高技术通讯 2023 年 第 33 卷 第 2 期:156-166

doi:10.3772/j.issn.1002-0470.2023.02.005

Bi-Attention:面向终端的细粒度识别网络加速方法①

钟巧灵② 汪 啸 张志斌 李 冰 程学旗

(中国科学院计算技术研究所网络数据科学与技术重点实验室 北京 100190)

(中国科学院大学计算机科学与技术学院 北京 100049)

摘 要 细粒度识别是针对具有微小差异的对象进行分类的图像识别任务,深度学习模型在细粒度识别任务上取得了较大的进步。然而现有的细粒度识别深度神经网络模型采用多个模型结构叠加,无法在手机、无人机等资源受限终端设备上部署。本文提出 Bi-Attention 细粒度识别模型加速方法,使用高效的 TensorSketch 运算以及权重共享机制,在 Stanford Cars 数据集上的准确率为 91.6%,且比现有最先进的模型提高 1.2%。本文提出一种结构化剪枝训练方法,通过 LASSO 正则化算法,在模型训练过程删除批归一化(BN)操作中不重要的扩展因子。实验结果表明,该剪枝方法可以降低 Bi-Attention 模型大小为原来的 1/4。

关键词 细粒度识别; Attention; 结构化剪枝;L1 正则化; 终端

0 引 言

细粒度识别是一种识别具有微小差异物体的图像分类任务[1],通常识别的对象是同一个大类之间的物体,如鸟的分类[2-3]、车型识别[4-5]等。目前细粒度识别模型[2-3,6-7]关注如何提高细粒度图像特征的获取,如采用多个细粒度级别的特征结合的方法并叠加多个模型。然而通过叠加多个网络模型的细粒度识别模型增加了模型的计算复杂度和空间复杂度,无法部署这些细粒度识别模型在智能手机、智能家居等资源受限的终端设备上[8-11]。

实验表明,复杂深度神经网络模型存在大量冗余的参数。文献[8,10-14]指出,给定一个训练收敛的神经网络模型,只需要约 5% 的参数,便能完成模型的推断并且能够重构出剩余的模型参数。在深度神经网络模型实际部署和推断过程中,大量的神经网络参数需要消耗更多的计算和存储资源。通过模型剪枝方法可以大幅降低模型的计算复杂度。但细粒度识别模型需要对具有细微差异的输入图像进行分类识别,传统卷积神经网络(convolutional neural network, CNN)删除网络权重的剪枝方法容易导致细粒度分类模型性能下降,无法满足模型部署和使用需求。因此,删除细粒度分类任务模型的冗余参数难度更大。

本文设计低计算复杂度模型算子和模型剪枝训练方法,部署细粒度识别网络模型在资源受限的终端设备上,设计 Bi-Attention 细粒度图像分类模型和训练时自动化模型压缩训练方法,在模型训练过程中完成模型剪枝并使模型最终收敛。本文的主要贡献包括以下 3 个方面:(1)设计了一种 Bi-Attention 细粒度图像识别模型,将现有的细粒度识别在 Stanford Cars 数据集上的准确率提高到 91.6%;(2)提出了一种基于 L1 正则化的自动结构化模型剪枝训练方法,压缩 Bi-Attention 细粒度识别模型;(3)在细粒度识别数据集 Stanford Cars、CUB Birds 和 FGVC Aircrafts 上,采用 Bi-Attention 模型压缩训练方法加速细粒度识别模型可以达到 4 倍的压缩比。

① 中国科学院战略性先导科技专项(A 类)(XDA19020400)资助项目。

② 男,1991 年生,博士生;研究方向:计算机软件与理论,深度学习加速;联系人,E-mail: zhongqiaoling@ict.ac.cn。

(收稿日期:2021-09-24)

1 相关工作

1.1 细粒度识别神经网络模型

目前细粒度图像识别相关研究主要采用强监督学习的方法训练和得到收敛的深度神经网络模型,模型训练时除了图像的类别标签外还需要额外的图像标注信息,因此在实际场景部署推断和模型训练过程中需要对应图像的标签数据。如文献[1]提出Part-based R-CNN,参考 RCNN 设计并组合各个关键局部特征的信息。从局部区域检测角度提取局部特征进行整合分类,类似的模型还有基于局部分割的Mask-CNN[15]。强监督学习的方法利用图像局部标注框、关键点等额外的标注信息对图像进行分类,涉及检测、分割、分类等多个阶段,整体流程比较复杂。

研究人员探索和尝试弱监督学习方法,不需要使用额外的局部标注信息。文献[16]提出在给定目标数据集的基础上,从一个大的源数据集里根据EMD(earth mover's distance)筛选出与目标数据集相似的子数据集。文献[17]提出采用汇合的方式对双路 CNN 提取的特征进行增广并采用 triplet loss 模型。总体而言,弱监督学习方式具有更好的通用性以及更少的数据依赖。本文提出的细粒度识别模型加速方法是一种弱监督学习方法下的模型加速框架,具有更强的适应性。

1.2 深度神经网络模型部署

细粒度识别模型实际大量部署在终端设备上,除了考虑模型性能之外,还需要考虑计算设备的计算能力,如终端环境[9]计算能力是数据中心计算设备计算能力的1/10 000。目前有大量的研究人员从事模型加速研究,包括模型计算、参数表示和模型剪枝等方法。

在模型计算方面,低秩近似通过多个小规模的低秩矩阵重构出原本稠密的大矩阵,早期的卷积网络模型存在大量的卷积和全连接操作,如 VGG16/19。对于单层卷积层,文献[18]采用低秩方法能够达到2 倍以上的加速,并且原始模型只有 1% 的准确率下降。随着网络深度增加,无论是卷积核近似还是矩阵分解都会带来较大的计算开销和误差,对最后精度损失影响较大。

