·图形图像处理·　　文章编号：1000-3428(2022)05-0242-09　　文献标志码：A　　中图分类号：TP391.41

面向轻量级卷积网络的激活函数与压缩模型

徐增敏[1,3]，陈　凯[2]，郭威伟[1,4]，赵汝文[1]，蒋占四[5]

（1.桂林电子科技大学 数学与计算科学学院，广西 桂林 541004；
2.杭州海康威视数字技术股份有限公司，杭州 310052；3.桂林安维科技有限公司，广西 桂林 541010；
4.中国通信建设集团设计院有限公司第四分公司，郑州 450052；5.桂林电子科技大学 机电工程学院，广西 桂林 541004）

摘　要：因卷积神经网络参数膨胀，导致模型训练时占用大量的计算资源和存储资源，从而限制其在边缘终端上的应用。依据深度可分离卷积模型MobileNet V1的设计思路，结合自门控函数和ReLU函数的特点，构建一种改进的激活函数和压缩神经网络模型MobileNet-rhs。将ReLU函数和swish函数分别作为分段线性函数，设计激活函数ReLU-h-swish，通过优化卷积单元结构，解决模型训练过程中难以激活部分神经元的问题，以减少特征信息丢失。构建一种剔除卷积核的压缩模型，从模型深处自下而上剔除2^n个卷积核，减少逐点卷积的参数量。在CIFAR-10和CIFAR-100数据集上进行实验，结果表明，引入ReLU-h-swish函数构建MobileNet-rhs模型的Top-1分类准确率为80.38%。相比MobileNet-rhs模型，压缩后MobileNet-rhs模型的参数量减少17.9%，其Top-1分类准确率仅降低2.28个百分点。此外，利用Tensorflow将该模型部署在安卓平台上，实现图像分类相册的应用。

关键词：manifold of interest变换；深度可分离卷积；逐点卷积；自门控函数；Kotlin协程

开放科学（资源服务）标志码（OSID）：

中文引用格式：徐增敏，陈凯，郭威伟，等. 面向轻量级卷积网络的激活函数与压缩模型[J]. 计算机工程，2022，48(5)：242-250.

英文引用格式：XU Z M，CHEN K，GUO W W，et al. Activation function and compression model for lightweight convolutional network[J]. Computer Engineering，2022，48(5)：242-250.

Activation Function and Compression Model for Lightweight Convolutional Network

XU Zengmin[1,3]，CHEN Kai[2]，GUO Weiwei[1,4]，ZHAO Ruwen[1]，JIANG Zhansi[5]

（1.School of Mathematics and Computing Science，Guilin University of Electronic and Technology，Guilin，Guangxi 541004，China；
2.Hangzhou Hikvision Digital Technology Co.，Ltd.，Hangzhou，310052，China；
3.Guilin Anview Technology Co.，Ltd.，Guilin，Guangxi 541010，China；
4.The fourth branch of China Communications Construction Group Design Institute Co.，Ltd，Zhengzhou，450052，China；
5.School of Mechanical and Electrical Engineering，Guilin University of Electronic and Technology，Guilin，Guangxi 541004，China）

【Abstract】The abundance of computing and storage resources required in model training to relieve the parameter expansion of a deep convolution neural network，limiting a network's application on edge terminals.Based on the design idea of the depthwise separable convolution model MobileNet V1，this study proposes an improved activation function and compressed neural network model that combines the characteristics of the self-gating function and the ReLU function.Taking the ReLU function and swish function as piecewise linear functions，the activation function ReLU-h-swish is designed.By optimizing the convolution unit structure to reduce the loss of feature information，it is difficult to activate some neurons during the process of model training.A compression model with convolution kernels removed is constructed.To compress the model，2^n convolution kernels are removed from the depths of the model from bottom to top to reduce the number of parameters of point-to-point convolution.The experimental results on the CIFAR-10 and CIFAR-100 datasets show that the Top-1 classification accuracy of the MobileNet-rhs model constructed by introducing the

基金项目：国家自然科学基金“视频侦查中基于深度学习的人体行为识别技术研究”(61862015)；广西重点研发计划项目“面向涉密场所的视频人体行为分析系统研发及应用”(AB17195025)；广西高校中青年教师科研基础能力提升项目“基于手机指纹识别身份认证系统研究”(2019KY0253)。

