# 基于通道相似度熵的卷积神经网络裁剪

耿丽丽[1,2]，牛保宁[1*]

(1. 太原理工大学计算机科学与技术学院(大数据学院)，山西 晋中 030600；2. 山西财经大学实验实训中心，山西 太原 030006)

**摘　要：** 卷积神经网络(CNN)中包含大量滤波器，参数训练以及存储占用大量内存资源。裁剪滤波器是减小网络规模、释放内存、提高计算速度的有效方法。现有滤波器裁剪方法的主要问题是将滤波器权值作为孤立的数值计算，裁剪小权值滤波器，保留权值大的滤波器，忽视了部分小权值滤波器在特征提取过程中的重要性。通过分析滤波器通道之间的相似性，提出一种基于通道相似度的滤波器熵值计算方法(FEC)。针对滤波器结构特征，对权值张量进行均值压缩，并证明其合理性。先计算滤波器通道距离判断通道之间的相似性，再根据通道相似度计算滤波器熵，由熵值大小进行滤波器排序，删除一定比例熵值较小的滤波器。实验设计针对不同卷积层采用不同的裁剪比例，在 CIFAR10 以及 ImageNet 标准数据集上对 VGG-16 和 ResNet-34 网络进行裁剪。实验结果表明：在基本保持原始准确度的情况下，分别减少了约 94%和 70%的参数数量；在目标检测网络 SSD 上参数数量减少了 55.72%，平均精度均值(mAP)提高了 1.04 个百分点。

**关键词：** 卷积神经网络；通道相似度；熵；滤波器；裁剪

**源代码链接：** https://github.com/genglily/keras-pruning

**中图分类号：** TP181　　**文献标志码：** A　　**DOI：** 10.19678/j.issn.1000-3428.0068284

# Convolutional Neural Network Pruning Based on Channel Similarity Entropy

GENG Lili[1,2], NIU Baoning[1*]

(1. College of Computer Science and Technology(College of Big Data), Taiyuan University of Technology, Jinzhong 030600, Shanxi, China;
2. Experimental Center, Shanxi University of Finance and Economics, Taiyuan 030006, Shanxi, China)

**【Abstract】** Convolutional Neural Network (CNN) contain a large number of filters, which occupy significant memory resources for training and storage. Pruning filters is an effective method to reduce the scale of networks, free up memory, and enhance computing speed. A primary issue with existing filter pruning methods is that they calculate the weights of filters in isolation, preserving the filter with the largest weight while often overlooking the importance of smaller weights in feature extraction. To address this, a Filter Entropy Calculation(FEC) method based on channel similarity is proposed. This method involves compressing the weight tensor by its mean value, a process whose rationality is substantiated by the structural characteristics of the filter. The similarity between channels is assessed by calculating the distance of filter channels, and filter entropy is determined based on this similarity. Filters are then ranked by their entropies, and a specific proportion of filters with lower entropy is removed. The experimental design uses different pruning ratios for different convolutional layers. Networks such as VGG-16 and ResNet-34 are pruned using the CIFAR10 and ImageNet standard datasets. Experimental results indicate that while the original accuracy is largely preserved, the number of parameters is reduced by approximately 94% and 70%, respectively. Additionally, on the Single Shot multibox Detector(SSD) framework, parameters decreased by 55.72%, and the mean Average Precision(mAP) improved by 1.04 percentage points.

**【Key words】** Convolutional Neural Network(CNN); channel similarity; entropy; filter; pruning

## 0　引言

卷积神经网络(CNN)是目前计算机视觉领域的主流模型，主要应用在图像分类、物体识别、图像分割检测、人脸识别等任务上，其准确率一般远超其他类型的神经网络。经典的图像分类卷积神经网络包括 AlexNet[1]、VGG[2]、Inception v1、v2、v4[3-5]以及残差网络 ResNet[6]等。这些深度网络模型在进行图像分类时虽然具有较高准确度，但随着网络层数增多，参数量以及计算复杂度大幅增加，需要多个 GPU 并行，导致网络模型训练效率低，占用了大量内存资源。随着网络规模的不断扩大，CNN 的应用受到了诸多限制，很难部署在资源受限的设备上。为了解决这一问题，产生了许多裁剪 CNN 的方法。

**收稿日期：** 2023-08-24　　**修回日期：** 2023-10-25

**基金项目：** 山西省重点研发计划(201903D421007)。

**通信作者 E-mail：** * niubaoning@tyut.edu.cn

HAN 等[7]采用剪枝、量化以及哈夫曼编码等技术对深度神经网络进行压缩，引起了模型压缩的热潮。之后出现了多种压缩方法，包括传统剪枝方法[8-10]、基于优化算法的剪枝方法[11-13]、基于深度学习的剪枝方法[14-15]等。裁剪是在一个更大的训练模型中寻找一个精确的子网[16]。通过裁剪，可以降低网络规模、减少参数量以及提高网络训练效率，同时还能保持网络的稳定性和准确率。

在裁剪网络的过程中，如何判断网络的冗余参数是影响裁剪效果的核心问题。CNN 的参数大部分都来自卷积层中的滤波器，因此，裁剪大多是针对滤波器的删除操作。本文将滤波器结构相似性与熵结合，定义通道距离，设计张量降维方法，高效计算通道相似性以及相似度熵，根据相似度熵值大小进行滤波器排序，删除一定比例的小熵值滤波器，实现 CNN 裁剪。

## 1 相关工作

一般裁剪方法都是以权值大小作为滤波器重要性的判断依据，文献[17]直接计算滤波器权值的 L1 范数，认为 L1 范数较大的滤波器是重要的，L1 范数较小的滤波器是不重要的，可以删除。文献[18]提出一种计算熵的方法(EP)来评估滤波器的重要性。熵在信息论中是衡量信息量的值，熵值大意味着所含信息丰富。该方法通过计算卷积层的激活通道概率，得到通道的熵值，由此决定相应滤波器的重要性，整个裁剪过程使用固定压缩率，保留排序前端的滤波器，其与 Apoz[19]方法相比，不同的是不直接删除激活值为零的通道，而是计算了通道的熵值，按照熵值大小排序，使用固定压缩率进行裁剪。文献[20]采用皮尔逊相关系数计算滤波器之间的相似度(LFC)，通过迭代方式将 2 个相似度最大的滤波器分为一组，然后裁剪掉每组中的一个滤波器。文献[21]采用狄利克雷(Dirichlet)分布作为先验和后验来学习通道的重要性，并删除了重要性较小的通道。

然而，目前先进的滤波器裁剪方法有几个缺陷：1)对滤波器通过权值排序和设置裁剪阈值进行裁剪，忽略了网络的实际输出，这会导致一些含有重要信息的滤波器被删除；2)裁剪阈值很难确定，并且通常从尝试和错误中得到。由于深度网络训练的计算量大且耗时，并且所有层的权值都是共享的，因此很难确定合适的全局裁剪阈值[22]；3)许多裁剪方法在不同的网络层使用相同的裁剪比例，而不同网络层的参数冗余度不同，因此，采用相同裁剪比例会造成参数裁剪过度或者裁剪不足[16]。

笔者认为卷积神经网络特征提取的丰富性和差异性更为重要，许多被认为不重要的小权值滤波器在特征提取中可能也有重要贡献。因为滤波器在对输入数据进行卷积操作时，不同通道的同一空间位置的卷积输出结果相似性是极高的，所以滤波器通道之间的差异性决定了特征提取的多样性，而不是由权值大小决定。