在参数表示方面,文献[19]尝试采用参数量化法来对模型权重参数近似替代,该方法的主要过程是在权重参数中选取参数代表,例如通过对权重聚类的方式获取,之后采用这些参数代表表示这一类别的具体权值;或采用 low-bit 参数表示方法,如用 uint8 替换 float 运算。文献[20]提出基于 k-means 算法的量化方法。文献[21]提出采用 8-bit 实值对 32-bit 浮点型。文献[22]提出采用 16-bit 定点数代替 32-bit 浮点型。然而 1-bit、2-bit 等量化后的模型需要特定的硬件处理,本文提出的细粒度识别模型加速方法可以和 8-bit、16-bit 量化方法结合使用。

在模型剪枝方面,神经网络模型剪枝[8,10-14]可以删除神经网络中多余的参数。为了使得压缩后的模型可以在硬件设备上高效执行,通道剪枝方式[23-24]可以使模型更加结构化,这样就可以利用高度优化的线性代数运算库(Basic Linear Algebra Subprograms, BLAS)线性代数运算库。但是当前的模型压缩算法集中通用的图像识别任务,而且细粒度图像识别任务需要更加精细的图像特征来执行识别细粒度图像。

2 Bi-Attention 细粒度识别模型

本节主要介绍细粒度识别模型 Bi-Attention 的模型架构及模型原理。

2.1 Bi-Attention 模型架构

Bi-Attention 模型以 Bilinear CNN 模型为基础框架,并嵌入基于 SENet 的注意力分支模块以及基于欧式距离的成对混淆损失函数。图 1 是 Bi-Attention 细粒度分类模型的架构,整个模型由 4 个基本模块构成。第 1 部分是数据模块,处理训练和验证过程数据输入的构造和预处理;第 2 部分是 Bilinear CNN 的卷积神经网络结构,处理核心特征的提取;第 3 部分是注意力分支,该分支用于提取 Bilinear 特征的注意力区域,即可能的含有高区分度局部特征;第 4 部分是损失函数计算,这一部分在训练阶段结合基础的 softmax 分类损失函数与基于欧式距离的成对混淆损失函数,作为模型的整体损失函数来训练模型。

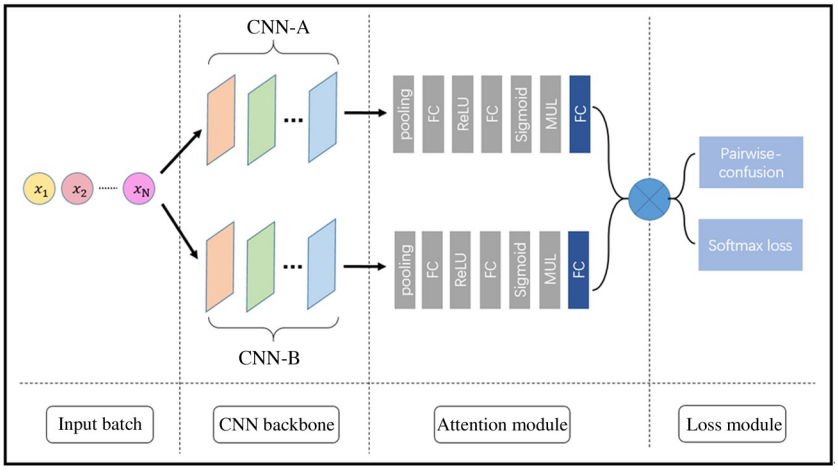


图 1　Bi-Attention 模型结构

基于注意力机制的 CNN 模型有较好的可解释性，其基本思路来源于人类的注意力模式，采取“先定位、再分类”的方式。获取细粒度图像包含高区分度特征的注意力区域，例如鸟的头部与翅膀，从而得到较好的网络特征，在保证模型准确率的同时还具有可解释性，能够进一步指导模型改进。但是注意力机制的网络模型大部分是多阶段（multi-stage）模型，训练过程复杂。本文提出的 Bi-Attention 深度模型架构，融合了注意力机制的可解释性与强可区分局部特征获取，另一方面利用度量学习的网络表示能力，防止模型过拟合，提高了模型的泛化能力。该模型整体上结构简洁规整，是一个一阶段、可端到端训练的网络模型。2.2 节将详细介绍 Bi-Attention 模型结构中的各个组成部分。

2.2　Bi-Attention 模型

2.2.1　双路 CNN 骨干网络

如图 1 所示，Bi-Attention 模型的基础骨干网络是对称的双路 CNN 网络，参考非对称双路 Bilinear CNN[4]。本文主要提出的是一种面向细粒度识别高效率神经网络框架，骨干网络的表达能力越强，越有助于提高模型最终的准确率。对于卷积神经网络，研究表明对于较深的 CNN 网络，其浅层特征提取的主要是图像的边缘、颜色信息；较深层的特征捕获的则是图像纹理层次的信息；末层的卷积特征表示更抽象的整体的图像特征。本文选取 ResNet 50 作为骨干网络，验证细粒度识别模型结构的准确率和效率。

对于图像分类而言，单独提取中、浅层的图像卷积特征没有太多的实际意义，其作用对于整体的 CNN 网络可以理解为是在协助网络处理图像特征转换，对于识别最终起主导作用的是深层的抽象特征。对于 Bi-Attention 的双路 CNN 网络，采用浅层网络权重共享的策略，减少网络的存储和计算资源的消耗，降低模型在执行前向操作和反向传播过程中的时间。本文提出的 Bi-Attention 框架采用全部卷积层共享机制，图 2 所示的是 ResNet 50 的全卷积共享权重机制。

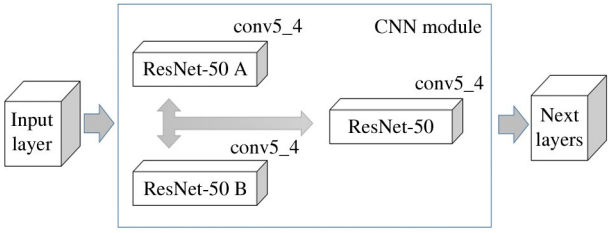


图 2　ResNet50 权重共享

2.2.2　注意力分支

参考 MAMC[6] 中提取图像注意力区域的方法，在 SE block 的子模块基础上增加一个全连接层形成的注意力提取子网络形成注意力网络子模块，如图 3所示。双路权重共享 CNN 子模块的输出特征首先经过 SE block 中的 2 层全连接层（FC）与 Sigmoid 层的“Squeeze”网路，学习特征通道的摘要信息；随后经过逐通道相乘与 FC 变换的“Excitation”网络，学习整合了不同通道信息的注意力特征。这

