24.7	91.14
20	PGMPF[13]	—	19.3	91.14
20	FPGM[14]	0.19	24.3	91.09
20	LPSR[15]	0.19	24.3	91.62
20	FD	0.17	22.0	92.06
20	FD-SKD	0.17	22.0	92.17
32	MIL[16]	—	47.0	90.74
32	SFP[17]	0.32	40.3	90.08
32	FPGM[14]	0.32	40.3	91.93
32	FD	0.32	40.3	92.45
32	FD-SKD	0.32	40.3	92.85
56	WACP[18]	0.35	46.3	93.17
56	NISP[19]	0.49	71.6	92.99
56	DAIS[20]	—	36.4	93.53
56	AFPrune[21]	—	60.8	93.45
56	FD	0.30	32.4	93.47
56	FD-SKD	0.30	32.4	93.62
110	AFPrune[21]	—	116	94.08
110	FPGM[14]	1.22	121	93.85
110	LFPC[22]	1.03	101	93.07
110	HRank[23]	1.07	106	93.36
110	FD	1.22	121	93.98
110	FD-SKD	1.22	121	94.15

3.4.2　在VGG上的实验结果

对于VGG，在CIFAR-10数据集上同样进行了FD-SKD算法测试，实验结果如表3所示。结果表明FD-SKD取得了VGG-16的最高准确率94.02%，并具有高效的压缩比。FD在未使用自知识蒸馏时同样优于其他方法，这归因于滤波衰减策略，该策略保留了部分预训练信息，使得模型准确性下降过程更加平滑。此外，在训练过程中引入自知识蒸馏会产生比没有使用时更好的结果。这种性能改进是因为教师模型能够向学生模型传递知识，从而防止

模型性能的急剧下降。因此，FD-SKD表现出比现有最先进方法更优越的特点。

3.4.3 收敛速度与性能的平衡性验证

为了验证FD-SKD方法能够平衡收敛速度和性能，在不同衰减率下进行验证实验。将ResNet-32在CIFAR-10数据集上的衰减速率固定为0.20，0.02和0。实验结果如表4所示。结果显示，当衰减速率较高（$d_r=0.20$）时，准确率较高，但需要更多的训练周期（周期为762）才能收敛到压缩目标。当衰减速率较低（衰减率为0.02）时，模型准确率较低（准确率为92.32%），但很快收敛到压缩目标。在这两种情况下，很难找到一个阈值在模型的准确率和收敛速度之间取得平衡。而FD-SKD可以轻松解决这个问题。与基准模型相比，FD-SKD可以在100个训练周期内完成训练过程，并且模型准确率提升了0.01个百分点。

表3 VGG-16的实验结果

Tab.3 Experimental results of VGG-16

Method	Acc/%	Acc drop/%	FLOPs/%
SFP[17]	93.33	0.57	37.3
HRank[23]	93.43	0.53	46.0
FiltDivNet[24]	92.79	0.72	30.2
CHIP[25]	93.86	0.10	41.9
PFEC[26]	93.40	0.15	65.8
WACP[18]	93.86	0.10	36.8
AFPruner[21]	93.67	0.23	40.4
CLR-RNF[27]	93.32	0.58	25.9
FD	93.94	−0.04	37.3
FD-SKD	94.02	−0.12	37.3

表4 不同衰减率下的实验结果

Tab.4 Experimental results of different decay rate

Decay rate	Epochs	Baseline/%	Acc/%	Difference/%
0.20	762	92.63	92.70	0.07
0.02	208	92.63	92.32	−0.31
0	83	92.63	91.89	−0.74
FD-SKD	100	92.63	92.64	0.01

3.5 消融实验结果分析

3.5.1 剪枝间隔的影响

在实验中，剪枝操作的间隔是一个时间步。不同的剪枝间隔可能会产生不同的实验结果。故有必要对不同的剪枝间隔进行实验以检验其对结果的影响。图5展示了在CIFAR-10上用不同的剪枝间隔训练ResNet-56的结果。由图5可知，当剪枝间隔大于5时，模型的准确率没有明显变化。这是因为在网络训练了5个以上的时间步后，模型逐渐稳定，准确率基本不变。当剪枝间隔小于5时，准确率随着剪枝间隔的增加而下降。这是因为当总训练次数固定时，剪枝间隔减少，使得模型有更多的训练次数搜索更优的结构。如果将剪枝间隔设置为1，准确率将提高到93.6%。这意味着可以通过缩短模型的剪枝间隔以获得更好的结果。