基于上述分析，提出一种基于通道相似度的滤波器熵值计算方法(FEC)。滤波器通道相似度熵值大小代表了滤波器特征提取的多样性程度。根据熵值大小进行滤波器排序，删除一定比例熵值较小的滤波器，实现由滤波器特征提取功能排序的 CNN 裁剪。与 EP 方法不同，本文方法先度量滤波器通道之间相似度，再由通道相似度计算滤波器熵值。与 LFC 方法不同，本文方法计算滤波器通道之间的马尔可夫距离，通过距离判断通道相似性，最终根据通道相似度熵裁剪滤波器。裁剪过程中不使用固定裁剪比例，而是针对不同宽度的卷积层分别设置不同的裁剪比例。实验结果证明，采用分层裁剪比例更容易获得高准确度的子网，子网在网络规模与准确度的综合性能上表现更优。

## 2 滤波器通道相似度熵

CNN 将图像的原始数据作为输入进行处理，因此，它是更直接、更有效的图像处理技术[23]。CNN 卷积层的作用是提取图像中局部区域的特征，卷积层中不同的滤波器提取不同的特征信息。滤波器相似度熵反映了滤波器提取特征的多样性程度，熵值越大，提取特征差别越大。卷积层中保留熵值大的滤波器，即保证特征映射提取的多样性。基于这一思想，本节介绍滤波器通道相似度的计算方法，描述滤波器张量压缩方法以及通道距离计算方法，并阐述通道相似度熵的计算过程及卷积层滤波器裁剪规则。

### 2.1 滤波器张量压缩及通道距离计算

CNN 的主要网络结构是卷积层、池化层以及全连接层，其中卷积层和全连接层含有大量参数，网络裁剪即对这 2 种网络层的参数进行压缩。CNN 中卷积层的数量较多，压缩卷积层可以大幅减少网络参数从而减小规模。卷积层由多个滤波器构成，滤波器是一个三阶张量，维度为高、宽、通道数。滤波器张量中数值即权值，$\boldsymbol{W} \in \mathbb{R}^{k\times k\times n}$ 表示滤波器张量，其中一个张量切片 $k\times k$ 表示滤波器的感受野，$n$ 为通道数。感受野决定了滤波器每次卷积的区域大小，而通道数与输入数据通道数一致，如输入是三通道彩色图片，则滤波器通道

数 $n=3$。相似度通常是通过距离计算来获得的，滤波器通道是一个二维矩阵，不同通道之间的距离定义为一个距离矩阵。

**定义 1** 通道距离矩阵。

对于滤波器 $\boldsymbol{W}$，$\boldsymbol{W}_{k\times k}^{n}$ 和 $\boldsymbol{W}_{k\times k}^{n'}$ 表示通道 $n$ 和 $n'$ 的权值矩阵。由权值矩阵同一空间位置的权值绝对距离所构成的矩阵为滤波器的通道距离矩阵。

$$\boldsymbol{C}^{nn'}=\boldsymbol{W}_{k\times k}^{n}-\boldsymbol{W}_{k\times k}^{n'}=\begin{bmatrix}|w_{11}^{n}-w_{11}^{n'}| & \cdots & |w_{1k}^{n}-w_{1k}^{n'}|\\ \vdots & & \vdots\\ |w_{k1}^{n}-w_{k1}^{n'}| & \cdots & |w_{kk}^{n}-w_{kk}^{n'}|\end{bmatrix} \tag{1}$$

本文采用马氏距离计算通道之间的相似度，相似度表示为 $\frac{1}{d^2}$，其中，$d^2$ 为通道之间的距离，计算公式如式(2)所示：

$$d^2=|(\boldsymbol{W}^{n}-\boldsymbol{W}^{n'})^{\mathrm{T}}\boldsymbol{S}^{-1}(\boldsymbol{W}^{n}-\boldsymbol{W}^{n'})| \tag{2}$$

$\boldsymbol{S}$ 为 2 个通道矩阵的协方差矩阵，计算公式如式(3)所示：

$$\boldsymbol{S}=E[(\boldsymbol{W}^{n}-E(\boldsymbol{W}^{n}))(\boldsymbol{W}^{n'}-E(\boldsymbol{W}^{n'}))] \tag{3}$$

由通道距离矩阵的定义可知：

$$d^2=|(\boldsymbol{W}^{n}-\boldsymbol{W}^{n'})^{\mathrm{T}}\boldsymbol{S}^{-1}(\boldsymbol{W}^{n}-\boldsymbol{W}^{n'})|=(\boldsymbol{C}^{nn'})^{\mathrm{T}}|\boldsymbol{S}^{-1}|\boldsymbol{C}^{nn'} \tag{4}$$

$d^2$ 的时间复杂度是 $O(k\times k\times n)$。从滤波器权值张量角度分析，对张量降维，将三维数据降为二维数据，由向量表示通道。计算向量的相似度相较于计算矩阵的相似度来说，时间复杂度更低。

目前的张量降维法都是由传统的矩阵降维法推广而来，矩阵即二阶张量。典型的张量降维法有 Tucker 分解、典范多项式(CP)分解、高阶奇异值分解等。这些张量降维法需要迭代求解得到变换矩阵，相较于本文的张量均值降维法计算更复杂。Tucker 分解将滤波器张量分解为 1 个核张量和 3 个模式矩阵，其计算复杂度取决于张量的大小和分解的秩。设三阶张量的维度为 $k\times k\times n$，分解的秩为 $r_1$、$r_2$、$r_3$，则 Tucker 分解的计算复杂度为 $O(k\times k\times n\times(r_1\times r_2\times r_3))$。CP 分解将滤波器张量分解为多个外积形式的矩阵相加。若分解的秩为 $r$，则 CP 分解的计算复杂度为 $O(k\times k\times n\times r)$。高阶奇异值分解将滤波器张量分解为多个矩阵相乘的形式，其计算复杂度通常较高。若分解的秩为 $r_1$、$r_2$、$r_3$，则计算复杂度为 $O(k\times k\times n\times(r_1\times r_2\times r_3))$。本文使用的均值降维法是一种简单的三阶张量降维法，其计算一个维度的均值，从而得到一个二阶矩阵，计算复杂度为 $O(k\times n)$，相对上述方法计算复杂度较低。此外，均值降维法计算所得二阶张量之间的马氏距离与三阶张量在降维前之间的距离相等，说明降维没有改变数据之间的关系。

针对滤波器权值张量，以其中一个维度 $k$ 为轴进行均值压缩，成为 $\boldsymbol{W}_{k\times n}$。通道 $n$ 则由向量 $\vec{\boldsymbol{W}}$ 表示，如式(5)所示：

$$\vec{\boldsymbol{W}}=\left(\frac{1}{k}\sum_{i=1}^{k}w_{i1},\frac{1}{k}\sum_{i=1}^{k}w_{i2},\cdots,\frac{1}{k}\sum_{i=1}^{k}w_{ik}\right) \tag{5}$$

此时，通道距离的计算公式如式(6)所示：

$$(d')^2=|(\vec{\boldsymbol{W}}^{n}-\vec{\boldsymbol{W}}^{n'})^{\mathrm{T}}(\boldsymbol{S}')^{-1}(\vec{\boldsymbol{W}}^{n}-\vec{\boldsymbol{W}}^{n'})| \tag{6}$$

