计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(8):49−55 [doi: 10.15888/j.cnki.csa.006481]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于统计分析的卷积神经网络模型压缩方法①

杨　扬, 蓝章礼, 陈　巍

(重庆交通大学 信息科学与工程学院, 重庆 400074)
通讯作者: 蓝章礼, E-mail: lzl7309@126.com

摘　要: 针对卷积神经网络中卷积层参数冗余, 运算效率低的问题, 从卷积神经网络训练过程中参数的统计特性出发, 提出了一种基于统计分析裁剪卷积核的卷积神经网络模型压缩方法, 在保证卷积神经网络处理信息能力的前提下, 通过裁剪卷积层中对整个模型影响较小的卷积核对已训练好的卷积神经网络模型进行压缩, 在尽可能不损失模型准确率的情况下减少卷积神经网络的参数, 降低运算量. 通过实验, 证明了本文提出的方法能够有效地对卷积神经网络模型进行压缩.

关键词: 卷积神经网络; 冗余; 裁剪; 统计分析; 模型压缩

引用格式: 杨扬,蓝章礼,陈巍.基于统计分析的卷积神经网络模型压缩方法.计算机系统应用,2018,27(8):49–55. http://www.c-s-a.org.cn/1003-3254/6481.html

Convolution Neural Network Model Compression Method Based on Statistical Analysis

YANG Yang, LAN Zhang-Li, CHEN Wei

(School of Information Science & Engineering, Chongqing Jiaotong University, Chongqing 400074, China)

Abstract: Aiming at the problem of convolutional layer parameter redundancy and low operation efficiency in convolutional neural network, a convolution neural network (CNN) model compression method based on statistical analysis is proposed in this paper. On the premise of ensuring a good ability of convolutional neural network to process information, the well-trained convolution neural network model is compressed by pruning the convolution kernels which have less influence on the whole model in the convolution layer, meanwhile, reducing the parameters of CNN without losing the model accuracy so as to reduce the amount of computation. Experiments show that the proposed method can effectively compress the convolution neural network model while maintaining a good performance.

Key words: convolution neural network; redundancy; pruning; statistical analysis; model compression

2012 年 Hiton[1]构建的深度卷积神经网络 AlexNet 才在图像分类领域取得了惊人的成绩, 卷积神经网络 (CNN) 在计算机视觉领域包括图像分类[1], 目标检测[2]、图像语义分割[3]、视频分类[4]得到了广泛的应用[5]. 之后, 层数更多、更加精细设计的深度卷积神经网络结构相继被提出, 比如 VggNet[6]、GoogLenet[7]、ResNet[8]在 ImageNet[9]上取得了更好的成绩. 除此之外, 卷积神经网络在人工智能[10], 自然语言处理[11], 故障诊断[12]有着广阔的应用前景. 这些深度卷积神经网络模型参数越来越多, 运算量越来越大, 对运算设备的内存、CPU、GPU 的配置要求越来越高. 当需要在运算和存储资源有限的微型设备上[13], 比如手机、嵌入式设备上使用卷积神经网络时, 除了准确率, 计算效率和模型的大小也是至关重要的.

模型压缩最早的研究为 OBD[14](Optimal Brain Surgeon) 和 OBS[15](Optimal Brain Surgeon), 通过泰勒

① 基金项目: 重庆市基础科学与前沿技术研究专项项目 (cstc2016jcyjA1953)
Foundation item: Special Research Project for Basic Science and Frontier Technology of Chongqing (cstc2016jcyjA1953)
收稿时间: 2017-12-10; 修改时间: 2018-01-04; 采用时间: 2018-01-16; csa 在线出版时间: 2018-07-28

