基于 SqueezeNet 的轻量化卷积神经网络 SlimNet

董艺威 于 津*
(汕头大学工学院计算机科学与技术系 广东 汕头 515063)

摘 要 结构参数量和计算量限制了卷积神经网络在移动设备上的应用。主要研究在尽量保持精度的前提下减少结构参数量和计算量。针对分组卷积引起的分组通道间不流通的问题,提出分组瓶颈;针对如何提升分类精度问题,提出奇异瓶颈;使用上述策略改进 SqueezeNet,提出轻量化结构 SlimNet。实验表明:引入分组瓶颈和奇异瓶颈具有有效性,提出的轻量化结构 SlimNet 在分类精度、结构参数量及计算量上均优于 SqueezeNet。

关键词 图像分类 卷积神经网络 轻量化 分组卷积 分组瓶颈 奇异瓶颈 SlimNet

中图分类号 TP399 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2018.11.039

LIGHT-WEIGHT CONVOLUTIONAL NEURAL NETWORK SLIMNET BASED ON SQUEEZENET

Dong Yiwei Yu Jin*
(*Department of Computer Science and Technology, College of Engineering, Shantou University, Shantou 515063, Guangdong, China*)

Abstract Structure parameters and computations limit the application of convolutional neural networks (CNNs) in mobile devices. We mainly studied how to reduce the amount of structural parameters and computations while keeping the accuracy as far as possible. The grouped bottleneck was proposed according to the interchannel congestion caused by grouped convolution, singular bottleneck was proposed to improve classification accuracy, and the light-weight structure, SlimNet, was proposed after improving SqueezeNet using the above strategies. The experiment results demonstrate the effectiveness of grouped bottleneck and singular bottleneck SlimNet is superior to SqueezeNet in terms of classification accuracy, structural parameters and computation.

Keywords Image classification Convolutional neural network Light-weight Group convolution Grouped bottleneck Singular bottleneck SlimNet

0 引 言

计算机视觉(Computer Vision)赋予计算机看见并理解世界的能力,包括图像分类、目标定位、目标识别、实例分割等任务。图像分类是计算机根据图像内容将其归类,是上述其他任务的基础,有重要的研究意义。

自 2012 年 Krizhevsky 等凭借 AlexNet[1] 赢得 ILSVRC12 的图像分类冠军以后,研究学者们开始关注卷积神经网络在图像分类中的应用。随后,在 AlexNet 的基础上,提出如 ResNet[2] 等很多表现优异的卷积神经网络,在 ILSVRC 图像分类中获得优异的成绩。通过设计更深更复杂的结构以获得更高分类精度(即深度学习)。上述卷积神经网络结构参数量高达数兆,识别一张图片需要上百亿次浮点运算次数。运行这些结构需要大量内存空间和计算资源,一般是工作站级别的设备。

现实中,人们希望将卷积神经网络部署到手机、无人机和无人驾驶汽车等移动设备中,用于图像分类和目标检测等计算机视觉任务。移动设备内存少、计算能力小。研究学者提出设计适用于移动设备的卷积神经网络,即轻量化结构。轻量化结构是指结构分类精度满足应用需求,结构参数量和计算量均未超出移动设备的能力。轻量化结构具有如下优势:一般采用空中下载 OTA 技术将训练好的结构部署(安装或更新)到移动设备,轻量化结构加快了部署过程,节省移动数

收稿日期:2018-05-28。**董艺威**,硕士,主研领域:轻量化卷积神经网络。**于津**,副教授。

据流量;移动设备内存少、计算量小,轻量化结构符合移动设备的硬件要求。

2016 年,Iandola 等[3]借鉴了 Inception 模块的设计思想,提出 Fire 模块,以此提出首个轻量化卷积神经网络结构 SqueezeNet。不同于 Inception 模块中对特征图进行独立的多尺度学习的做法,Fire 模块先对特征图进行维度压缩,然后对这些特征图做多尺度学习后进行拼接。此外,Fire 模块仅采用 1×1 卷积和 3×3 卷积,并未使用更大的卷积,有效减少结构参数量。在维持 AlexNet 的分类精度情况下,SqueezeNet 的结构参数量仅为 AlexNet 的 1/50。SqueezeNet 存在以下两点不足:AlexNet 是 2012 年提出的结构,分类精度低于近些年的结构,没有可比性;虽然 SqueezeNet 结构参数量较小,但与 AlexNet 相比其所占计算资源几乎相同。