可以从高维度基础卷积特征聚类角度来理解,即相似通道的特征图聚类到一起,反映到输入图像上则是图像的局部可区分特征往往集中在图像的某一处。

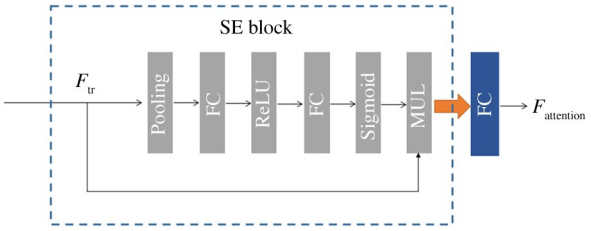


图 3　Attention 分支网络结构图

2.2.3　紧凑的 Bilinear 特征

在计算双路对称 CNN 的 Bilinear 特征过程中,双路 CNN 特征求外积之后得到的特征向量维度太高,达到 10 万量级,特征向量计算过程对于内存和运算带来的资源开销极大,因此需要降低 Bilinear 特征的维度并获取紧凑的 Bilinear 特征。给定注意力分支输出特征 F_1、F_2,维度都是 n,设计 Tensor-Sketch 算法对 F_1、F_2 作压缩的 Bilinear 汇合,输出的特征维度是 $d(d << n)$。TensorSketch 算法细节如算法 1 所示。

算法 1　Tensor Sketch 算法

输入:$F_1, F_2 \in R^n$
输出:$\Phi(F_1, F_2) \in R^d$ (d 是用户特征维度)
for $k \leftarrow 1 \cdots 2$ do
　if h_k, s_k 没有初始化,then
　　for $i \leftarrow 1 \cdots n$ do
　　从 $\{1 \cdots d\}$ 中随机抽样赋值给 $h_k[i]$
　　从 $\{-1, +1\}$ 中随机抽样赋值给 $s_k[i]$
　　done
　$F'_k = \psi(F_k, h_k, s_k, n)$
done
$\Phi = FFT^{-1}(FFT(F_1') \odot FFT(F_2'))$
函数 $\psi(F, h, s, n)$:
　$y = [0, \cdots, 0]$
　for $i \leftarrow 1 \cdots n$ do
　　$y[h[i]] = y[h[i]] + s[i] \times F[i]$
　return y

算法 1 中的 FFT 是快速傅立叶变换(fast Fourier transformation, FFT)算法,⊙是每个元素之间的点乘操作。上述算法是从核函数的角度对双线性池化(Bilinear pooling)的数学抽象,可以在保证准确率几乎无损的情况下将 Bilinear 特征的维度降到 1000 的数量级。

2.2.4　成对混淆损失

本文参考成对混淆损失方法,通过对每一批的训练数据添加成对的混淆约束,使其减少对同类别图像的距离,增大其与不同类别图像的距离。具体过程如下:给定输入数据 x 和标签 y,模型权重为 θ,模型表示为分布函数 $p_\theta(y \mid x)$,对于任意一批输入图像中 x_1、x_2 的成对混淆损失 $H_{EC}(p_\theta(y \mid x_1), p_\theta(y \mid x_2))$ 表示如式(1)所示。

$$H_{EC}(p_\theta(y \mid x_1), p_\theta(y \mid x_2)) = \sum_{i=1}^{N} (p_\theta(y \mid x_1), p_\theta(y \mid x_2))^2 \tag{1}$$

其中 N 为批数据的批大小(batch size),网络模型参数表示为 θ。

从式(1)中可以看出,成对混淆损失实际上是对输入图像对之间的不同类图像添加了基于欧式距离的相似性约束,该约束用于控制图像类别之间的距离。而整体的网络损失函数 L 由 Softmax 交叉熵损失 L_{CE} 和成对混淆损失 H_{EC} 共同构成。

$$L = L_{CE} + \lambda \times H_{EC} \tag{2}$$

其中 λ 作为正则参数调整成对混淆损失 H_{EC} 在整体损失函数中的比重,λ 越大模型对批数据中不同类图像的距离约束越紧。

2.3　骨干网络模型选择

Bi-Attention 中的骨干网络选择采用深度卷积神经网络模型,在模型训练过程中,基于 ImageNet 的预训练模型权重开展模型权重的细粒度识别模型训练。本文采用 ResNet 50 验证 Bi-Attention 模型框架效果。理论上,增加骨干网络模型的深度以及宽度,Bi-Attention 模型的准确率会更高。

3　Bi-Attention 模型压缩训练

针对第 2 节中描述的 Bi-Attention 细粒度识别模型,本节主要介绍如何针对上述模型进行训练,改进模型训练过程。

3.1 Bi-Attention 模型压缩

Bi-Attention 中的骨干网络需要消耗大量的计算资源和存储资源,如 ResNet 50 执行一次前向操作需要 4 GFLOPs 和 103 MB 的特征存储空间。为了部署 Bi-Attention 模型到手机等资源受限设备上,本文设计在训练时剪枝的模型训练方法,删除 Bi-Attention 中双路结构方式引入的冗余的权重参数以及 Bi-Attention 架构中主干网络的冗余计算。文献[25]总结了 2 种剪枝方法,结构化剪枝(删除过滤器 filter、通道 channel 或网络子结构 sub-network)和非结构化剪枝(删除单个权重,不考虑网络结构)。非结构化剪枝需要特定的硬件支持才能加速剪枝后的网络模型结构,为了能够在现有的硬件设备和软件系统上运行模型,减少对硬件设备的依赖,本文设计结构化剪枝训练算法。