$\boldsymbol{S}'$ 的计算公式如式(7)所示：

$$\boldsymbol{S}'=E[(\vec{\boldsymbol{W}}^{n}-E(\vec{\boldsymbol{W}}^{n}))(\vec{\boldsymbol{W}}^{n'}-E(\vec{\boldsymbol{W}}^{n'}))] \tag{7}$$

由于张量维度减少一维，因此计算复杂度降低一个数量级，成为 $O(k\times n)$。下面证明一个结论：张量通道之间相似性在均值降维前后保持不变，均值降维没有改变数据之间的关系，即证明 $(d')^2=d^2$。

证明：$k\in\mathbb{R}$ 且 $k>0$。

由式(5)可知：

$$\vec{\boldsymbol{W}}^{n}=\frac{1}{k}\boldsymbol{W}^{n},\ \vec{\boldsymbol{W}}^{n'}=\frac{1}{k}\boldsymbol{W}^{n'} \tag{8}$$

将式(8)代入式(6)可得：

$$(d')^2=\left|\left(\frac{1}{k}\boldsymbol{W}^{n}-\frac{1}{k}\boldsymbol{W}^{n'}\right)^{\mathrm{T}}(\boldsymbol{S}')^{-1}\left(\frac{1}{k}\boldsymbol{W}^{n}-\frac{1}{k}\boldsymbol{W}^{n'}\right)\right|=\frac{1}{k^2}|(\boldsymbol{W}^{n}-\boldsymbol{W}^{n'})^{\mathrm{T}}(\boldsymbol{S}')^{-1}(\boldsymbol{W}^{n}-\boldsymbol{W}^{n'})| \tag{9}$$

由通道距离矩阵的定义可知：

$$(d')^2=\frac{1}{k^2}(\boldsymbol{C}^{nn'})^{\mathrm{T}}|(\boldsymbol{S}')^{-1}|\boldsymbol{C}^{nn'} \tag{10}$$

将式(5)代入式(7)可得：

$$\boldsymbol{S}'=E[(\vec{\boldsymbol{W}}^{n}-E(\vec{\boldsymbol{W}}^{n}))(\vec{\boldsymbol{W}}^{n'}-E(\vec{\boldsymbol{W}}^{n'}))]=\frac{1}{k^2}E[(\boldsymbol{W}^{n}-E(\boldsymbol{W}^{n}))(\boldsymbol{W}^{n'}-E(\boldsymbol{W}^{n'}))] \tag{11}$$

$$|(\boldsymbol{S}')^{-1}|=k^2|(E[(\boldsymbol{W}^{n}-E(\boldsymbol{W}^{n}))(\boldsymbol{W}^{n'}-E(\boldsymbol{W}^{n'}))])^{-1}| \tag{12}$$

由式(3)和式(12)可得：

$$|(\boldsymbol{S}')^{-1}|=k^2|\boldsymbol{S}^{-1}| \tag{13}$$

将式(13)代入式(10)可得：

$$(d')^2=\frac{1}{k^2}(\boldsymbol{C}^{nn'})^{\mathrm{T}}k^2|\boldsymbol{S}^{-1}|\boldsymbol{C}^{nn'}=(\boldsymbol{C}^{nn'})^{\mathrm{T}}|\boldsymbol{S}^{-1}|\boldsymbol{C}^{nn'} \tag{14}$$

由式(4)与式(14)可得 $(d')^2=d^2$，证毕。

计算张量间距离的均值降维法可以推广到高阶张量。对于卷积神经网络中的滤波器裁剪而言，滤

波器张量均值降维法比分解形式的迭代求解典型方法的运算复杂度更低，计算效率更高。

### 2.2 通道相似度熵

使用滤波器对输入数据进行卷积操作时，在同一空间位置但不同通道的卷积输出结果是相似的。图 1 表示了三通道滤波器中不同通道同一空间位置的对应关系。由于特征提取是各通道分别对输入数据进行卷积之后叠加得到的，因此不同通道在输入数据同一位置卷积之后得到的是相似特征。

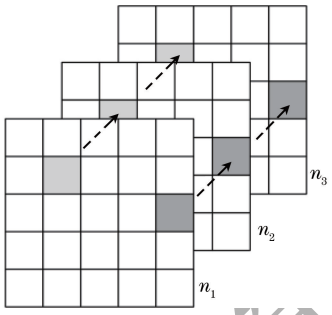


**图 1 滤波器通道空间位置**

**Fig. 1 Spatial position of the filter channel**

根据信息熵的计算公式将滤波器通道相似度熵定义为：

$$G(\boldsymbol{D})=-\sum_{i=1}^{N}(P(\boldsymbol{D}_i)\,\mathrm{lb}\,P(\boldsymbol{D}_i)) \tag{15}$$

式中：$\boldsymbol{D}$ 为滤波器通道相似度向量；$P(\boldsymbol{D}_i)$ 为 $\boldsymbol{D}$ 中不同类元素出现的概率；$N$ 表示元素分类的数量。

滤波器熵值具体计算步骤如下：

1）计算 $n$ 个通道两两之间的距离 $d_{nn'}$，$n\neq n'$。每个滤波器中通道相似度构成一个向量 $\boldsymbol{D}$，其中有 $\frac{n(n-1)}{2}$ 个分量。

$$\boldsymbol{D}=\underbrace{\left(\frac{1}{d_{12}},\cdots,\frac{1}{d_{1n}},\frac{1}{d_{23}},\cdots,\frac{1}{d_{2n}},\cdots,\frac{1}{d_{n(n-1)}}\right)}_{\frac{n(n-1)}{2}} \tag{16}$$

2）计算 $\boldsymbol{D}$ 中不重复元素的数量 $N$。

3）计算每个不重复元素在 $\boldsymbol{D}$ 中出现的概率 $P(\boldsymbol{D}_i)$。

$$P(\boldsymbol{D}_i)=\frac{\boldsymbol{D}_i}{\sum_{i=1}^{N}\boldsymbol{D}_i} \tag{17}$$

4）计算滤波器通道相似度熵：

$$G(\boldsymbol{D})=-\sum_{i=1}^{N}(P(\boldsymbol{D}_i)\,\mathrm{lb}\,P(\boldsymbol{D}_i))=-\sum_{i=1}^{N}\left(\frac{\boldsymbol{D}_i}{\sum_{i=1}^{N}\boldsymbol{D}_i}\mathrm{lb}\,\frac{\boldsymbol{D}_i}{\sum_{i=1}^{N}\boldsymbol{D}_i}\right) \tag{18}$$

$G(\boldsymbol{D})$ 即滤波器熵值。

不同滤波器中通道相似性决定了滤波器特征提取的多样性，一般卷积神经网络提升特征表达能力的手段主要是增加滤波器数量。在数量众多的滤波器中保留熵值大的滤波器，通过度量通道之间相似度，采用信息熵理论计算滤波器通道相似熵。熵值大，滤波器特征提取差异大；熵值小，滤波器特征提取差异小。

CNN 卷积层的滤波器数量通常根据经验值设定，一般取 32、64、128、256、512、1 024 等值。越深的卷积层，滤波器数量越多，CNN 成为一个金字塔型。裁剪滤波器不改变网络的深度，即卷积层数 $I$，裁剪将按照滤波器熵值大小减少每一层滤波器数量，使网络成为瘦金字塔型(S-CNN)，如图 2 所示。