作者简介：徐增敏(1981—)，男，副教授、博士，主研方向为人工智能、计算机视觉、人体行为分析；陈　凯(通信作者)，工程师；郭威伟，学士；赵汝文，讲师、硕士；蒋占四，教授、博士生导师。

收稿日期：2021-05-06　　**修回日期：**2021-07-04　　**E-mail：**chenkai34@hikvision.com

ReLU-h-swish function is 80.38%. Compared to the MobileNet-rhs model, the parameters of the compressed MobileNet-rhs model are reduced by 17.9%, and the Top-1 classification accuracy is reduced by only 2.28 percentage points. In addition, Tensorflow is used to deploy the model on the Android platform, which realizes the application of image classification album.

【**Key words**】transformation of manifold of interest; depthwise seperable convolution; pointwise convolution; self-gating function; Kotlin coroutine

DOI: 10. 19678/j. issn. 1000-3428. 0061550

0　概述

卷积神经网络的权重共享网络是用于识别二维图像的感知器。研究人员不断增加网络层数，以获得更优的识别性能，如从最初仅7层的AlexNet[1]演化到16层的VGG[2]、22层的GoogleNet[3]和152层的ResNet[4]，且模型训练时占用大量的硬件资源。

在移动互联网时代，人们对移动设备的使用频率已经超过桌面设备。因此，仅利用移动端边缘设备硬件资源就能快速运行的深度学习模型应运而生，如SqueezeNet[5]、MobileNet[6]、ShuffleNet[7]等轻量级卷积神经网络。

传统的云端部署模型通过移动终端与云端的交互，满足移动应用场景的需求。由于数据在传输过程中受网络延迟、带宽等因素的影响，因此该技术的发展受到限制。随着移动设备计算能力和存储能力的提升，MobileNet等轻量级卷积神经网络应运而生，模型可以直接部署在移动设备或嵌入式设备上，并且具有较优的精度。微小的体积和高效的运行速度使得模型在边缘终端中的应用更易实现。因此，性能较优的边缘终端轻量级神经网络成为研究热点。

2017年，国内的旷视科技推出一个高效运算且网络参数较小的轻量级卷积神经网络ShuffleNet[7]，并提出通道混洗和点态组卷积。通道混洗技术使得特征能够跨通道进行学习；点态组卷积技术能够加快网络运算，在提高计算效率的同时使网络更加轻量化。2018年，旷视科技推出ShuffleNet的改进版本ShuffleNet V2[8]，利用新标准衡量目标检测模型的运行速度，在相同复杂度的情况下提升精度并加快运算速度。文献[9]构建一种轻量型卷积神经网络，通过对所有的疑似目标切片进行精确分类，以确定目标种类，从而识别空中红外目标。文献[10]利用轻量化的YOLO卷积神经网络对视频首帧进行目标识别，通过结合KCF目标跟踪算法与感知哈希算法对完成识别的目标进行跟踪与矫正。优化后的算法能够对复杂目标进行实时识别，具有较强的自适应能力。文献[11]提出一种基于轻量化深度网络的舰船目标识别方法，通过将深度可分离卷积和多尺度语义信息相融合，并对其进行改进，从而完成目标识别。在自建目标数据集保证Top-5准确率达到93.5%的情况下，该方法降低了模型参数量与计算量。文献[12]提出一种基于深度分离卷积、分组卷积等轻量化的高效卷积方式，设计用于图像特征提取的不变分辨率卷积模块和下采样模块，以此构建深度主干网络，并对网络进行剪枝。

文献[13]提出一种面向资源受限平台应用的轻量化特征提取结构DResNet，该方法的综合性能均优于MobileNet、ShuffleNet、MobileNet V2[14]等轻量化方法。文献[15]提出基于注意力机制的轻量化算法，该算法对网络进行剪枝后，能够有效地减少冗余参数。文献[16]提出一种模型规模小、计算复杂度低的预训练SqueezeNet模型。文献[17]提出一种针对受限环境的轻量级简化密码算法LAES，在处理时间和随机性方面具有一定优势。文献[18]提出一种Levy飞行优化算法，以优化DCNN的网络结构，LFOA算法减少了DCNN隐含层的神经元数目和音频输入特征的数目，提高了分类精度。该模型提高了6种故障分类的准确率，为车辆健康状况识别构建一种新的研究模型。