本文构建的细粒度识别模型 Bi-Attention 的骨干网络是 ResNet-50(也可以替换成其他深度卷积神经网络模型)。当下先进的卷积神经网络模型由卷积运算、批归一化(batch normalization,BN)和激活函数等运算构成,且 BN 操作与卷积运算作为一个整体出现(BN 操作在卷积运算之前或之后),关于 BN 中的通道因子与模型计算复杂度的关系见 3.2节中 BN 通道因子的描述。本文针对 BN 操作剪枝方法,在模型训练过程中自动化确定剪枝间隔并完成模型剪枝,具体过程见 3.3 节基于 L1 正则化的剪枝训练算法。

3.2 批归一化的通道因子

假定模型的特征格式是 NCHW,其中 N 是 batch size、C 是通道(channel)数目、H 是高度(height)、W 是宽度(width),其卷积权重的参数维度是 K_{n}、K_{c}、K_{h}、K_{w},卷积的输出特征维度分别是 $Z_{\mathrm{out\_}n}$、$Z_{\mathrm{out\_}c}$、$Z_{\mathrm{out\_}h}$、$Z_{\mathrm{out\_}w}$,则该卷积运算的浮点运算数目为 $Z_{\mathrm{out\_}n} \times Z_{\mathrm{out\_}c} \times Z_{\mathrm{out\_}h} \times Z_{\mathrm{out\_}w} \times K_{\mathrm{n}} \times K_{\mathrm{c}} \times K_{\mathrm{h}} \times K_{\mathrm{w}}$,并且 $Z_{\mathrm{out\_}c} = K_n$。根据批归一化(BN)[44]描述,BN 的形式如式(3)所示。

$$Z_{\mathrm{out}} = \gamma \times \frac{Z_{\mathrm{in}} - \mu_\theta}{\sqrt{\sigma_\theta^2 + \varepsilon}} + \beta \tag{3}$$

其中,Z_{out} 是 BN 操作的输出特征,Z_{in} 是 BN 操作的输入特征同时也是卷积运算操作的输出特征;μ_θ 与 σ_θ 表示输入的批数据的均值与标准差,γ 与 β 是 BN 层的学习参数,分别表示通道因子与偏移。通道因子 γ 的维度是 $Z_{\mathrm{in\_}c}$,如果减少通道因子 γ,则减少了 BN 操作之前卷积运算的权重参数 K_n,从而减少了卷积浮点运算。

3.3 L1 正则化自动化迭代剪枝训练算法

文献[26]指出在神经网络模型训练初始迭代阶段,存在一个子网络结构,该子网络结构参数更少并且准确率与原始复杂稠密模型一致。针对随机初始化的 Bi-Attention 模型执行训练时迭代式结构化剪枝,给定指定的稀疏度,自动化完成通道剪枝过程。

针对神经网络 $f(w; x)$,模型权重是 w,模型的输入是 x,在模型剪枝训练过程中第 t 次迭代的损失结果为 l_t,总的迭代次数为 N_T。用通道因子 γ 的 mask 矩阵来表示 γ 是否被删除,如果 mask 矩阵中的值为 1,则表示对应的通道因子 γ 保留;如果为 0,则表示删除。利用 L1 正则稀疏化通道因子 γ 的损失函数表示为式(4)。

$$L = \sum_{x,y} l(f(w; x), y) + \lambda \times \sum_i |\gamma_i| \tag{4}$$

其中 λ 是用来控制 L1 正则化的约束力度,是需要调试的参数,l 是模型的损失函数。具体的剪枝训练算法如算法 2 所述。

通过算法 2 得到的是一个剪枝后的模型,这里采取的策略是忽略 BN 操作中的偏移因子。在算法 2 中,记录模型剪枝算法过程中的损失函数变化,如果模型的损失函数输出变化小于 ϵ,则利用 loss 损失函数输出变化时刻的迭代次数来记录模型自动剪枝的间隔 T。

算法 2 Bi-Attention 迭代式剪枝训练算法

输入:Bi-Attention 模型,剪枝率队列 P,阈值 ϵ

输出:剪枝后的 Bi-Attention 模型

```
记录 loss 变化 Δl
初始化剪枝间隔 T
for t ← 1…N_T do
    if Δl < ε, then
        更新剪枝间隔 T 为 t
done
```

```
for t ← t_last ⋯ N_T do
    if t mod T = = 0 and P 不为空,then
        从剪枝率队列中选择一个剪枝率 p
        获取全局的通道因子 γ 并排序
        计算所有通道因子 γ 中第 p 大小的值 γ_p
        更新 mask 矩阵对应值为 1 如果大于 γ_p
    用随机梯度下降(stochastic gradient descent,SGD)训练模型
done
```

4 实验结果与分析

本节主要验证细粒度识别模型 Bi-Attention 在细粒度图像数据集 Stanford Cars、CUB Birds 和 FGVC Aircrafts 上的效果。

4.1 性能度量标准

本文实验验证和对比性能指标包括模型准确率、模型参数量、模型压缩比和模型 FLOPs,这些指标计算方式如下。

(1) 模型准确率表示压缩模型在细粒度识别数据集上的识别准确率。

$$accuracy = \frac{\#ofaccuratesamples}{\#oftotalsamples}$$

(2) 模型参数量表示深度学习模型参数的大小,以单精度浮点数数据类型为存储单位。

(3) 模型压缩比表示模型压缩前与压缩后模型在模型参数量大小上的比例。

$$compression\ rate = \frac{original\ size}{compressed\ size}$$

(4) 模型 FLOPs 表示深度学习模型的浮点数运算次数,包括乘法和加法操作。

4.2 实验测试数据集

测试和验证 Bi-Attention 模型的数据集如表 1 所示。Stanford Cars 是车型细粒度识别数据集,各类别图像数据量大小不一致,整体数量呈高斯分布,输入大小约为 650×450 像素;CUB Birds 是鸟类品种的细粒度识别数据集,共有 200 类,每个类别大约各 60 张,平均输入大小大约为 650×450 像素;FGVC Aircrafts 是飞行器品种的细粒度识别数据集,共有 100 个子类,每个子类共 100 张图像,平均分辨率大约为 1000×700 像素。

表 1 细粒度图像识别数据集

数据集	规模	训练/测试	种类
Stanford Cars	16 185	8144/8041	196
CUB Birds	11 788	5994/5794	200
FGVC Aircrafts	10 000	3334/3333	100