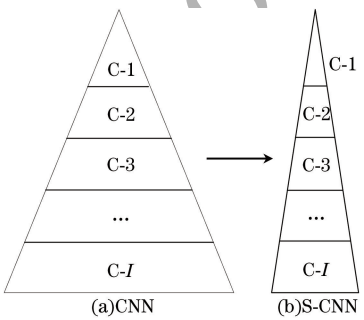


**图 2 裁剪后网络宽度变化**

**Fig. 2 The change of network width after pruning**

S-CNN 网络具有原始 CNN 的深度，保留了深度 CNN 网络逐层提取不同特征、逐步泛化抽象的能力，同时极大地减少了网络参数，压缩了网络规模。神经网络需要增加参数量以保证其稳健性，提高训练效率以及泛化能力[24]，裁剪是在网络训练之后进行的，因此不会对网络准确度产生明显影响。

滤波器按照其 $G(\boldsymbol{D})$ 值大小排序，选取一定比例的较小 $G(\boldsymbol{D})$ 值滤波器进行裁剪。按照比例裁剪每一层滤波器，关键在于裁剪比例的选取。目前还没有发现针对网络深度以及宽度计算裁剪比例的研究，多数研究认为裁剪比例设置在 50%～70%得到的 S-CNN 性能最佳，一般的裁剪方法都选择一个固定比例进行裁剪。通常的网络结构设计随着网络层次加深，每一层的滤波器数量也成倍增加，网络越深越宽，冗余越大。固定比例裁剪方法会造成浅层裁剪过量或者深层裁剪不足的问题。笔者认为裁剪比例应该随着网络加深加宽而增大，应采取不同比例的裁剪策略。下文通过几个比较实验确定裁剪比例的选择范围，实验结果也验证了采用不同比例进行裁剪更满足实际需求，能获得更稳健的 S-CNN。

## 3 实验结果及分析

实验在 Keras 平台上进行，采用了 CIFAR10[25] 与 ImageNet[26] 基准图像识别数据集，分别计算 VGG-16、ResNet-34 网络的滤波器通道相似度熵并且实现网络压缩。为了比较 CNN 模型的性能，采用准确度(Acc)、参数量(Parameter)以及浮点计算量(FLOPs)这 3 个指标评价网络压缩效果，其中，Acc 表示网络的预测性能，Parameter 的减少量表示网络压缩程度，FLOPs 的降低则代表网络的运算量减少以及推理速度提高。此外，在目标检测网络(SSD)上也验证了裁剪方法的有效性。网络压缩程度大同时保证网络的预测性能是网络压缩需要解决的重要问题，将通过实验验证使用本文算法进行大比例网络压缩后，S-CNN 仍能保持原网络预测性能。

### 3.1 VGG 裁剪

VGG-16 网络是牛津大学和 Google 公司共同研发的深度 CNN，其网络结构整齐，通过不断增加卷积层来加深网络结构，准确率比之前的 CNN 有较大提高。VGG-16 中有 16 个含有参数的网络层、13 个卷积层和 3 个全连接层。在 13 个卷积层中，第 1 层(CL_1)和第 2 层(CL_2)各有 64 个滤波器，第 3 层(CL_3)和第 4 层(CL_4)各有 128 个滤波器，第 5～7 层(CL_5～CL_7)每层有 256 个滤波器，第 8～13 层(CL_8～CL_13)每层有 512 个滤波器。

实验执行环境采用单个 GPU(NIVIDIA GeForce GTX1080Ti)，训练网络 Epoch 设置为 200，学习率设置为 0.01，以交叉熵损失作为损失函数。同时，使用随机梯度下降算法(SGD)作为优化器来更新参数，并在损失函数中添加 L2 正则化项来控制参数的大小。采用不同的滤波器删除比例，即裁剪比例，实验结果如表 1～表 4 所示，其中，“CL_*X*”列表示卷积层，“Filters”列为删除后剩下的滤波器数量，“Acc”是指网络的准确度，一般在分类任务中表示计算的准确率，“Acc±”列是准确度变化值，“Parameters/$10^6$”、“Parameters↓/%”和“FLOPs↓/%”列分别为压缩 CL_*X* 层之后参数量、参数压缩百分比以及 FLOPs 压缩百分比。

**表 1 CIFAR10 数据集上 VGG-16 裁剪比例设置为 80%的实验结果**

**Table 1 Experimental results with VGG-16 pruning ratios set to 80% on the CIFAR10 dataset**

| CL_*X* | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| CL_13 | 103 | 0.008 2+ | 13.17 | 13.75 | 13.51 |
| CL_12 | 103 | 0.001 8+ | 10.90 | 28.62 | 28.57 |
| CL_11 | 103 | 0.011 1+ | 8.64 | 43.42 | 43.24 |
| CL_10 | 103 | 0.002 8+ | 6.37 | 58.28 | 58.30 |
| CL_9 | 103 | 0.014 6+ | 4.11 | 73.08 | 73.09 |
| CL_8 | 103 | 0.015 5+ | 2.78 | 81.79 | 81.78 |
| CL_7 | 52 | 0.011 1+ | 2.12 | 86.12 | 86.10 |
| CL_6 | 52 | 0.007 3− | 1.56 | 89.78 | 89.85 |
| CL_5 | 52 | 0.003 4+ | 1.22 | 92.01 | 92.01 |
| CL_4 | 26 | 0.003 6− | 1.06 | 93.06 | 93.09 |
| CL_3 | 26 | 0.039 1− | 0.98 | 93.58 | 93.63 |
| CL_2 | 13 | 0.069 6− | 0.93 | 93.91 | 93.90 |
| CL_1 | 13 | 0.042 8− | 0.93 | 93.91 | 93.94 |

**表 2 CIFAR10 数据集上 VGG-16 裁剪比例设置为 80%-70%-60%的实验结果**

**Table 2 Experimental results with VGG-16 pruning ratios set to 80%-70%-60% on the CIFAR10 dataset**

| CL_*X* | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| CL_4 | 39 | 0.017 0− | 1.08 | 92.93 | 92.93 |
| CL_3 | 39 | 0.000 6+ | 0.99 | 93.52 | 93.47 |
| CL_2 | 26 | 0.053 0− | 0.96 | 93.71 | 93.71 |
| CL_1 | 26 | 0.026 4− | 0.95 | 93.78 | 93.79 |

在表 1 中，所有的卷积层都删除了 80%的滤波器，参数量减少了 93.91%，准确度降低了 0.042 8，FLOPs 减少了 93.94%。表 2 则保持表 1 中 CL_5～CL_13 的压缩结果，将 CL_4、CL_3 的裁剪比例设置为 70%，将 CL_2、CL_1 的裁剪比例设置为 60%，结果显示网络参数量减少了 93.78%，准确度降低了 0.026 4，FLOPs 减少了 93.79%。

表 3 CIFAR10 数据集上 VGG-16 裁剪比例设置为 80%-60%-50%的实验结果

Table 3 Experimental results with VGG-16 pruning ratios set to 80%-60%-50% on the CIFAR10 dataset

| CL_X | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| CL_4 | 52 | 0.020 0— | 1.10 | 92.80 | 92.82 |
| CL_3 | 52 | 0.001 5— | 1.02 | 93.32 | 93.32 |
| CL_2 | 32 | 0.042 9— | 0.99 | 93.52 | 93.55 |
| CL_1 | 32 | 0.028 4— | 0.98 | 93.58 | 93.63 |