面向移动终端与嵌入式设备的模式识别，以及轻量级卷积神经网络的结构调优是目前的研究热点。本文提出一种改进的激活函数与压缩模型。基于manifold of interest分析激活函数，并结合ReLU函数和swish函数的优点，设计激活函数ReLU-h-swish，以解决网络训练过程中部分神经元无法被激活的问题，同时提出一种剔除卷积核的模型压缩方法，减少参数量，实现模型压缩的目的。

1　轻量级卷积神经网络研究

1.1　MobileNet V1模型

2017年，针对移动终端和嵌入式视觉应用，谷歌推出卷积神经网络模型MobileNet V1[6]。该模型利用深度可分离卷积构造轻量级权重深度神经网络，其结构呈流线型，包含2个能够权衡准确率和延迟的全局超参数。该超参数使得模型构造器可以根据特定问题选择合适大小的模型。MobileNet的主要原理是利用深度可分离卷积代替传统卷积，即利用深度卷积结合逐点卷积的方式，能够有效保证信息流通畅，并且减少网络权值参数量，达到模型压缩的效果。

在以组为单位的传统卷积神经网络中，采用group convolution卷积方式，即一组卷积核负责一组特征图像。基于group原理，MobileNet将Depth-Wise Convolution看作是特殊的group convolution操作，一个卷积核作用于一组特征图像和一个通道，即将每个通道看作一个组，每个卷积核仅在特定的通道上进行卷积，使得卷积

计算仅指向特定组的输入，从而减少卷积的计算量。这样的设计适用于移动端计算资源不足的情况，提高移动终端正向的计算效率。

1.2 激活函数的改进

1.2.1 ReLU函数分析

在卷积神经网络训练时，ReLU函数会使实际数据进入网络时，无法激活部分神经元，即“神经元静默”现象，同时使用ReLU函数变换可能存在2个问题：1）如果当前激活空间内有较完整的manifold of interest时，经过ReLU函数变换后可能会使激活空间坍塌，导致特征信息丢失；2）如果经过ReLU函数变换后输出非零值，那么在输入与输出之间进行线性变换，即输入空间的一部分映射到全维输出，因此ReLU函数具有线性分类器的作用。

经过激活层后的张量被称为manifold of interest。卷积神经网络将ReLU作为激活函数，就不可避免地损失该通道内的部分信息。输入数据映射的维度越高，还原特征空间分布的效果越好，从而保留更多的原始信息。manifold of interest的ReLU变换示例如图1所示。

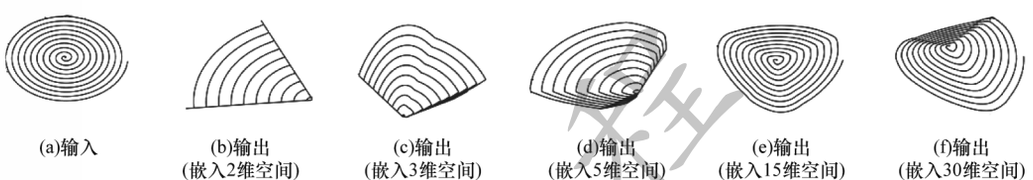


图1 manifold of interest的ReLU变换示例

Fig.1 ReLU transformations examples of manifold of interest

根据文献[14]，假设有一个2维的输入数据，其manifold of interest变换如图1(a)所示。随机矩阵$\boldsymbol{T}$将高维数据嵌入到n维空间中，并与上一个ReLU进行变换，之后再将$\boldsymbol{T}^{-1}$投影到2维平面上。该方法能够获得嵌入到n维空间manifold of interest的图形，便于与输入数据的manifold of interest进行对比。

当n=2时，将manifold of interest嵌入到2维空间中。经过上述变换，可以得到图1(b)。从图1(b)可以看出，2维空间中的manifold of interest发生了形变，投影到二维平面上的图像中心点坍塌，从而导致信息丢失。

当n=3和n=5时，将高维数据分别嵌入到3维和5维空间，经过ReLU变换后按上述方法投影至2维平面上，得到图1(c)和图1(d)，可以看出，丢失的信息逐渐恢复，但是存在信息重叠的问题。