为了降低结构计算量,2014 年,Sifre 等[4]首次提出深度可分卷积(Depthwise Separable Convolution),以此改进 AlexNet,在保持分类精度的情况下,减少了结构的训练时间和参数量。受 Sifre 启发,文献[5]采用深度可分卷积替换 InceptionV3 中的常规卷积(Standard Convolution)。同时指出去掉 3×3 卷积和 1×1 卷积之间的 ReLU 能够加速训练并提高分类精度,以此提出 Xception[6],在相同参数量的情况下,分类精度上胜过 InceptionV3。

受到 Xception 启发,2017 年谷歌的 Howard 等借用深度可分卷积代替常规卷积,借鉴 VGG[7]中逐层设计的思想,提出了 MobileNet[8]。MobileNet 的主要工作在于使用深度可分卷积替代常规卷积来降低卷积神经网络的参数量和计算量,设计面向移动设备的轻量化结构。实验证明,与 VGG 相比,在很小的分类精度损失情况下,MobileNet 参数量是 VGG 的 3%,计算量是 VGG 的 4%。使用深度可分卷积的 MobileNet 在轻量化设计中起到了启蒙作用。但 MobileNet 存在以下问题:采用 VGG 的直筒结构性价比较低,ResNet 等结构已证明通过残差学习能提升结构分类精度。

深度可分卷积包含深向卷积(Depthwise Convolution)和点向卷积(Pointwise Convolution)。深向卷积是单通道学习特征,点向卷积使得各个通道间信息流通。其中 1×1 卷积占据了大部分的计算量。在 MobileNet 中 1×1 卷积在参数量占 74.59%,在计算量上占 94.86%。

为了减少 1×1 卷积所占计算量比例,2017 年旷视科技的孙剑等在 1×1 卷积中引入分组卷积,提出 1×1 分组卷积(Pointwise Group Convolution),但分组卷积导致分组通道间信息不流通。他们提出在结构中添加通道交叉操作使得分组通道间信息重新流通。文献[9]在 MobileNet 的基础上,添加了残差学习,提出 ShuffleNet。ShuffleNet 在分类精度、结构参数量和计算量上均优于 MobileNet。

2018 年,谷歌的 Sandler 等[10]对 MobileNet 进行改进,提出 MobileNetV2。他们提出反转残差模块(Inverted Residuals),其特点是先扩张后卷积再压缩。这样做的目的是为深度可分卷积提供更多的通道,从而提高结构的学习能力。他们还发现去掉通道压缩层的非线性变换能够提升分类精度。实验证明,和 MobileNet 相比,MobileNetV2 在分类精度、结构参数量和计算量上均有了改善。在 ImageNet-1k 数据集上,MobileNetV2 分类精度提升 2%,参数量减少 30%,计算量减少 50%。

本文提出两种设计策略:分组瓶颈和奇异瓶颈,并使用上述策略改进 SqueezeNet,提出轻量化结构 SlimNet。与 SqueezeNet 相比,SlimNet 在 Flowers5 数据集上分类精度提高 17%,在 Simpsons13 数据集上分类精度提高 9%,在结构参数量上降低 34%,在计算量上降低 75%。

1　轻量化结构 SlimNet

SqueezeNet 在分类精度和结构计算量上与 AlexNet 持平,其结构参数量为 1.24 兆,而 AlexNet 结构参数量为 61.10 兆。SqueezeNet 计算量与 AlexNet 持平,不利于部署在移动设备上。经过推导 Fire 模块的计算量,发现导致计算量过高的原因有:压缩层输入通道数量较大,用于降低通道数量的 1×1 卷积操作占计算比例高;扩展层输出通道数量较大,用于特征学习的 1×1 和 3×3 卷积操作计算较多。本文针对如何降低计算量而展开研究。

1.1　分组瓶颈

为改进结构以减少结构计算量,提出分组卷积,它能够有效降低结构的参数量和计算量。常规卷积对输入整体进行卷积操作,如图 1(a)所示。分组卷积中的分组是针对输入通道而言,在图 1(b)中,输入通道分为两组。每组输入独立进行卷积操作,分组得到的输出拼接后作为整体输出。