以上 3 个数据集均只采用数据标签信息作为监督信息训练,不采用任何边界框、部分标注等其他额外标注信息。

4.3 Bi-Attention 细粒度识别准确率分析

本次实验过程中,批大小设置为 32,模型的稀疏度为 0;对于训练数据,预处理方法采用去均值、减方差、固定 256 像素等比例缩放和 224 像素随机裁剪,对于测试数据将 224 像素随机裁剪改成 224 像素中心裁剪。模型的优化方法均为动量 SGD,训练过程学习率(learning rate, LR)调度策略为指数衰减策略,最大迭代次数设为 80 000,训练轮次(epoch)为 100 轮次。

图 4 和图 5 给出 Bi-Attention 模型在 Stanford Cars 数据集上训练过程的损失函数以及准确率变化图。从图4和图5中可以看到,在模型训练过程中

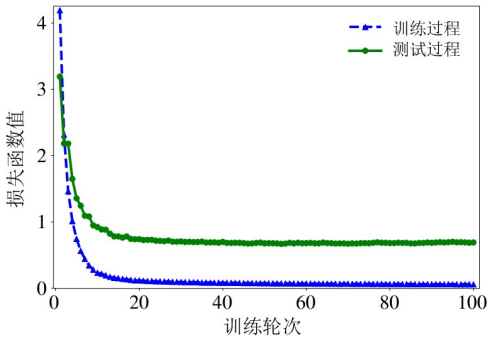


图 4 训练过程中的损失函数

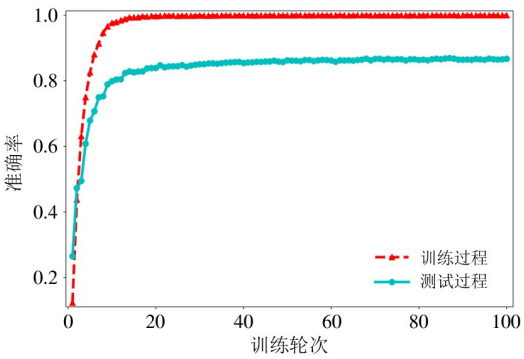


图 5 训练过程中的准确率函数

损失函数值在初始 epoch 阶段是一个较高的数值，与之对应的模型准确率也很低；随着训练的进行，模型逐渐收敛，大约在第 20 个 epoch 开始趋于稳定；Bi-Attention 模型在测试集上 top-1 准确率为 85.8% 左右，top-5 准确率为 96.5%，由此验证了 Bi-Attention 模型的可行性与有效性。

进一步地，本文将 Bi-Attention 模型和主流细粒度图像识别模型做了相应的对比实验，验证 Bi-Attention 模型在精度上的提升。本次实验在数据部分与现有的主流细粒度识别方法的配置保持一致，采用 batch size 为 64，输入图像为 448 × 448 像素的输入数据设置，实验结果准确率对比数据如表 2 所示。

表 2 Bi-Attention 模型与其他模型的准确率对比

	Stanford Cars	CUB birds	FGVC Aircrafts
ResNet 50[6]	88.4%	77.3%	84.6%
Bilinear CNN[4]	90.3%	80.9%	85.1%
MAMC-SE[6]	89.6%	78.9%	85.4%
PC-ResNet[27]	90.5%	81.2%	85.2%
PC-Bilinear[27]	90.4%	82.1%	85.7%
Bi-Attention	**91.6%**	**82.3%**	**88.0%**

从表 2 中可以看出，Bi-Attention 模型在 Stanford Cars、CUB Birds 和 FGVC Aircrafts 数据集上的准确率分别为 91.6%、82.3% 和 87.0%，与其他模型相比均为最优。

表 3 对比了深度复杂神经网络模型的深度和宽度对于细粒度识别性能的影响。通过表 3 可以发现，Bi-Attention 的准确率与 MAMC-ResNet 101、PC-DensNet 161 等模型仍有差距：在 CUB Birds 数据集上比 PC-DenseNet 161 差 4.5%，在 Stanford Cars 数据集上比 PC-DenseNet 161 差 1.2%，在 FGVC Aircrafts 数据集上比 PC-DenseNet 161 差 1.3%。本质上是 PC-DenseNet 161 和 MAMC-ResNet 101 的骨干网络分别是 DenseNet 161 和 ResNet 101，其中 DenseNet 161 和 ResNet 101 网络模型深度和宽度明显高于 ResNet 50，本文中的 Bi-Attention 模型骨干网络结构是 ResNet 50。

本文提出 Bi-Attention 模型主要是为了能够在资源受限场景下部署细粒度识别模型，DenseNet 161 和 ResNet 101 计算复杂度和空间复杂度明显高于 ResNet 50，无法在手机等终端场景下有效部署。在未来实验中，将进一步选取ResNet101等骨干网络验证 Bi-Attention 模型结构以及训练方法在资源受限场景下的有效性和可行性。

表 3 Bi-Attention 模型与 SOTA 模型准确率对比

	模型	准确率	最好准确率	Bi-Attention 准确率
Stanford Cars	MAMC-ResNet-101[6]	93.0%	93.0%	91.6%
	PC-DenseNet-161[27]	92.8%		
CUB Birds	PC-DenseNet-161[27]	86.8%	86.8%	82.3%
	MAMC-ResNet-101[6]	85.2%		
FGVC Aircrafts	PC-DenseNet-161[27]	89.3%	89.3%	88.0%
	MAMC-ResNet-101	-		

4.4 L1 正则化模型压缩训练算法分析

本节分析 Bi-Attention 模型在不同稀疏度下的模型训练收敛的结果。

准确率分析：根据算法 2 中的迭代剪枝算法，对原始 Bi-Attention 模型的 BN 层的稀疏化通道因子 γ 添加 L1 正则训练模型。输入一个随机初始化的模型，根据 γ 值的大小对 Bi-Attention 模型进行剪枝，在训练过程中，依托式(4)设置以及控制 L1。本文中实验 λ 设置为 0.0001，同时设置全局剪枝比例为 40%。