展开, 分析参数的扰动对损失函数的影响, 以此确定参数的重要性, 决定参数保留或者裁剪. Han 等人[16]通过裁剪、量化、压缩卷积神经网络模型参数, 大幅度减少模型的大小, 并且没有降低模型的准确率. 文献[17]指出非结构的稀疏无法利用现有的硬件进行加速, 提出了一种在目标函数上增加 group lasso 进行结构化稀疏的学习方式. 文献[18]通过一个训练好的较大的模型来训练一个较小的模型, 将较大的模型学到的知识迁移到较小的模型中. Network in Network[19]除了对卷积层进行了改进, 还提出了全局平均化的方法, 解决了全连接层参数数量多的问题, 并被 GoogleNet 和 ResNet 等采用. 目前卷积层的卷积核大小、卷积核数量对于不同的数据集参数设置不同, 需要大量实验, 有一定经验成分, 卷积层用一般包含了足够多数目的卷积核, 存在冗余, 已有相关文献通过实验证明在裁剪部分不重要的卷积核后, 再训练整个卷积神经网络 (或者逐层裁剪、逐层训练) 可以在尽量不损失准的条件下对卷积神经网络进行压缩[20−22]. Wen[20]等人通过定义 APoZ (Average Percentage of Zeros) 来统计每一个卷积核中激活为 0 的比例, 以为评估一个卷积核是否重要, 主要是用在最后一层卷积层, 以此减少全连接层的参数数量. 文献[21]通过将一定样本输入卷积神经网络, 计算特征图的各类参数, 对活性低的特征图通道裁剪. 文献[22]通过将卷积核参数的 L1 范数作为评价一个卷积核重要性的依据, 将不重要的卷积核裁剪.

由于文献[19]提出的全局平均池化在一定会程度上解决了传统卷积神经网络参数多的问题. 本文的主要解决的是卷积层的压缩, 在调研了卷积神经网络训练过程中的规律的基础上, 提出了基于卷积核的标准差作为卷积核重要性指标进行卷积核裁剪的方法, 并和文献[22]相结合的方法, 通过实验, 本文提出的方法能和文献[22]互补, 综合两种评价指标综合进行卷积核裁剪能保留对分类更有作用的卷积核.

1 卷积神经网络的原理

1958 年, Hubel 和 Wiesel 等人[23]发现了生物视觉系统的信息处理方式, 视觉信息从视网膜传递到大脑是通过多层的感受野激活完成的. 1998 年, Lecun 等人[24]提出的 LeNet-5, 如图 1 所示, LeNet-5 由两层卷积层和两层池化层交替将输入图像转换成一系列特征图, 再连接三层全连接层对提取的特征分类. 卷积层的卷积核实现了局部感受野和特征提取的功能, 将局部区域信息通过卷积核的卷积运算, 再经过激活函数、池化, 将低层的激活信息传递到高层. 以往的人工设计的特征有良好的特征表达能力, 例如 HOG[25], SIFT[26], 但这些人工设计的特征缺乏良好的泛化能力. 池化层也称下采样层, 能在一定程度上保持特征的尺度不变性并对特征图降维. Lenet 提出后, 在图像分类领域没有取得实质的进展和突破, 直到 2012 年 Hiton 及其学生 Alex 构建的深度卷积神经网络 AlexNet 在 ImageNet 上取得了显著的成绩, 主要原因是训练的改进, 在网络的训练中加入了权重衰减、Droupout[27]、Batch Normalization[28]等技术, 更关键的是计算机计算能力的提升, GPU 加速技术的发展, 使得在计算机可以高效地实现卷积的运算. 之后, 更复杂, 准确率更高的深度卷积神经网络被提出.

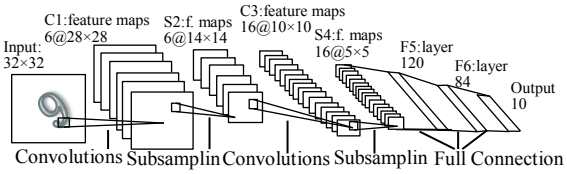

图 1 LeNet-5 结构图

如图 2 所示, 为一个 5×5×3 的卷积核作用在一个 32×32×3 的图像 (也可能为 32×32×3 的特征图) 上, 5×5×3 的卷积核与图像的 5×5×3 区域点乘再加上偏置 (bias) 经过激活函数, 产生一个运算结果, 即图中的小圆球. 卷积核在图像的所有局部区域以步长为 1 滑动并卷积, 得到一个 28×28×1 的特征图 (feature map).