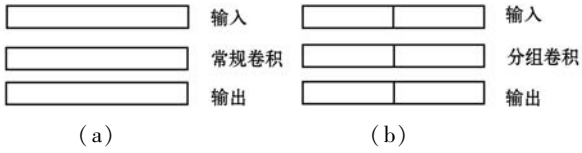

图 1　常规卷积和分组卷积

当输入为 $W1 \times H1 \times C1$($W1$ 代表 Width,指输入通道宽度;$H1$ 代表 Height,指输入通道高度;$C1$ 代表

Channel,指输入通道数量),卷积核大小为 $h \times w$,一共有 $C2$ 个,则输出数据为 $H2 \times W2 \times C2$。对于常规卷积和分组卷积,其参数量和计算量如表 1 所示,其中分组数量为 g。

表 1 常规卷积和分组卷积对比

	参数量	计算量
1	$C1 \times C2 \times 9$	$2 \times C1 \times C2 \times h \times w \times H2 \times W2$
2	$(C1/g) \times (C2/g) \times 9 \times g$	$2 \times (C1/g) \times (C2/g) \times h \times w \times H2 \times W2 \times g$
r	$1:(1/g)$	$1:(1/g)$
注:表中 1 指常规卷积,2 指分组卷积,r 指比率		

从表 1 可见,分组卷积有效减少了结构的参数量和计算量,但存在分组通道间信息不流通的问题。

为了解决分组卷积引起的分组通道信息不流通问题,本文研究近年来的卷积神经网络结构,发现 ResNet 中的瓶颈模块或许能够解决这个问题。瓶颈模块如图 2 所示。瓶颈模块先使用 1 ×1 卷积缩减输入通道数量(通道数量由 256 变为 64),再对 64 个通道进行 3 ×3 卷积以学习特征,最后使用 1 ×1 卷积恢复通道数量(通道数量由 64 变为 256)。在模块中通道数量经历了先减少再不变最后增加的过程,类似于瓶颈,所以叫瓶颈模块。这样做的目的是减少 3 ×3 卷积的输入和输出通道数量(均为 64),以减少结构的训练时间。图 2 中右边一条线连接输入和输出,为恒等映射,其目的是使用残差学习。

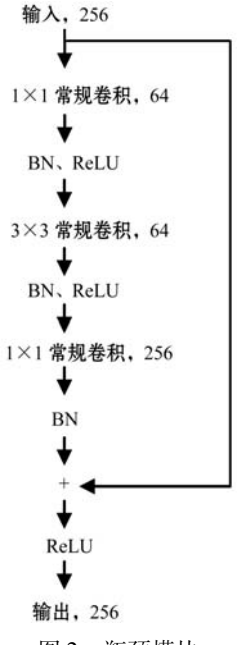

图 2 瓶颈模块

因瓶颈模块中 1 ×1 卷积的输入或输出通道数量较大,3 ×3 卷积的输入和输出通道数量较小,故本文提出 1 ×1 卷积使用分组卷积以减少结构的参数量和计算量,3 ×3 卷积使用常规卷积以使得分组通道间信息重新流通,并命名为分组瓶颈,如图 3 所示。使用分组瓶颈可解决分组通道间信息不流通的问题,而不需要额外操作(ShuffleNet 使用通道交叉操作来解决分组通道间信息不流通的问题)。

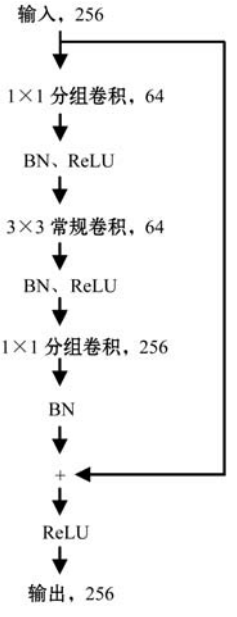

图 3 本文提出的分组瓶颈

1.2 奇异瓶颈

2017 年 Sandler 等在 MobileNetV2 中提出线性瓶颈的设计策略。线性瓶颈指,删除用于通道缩减的 1 ×1 卷积后的非线性变换能够提高分类精度,如图 4 所示。在线性瓶颈中,下面的 1 ×1 卷积没有非线性变换(即 ReLU)。由此本文提出舍去用于通道扩增的 1 ×1 卷积后的非线性变换,提升了分类精度,并通过实验验证上述想法的可行性。由于改进后的模块仅一次非线性变换,故叫奇异瓶颈。在图 5 中,舍去上面的 1 ×1 卷积的非线性操作。