表 4 CIFAR10 数据集上 VGG-16 裁剪比例设置为 80%-50%-40%的实验结果

Table 4 Experimental results with VGG-16 pruning ratios set to 80%-50%-40% on the CIFAR10 dataset

| CL_X | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| CL_4 | 64 | 0.007 5— | 1.12 | 92.67 | 92.70 |
| CL_3 | 64 | 0.000 6— | 1.05 | 93.12 | 93.17 |
| CL_2 | 39 | 0.010 6— | 1.02 | 93.32 | 93.36 |
| CL_1 | 39 | 0.035 4— | 1.01 | 93.39 | 93.44 |

同样，表 3、表 4 也保持了表 1 中 CL_5～CL_13 的结果，分别将 CL_4、CL_3 的裁剪比例设置为 60%、50%，将 CL_2、CL_1 的裁剪比例设置为 50%、40%。结果显示，表 3 中参数量减少了 93.58%，准确度降低了 0.028 4，FLOPs 减少了 93.63%，表 4 中参数量减少了 93.39%，准确度降低了 0.035 4，FLOPs 减少了 93.44%。

图 3 是压缩 CL_1～CL_4 后的准确度折线图（彩色效果见《计算机工程》官网 HTML 版，下同），可以看出，不同的裁剪比例对网络准确度的影响是不同的，较大的裁剪比例对网络造成的损伤较大，准确度降低明显，但结果显示裁剪比例小不是减少准确度损失的前提，在实验中，准确度降低最少的是裁剪比例设置为 80%-70%-60%的情况，这也充分说明了网络参数存在一定冗余，删除过多或者过少对准确度的影响相对都较大。

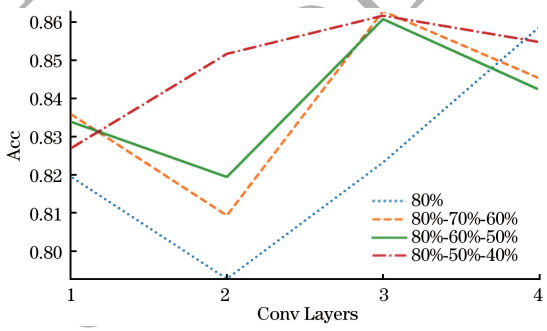


图 3 不同裁剪比例下网络准确度比较

Fig.3 Comparison of accuracy under different pruning ratios

与现有的一些压缩方法相比，本文 FEC 方法在较大压缩率的设置下对网络准确度造成的影响较小。L-OBS[27]方法压缩了 92.24%参数，准确度降低了 0.85，而本文 FEC 方法在 80%压缩率下参数减少了 93.91%，准确度只降低了 0.04。LWM 方法准确度虽然增加了 0.13，但参数压缩率只有 64%。Dirichlet 方法参数压缩率大于本文方法，参数量为 $7.3\times10^5$，但网络分类错误率比 FEC 高。GAL[28]方法在压缩率以及分类错误率上的表现都弱于 FEC。HRank[29]在准确度上的表现与本文方法基本一致，压缩率要小于本文方法。FEC 适用于对网络规模要求较小的设备，同时能够基本保持原网络准确度。表 5 列出了 CIFAR10 数据集上本文 FEC 方法与典型裁剪方法的性能比较，其中，Error 表示网络错误率。

表 5 CIFAR10 数据集上 VGG-16 裁剪后的实验结果比较

Table 5 Comparison of experimental results on CIFAR10 dataset after VGG-16 pruning

| Pruning methods | Error/% | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| FEC(80%) | 6.92 | 0.04— | 0.93 | 93.91 | 93.94 |
| FEC(80%-70%-60%) | 6.26 | 0.03— | 0.95 | 93.78 | 93.79 |
| FEC(80%-60%-50%) | 6.43 | 0.03— | 0.98 | 93.58 | 93.63 |
| FEC(80%-50%-40%) | 6.85 | 0.04— | 1.01 | 93.39 | 93.44 |
| L-OBS | 7.89 | 0.85— | 1.21 | 92.24 | — |
| LWM | 6.75 | 0.13+ | 5.40 | 64.00 | 34.00 |
| Dirichlet | 8.63 | 0.78— | 0.73 | 95.32 | 37.58 |
| GAL | 7.97 | 1.93+ | 3.36 | 77.60 | 39.60 |
| LFC | 7.02 | 0.51— | — | — | 81.93 |
| HRank | 6.79 | 0.02— | 1.78 | 92.00 | 76.50 |

在 ImageNet 数据集上对 VGG-16 进行裁剪，参考 CIFAR10 的裁剪结果，裁剪比例设置为 80%-70%-60%，其中 CL_5～CL_13 裁剪比例为 80%，CL_3、CL_4 裁剪比例为 70%，CL_1、CL_2 裁剪比例为 60%，结果如表 6 所示。可以看出，裁剪之后，网络准确度降低了 0.11，参数量减少了 93.85%。

**表 6 ImageNet 数据集上 VGG-16 裁剪比例设置为 80%-70%-60%的实验结果**

**Table 6 Experimental results with VGG-16 pruning ratios set to 80%-70%-60% on the ImageNet dataset**

| CL_*X* | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| CL_8～CL_13 | 103 | 0.021 0− | 2.86 | 82.23 | 85.36 |
| CL_5～CL_7 | 52 | 0.016 0− | 1.29 | 91.99 | 93.43 |
| CL_3、CL_4 | 64 | 0.114 1− | 1.05 | 93.48 | 95.78 |
| CL_1、CL_2 | 39 | 0.110 0− | 0.99 | 93.85 | 96.27 |

为了验证 FEC 方法在不同量级和复杂程度的数据集上对 VGG-16 网络裁剪比例设置的一致性，在 ImageNet 数据集上仍然采取 80%-60%-50%和 80%-50%-40%的裁剪比例进行实验。实验结果显示，当裁剪比例设置为 80%-60%-50%时，Acc 减少了 0.133 8，当裁剪比例设置为 80%-50%-40%时，Acc 减少了 0.224 1，即其中裁剪效果优劣依次是 80%-70%-60% 最优，80%-60%-50% 次优，80%-50%-40%最差，此结果与在 CIFAR10 数据集上的实验效果一致，当裁剪比例设置为 80%-70%-60%时，VGG-16 网络的裁剪结果相对最优。

上述结果表明，对于大型数据集，FEC 方法仍然具有有效性，数据集规模对裁剪性能影响较小。

### 3.2 ResNet 裁剪

VGG 网络只有标准卷积结构，上述实验证明 FEC 适用于此结构。本节使用 FEC 方法压缩 ResNet 残差网络，残差网络相较于 VGG 增加了残差块，ResNet-34 参数量比 VGG-16 略多，但网络准确度更高。实验只压缩 ResNet-34 中的标准卷积层，不改变残差块的结构。训练参数 Epoch 设置为 500，学习率设置为 0.000 1，并以交叉熵作为损失函数。同时，优化器使用随机梯度下降算法更新参数，在损失函数中添加 L2 正则化项控制参数，滤波器删除比例同样设置了 4 种不同的组合，表 7 中 CL_*X* 比例全部为 80%，表 8 中分别将 CL_30～CL_35、CL_17～CL_28、CL_8～CL_15、CL_2～CL_6 卷积层分段，删除比例分别设置为 80%-70%-60%-50%，表 9 和表 10 采用同样的分段，将删除比例减小，分别设置为 80%-60%-50%-40% 以及 80%-50%-40%-30%。