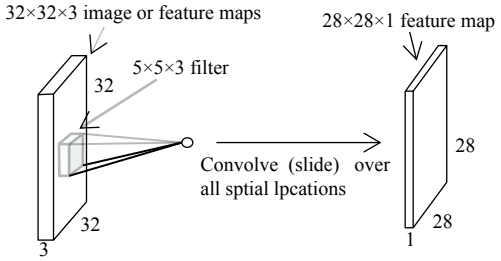

图 2 一个卷积核作用在图像或者特征图上

同理, 如图 3 所示, 32×32×3 的图像经过一个有 6 个 5×5×3 的卷积核的卷积层, 产生了 28×28×6 的特征图. 在实际中, 滑动的步长不一定为 1, 有时为了保持卷积后特征图大小不变或者取整, 会对特征图的边界进行填充 (padding), 特征图通过卷积层后, 会接着通过

池化层 (pooling layer), 将得到的特征图输入到下一层卷积层.

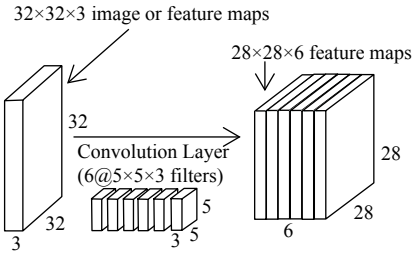

图 3 6 个卷积核作用在图像或者特征图上

2 卷积核的裁剪

近年来, 为了让卷积神经网络达到更好的效果, 卷积神经网络朝着更深更复杂的方向发展. 而另一方面, 增强深度神经网络的运算效率, 在不损失精度的情况下, 对深度学习训练得到的模型进行优化压缩也有着大量的研究. 本文参照文献[21,22]提出的卷积核裁剪方式, 针对已经训练好的卷积神经网络模型, 对卷积层中数个不重要的卷积核裁剪, 同时裁剪对应的特征图, 最后对裁剪后的模型进行再训练, 恢复模型的性能.

2.1 裁剪方式

为了便于说明裁剪卷积核的方式, 如图 4 所示, $X_k \in R^{H_K \times W_K \times N_K}$代表第 k 层的特征图, H_k、W_k、N_k 分别是其高、宽、维度, Conv $k \in R^{s \times s \times N_K \times N_{K+1}}$为第 k 层卷积层, $F_{i,j} \in R^{s_k \times s_k}$为其中一个 2D 的单层卷积核, i=1, 2, ⋯, N_k, j=1, 2, ⋯, N_{k+1}, s_k 为第 k 层卷积层卷积核的高和宽, N_k、N_{k+1} 分别为上一层的特征图的维度和下一层特征图的维度. X_k 层特征图通过第 k 层卷积层 *Conv k* 得到了 k+1 层特征图 X_{k+1}, 假设在卷积层中删除第 j 个卷积核, 即图中 Conv k 的 j 列 (灰色), 同时也裁剪了 X_{k+1} 层特征图的第 j 个特征图. 减少的参数 $s_k^2 \times N_k$, 减少的计算量为$N_k \times s_k^2 \times W_{k+1} \times H_{k+1}$, 在 k+2 层特征图 X_{k+2} 的计算过程中额外将少的乘法运算量为$N_{k+2} \times s_{k+1}^2 \times H_{k+2} \times W_{k+2}$. 当在第 k 层卷积层中裁剪 m($0 \leqq m < n_{k+1}$) 个卷积核时, 减少的参数为 $m \times s_k^2 \times N_k$, 减少的乘法运算量为$m \times N_k \times s_k^2 \times W_{k+1} \times H_{k+1}$.