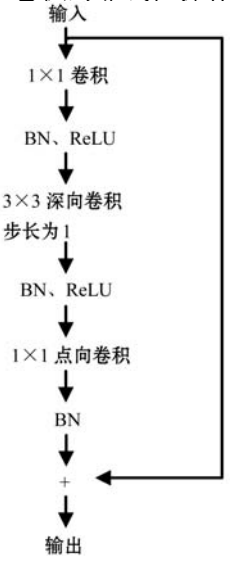

图 4 线性瓶颈

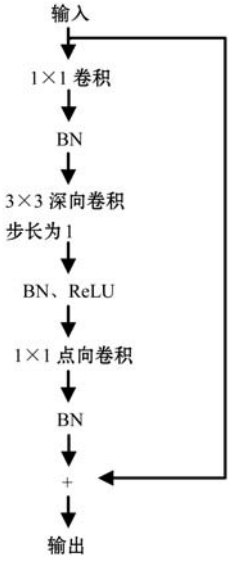

图 5 本文提出的奇异瓶颈

1.3 SlimNet

本文使用上述策略改进 SqueezeNet,并引入批归一化 BN(Batch Normalization)[11],改进 SqueezeNet 的 Fire 模块。改进 SqueezeNet 的整体结构后级联 Slim 模块,增加池化层和 Slim 模块的数量,添加 Softmax,提出 SlimNet。

Fire 模块没有采用瓶颈设计,为了使用分组瓶颈进行改进,先将 Fire 模块改进成瓶颈模块,再引入分组瓶颈。具体讲,减小 Fire 模块中 3×3 卷积的输出通道数量,使其有输入输出通道数量相同,后接 1×1 卷积用于扩增通道数量,形成瓶颈模块。使用分组瓶颈,对模块中的 1×1 卷积采用分组卷积。使用奇异瓶颈,舍去 1×1 卷积的非线性变换。此外,发现 Fire 模块中没有批归一化。批归一化能够加速训练过程,提高分类精度。添加批归一化后,本文提出 Slim 模块,如图 6 所示。Slim 模块由压缩层、卷积层和扩展层组成,设有四个超参数,分别是压缩层和扩展层的分组组数 g、压缩层卷积核数 s、卷积层卷积核数 c 和扩展层卷积核数 e,其中 $s=c$, $s<e$, $c<e$。

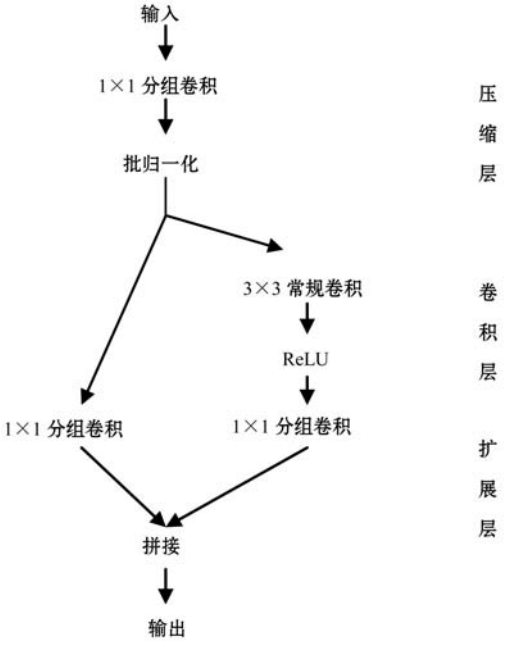

图 6　Slim 模块

将 Slim 模块级联形成卷积阶段以进行图像特征学习,后接分类器用于图像分类。由于 Slim 模块和 Fire 模块输入和输出通道的宽、高、数量都相同,本文按照 SqueezeNet 整体结构的方式级联 Slim 模块,并在结构中额外增加一个池化层和两个 Slim 模块。SqueezeNet 最后一层卷积层采用 1×1 卷积用以改变输出分类数量,将通道数量从 512 扩增至 1 000。该操作占用大量的计算,于是去掉该卷积层,增加全连接层用于改变输出分类数量。此外,在结构最后添加 Softmax 函数。Softmax 函数用于多分类过程,将结构的输出值映射到 0~1 的范围内,且输出之和为 1,可以看作是概率。SlimNet 的整体结构如表 2 所示。