模型训练过程的其他参数配置如下：batch size 设置为 32；输入图像大小设置为 224，采用减均值、去方差处理的方法预处理方法；学习方法为动量法(momentum)的 SGD，momentum 大小设置为 0.9，学习率为 0.01，权重衰减为 0.0005，学习率衰减策略为指数衰减策略，最大迭代次数为 80 000，总训练

epoch 为 100。设置训练时剪枝迭代算法 2 中的参数 ϵ 为0.5,针对 Bi-Attention 执行训练时剪枝训练的结果如表 4 所示。

从表 4 中可以得出,当剪枝率为 40% 时,Bi-Attention 模型在测试集上准确率损失与原始模型相比平均控制在 1% 以内,在 Stanford Cars 数据集上几乎没有准确率损失,在 FGVC Aircrafts 数据集上反而有细微的提升。因为 Bi-Attention 的骨干网络 ResNet 50 初始化为基于 ImageNet 数据集预训练的权重,模型在训练初始阶段的 epoch 时刻开始执行剪枝而不会影响模型的准确率。同时因 Bi-Attention 模型的结构主体是由 ResNet 50 构成,辅以 SENet 的注意力网络结构,模型主体是一个线性结构,并且 BN 网络层应用广泛,可以更好地开展对于 L1 正则的稀疏化通道剪枝。

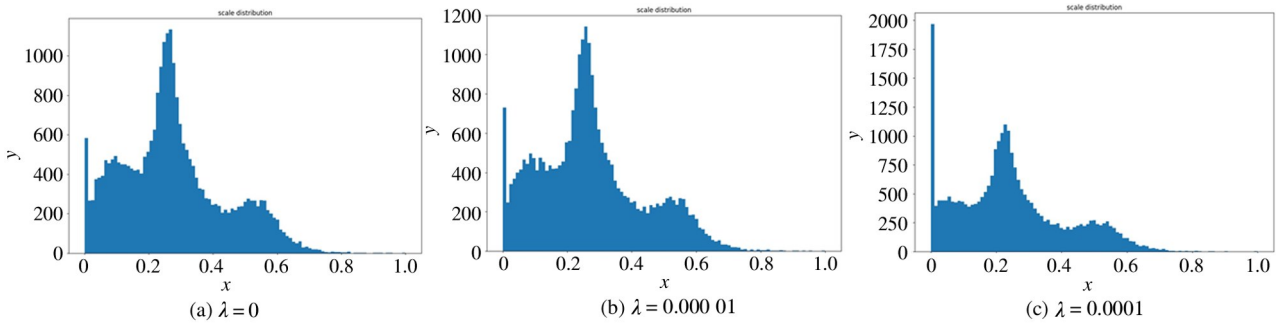
表 4　Bi-Attention 模型剪枝准确率降低对比

	原始模型	剪枝后模型	准确率损失
Stanford Cars	86.9%	86.7%	0.2%
CUB Birds	76.3%	75.2%	1.1%
FGVC Aircrafts	80.1%	80.5%	-0.4%

稀疏化通道因子分布:为了验证 L1 正则对稀疏化通道因子的影响,本实验在 Stanford Cars 数据集上设置 3 组不同的 L1 正则参数 λ, 分别为 0、0.0001和 0.000 01,其他实验参数设定与前面保持一致。观察在不同大小 λ 的情况下的稀疏化通道因子的参数分布,可以发现不同强度的 L1 正则对网络参数的稀疏化程度的影响。

图 6(a)描述的是没有添加 L1 正则约束的原始 Bi-Attention 模型稀疏化通道因子 γ 的分布,数据表明原始 Bi-Attention 模型网络参数大多以 0.3 为中心偏向 0 聚合。图 6(b)和(c)分别表示 λ =0.000 01 和 λ =0.0001 的 γ 的分布,当 λ 值增大时,对应的图中分布中心更向 0 靠近聚合,模型参数的稀疏化程度越高,当 λ =0.0001 时靠近 0 值的参数最多。这与式(4)的优化目标一致,也从侧面说明了算法的可行性。

基于 L1 正则的稀疏化通道训练时剪枝方法是根据这些不同程度的稀疏化参数因子对 Bi-Attention 模型进行剪枝。稀疏化程度越大,删除的参数越多,模型越小,对模型的准确率损失越大,反之亦



(a) λ = 0　(b) λ = 0.000 01　(c) λ = 0.0001

图 6　稀疏化通道因子分布图

然。因此寻找合适的 λ 值也是 Bi-Attention 模型训练算法的重要组成部分。

4.5　L1 正则化模型压缩效果分析

在 3 个数据集上分别验证在不同稀疏度下 Bi-Attention 模型训练效果,每组实验中分别设置不同的全局剪枝阈值 40% 、50% 和 60% 。

实验对比分析分别测量模型在不同的数据集和不同的全局剪枝阈值下的模型准确率、模型参数量和 FLOPs 来说明压缩算法的性能。模型训练过程参数设置与 4.4 节中的配置保持一致,表 5 ~ 表 7 是具体的实验数据。

从表 5 ~ 表 7 中可以得知在 3 个测试数据集上,对于 40% 和 50% 的全局剪枝,其对准确率基本没有影响,一些数据集上还有细微的提升,对应的参数量与 FLOPs 的剪枝比例平均分别为 20% 与 37% 。对于 60% 的剪枝,整体数据集的准确率损失平均在

2.5% 左右，但 Bi-Attention 模型对应的参数量与 FLOPs 的剪枝率均达到了接近于基础模型的 50%。整体来看，较小的全局阈值对准确率几乎没有影响甚至有所提升，较大的权值剪枝比例对整体模型的准确率有 2% ~3% 的模型损失，但是压缩比例能达到原来的一半，Bi-Attention 模型性能上损失较小。

表 5　Stanford Cars 上 Bi-Attention 模型剪枝性能对比

	准确率	参数个数(M)	减少	FLOPs(G)	减少
原始模型	86.9%	29.18	0	3.88	0
删除 40% 权重	87.1%	24.02	17.7%	3.03	20.3%
删除 50% 权重	86.7%	18.55	36.4%	2.42	36.3%
删除 60% 权重	84.5%	15.19	47.9%	1.95	48.7%