2.2 卷积核的评价指标

为了决定一个卷积层中某个卷积核的重要程度, 文献[20]通过定义 APoZ (Average Percentage of Zeros), 即一个卷积核中来统计每一个卷积核中激活为 0 的比例, 以为评估一个卷积核是否重要, 主要是用在最后一层卷积层, 以此减少全连接层的参数数量. 文献[21]通过输入样本, 通过计算特征图的相关参数确定卷积核的重要程度, 认为对不同样本得到类似特征图的卷积核为冗余的卷积核. 通过文献[22]在提出以卷积核的 L1 作为卷积核重要程度的评价指标, 认为裁剪 L1 范数较小的卷积核对整个模型影响较小.

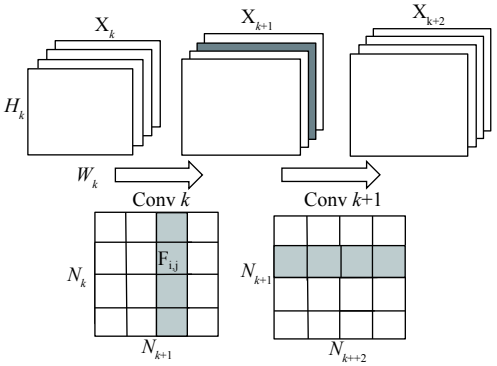

图 4 卷积核的裁剪方式

卷积神经网络训练过程中, 卷积层中参数的标准差 (或方差) 逐渐增大, 分布范围逐渐扩大, 参数之间的差异性逐渐明显, 如图 5、图 6 所示.

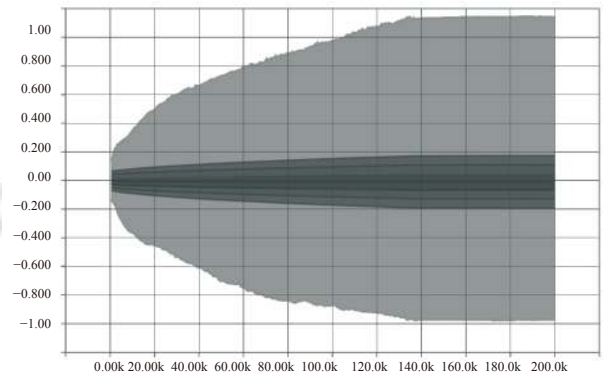

图 5 Conv 2 参数在训练过程中的分布

(从上到下的曲线依次为[max, μ+1.5ρ, μ+ρ, μ+0.5ρ, μ, μ–0.5ρ, μ–ρ, μ–1.5ρ, min], μ 为全部卷积核的均值, ρ 为标准差, max 为最大值, min 为最小值)

本文认为, 卷积神经网络通过训练, 标准差或者方差更大的卷积核学习到了更明显的局部特征, 因此提出了基于标准差的卷积核裁剪方法, 克服了文献[20,21]需要输入样本, 统计特征图各类参数需要额外大量计算量的缺点, 同时也避免了文献[22]只保留 L1 范数较大的卷积核, 而没有考虑到卷积核提取特征的能力与

参数的分布有关. 本文了还将以卷积核标准差作为卷积核重要性指标与文献[22]提出的以卷积核 L1 范数作为卷积核重要性指标相结合, 即将卷积核的 L1 范数和标准差结合作为卷积核重要性的评价指标, 对卷积核进行裁剪.