当n=15和n=30时，按同样方法得到图1(e)和图1(f)，可以看出，相比在2维空间和3维空间等低维空间，在高维空间使用ReLU函数进行变换后恢复了较多的特征信息。

因此，在网络训练时使用ReLU激活函数可能会使激活空间坍塌，造成特征信息丢失。在改进MobileNet网络结构时，通过对ReLU激活函数进行改进，或者使用其他性能更优的激活函数替换ReLU激活函数，以提高模型的精度。

1.2.2 RReLU函数分析

传统ReLU函数将负半轴的响应设置为0，部分神经元在训练中“静默”。如果在ReLU函数的基础上引入一个额外的可训练参数，用于控制负半轴的响应，那么其改进函数RReLU(Randomized_Leaky_ReLU)[19]如式(1)所示：

$$\mathrm{RReLU}(x)=\begin{cases}x, x>0\\ a_i x, x\leqslant 0\end{cases} \tag{1}$$

其中：x为非线性激活函数f在第i个通道的输入；a_i控制负值部分斜率的系数，即需要在训练过程中指定的参数，a_i中的下标i为它允许非线性激活函数在不同通道上的变化，该变化可以相同也可以不同。

RReLU函数需要在训练过程中确定引入的参数，Leaky_ReLU[20]函数如式(2)所示：

$$\mathrm{Leaky\_ReLU}(x)=\begin{cases}x, x>0\\ \dfrac{x}{a}, x<0\end{cases} \tag{2}$$

当式(1)中的a_i是一个可学习的参数时，RReLU函数如式(3)所示：

$$f(y_i)=\max(0,y_i)+a_i\min(0,y_i) \tag{3}$$

当$a_i=0$时，式(1)为传统的ReLU函数；当a_i为较小的固定值时（例如$a_i=0.01$），RReLU函数变为Leaky_ReLU函数。由于Leaky_ReLU的导数不为0，因此负值输入都有一个梯度，减少“神经元静默”现象的发生。相比整个网络的参数量，RReLU激活函数只增加了较少的参数量。

1.2.3 自门控函数分析

自门控函数[21]是谷歌大脑推出的激活函数，如式(4)和式(5)所示：

$$\mathrm{swish}(x)=x\times\sigma(\beta x) \tag{4}$$

$$\sigma(z)=(1+\mathrm{e}^{-z})^{-1} \tag{5}$$

自门控函数图像如图2所示。当$\beta=1$时，自门控函数等价于Sigmoid加权线性单元，且适用于强化学习。当$\beta=0$时，自门控函数变换成为比例线性函数，$f(x)=x/2$。当$\beta\to\infty$时，自门控函数变换为ReLU函数，说明可以将自门控函数swish看作是一个平滑函数。该函数在线性函数和ReLU函数之间进行线性插值。如果将β设置为可训练的参数，那么插值度可由模型自身进行控制。

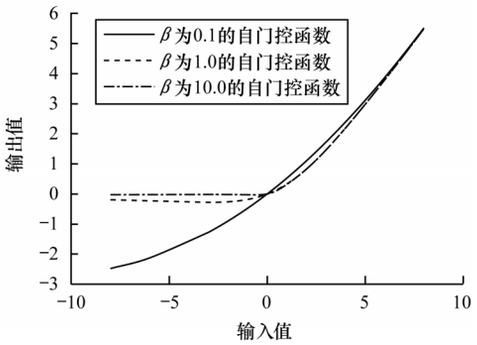


图2　自门控函数图像

Fig.2　Self-gating function images

swish函数本质上是Sigmoid函数的变形，存在收敛缓慢的问题。尽管这种非线性函数能够有效提高精度，但是对于嵌入式环境，其计算成本较大。因此，本文对swish函数进行改进，使其适用于轻量级网络，采用ReLU函数替换Sigmoid函数，如式(6)所示：

$$\sigma(x)=\frac{\mathrm{ReLU6}(x+3)}{6} \tag{6}$$

ReLU6函数是最大输出限制为6的ReLU函数，在移动设备float16低精度时，其具有较优的数值分辨率。如果对ReLU函数激活范围不加以限制，其输出范围为0至正无穷。当激活函数的输出值较大时，float16难以精确地描述大范围的数值，从而导致精度降低。改进的swish自门控函数称为hard-swish函数[22]，如式(7)所示：

$$\text{hard-swish}(x)=x\frac{\mathrm{ReLU6}(x+3)}{6} \tag{7}$$

hard-swish函数的图像与原swish函数图像相似。相比swish函数，在移动设备上部署带有hard-swish函数的模型具有更多的优势。网络层数越深，分辨率越低，每层的激活内存就会减半，应用非线性函数的成本降低[22]。