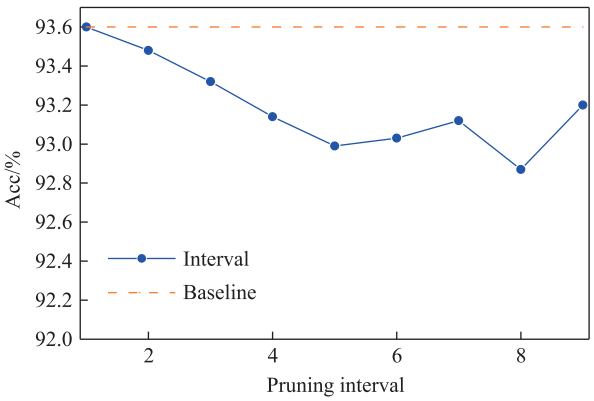


图5 剪枝间隔对准确率的影响

Fig. 5 Effect of pruning interval on accuracy

3.5.2 滤波衰减次数的影响

探讨滤波衰减次数对准确率的影响。图6展示了在CIFAR-10上用不同的滤波衰减次数训练ResNet-56的结果，图中Proportion表示滤波衰减次数占总训练次数的比例。由图可知，当Proportion小于0.4时，准确率浮动较小，性能稳定。这是因为衰减策略的使用频次不高，导致算法的搜索空间较小。当Proportion大于0.6时，准确率大幅下降。这是因为模型的剪枝次数过于频繁，使得模型不能及时更新参数以适应新的结构。当Proportion在（0.4，

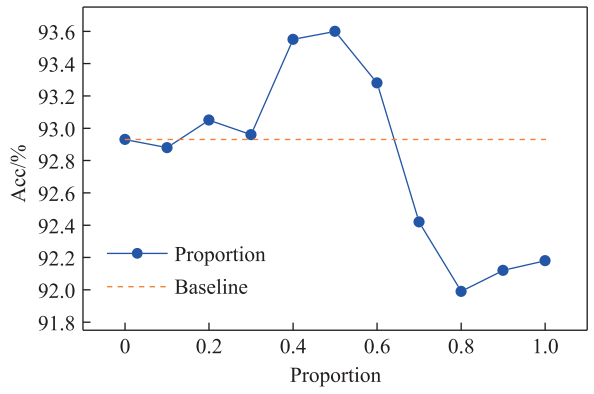


图6 滤波衰减比例对准确率的影响

Fig. 6 Effect of filter attenuation ratio on accuracy

0.6)内，模型能够获得很好的性能。尤其是当其等于0.5时，模型的准确率达到了93.6%。

3.5.3　滤波筛选准则的影响

探讨3种不同的滤波器选择标准对准确率的影响，3种标准分别为 L_1 范数、L_2 范数和几何中位数。图7展示了使用3种的滤波筛选准则在CIFAR-10上训练ResNet-56并在不同剪枝率下的实验结果。由图可知，当剪枝率低于0.2时，L_1 和 L_2 都提高了模型的准确率。在这种情况下，FD-SKD不仅压缩了模型，还提高了模型的精度。当剪枝率大于0.2而小于0.4时，3种标准都能表现出与基线模型近似的准确率。当剪枝率大于0.7时，L_2 与其他两种方法相比存在较大的性能差距。此外，模型的准确性呈现出明显的下降趋势，这表明模型不能仅追求压缩大小，还要考虑其性能的变化。综上，剪枝率在0.3~0.6更具有实际应用效果。

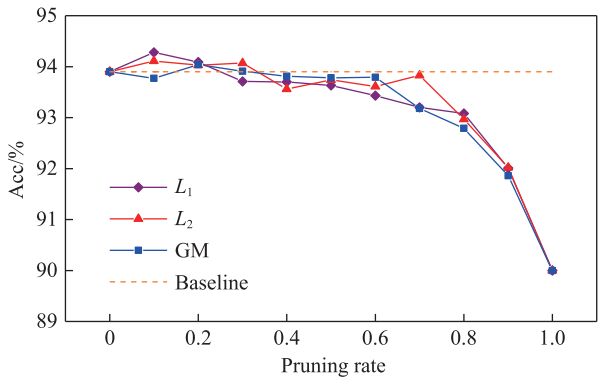


图7　滤波筛选准则对准确率的影响

Fig.7　Effect of filtering screening criteria on accuracy

3.5.4　退火方案的影响

将线性函数和Sigmoid函数分别作为退火函数在ResNet-56上进行实验。实验结果如表5所示，在FLOPs为41.1%时，线性函数表现出更好的实验结果。此时，Sigmoid函数实现了93.45%的准确率，而线性函数实现了93.61%的准确率。这表明FD-SKD在线性函数下可以实现更好的性能。如果从预训练模型开始剪枝，当FLOPs下降为28.4%时，与Sigmoid函数相比，使用线性函数准确率仍然可以提升0.15个百分点。

表5　不同退火方案的实验结果

Tab.5　Experimental results of different annealing schemes

Annealing function	Initpretrain	FLOPs/%	Acc/%
Sigmoid	N	41.1	93.45
Linear	N	41.1	93.61
Sigmoid	Y	28.4	93.50
Linear	Y	28.4	93.65

4　结论

1）提出了一种名为基于滤波衰减的自知识蒸馏压缩算法FD-SKD，用于加速深度卷积神经网络。相比于传统的剪枝算法，FD-SKD考虑了冗余滤波器的预训练信息，减小了剪枝前后的模型差异，使得模型更加高效、稳定。

2）滤波衰减策略能够在训练过程中动态控制衰减速率以保证模型的收敛速度和性能，解决了传统剪枝方法训练耗时长和性能低下的问题。更重要的是，FD-SKD可以应用于任何滤波器剪枝方法。

参考文献：

[1] 裴莹玲, 罗晖, 张诗慧, 等. 基于改进Faster R-CNN的高铁扣件检测算法[J]. 华东交通大学学报, 2023, 40(1): 75-81.
PEI Y L, LUO H, ZHANG S H, et al. High-speed railway fastener detection algorithm based on improved Faster R-CNN[J]. Journal of East China Jiaotong University, 2023, 40 (1): 75-81.