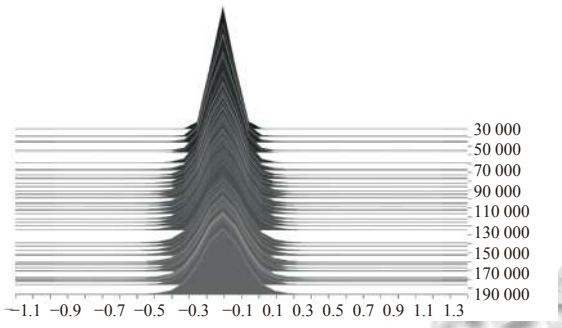

图 6 Conv 2 参数在训练过程中的统计直方图

$$std(F_j)=\sqrt{\sum_{i=1}^{n_k}\sum_{s\times s}\left(F_{i,j}-\frac{1}{s\times s\times n_k}\sum_{i=1}^{n_k}\left(\sum_{s\times s}F_{i,j}\right)\right)^2} \quad (1)$$

$$L_1(F_j)=\sum_{i=1}^{n_k}\left(\sum_{s\times s}|F_{i,j}|\right) \quad (2)$$

$$sl(F_j)=\frac{std(F_j)}{\sum_{j=1}^{n_{k+1}}std(F_j)}+\lambda\times\frac{L_1(F_j)}{\sum_{j=1}^{n_{k+1}}L_1(F_j)} \quad (3)$$

式 (1) 为一个卷积层中第 j 个卷积核的标准差计算公式, $\sum_{s\times s}F_{i,j}$代表对 $F_{i,j}$ 中 $s\times s$ 个元素进行求和. 式 (2) 为卷积层第 j 个卷积核的 L1 范数计算公式, $\sum_{s\times s}|F_{i,j}|$代表对 $F_{i,j}$ 中 $s\times s$ 个元素的绝对值进行求和, 即 $F_{i,j}$ 的 L1 范数, n_k 个单层卷积核 $F_{i,j}$ 的 L1 范数和即为卷积核 F_j 的 L1 范数. 式 (3) 同时考虑了卷积核的 L1 范数和标准差, 为了防止 L1 范数和标准差计算结果相差过大, 对卷积核 L1 范数和标准差进行了归一化处理, 参数 λ 调节卷积核 L1 范数和标准差的相对重要程度. 当 λ=1 时, 表示卷积核 L1 范数和标准差同等重要, 当 λ<1 时, 表示卷积核标准差比 L1 范数更重要, 当 λ>1 时, 表示卷积核 L1 范数比标准差更重要.

3 实验分析

为了验证本文所提出的卷积核裁剪方法的正确性和有效性, 本文在 MNIST 和 Cifar-10 数据集上分别设计了有两层卷积层和三层卷积层的卷积神经网络进行了实验. 实验环境: Ubuntu 16.04, Python3.6, Tensorflow 1.2, 计算机 CPU 为 6700 hq, GPU 为 GTX 960 m (4 G 显存), 内存为 8 G.

对 MNIST 数据集和 Cifar-10 数据集分别设计了两层卷积层和三层全连接、三层卷积层和三层全连接层结构的卷积神经网络, 如表 1 所示, *Conv k* 代表第 *k* 层卷积层, (3, 3, 1, 32) 代表该层有 32 个 3×3×1 的卷积核. *FC k* 代表第 *k* 层全连接层, (3136, 200) 代表输入一个 3136 维的数组, 输出一个 200 维的数组. 由于 cifar-10 数据集更复杂, 所以在训练过程中采用了数据增强和权重衰减. 两个卷积神经网络都采用了交叉熵损失函数计算代价函数. 在训练完成后, 在 MNIST 数据集上的正确率达到了 99.02%, 在 Cifar-10 上的正确率达到了 86.56%.

表 1 所设计的卷积神经网络的结构

	ConvNet For MNIST	ConvNet For Cifar-10
Conv 1	(3, 3, 1, 32)	(5, 5, 3, 64)
Conv 2	(3, 3, 32, 64)	(5, 5, 64, 96)
Conv 3	无	(5, 5, 64, 96)
*FC*1	(3136, 200)	(576, 382)
FC 2	(200, 100)	(382, 192)
FC 3	(100, 10)	(192.10)
总参数数量	667 326	608 136

3.1 据方差裁剪卷积核对比

从图 7 和图 8 可以看出, 在用 MNIST 数据集训练得到的带有两层卷积层的神经网络中, 裁剪标准差较小的卷积核能比裁剪标准较大的卷积核保留的准确率更高, 即标准差较大的卷积核比方差较小的卷积核更重要, 证实了本文的设想, 即标准差较大的卷积核在训练的过程中学到了更为重要的局部特征.