**表 7 CIFAR10 数据集上 ResNet-34 裁剪比例设置为 80%的实验结果**

**Table 7 Experimental results with ResNet-34 pruning ratios set to 80% on the CIFAR10 dataset**

| CL_*X* | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| CL_35 | 103 | 0.163 5+ | 17.54 | 17.69 | 17.68 |
| CL_33 | 103 | 0.177 5+ | 13.77 | 35.38 | 35.36 |
| CL_30 | 103 | 0.176 6+ | 10.94 | 48.66 | 48.62 |
| CL_28 | 52 | 0.119 9+ | 9.99 | 53.12 | 53.04 |
| CL_26 | 52 | 0.170 6+ | 9.06 | 57.48 | 57.46 |
| CL_24 | 52 | 0.151 4+ | 8.12 | 61.89 | 61.88 |
| CL_22 | 52 | 0.180 2+ | 7.18 | 66.31 | 66.30 |
| CL_20 | 52 | 0.179 3+ | 6.24 | 70.72 | 70.72 |
| CL_17 | 52 | 0.179 6+ | 5.53 | 74.05 | 74.12 |
| CL_15 | 26 | 0.176 9+ | 5.29 | 75.18 | 75.41 |
| CL_13 | 26 | 0.170 5+ | 5.06 | 76.26 | 76.24 |
| CL_11 | 26 | 0.111 7+ | 4.82 | 77.38 | 77.35 |
| CL_8 | 26 | 0.098 0+ | 4.65 | 78.18 | 78.18 |
| CL_6 | 13 | 0.157 7+ | 4.59 | 78.46 | 78.45 |
| CL_4 | 13 | 0.122 7− | 4.53 | 78.74 | 78.73 |
| CL_2 | 13 | 0.108 4+ | 4.47 | 79.02 | 79.01 |

表 8 CIFAR10 数据集上 ResNet-34 裁剪比例设置为 80%-70%-60%-50%的实验结果

Table 8 Experimental results with ResNet-34 pruning ratios set to 80%-70%-60%-50% on the CIFAR10 dataset

| CL_X | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| Conv_28 | 77 | 0.098 6+ | 10.11 | 52.56 | 52.49 |
| Conv_26 | 77 | 0.189 3+ | 9.29 | 56.41 | 56.35 |
| Conv_24 | 77 | 0.188 6+ | 8.46 | 60.30 | 60.22 |
| Conv_22 | 77 | 0.189 2+ | 7.64 | 64.15 | 64.09 |
| Conv_20 | 77 | 0.188 4+ | 6.81 | 68.04 | 67.96 |
| Conv_17 | 77 | 0.180 9+ | 6.19 | 70.95 | 70.99 |
| Conv_15 | 52 | 0.184 1+ | 6.02 | 71.75 | 71.82 |
| Conv_13 | 52 | 0.181 3+ | 5.84 | 72.60 | 72.65 |
| Conv_11 | 52 | 0.182 0+ | 5.67 | 73.39 | 73.48 |
| Conv_8 | 52 | 0.183 7+ | 5.53 | 74.05 | 74.03 |
| Conv_6 | 32 | 0.181 6+ | 5.49 | 74.24 | 74.12 |
| Conv_4 | 32 | 0.176 0+ | 5.46 | 74.38 | 74.42 |
| Conv_2 | 32 | 0.177 0+ | 5.42 | 74.57 | 74.60 |

表 9 CIFAR10 数据集上 ResNet-34 裁剪比例设置为 80%-60%-50%-40%的实验结果

Table 9 Experimental results with ResNet-34 pruning ratios set to 80%-60%-50%-40% on the CIFAR10 dataset

| CL_X | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| Conv_28 | 103 | 0.174 1+ | 10.23 | 51.99 | 51.93 |
| Conv_26 | 103 | 0.177 1+ | 9.53 | 55.28 | 55.25 |
| Conv_24 | 103 | 0.185 8+ | 8.82 | 58.61 | 58.84 |
| Conv_22 | 103 | 0.177 8+ | 8.12 | 61.90 | 61.88 |
| Conv_20 | 103 | 0.186 1+ | 7.41 | 65.23 | 65.19 |
| Conv_17 | 103 | 0.184 6+ | 6.88 | 67.71 | 67.68 |
| Conv_15 | 64 | 0.179 7+ | 6.73 | 68.42 | 68.51 |
| Conv_13 | 64 | 0.172 5+ | 6.58 | 69.12 | 69.06 |
| Conv_11 | 64 | 0.179 3+ | 6.44 | 69.78 | 69.89 |
| Conv_8 | 64 | 0.185 6+ | 6.33 | 70.30 | 70.44 |
| Conv_6 | 39 | 0.171 8+ | 6.29 | 70.48 | 70.50 |
| Conv_4 | 39 | 0.180 8+ | 6.27 | 70.58 | 70.64 |
| Conv_2 | 39 | 0.175 9+ | 6.24 | 70.72 | 70.77 |

表 10 CIFAR10 数据集上 ResNet-34 裁剪比例设置为 80%-50%-40%-30%的实验结果

Table 10 Experimental results with ResNet-34 pruning ratios set to 80%-50%-40%-30% on the CIFAR10 dataset

| CL_X | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| Conv_28 | 128 | 0.186 6+ | 10.35 | 51.43 | 51.38 |
| Conv_26 | 128 | 0.148 2+ | 9.76 | 54.20 | 54.14 |
| Conv_24 | 128 | 0.175 9+ | 9.17 | 56.97 | 56.91 |
| Conv_22 | 128 | 0.165 8+ | 8.58 | 59.74 | 59.67 |
| Conv_20 | 128 | 0.186 4+ | 7.98 | 62.55 | 62.43 |
| Conv_17 | 128 | 0.182 1+ | 7.55 | 64.57 | 64.64 |
| Conv_15 | 77 | 0.179 0+ | 7.43 | 65.13 | 65.19 |
| Conv_13 | 77 | 0.191 8+ | 7.31 | 65.70 | 65.75 |
| Conv_11 | 77 | 0.152 8+ | 7.19 | 66.26 | 66.30 |
| Conv_8 | 77 | 0.185 3+ | 7.10 | 66.68 | 66.85 |
| Conv_6 | 45 | 0.168 7+ | 7.08 | 66.78 | 66.89 |
| Conv_4 | 45 | 0.184 2+ | 7.06 | 66.87 | 66.91 |
| Conv_2 | 45 | 0.184 0+ | 7.04 | 66.96 | 67.02 |

表 7～表 10 中的数据显示，上述 4 种裁剪比例设置下的 S-CNN 的准确度相比原始网络都有所增加：表 7 中裁剪完 CL_2 层之后 S-CNN 准确度最终增加了 0.108 4，参数量减少了 79.02%，FLOPs 减少了 79.01%；表 8 中 S-CNN 准确度最终增加了 0.177 0，参数量减少了 74.57%，FLOPs 减少了 74.60%；表 9 中 S-CNN 准确度增加了 0.175 9，参数量减少了 70.72%，FLOPs 减少了 70.77%；表 10 中准确度最终增加了 0.184 0，参数量减少了 66.96%，FLOPs 减少了 67.02%。实验结果表明，FEC 同样适用于残差网络，只对网络中的标准卷积结构进行裁剪不会降低网络准确度。

经过计算得到不同裁剪比例的准确度方差分别为：$\sigma_1^2=0.006\ 0$(80%)；$\sigma_2^2=0.002\ 1$(80%-70%-60%-50%)；$\sigma_3^2=0.001\ 7$(80%-60%-50%-40%)；$\sigma_4^2=0.001\ 8$(80%-50%-40%-30%)。$\sigma_3^2$ 最小，即实验中裁剪比例设置为 80%-60%-50%-40%时，准确度变化更加稳定。