[2] HAN S, MAO H, DAILY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding[C]//San Juan: 2016 International Conference on Learning Representations (ICLR), 2016.

[3] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. Computer Science, 2015, 14 (7): 38-39.

[4] ROMERO A, BALLAS N, KAHOU S E, et al. FitNets: hints for thin deep nets[C]//San Diego: 2015 International Conferenceon Learning Representations (ICLR), 2015.

[5] CHEN T, GOODFELLOW I, SHLENS J. Net2Net: accelerating learning via knowledge transfer[C]//San Juan: 2016 International Conference on Learning Representations (ICLR), 2016.

[6] CHEN H, WANG Y, XU C, et al. Addernet:do we really need multiplications in deep learning?[C]//Seattle: 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[7] 王炯. 基于结构化剪枝的深度神经网络模型压缩方法研究[D]. 南京: 南京邮电大学, 2022.
WANG J. Research on deep neural network model compression method based on structured pruning[D]. Nanjing: Nanjing University of Posts and Telecommunic-

ations, 2022.

[8] GUO Y W, YAO A B, CHEN Y R. Dynamic network surgery for efficient dnns [C]//Barcelona: 2016 Advances in Neural Information Processing Systems (NIPS), 2016.

[9] AGHASI A, ABDI A, NGUYEN N, et al. Net-trim: convex pruning of deep neural networks with performance guarantee[C]//Long Beach: 2017 Advances in Neural Information Processing Systems (NIPS), 2017.

[10] CHEN Y, FAN H, XU B, et al. Drop an octave:reducing spatial redundancy in convolutional neural networks with octave convolution[C]//Los Angeles: 2019 IEEE International Conference on Computer Vision (CVPR), 2019.

[11] 何俊杰. 面向深度卷积神经网络的模型压缩与加速方法研究[D]. 杭州: 浙江大学, 2022.
HE J J. Research on compression and acceleration methods for deep convolutional neural networks[D]. Hangzhou: Zhejiang University, 2022.

[12] WANG Y, ZHANG X, XIE L, et al. Pruning from scratch [C]//New York: 2020 AAAI Conference on Artificial Intelligence(AAAI), 2020.

[13] CAI L, AN Z, YANG C, et al. Prior gradient mask guided pruning-aware fine-tuning[C]// Palo Alto: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022.

[14] HE Y, LIU P, WANG Z, et al. Filter pruning via geometric median for deep convolutional neural networks acceleration[C]//Los Angeles: 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[15] ZHANG K, LIU G. Layer pruning for obtaining shallower resnets[J]. IEEE Signal Processing Letters, 2022, 29: 1172-1176.

[16] DONG X, HUANG J, YANG Y, et al. More is less: a more complicated network with less inference complexity[C]//Hawaii: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[17] HE Y, KANG G, DONG X, et al. Soft filter pruning for accelerating deep convolutional neural networks[C]// Stockholm: 2018 International Joint Conference on Artificial Intelligence (IJCAI), 2018.

[18] DONG Z, DUAN Y, ZHOU Y, et al. Weight-adaptive channel pruning for CNNs based on closeness-centrality modeling[J]. Applied Intelligence, 2024, 54(1): 201-215.

[19] YU R, LI A, CHEN C F,et al. NISP: pruning networks using neuron importance score propagation[C]//Salt Lake: 2018 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018.

[20] GUAN Y, LIU N, ZHAO P, et al. DAIS: automatic channel pruning via differentiable annealing indicator search [J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(12): 9847-9858.

[21] XUE Y, YAO W, PENG S, et al. Automatic filter pruning algorithm for image classification[J]. Applied Intelligence, 2024, 54(1): 216-230.

[22] HE Y, DING Y, LIU P, et al. Learning filter pruning criteria for deep convolutional neural networks acceleration [C]//Seattle: 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

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

[24] LEI P, LIANG J, ZHENG T, et al. Compression of convolutional neural networks with divergent representation of filters[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(3): 4125-4137.

[25] SUI Y, YIN M, XIE Y, et al. CHIP: channel independence based pruning for compact neural networks[C]//Online: 2021 Advances in Neural Information Processing Systems (NIPS), 2021.

[26] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for effcient convNets[C]//Toulon: 2017 International Conference on Learning Representations (ICLR), 2017.

[27] LIN M, CAO L, ZHANG Y, et al. Pruning networks with cross-layer ranking & k-reciprocal nearest filters[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(11): 9139-9148.

通信作者：熊李艳（1968—），女，教授，硕士生导师，研究方向为交通大数据。E-mail：445935939@qq.com。

（责任编辑:吴海燕）