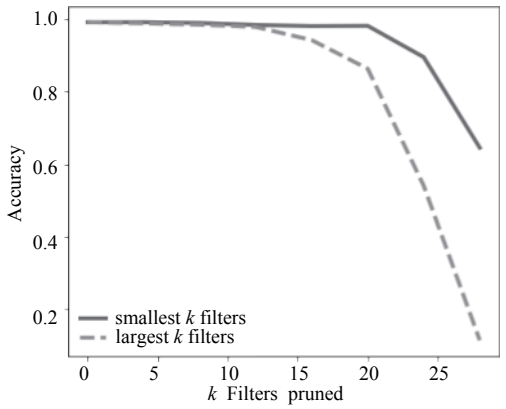

图 7 裁剪针对 MNIST 训练的卷积神经网络第一层卷积层中的卷积核

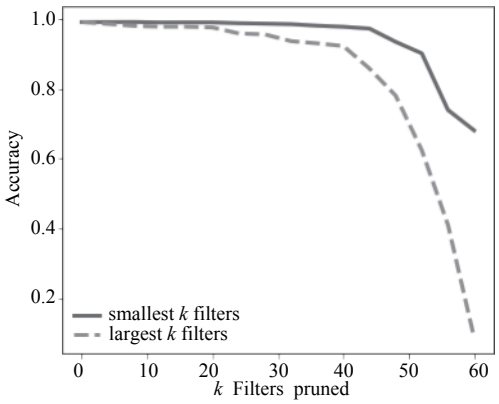

图 8 裁剪针对 MNIST 训练的卷积神经网络第二层卷积层中的卷积核

如图 9-图 11 可以得出, 在 Cifar-10 上也有着类似的结果, 在针对 MNIST 数据集设计的卷积神经网络中, 在裁剪甚至 50% 的卷积核时, 准确率没有明显降低. 而针对 Cifar-10 数据集设计的卷积神经网络, 裁剪少数卷积核准确率也会明显降低. 一方面是因为 MNIST 数据集较为简单, Cifar-10 数据集较为复杂, 其次是因为针对 MNIST 数据集设计的卷积神经网络卷积核设置得较多, 这也说明了, 在训练好的卷积神经网络模型中, 如果在一个卷积层中裁剪一定的卷积核而准确率没有明显降低, 说明这一层的卷积核设置的过多, 可以对卷积核进行裁剪.

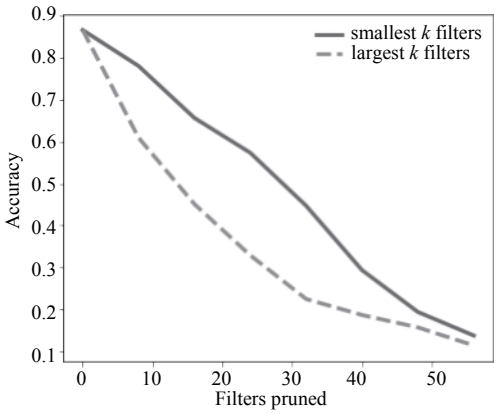

图 9 裁剪针对 Cifar-10 训练的卷积神经网络第一层卷积层中的卷积核

3.2 对比实验

为了验证本文提出的方法的有效性, 本文在分别针对 MNIST 和 Cifar-10 训练好的卷积神经网络模型与其他的裁剪方式进行了对比, 式 (3) 中 λ 取值为 1. 卷积神经网络在裁剪后, 性能会有一定的下降, 为了恢复性能, 一般会在对裁剪后的模型进行再训练, 再训练的迭代次数一般没有从初始状态训练多. 24-48 代表经过裁剪, 第一层卷积层的卷积核保留的个数为 24, 第二层保留的个数为 48, 其余情况以类似的方式表示.