表 6　CUB Birds 上 Bi-Attention 模型剪枝性能对比

	准确率	参数个数(M)	减少	FLOPs(G)	减少
原始模型	76.3%	29.18	0	3.88	0
删除 40% 权重	75.9%	24.05	17.5%	3.02	22.2%
删除 50% 权重	75.2%	18.56	36.4%	2.42	36.3%
删除 60% 权重	73.8%	15.21	47.9%	1.95	48.7%

表 7　FGVC Aircrafts 上 Bi-Attention 模型剪枝性能对比

	准确率	参数个数(M)	减少	FLOPs(G)	减少
原始模型	80.1%	29.18	0	3.88	0
删除 40% 权重	80.2%	23.83	18.3%	3.03	20.3%
删除 50% 权重	80.5%	18.38	37.0%	2.42	36.3%
删除 60% 权重	77.1%	15.01	48.6%	1.95	48.7%

对比模型的实际静态存储在剪枝前后的变化。由于在上述 3 个数据集上的模型整体大小相近，在 Stanford Cars 数据集上测试 Bi-Attention 模型剪枝变化，实验结果如图 7 所示。随着全局剪枝阈值比例的增大，模型实际存储空间逐渐减小。

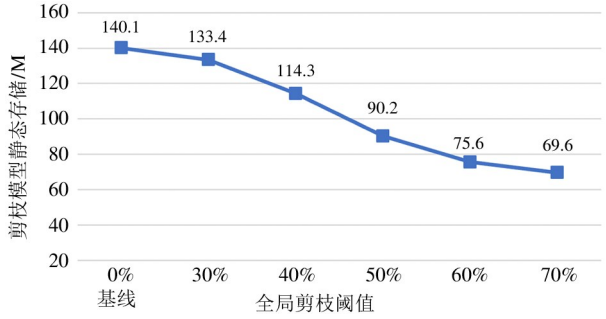


图 7　不同剪枝阈值下的剪枝模型大小

测试验证对 Bi-Attention 模型执行多次迭代剪枝压缩训练，通过迭代剪枝可以进一步压缩训练 Bi-Attention 模型以得到更大的压缩比和精度损失均衡。根据算法 2 的描述，通过输入不同轮次的剪枝率，观测每一轮剪枝的 Bi-Attention 模型准确率与压缩比。本次实验每一轮剪枝设置的全局剪枝阈值均为 50%，其他设置与前文保持一致，实验结果如图 8 和图 9 所示。

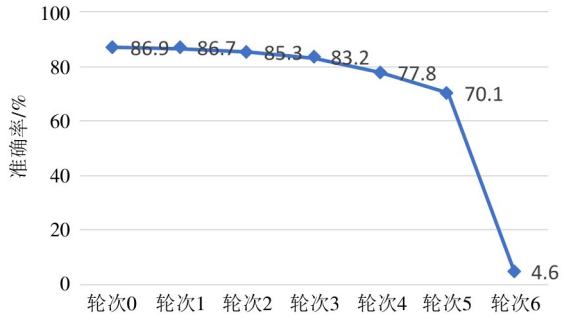


图 8　多次剪枝准确率变化曲线

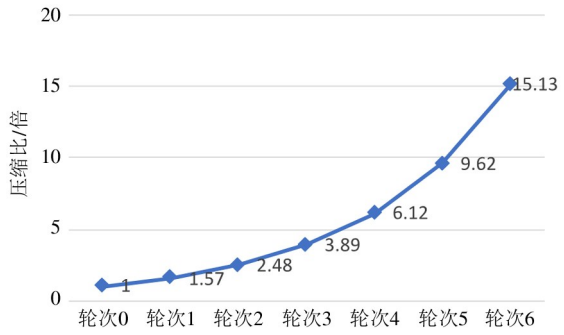


图 9　多次迭代压缩比变化曲线

从图 8 和图 9 中可知,对于 Bi-Attention 模型的压缩,随着剪枝轮次的增加,模型的压缩比逐渐增大,模型的准确率逐渐减小,当超过某一轮次(轮次 5)时,准确率出现断崖式下跌,这说明已经到达模型可接受剪枝的最大阈值。另外还可以发现,对于 Bi-Attention 模型,在准确率损失可接受的范围 5% 之内,模型最大的压缩比可以达到接近 4 倍的压缩,Bi-Attention 模型大小将减少为原始的 25%,进一步验证了 Bi-Attention 模型压缩训练算法的实用性。

本文测试验证 Bi-Attention 模型加速方法在树莓派 Model 4B 上的性能,主要测试指标为延迟。表 8描述了 Bi-Attention 模型在 Stanford Cars 数据集上不执行剪枝、删除 40% 权重以及删除 50% 权重的模型推断速度。实验表明,本文提出的模型剪枝方法可以提高识别速度约 2 倍,同时模型的性能基本不变。

表 8　Bi-Attention 模型在树莓派上的运行速度

	延迟/ms	准确率/%
原始模型	186	86.9
删除 40% 权重	112	87.1
删除 50% 权重	98	86.7

5　结　论

在实际部署细粒度图像识别模型过程中,需要兼顾神经网络模型性能和模型运行速度。本文设计了 Bi-Attention 细粒度识别模型,并设计基于 L1 正则的训练时剪枝训练方法,减少 Bi-Attention 细粒度模型运行时的时间和空间消耗。当稀疏度为 0 时,在 Stanford Cars、CUB Birds 和 FGVC Aircrafts 数据集上的准确率分别为 91.6%、82.3% 和 87.0%,相比较其他模型准确率提高 1% ~4%。基于 L1 正则化的自动化模型压缩训练方法,在保证 Bi-Attention 模型准确率基本保持不变的同时,实现了最多 4 倍的压缩效果比。