1.2.4　ReLU-h-swish函数

由于修正单元在x神经元输入的半个区间内为线性函数，在另一半区间为非线性函数，因此ReLU相当于x的一个分段线性函数[23]。研究人员认为修正神经元具有脑神经的稀疏激活性，尽管其具有严格的非线性和在零处的不可微性，但是相比双曲正切网络，其具有相等或更优的性能，并接近脑神经接收信号的激活模型[24]。

本文对ReLU、RReLU和swish函数进行分析，通过对ReLU函数引入可训练参数，以解决神经元静默的问题。对于swish函数，本文通过将函数中的Sigmoid函数替换为ReLU6，以减少在移动设备上的计算成本。因此，本文结合ReLU函数与hard-swish函数的优点，提出全新的激活函数ReLU-h-swish，如式(8)所示：

$$\text{ReLU-h-swish}(x)=\begin{cases}x, x>0\\ x\dfrac{\mathrm{ReLU6}(x+3)}{6}, x\leqslant 0\end{cases} \tag{8}$$

其中：当$x>0$时，因纯线性输入值x导数固定为1，其收敛速度比Sigmoid、Tanh函数快，从而解决在训练过程中swish函数收敛速度过慢的问题；当$x\leqslant 0$时，为缓解ReLU出现神经元静默，以及激活值过大影响轻量级卷积网络模型体积的问题，采用hard-swish函数抑制激活值的线性增长。

虽然hard-swish函数具有较少的计算量，但是其消除了近似Sigmoid函数不同可能造成的数值精度损失。而且，当卷积网络层数加深时，应用非线性的计算开销会逐渐下降，每层的激活内存通常会在分辨率下降时减半。因此通过减少内存访问次数，以大幅降低延迟开销，解决swish函数在反向传播时求解误差梯度涉及大量参数计算的问题，从而达到加快收敛速度的目的[22]。

swish函数、hard-swish函数和ReLU-h-swish函数图像对比如图3所示，本文提出的ReLU-h-swish激活函数图像与swish函数和hard-swish函数十分接近。当接收到正值输入时，ReLU-h-swish激活函数相比于swish函数和hard-swish函数能更快速地收敛；当接收到负值输入时，函数图像为非单调凸函数，能够以较低的计算成本应用在移动设备上。

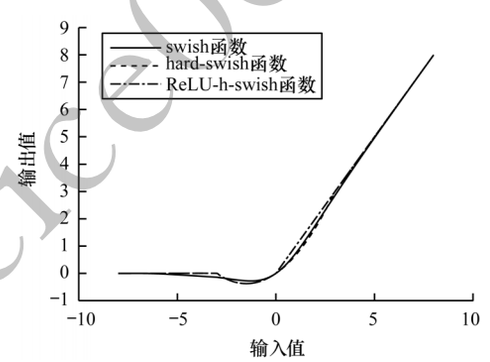


图3　3个函数图像对比

Fig.3　Comparison of 3 function images

1.3　卷积信息完整性分析

一个标准卷积的操作(D_K,D_K,M,N)表示为：卷积核的宽高为(D_K,D_K)；M为输入的通道数；N为输出的通道数，即卷积核个数。卷积前输入特征尺寸为(D_F,D_F,M)，卷积后输出特征的尺寸为(D_F,D_F,N)。在卷积过程中，每个卷积核对图像区域进行$D_F\times D_F$次扫描，因此标准卷积参数量为$(D_K\times D_K\times M)\times N$，计算量为$(D_K\times D_K\times M)\times N\times D_F\times D_F$。