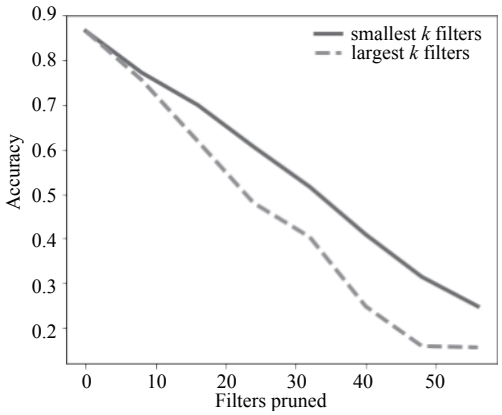

图 10 裁剪针对 Cifar-10 训练的卷积神经网络第二层卷积层中的卷积核

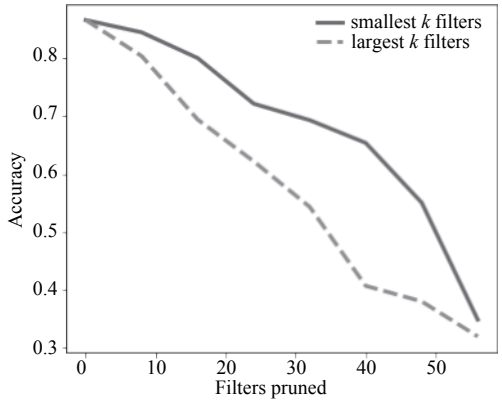

图 11 裁剪针对 Cifar-10 训练的卷积神经网络第三层卷积层中的卷积核

通过表 2 可以看出, 对 MNIST 数据集设计的两层卷积神经网络分别裁剪后, 通过方差作为评价卷积核的裁剪能保留更多的准确率, 同时通过再训练后得到的准确率也比文献[22]的方式高. 通过表 3 可以看出, 针对 Cifar-10 设计的三层卷积神经网络中, 在 48-64-64 这种方式裁剪时, 文献[22]提出的 L1 范数作为卷积核的评价指标保留了较大的准确率, 而在 32-48-48 这种裁剪方式时, 本文提出的方法保留了较高的准确率, 而通过再训练准确率会有所损失. 选取 L1+std 作为卷积核的评价指标进行裁剪时, 裁剪后的准确率在

L1 和 std 之间. 因此, 式 (3) 中的参数 λ 可以通过卷积核裁剪的比例动态动态调整, 在裁剪卷积核数量过多时, 适当增大 λ, 可以保留更高的准确率.

表 2 裁剪针对 MNIST 数据集训练的两层卷积层神经网络

		准确率 (%)		减少的参数
		裁剪后	再训练后	
24-48	L1	98.57	99.09	164 912 (24.71%)
	std	98.62	99.39	
	L1+std	99.08	99.23	
16-32	L1	94.97	99.16	327 510 (49.08%)
	std	95.76	99.28	
	L1+std	95.75	99.21	

表 3 裁剪针对 Cifar-10 数据集训练的三层卷积层神经网络

		准确率 (%)		减少的参数
		裁剪后	再训练	
48-64-64	L1	54.41	86.62	206 080 (33.89%)
	std	54.03	86.57	
	L1+std	54.28	86.59	
32-48-48	L1	26.65	81.24	245 680 (40.40%)
	std	28.59	81.36	
	L1+std	27.04	81.29	

4 结论与展望

本文从卷积神经网络训练过程中参数的统计特征出发, 提出了一种基统计分析裁剪卷积核的卷积神经网络模型压缩方法. 通过在针对 MNIST 和 Cifar-10 所设计的两个卷积神经网络中进行裁剪实验, 本文提出的标准差较大的卷积学习到了更显著的局部特征的设想是正确的, 在与类似的裁剪方式的对比中, 本文提出的方法在裁剪较多的卷积核时保留更高的准确率, 根据裁剪的比例动态调整卷积核 L1 范数和标准差的相对重要程度, 可以使得裁剪的结果更稳定. 在后续的研究中, 将进一步研究卷积核通道间的裁剪和利用特征图使得裁剪结果最优.