表 2　Slim 整体结构

层名称/类型	输出尺寸	s	c	e
输入	224×224×3			
卷积层 1	111×111×64			
最大池化 1	55×55×64			
Slim2	55×55×128	16	16	64
Slim3	55×55×128	16	16	64
最大池化 3	27×27×128			
Slim4	27×27×256	32	32	128
Slim5	27×27×256	32	32	128
最大池化 5	13×13×256			
Slim6	13×13×384	48	48	192
Slim7	13×13×384	48	48	192
Slim8	13×13×512	64	64	256
Slim9	13×13×512	64	64	256
最大池化 9	6×6×512			
Slim11	6×6×512	64	64	256
Slim12	6×6×512	64	64	256
平均池化 12	1×1×512			
全连接层	分类种类			
Softmax	分类种类			

2　实验分析

考虑到结构针对不同数据集的普遍适用能力,本文采用两个不同的数据集:Flowers5[12] 和 Simpsons13[13],如表 3 所示。

表 3　数据集概况

数据集	大小	数量	种类	分辨率
Flowers5	240 MB	4 322 张	5	320×240
Simpsons13	440 MB	16 615 张	13	340×256

实验环境如下:操作系统为 Ubuntu 17.10,显卡驱动为 NVIDIA 384.81,CUDA 9.0,PyTorch 0.3.0。使用 Torchsummary 记录结构的参数量。使用 R 语言环境下的 Tidyverse 库绘制实验折线图。

超参数设置如下:输入图像分辨率为 224×224,

批大小为 16 个,损失函数为交叉熵,优化方法为随机梯度下降,学习速率为 0.001,动量为 0.9,步长为 8,Gamma 值为 0.1,趟数为 50。

使用测试集分类精度、结构参数量和结构计算量作为评估指标。分类精度选取结构在训练过程中最高的测试集分类精度,以度量结构的能力上限。本文使用 Torchsummary 来记录结构的参数量。结构计算量指结构推断一张图片所需浮点数运算次数 GFLOPI (Giga Floating-Point Operations Per Image)。通过计算得到结构计算量,精简已有公式[14]后得到如下公式:

$$C_{\text{omplexity}} = 2 \times C1 \times C2 \times w \times h \times W2 \times H2 \quad (1)$$

式中:$C_{\text{omplexity}}$ 指结构计算量;$C1$ 指输入通道数量,$C2$ 指输出通道数量;w 和 h 指卷积核的宽和高;$W2$ 和 $H2$ 指输出通道的宽和高。

实验首先使用分组瓶颈和奇异瓶颈修改已有结构,通过评估指标分析策略的有效性。然后对比不同分组数量下 SlimNet 的评估指标,以分析分组数量对 SlimNet 的影响。最后将 SlimNet 同其他结构进行比较,突出 SlimNet 的实用价值。

2.1 分组瓶颈的验证

选取采用瓶颈结构的 ResNet 和 SlimNet 进行验证。其中 ResNet 层数为 50 层,分组数量均取 4。实验分别在有无分组瓶颈两种情况下,将各个结构在两个真实数据集上进行训练,得到最高测试集分类精度,如表 4 所示。分组瓶颈对分类精度影响不大。在 ResNet 中引入分组瓶颈,分类精度没有明显变化,说明分组瓶颈有效解决了分组卷积引起的分组通道间信息不流通的问题。SlimNet 中引入分组瓶颈后,发现在 Simpsons13 数据集上分类精度有所降低。由此可见,不同于 ResNet 中“1”字型结构, SlimNet 中“人”字型结构抑制了分组瓶颈的作用。

表 4 分组瓶颈对分类精度的影响

结构	Flowers5		Simpsons13	
	无	有	无	有
ResNet	0.688	0.687	0.908	0.910
SlimNet	0.785	0.787	0.890	0.876
注:无表示没有使用分组瓶颈,有表示使用分组瓶颈				