参考文献

[1] ZHANG N, DONAHUE J, GIRSHICK R, et al. Part-based R-CNNs for fine-grained category detection[C]// European Conference on Computer Vision. Zurich: Springer, 2014: 834-849.
[2] BRANSON S, VAN HORN G, BELONGIE S, et al. Bird species categorization using pose normalized deep convolutional nets[EB/OL]. (2014-06-11)[2021-09-24]. https://arxiv.org/pdf/1406.2952.pdf.
[3] ZHANG X, XIONG H, ZHOU W, et al. Picking deep filter responses for fine-grained image recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016: 1134-1142.
[4] LIN T Y, ROYCHOWDHURY A, MAJI S. Bilinear CNN models for fine-grained visual recognition[C]// Proceedings of the IEEE International Conference on Computer Vision. Santiago: IEEE, 2015: 1449-1457.
[5] KRAUSE J, JIN H, YANG J, et al. Fine-grained recognition without part annotations[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015: 5546-5555.
[6] SUN M, YUAN Y, ZHOU F, et al. Multi-attention multi-class constraint for fine-grained image recognition[C]// Proceedings of the European Conference on Computer Vision (ECCV). Munich: Springer, 2018: 805-821.
[7] FU J, ZHENG H, MEI T. Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017: 4438-4446.
[8] CHENG Y, WANG D, ZHOU P, et al. A survey of model compression and acceleration for deep neural networks[EB/OL]. (2017-10-30)[2021-09-24]. https://arxiv.org/pdf/1710.09282v2.pdf.
[9] SAINATH T N, KINGSBURY B, SINDHWANI V, et al. Low-rank matrix factorization for deep neural network training with high-dimensional output targets[C]// 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver: IEEE, 2013: 6655-6659.
[10] HE Y, LIN J, LIU Z, et al. AMC: AutoML for model compression and acceleration on mobile devices[C]// Proceedings of the European Conference on Computer Vision (ECCV). Munich: ECCV, 2018: 784-80.
[11] LOUIZOS C, ULLRICH K, WELLING M. Bayesian compression for deep learning[C]// Advances in Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017: 3288-329.

[12] WEN W, WU C, WANG Y, et al. Learning structured sparsity in deep neural networks[C]// Advances in Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016: 2074-2082.
[13] LUO J H, WU J, LIN W. Thinet: a filter level pruning method for deep neural network compression[C]// Proceedings of the IEEE International Conference on Computer Vision. Venice: IEEE, 2017: 5058-506.
[14] KIM Y D, PARK E, YOO S, et al. Compression of deep convolutional neural networks for fast and low power mobile applications[EB/OL]. (2016-02-24)[2021-09-24]. https://arxiv.org/pdf/1511.06530.pdf.
[15] WEI X S, XIE C W, WU J, et al. Mask-CNN: localizing parts and selecting descriptors for fine-grained bird species categorization[J]. Pattern Recognition, 2018, 76: 704-714.
[16] CUI Y, SONG Y, SUN C, et al. Large scale fine-grained categorization and domain-specific transfer learning[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 4109-4118.
[17] XIONG F, GOU M, CAMPS O, et al. Person re-identification using kernel-based metric learning methods[C]// European Conference on Computer Vision. Zurich: ECCV, 2014: 1-16.
[18] DENTON E L, ZAREMBA W, BRUNA J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]// Advances in Neural Information Processing Systems. Montreal: MIT Press, 2014: 1269-1277.
[19] ZHU C Z, HAN S, MAO H Z, et al. Trained ternary quantization[EB/OL]. (2017-02-23)[2021-09-24]. https://arxiv.org/pdf/1612.0/064.pdf.
[20] HAN S, MAO H Z, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization andhuffman coding[EB/OL]. (2016-02-15)[2021-09-24]. https://arxiv.org/pdf/1510.00149.pdf.
[21] VANHOUCKE V, SENIOR A, MAO M Z. Improving the speed of neural networks on CPUs[C]// Deep Learning and Unsupervised Feature Learning. Granada: NIPS, 2011:1-8.
[22] GUPTA S, AGRAWAL A, GOPALAKRISHNAN K, et al. Deep learning with limited numerical precision[C]// International Conference on Machine Learning. Lille: JMLR, 2015: 1737-1746.
[23] PENG H, WU J, CHEN S, et al. Collaborative Channel Pruning for Deep Networks[C]// International Conference on Machine Learning. Long Beach: JMLR, 2019: 5113-5122.
[24] ZHUANG Z, TAN M, ZHUANG B, et al. Discrimination-aware channel pruning for deep neural networks[C]// Advances in Neural Information Processing Systems. Montreal: Curran Associates Inc., 2018: 875-886.
[25] ACHILLE A, ROVERE M, SOATTO S. Critical learning periods in deep networks[C]// International Conference on Learning Representations. New Orleans: ICLR, 2019.
[26] FRANKLE J, SCHWAB D J, MORCOS A S. The early phase of neural network training[EB/OL]. (2020-02-24)[2021-09-24]. https://arxiv.org/pdf/2002.10365v1.pdf.
[27] DUBEY A, GUPTA O, GUO P, et al. Pairwise confusion for fine-grained visual classification[C]// Proceedings of the European Conference on Computer Vision (ECCV). Munich: ECCV, 2018: 70-86.

Bi-Attention: acceleration method for fine-grained classification network toward edges

ZHONG Qiaoling, WANG Xiao, ZHANG Zhibin, LI Bing, CHENG Xueqi
(Key Laboratory of Web Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)
(School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049)

Abstract

A fine-grained classification task is an image classification task that recognizes objects with minor differences. Deep learning models have achieved great improvement in these fine-grained classification tasks. However, the existing fine-grained networks ensemble multiple models. It is hard to deploy these networks on the edges with limited resources, such as mobile phones and drones. In this paper, a method to accelerate model for fine-grained classification toward edges is proposed, called Bi-Attention. The efficient TensorSketch operation and share weights are used within the model. It can be achieved an accuracy of 91.6% and a better accuracy of 1.2% than the existing state-of-the-art models on Stanford Cars dataset. A structured-pruning training method is proposed to prune the unimportant scale factor in batch normalization (BN) through LASSO regularization. The experimental results show that it can be reduced the size of Bi-Attention model up to 1/4.

Key words: fine-grained classification, Attention, structured pruning, LASSO regularization, edge