参考文献

1 Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25. 2012, 60(2): 1097–1105.

2 Girshick R. Fast R-CNN. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1440–1448.

3 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA. 2015. 3431–3440.

4 Karpathy A, Toderici G, Shetty S, *et al*. Large-scale video classification with convolutional neural networks. Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. 2014. 1725–1732.

5 李彦冬, 郝宗波, 雷航. 卷积神经网络研究综述. 计算机应用, 2016, 36(9): 2508–2515, 2565. [doi: 10.11772/j.issn.1001-9081.2016.09.2508]

6 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Computer Science, 2014.

7 Szegedy C, Liu W, Jia YQ, *et al*. Going deeper with convolutions. Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA. 2015. 1–9.

8 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 770–778.

9 Russakovsky O, Deng J, Su H, *et al*. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 2015, 115(3): 211–252. [doi: 10.1007/s11263-015-0816-y]

10 Silver D, Schrittwieser J, Simonyan K, *et al*. Mastering the game of go without human knowledge. Nature, 2017, 550(7676): 354–359. [doi: 10.1038/nature24270]

11 Johnson R, Zhang T. Semi-supervised convolutional neural networks for text categorization via region embedding. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal, Canada. 2015. 919–927.

12 Guo XJ, Chen L, Shen CQ. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. Measurement, 2016, 93: 490–502. [doi: 10.1016/j.measurement.2016.07.054]

13 雷杰, 高鑫, 宋杰, 等. 深度网络模型压缩综述. 软件学报, 2018, 29(2): 251–266. [doi: 10.13328/j.cnki.jos.005428]

14 Cun LY, Denker JS, Solla SA. Optimal brain damage. In: Jordan MI, LeCun Y, Solla SA, eds. Advances in Neural Information Processing Systems 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1990. 598–605.

15 Hassibi B, Stork DG. Second order derivatives for network pruning: Optimal brain surgeon. In: Hanson SJ, Cowan JD, Giles CL, eds. Advances in Neural Information Processing Systems 5. Morgan Kaufmann, San Mateo, CA, USA. 1993.

164–171.

16 Han S, Mao HZ, Dally WJ. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. Fiber, 2015, 56(4):3-7.

17 Wen W, Wu CP, Wang YD, *et al*. Learning structured sparsity in deep neural networks. Advances in Neural Information Processing Systems 29. 2016. 2074–2082.

18 Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network. Computer Science, 2015, 14(7):38–39.

19 Lin M, Chen Q, Yan SC. Network in network. arXiv: 1312.4400, 2013.

20 Hu H, Peng R, Tai Y W, *et al*. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. arXiv: 1607.03250, 2016.

21 Polyak A, Wolf L. Channel-level acceleration of deep face representations. IEEE Access, 2015, 3: 2163–2175. [doi: 10.1109/ACCESS.2015.2494536]

22 Li H, Kadav A, Durdanovic I, *et al*. Pruning filters for efficient convnets. arXiv: 1608.08710, 2016.

23 Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology, 1962, 160(1): 106–154. [doi: 10.1113/jphysiol.1962.sp006837]

24 Cun LY, Bottou L, Bengio Y, *et al*. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278–2324. [doi: 10.1109/5.726791]

25 Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA. 2005. 886–893.

26 Lindeberg T. Scale invariant feature transform. Scholarpedia, 2012, 7(5): 10491. [doi: 10.4249/scholarpedia.10491]

27 Srivastava N, Hinton GE, Krizhevsky A, *et al*. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 2014, 15(1): 1929–1958.

28 Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning. Lille, France. 2015. 448–456.