同时,实验记录结构参数量,并计算结构计算量,如表 5 所示。分组瓶颈有效降低结构参数量和计算量。在 ResNet 中,分组卷积降低了 27% 的参数量和 34% 的计算量。在 SlimNet 中,分组卷积降低了 25% 的参数量和 40% 的计算量。

表 5 分组瓶颈对结构参数量和计算量的影响

结构	参数量/兆		计算量/GFLOPI	
	无	有	无	有
ResNet	25.56	18.54	7.30	4.78
SlimNet	1.09	0.82	0.30	0.18
注:无指没有分组瓶颈,有指使用分组瓶颈				

在对分类精度影响不大的情况下,分组瓶颈大幅度降低结构的参数量和计算量,是一种有效的轻量化设计策略。

2.2 奇异瓶颈的验证

选择 ResNet、ShuffleNet、MobileNetV2 和 SlimNet,通过实验比较奇异瓶颈对分类精度的影响。其中 ResNet 层数为 50 层。实验分别在有无奇异瓶颈两种情况下,将各个结构在两个真实数据集上进行训练,得到最高测试集分类精度,如表 6 所示。

表 6 奇异瓶颈对分类精度的影响

结构	Flowers5		Simpsons13	
	无	有	无	有
ResNet	0.688	0.768	0.908	0.926
ShuffleNet	0.748	0.778	0.920	0.913
MobileNetV2	0.746	0.782	0.927	0.938
SlimNet	0.749	0.787	0.851	0.876
注:无指没有奇异瓶颈,有指使用奇异瓶颈				

从表 6 可见,奇异瓶颈在各个结构上均能提升分类精度。在 Flowers5 数据集上,奇异瓶颈使得分类精度提高 4.8%。在 Simpsons13 数据集上,奇异瓶颈使得分类精度提高 1.2%。奇异瓶颈对分类精度的提高幅度与其自值有关,即分类精度越高,提高幅度越小。值得注意的是,对于 ShuffleNet,在 Simpsons13 数据集上,无奇异瓶颈的情况下有着更高的分类精度,这可能和 ShuffleNet 自身结构有关。

2.3 分组数量的影响

分组数量分别选择 1、2、4、8 和 16。其中,分组数量为 1 相当于常规卷积。分组数量均选择 2 的指数是为了能够整除通道数量。最大选择 16 是因为 SlimNet 中最小的通道数量为 16。实验结果如表 7 所示。

表 7 分组数量对 SlimNet 的影响

分组数量	分类精度		参数量/兆	计算量/GFLOPI
	Flowers5	Simpsons13		
1	0.785	0.890	1.09	0.30
2	0.772	0.885	0.91	0.22

续表 7

分组数量	分类精度		参数量/兆	计算量/GFLOPI
	Flowers5	Simpsons13		
4	0.787	0.876	0.82	0.18
8	0.768	0.865	0.78	0.16
16	0.766	0.864	0.75	0.15

从表 7 可见,分组数量从 1 增加到 16,在 Flowers5 数据集上分类精度降低 2%,在 Simpsons13 数据集上分类精度降低3%,训练时间也相应增加3 分钟和8 分钟,参数量减少 32%,计算量降低 50%。由此可见,损失少量的分类精度,可以得到参数量和计算量的大幅度降低。在分组数量从 4 增加到 8 时,SlimNet 在 Simpsons13 数据集上的分类精度降低 1.1%,而结构参数量降低 5%,结构计算量降低 11%。由此可见,当分组数量从 4 增加 16 的过程中,参数量和计算量的减少幅度逐渐降低。为了保持较高分类精度,本文建议分组数量为 4 较佳。

值得注意的是,在 Flowers5 数据集上,分组数量为 4 的分类精度高于分组数量为 1 和 2 的,这是因为 Flowers5 数据集自身存在问题,SlimNet 在 Flowers5 数据集上的训练过程不稳定,有较大的起伏(具体原因在下节分析)。但从实验角度考虑,Flowers5 数据集能一定程度上反映结构特性,且训练时间较短,便于结构多次改进和实验的重复迭代,所以本文依旧选择了 Flowers5 数据集。