图 4 更加清晰直观地显示了裁剪过程中准确度的变化情况。图 4(a)～图 4(d)分别显示了 4 种裁剪比例设置下的准确度变化，其中，蓝色折线代表不同卷积层裁剪之后的准确度增加值，红色虚线是线性拟合的变化趋势线。由线性拟合线可以看出，当各层裁剪比例统一为 80%时[图 4(a)]，随着裁剪的进行，网络准确度增加值总体呈下降趋势，对于含有滤波器数量较少的 CL_2～CL_6 卷积层裁剪，选取的比例值大，网络准确度增加值小，尤其在裁剪 CL_4 层时，准确度相较于原始网络降低了 0.122 7，裁剪 CL_2 层之后，对网络进行微调恢复，使其准确度最终增加了 0.108 4。在采用分段裁剪比例方法的图 4(b)、图 4(c)、图 4(d)中，红色拟合线则显示网络准确度增加值随着裁剪呈上升趋势，说明针对滤波器数量不一样的卷积层，采用不同裁剪比例可以更大程度地增加网络准确度。比较上述分图的裁剪结果可知，当裁剪比例选用 80%-50%-40%-30%的设置时，裁剪效果较优。这一结果进一步说明了含有滤波器数量越多的卷积层，其参数冗余度越大，裁剪比例的设置应该随着滤波器数量的增多而增大。

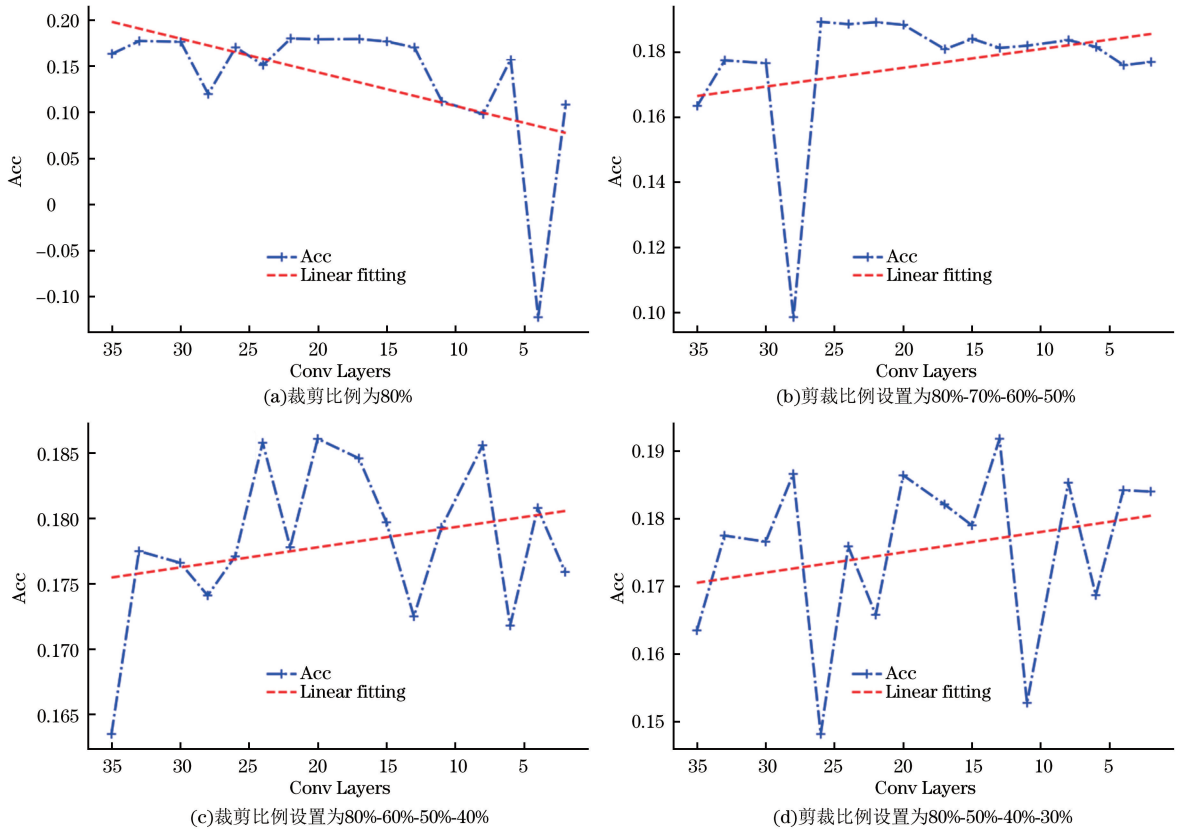


**图 4 准确度变化趋势**

**Fig. 4 Trend of accuracy changes**

此外，使用 ImageNet 数据集训练 ResNet-34，并随机选择其中 100 个类别，进行 1 000 个 Epochs 的训练，并以交叉熵损失作为损失函数。同时，优化器使用 SGD，正则化项为 L2。选择了每个残差块中的前 2 层标准卷积进行裁剪，根据 ResNet-34 在 CIFAR10 上的准确度方差最小值 $\sigma_3^2=$

0.001 7，将裁剪比例设置为 80%-50%-40%-30%可以获得较好的裁剪性能。裁剪之后，网络参数数量减少了 74.67%，准确度则提高了 0.114 0，如表 11 所示。

**表 11 ImageNet 数据集上 ResNet-34 裁剪比例设置为 80%-50%-40%-30%的实验结果**

**Table 11 Experimental results with ResNet-34 pruning ratios set to 80%-50%-40%-30% on the ImageNet dataset**

| CL_*X* | Filters/个 | Acc± | Parameter/$10^6$ | Parameter↓/% | FLOPs↓/% |
|---|---|---|---|---|---|
| CL_30～CL_35 | 103 | 0.0610+ | 11.54 | 45.97% | 48.89 |
| CL_17～CL_28 | 128 | 0.2761+ | 6.42 | 69.93% | 70.78 |
| CL_8～CL_15 | 77 | 0.0840+ | 5.56 | 73.97% | 75.69 |
| CL_2～CL_6 | 45 | 0.1140+ | 5.41 | 74.67% | 76.58 |

本文结合标准卷积神经网络 VGG-16 以及残差结构网络 ResNet-34 的实验结果，给出在实际进行具体的 CNN 裁剪时，可以按照卷积层滤波器数量设置不同的裁剪比例范围，如表 12 所示。

**表 12 滤波器数量对应裁剪比例设置**

**Table 12 The number of filters corresponds to the pruning ratio** %

| Filters/个 | Pruning ratio |
|---|---|
| 1 024 | 70～80 |
| 512 | 70～80 |
| 256 | 50～70 |
| 128 | 40～60 |
| 64 | 30～40 |

卷积神经网络滤波器数量的设置一般是根据经验值设置为 64、128、256、512、1 024 等值，目前的研究大多由实验结果来判断裁剪方法的有效性，裁剪比例设置对网络准确性影响的理论性证明将是未来的一个研究方向。