2.4　结构比较

选择了 AlexNet、ResNet、SqueezeNet、ShuffleNet、MobileNetV2 作为对比结构。ResNet 选用 18 层是因为在实验数据集上分类精度较高,具有可比性。SlimNet 分组数量为 4。

首先,在 Flowers5 数据集上进行比较,各个结构的训练过程如图 7 所示。结构均在 20 趟(Epoch)后趋于收敛。其中 AlexNet 和 SqueezeNet 在 30 趟后几乎无波动,而其余结构一直保持上下波动,这是由于其余结构中添加了批归一化层导致每趟训练改变每层的输入使得分类精度无法稳定在一个值。在 ImageNet-1K 数据集上 AlexNet 和 SqueezeNet 分类精度相同,在 Flowers5 数据集上分类精度不同,说明 Flowers5 数据集自身存在问题影响到了两个结构的分类精度。SlimNet 取得最高的分类精度可能是受到 Flowers5 数据集影响,不能真实反映 SlimNet 在其他数据集上的分类精度。从整体上看,相较于其他结构,SlimNet 在 Flowers5 数据集上分类精度最高。

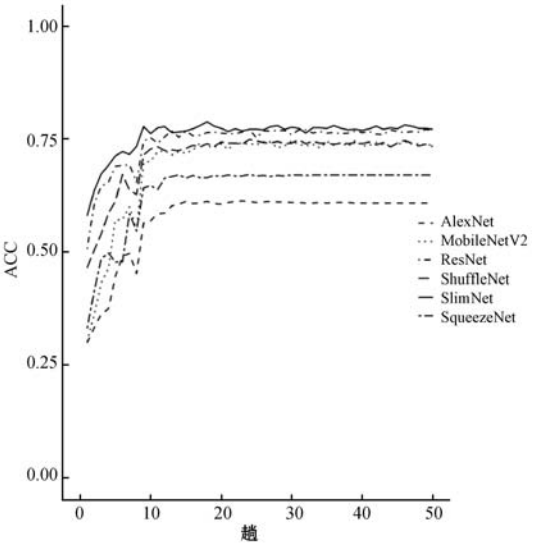

图 7　各结构在 Flowers5 数据集上的训练过程

同时,在 Simpsons13 数据集上进行比较,各个结构的训练过程如图 8 所示。使用批归一化的 ResNet、ShuffleNet、MobileNetV2 和 SlimNet 在 20 趟后趋于收敛。未使用批归一化的 AlexNet 和 SqueezeNet 在 30 趟后趋于收敛。批归一化能够加速结构训练过程,因此本文在改进过程中引入批归一化,分类精度由低到高依次是 AlexNet、SqueezeNet、SlimNet、ResNet、ShuffleNet 和 MobileNetV2。可见 SlimNet 比 SqueezeNet 有更高的分类精度。此时,AlexNet 和 SqueezeNet 分类精度几乎相同,说明 Simpsons13 数据集和 ImageNet-1K 数据集特性接近,则 Simpson13 能够真实反映 SlimNet 的分类精度。从整体上看,相较于其他结构,SlimNet 在 Simpsons13 数据集上能收敛到较好的分类精度。

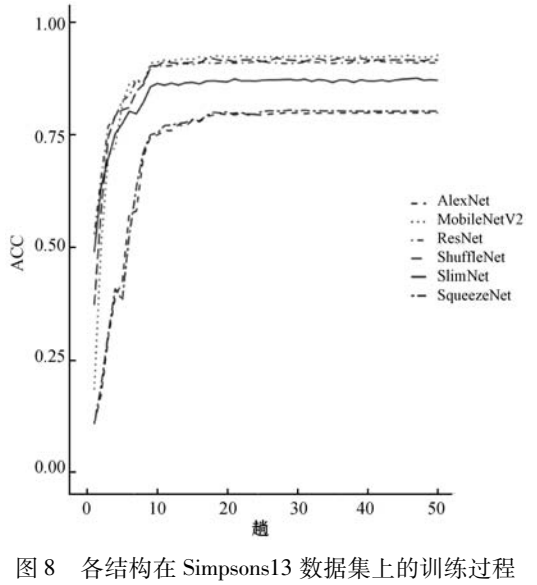

图 8　各结构在 Simpsons13 数据集上的训练过程

两个数据集中,发现各结构在第 8 趟前后均出现不同程度的分类精度突然下降的现象。在第 1 趟时,ShuffleNet 和 SlimNet 均有较高的分类精度,ResNet 次之,MobileNetV2、AlexNet 和 SqueezeNet 最低。需要指

出,MobileNetV2 在两个数据集上的表现差异较大,因为其对数据集的质量有着较高的要求,不利于在现实任务中应用。

比较各个结构的参数量和计算量,如表 8 所示。AlexNet 有最大的参数量,计算量却居中。AlexNet 中三个全连接层的参数占据大部分参数空间,全连接层导致结构参数多。但其结构层数(8 层)较少,卷积部分参数少,故计算量不大。ResNet 有着适中的参数量,计算量却特别大,这是由于 ResNet 使用全局平均池化层(全局平均池化层在 Network In Network 中最早提出)取代全连接层,大大减少结构参数量,但 ResNet 使用了过多的层数(18 层)进行 3×3 卷积,使得计算量过大。SqueezeNet 使用较少 3×3 卷积使得参数量很小,但有 18 层,其中包含大量的 1×1 卷积使得计算量过大。ShuffleNet 使用分组卷积和深度可分卷积,结构参数量和计算量均较小。MobileNetV2 一方面使用深度可分卷积以减少参数量和计算量,另一方面为了增加分类精度,使用反转残差在模块内增加通道数量导致参数量和计算量的增加,故参数量和计算量居中。SlimNet 使用分组瓶颈,使得结构的参数量和计算量均为最小。

表 8 各结构的参数量和计算量比较

结构	参数量/兆	计算量/GFLOPI
AlexNet	61.10	0.72
ResNet	11.69	3.47
SqueezeNet	1.24	0.70
ShuffleNet	1.88	0.30
MobileNetV2	6.06	0.96
SlimNet	0.82	0.18

和 SqueezeNet 相比,SlimNet 在 Flowers5 数据集的分类精度提高 17%,在 Simpsons13 数据集的分类精度提高 9%,参数量减少了 34%,计算量上减少了 75%。实验证明,本文提出的 SlimNet 具有一定的实用价值。

3 结 语

分组瓶颈和奇异瓶颈使得本文提出的策略改进 SqueezeNet 的轻量化结构 SlimNet 在分类精度、结构参数量和计算量上均优于 SqueezeNet。

但 SlimNet 在分类精度上稍低于 ShuffleNet 和 MobileNetV2。这是由于 SlimNet 和 SqueezeNet 采用了相近的整体结构,可改进整体结构以提高分类精度。本文的实验数据集包含图片类别(分别是 5 和 13)稍小,在 ImageNet-1K 数据集(种类为 1 000)上分类精度未知因此可提高实验环境,以测量 SlimNet 在 ImageNet-1K 上的分类精度。未来还可进一步研究 SlimNet 在目标检测等计算机视觉其他领域中的可行性。

参 考 文 献

[1] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]//International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.

[2] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015: 770-778.

[3] Iandola N, Han S, Moskewicz W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size [EB]. eprint arXiv:160207360, 2016.

[4] Sifre L, Mallat S. Rigid-Motion Scattering for Texture Classification[EB]. eprint arXiv:1403.1687, 2014.

[5] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015:2818-2826.

[6] Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions[EB]. eprint arXiv:1610.02357, 2016.

[7] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[EB]. eprint arXiv:1409.1556, 2014.

[8] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[EB]. eprint arXiv:1704.04861, 2017.

[9] Zhang X, Zhou X, Lin M, et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices [EB]. eprint arXiv:1707.01083, 2017.

[10] Sandler M, Howard A, Zhu M, et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks[EB]. eprint arXiv: 1801.04381, 2018.

[11] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[EB]. eprint arXiv:1502.03167, 2015.

[12] Mamaev A. Flowers recognition[EB/OL]. https://www.kaggle.com/ alxmamaev/flowers-recognition/data.

[13] Alexattia. The Simpsons Characters Data[EB/OL]. https://www.kaggle.com/alexattia/the-Simpsons-characters-dataset/data.

[14] Molchanov P, Tyree S, Karras T, et al. Pruning Convolutional Neural Networks for Resource Efficient Inference [EB]. eprint arXiv:1611.06440, 2016.