### 3.3 SSD 裁剪

SSD 是经典的目标检测算法，其主干网络特征层由改进的 VGG 网络构成，属于含有标准卷积层的神经网络。SSD 训练集采用 VOC2007 数据集，VOC2007 数据集有 20 个类，可以检测并标注 Aeroplane、Bicycle、Bird、Boat 等类别。初始 SSD 的参数量为 $2.629\times10^7$，平均精度均值(mAP)为 76.03%，按照表 12 滤波器数对应最小比例值进行裁剪，裁剪卷积层滤波器如表 13 所示。其中，“Before pruning”和“After pruning”列分别为裁剪前后各卷积层的滤波器数量。因为 FEC 方法适用于标准卷积结构，所以 SSD 中只裁剪标准卷积层。

SSD 裁剪之后参数量减少为 $1.164\times10^7$，重新进行网络训练后得到 mAP 为 77.07%。裁剪后参数量降低了 55.72%，mAP 提高了 1.04 个百分点，说明卷积网络参数存在极大冗余，采用 FEC 方法删除冗余参数对网络进行瘦身的同时可以保持网络的性能。

**表 13 SSD 裁剪前后滤波器数量**

**Table 13 Number of filters before and after pruning SSD**

单位：个

| CL_*X* | Before pruning | After pruning |
|---|---|---|
| CL_1 | 64 | 45 |
| CL_2 | 128 | 77 |
| CL_3 | 256 | 128 |
| CL_4 | 512 | 154 |
| CL_5 | 512 | 154 |
| CL_6_1 | 256 | 128 |
| CL_6_2 | 512 | 154 |
| CL_7_1 | 128 | 77 |
| CL_7_2 | 256 | 128 |
| CL_8_1 | 128 | 77 |
| CL_8_2 | 256 | 128 |
| CL_9_1 | 128 | 77 |

VGG、ResNet 以及 SSD 网络裁剪的实验结果说明，滤波器通道相似度熵值计算方法适用于不同任务神经网络中的标准卷积层裁剪。

## 4 结束语

判断滤波器在 CNN 中的重要性是网络裁剪的关键步骤，目前多数方法是以权值大小作为滤波器重要性的判断依据，这些方法基于大权值对损失函数贡献大的假设。本文提出的基于通道相似度的滤波器熵值计算方法，分析滤波器的结构特征，判断通道之间的相似性，根据通道相似度计算滤波器熵，由熵值大小进行滤波器排序，删除一定比例熵值较小的滤波器。该方法参数压缩率高、准确度损失小、计算速度快且易于实现，通过在分类网络 VGG-16、ResNet-34 以及目标检测网络 SSD 上进行对比实验与分析，充分说明了深度 CNN 在深层卷积层存在巨大的参数冗余，裁剪冗余参数之后能够提高网络性能，降低网络规模，在资源受限设备中具有很好的应用前景。未来将继续探索滤波器裁剪的新技术，研究针对多种卷积结构的裁剪方法，进一步优化提高压缩后的网络性能。

## 参考文献


[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[2] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. [2023-07-10]. https://arxiv.org/abs/1409.1556.

[3] SZEGEDY C, LIU W, JIA Y Q, et al. Going deeper with convolutions[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2015: 1-9.

[4] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the inception architecture for computer vision [EB/OL]. [2023-07-10]. https://arxiv.org/abs/1512.00567.

[5] SZEGEDY C, IOFFE S, VANHOUCKE V, et al. Inception-v4, Inception-ResNet and the impact of residual connections on learning[C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence. New York, USA: ACM Press, 2017: 4278-4284.

[6] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2016: 770-778.

[7] HAN S, MAO H Z, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding[J]. Fiber, 2015, 56(4):3-7.

[8] CARREIRA-PERPINAN M A, IDELBAYEV Y. "Learning-Compression" algorithms for neural net pruning[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2018: 8532-8541.

[9] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2323.

[10] LUO J H, WU J, LIN W. ThiNet: a filter level pruning method for deep neural network compression[C]// Proceedings of IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2017: 5068-5076.

[11] DONG X, CHEN S Y, PAN S J. Learning to prune deep neural networks via layer-wise optimal brain surgeon[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. New York, USA: ACM Press, 2017: 4860-4874.

[12] MITTAL D, BHARDWAJ S, KHAPRA M M, et al. Studying the plasticity in deep convolutional neural networks using random pruning[J]. Machine Vision and Applications, 2019, 30(2): 203-216.

[13] YANG T J, CHEN Y H, SZE V. Designing energy-efficient convolutional neural networks using energy-aware pruning[C]// Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2017: 6071-6079.

[14] HU Y, SUN S, LI J, et al. A novel channel pruning method for deep neural network compression[EB/OL]. [2023-07-10]. https://arxiv.org/abs/1805.11394.

[15] MOLCHANOV P, TYREE S, KARRAS T, et al. Pruning convolutional neural networks for resource efficient transfer learning[C]//Proceedings of the 5th International Conference on Learning Representations. [S. l.]: ICLR, 2017: 1-17.

[16] 耿丽丽,牛保宁. 深度神经网络模型压缩综述[J]. 计算机科学与探索, 2020, 14(9): 1441-1455.
GENG L L, NIU B N. Survey of deep neural networks model compression[J]. Journal of Frontiers of Computer Science and Technology, 2020, 14(9): 1441-1455. (in Chinese)

[17] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[C]// Proceedings of International Conference on Learning Representations. [S. l.]: ICLR, 2017: 1-13.

[18] LUO J H, WU J. An entropy-based pruning method for CNN compression[EB/OL]. [2023-07-10]. https://arxiv.org/abs/1706.05791v1.

[19] HU H, PENG R, TAI Y W, et al. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures[EB/OL]. [2023-07-10]. https://arxiv.org/abs/1607.03250v1.

[20] SINGH P, VERMA V K, RAI P, et al. Leveraging filter correlations for deep model compression[C]//Proceedings of IEEE/CVF Winter Conference on Applications of Computer Vision. Washington D. C., USA: IEEE Press, 2020: 824-833.

[21] ADAMCZEWSKI K, PARK M J. Dirichlet pruning for convolutional neural networks[C]//Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. Washington D. C., USA: IEEE Press, 2021: 3637-3645.

[22] GENG L L, NIU B N. Pruning convolutional neural networks via filter similarity analysis[J]. Machine Learning, 2022, 111(9): 3161-3180.

[23] 周林勇,谢晓尧, 刘志杰, 等. 卷积神经网络池化方法研究[J]. 计算机工程, 2019, 45(4): 211-216.
ZHOU L Y, XIE X Y, LIU Z J, et al. Research on pooling method of convolution neural network[J]. Computer Engineering, 2019, 45(4): 211-216. (in Chinese)

[24] BUBECK S, SELLKE M. A universal law of robustness via isoperimetry[J]. Journal of the ACM, 70(2): 10.

[25] KRIZHEVSKY A. Learning multiple layers of features from tiny images[EB/OL]. [2023-07-10]. http://www.cs.toronto.edu/~kriz/cifar-10-binary.tar.gz.

[26] DENG J, DONG W, SOCHER R, et al. ImageNet: a large-scale hierarchical image database[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2009: 248-255.

[27] DONG X, CHEN S Y, PAN S J. Learning to prune deep neural networks via layer-wise optimal brain surgeon[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. New York, USA: ACM Press, 2017: 4860-4874.

[28] LIN S H, JI R R, YAN C Q, et al. Towards optimal structured CNN pruning via generative adversarial learning[C]//Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2019: 2785-2094.

[29] LIN M, JI R, WANG Y, et al. HRank: filter pruning using high-rank feature map[EB/OL]. [2023-07-10]. https://arxiv.org/abs/2002.10179.


编辑 金胡考