根据MobileNet V1设计的网络结构，深度可分离卷积将标准卷积操作分解为深度卷积和逐点卷积[6]。其中深度卷积负责滤波，尺寸为$(D_K,D_K,1)$，共M个输入通道，作用在输入的每个通道上，逐点卷积负责转换通道，尺寸为$(1,1,M)$，共N个输出通道，作用在深度卷积的输出特征图上。MobileNet的

D_K=3,且每个卷积层的输出通道N与下一个卷积层的输入通道数M相等。

因此,MobileNet V1中每个深度卷积层的参数量为(D_K×D_K×1)×M,每个逐点卷积层的参数量为(1×1×M)×N。每个深度可分离卷积层的参数量是标准卷积层的$\frac{(D_K\times D_K\times 1)\times M+(1\times 1\times M)\times N}{(D_K\times D_K\times M)\times N}=\frac{1}{N}+\frac{1}{D_K^2}$。

假设M=N,深度卷积为g=M=N的分组卷积,没有直接将g组结果进行拼接,因此深度卷积参数量是标准卷积的1/N。然而逐点卷积采用1×1卷积将g组结果进行拼接,逐点卷积参数量是标准卷积的$1/D_K^2$。

MobileNet V1中不同卷积的参数量所占模型总参数量的比例分别为[6]:1×1逐点卷积占74.59%,3×3深度卷积占1.06%,3×3标准卷积占0.02%,全连接层占24.33%。模型95%的计算时间用于1×1逐点卷积。

1.4 MobileNet V1模型的压缩方法

卷积神经网络模型的参数量越少,所需的存储空间就越小,计算量也会减少。随着参数量的减少,模型的精度也会降低。因此,轻量级卷积神经网络的模型压缩需要在精度与参数量之间进行权衡。

MobileNet V1共有28层卷积。第1层标准卷积是3×3卷积,包含的参数比较少,因此去除这层对于整体参数量的影响不大,不对第1层卷积进行压缩。深度卷积比标准卷积的计算量少,并且深度卷积具有相同的输入通道与输出通道,因此移除深度卷积核对整体参数量影响较小。本文将模型压缩的目标放在了逐点卷积上,相应的压缩流程分为4个步骤:1)训练模型原型,记录参数量;2)从一层逐点卷积开始,核对卷积核大小与输入尺寸;3)删去2^n(n=1,2,…,n)个卷积核,重新训练模型并记录参数量,对比参数减少量;4)重复步骤2,直到模型精度与参数量保持平衡。

综上所述,本文所提模型压缩方法仅剔除部分逐点卷积的卷积核个数。如果在模型压缩过程中删除的逐点卷积核个数为$\tilde{N}$,那么模型有效信息的完整性可表示为:(总参数量 − 1×1×M×$\tilde{N}$)/总参数量。随着被删去的逐点卷积核个数增加,模型的精度将逐渐下降。

在删除不必要卷积核的过程中,下一层卷积层的输入应与上一层的输出相对应。由于MobileNet V1轻量网络结构设计的原因,在深度卷积之后会再接一个批量归一化层,因此在保持卷积层输出与下一层卷积层输入对应的同时,还需注意批量归一化层中对应的参数。

在训练模型的权重参数过程中,有些卷积核参数值趋近于0,尽管这些参数对于模型的精度影响较小,但是需要较多的时间学习。因为不必要的网络参数量会增加模型计算量和存储空间,所以本文尝试去除数值较小的卷积核,以达到模型压缩的目的。

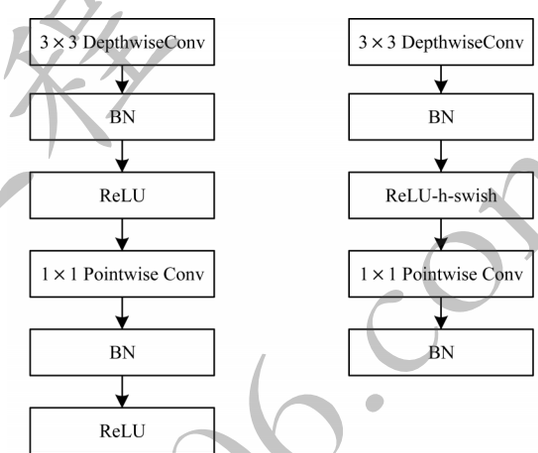
2 实验结果与分析

本节主要对比模型压缩前后的效果,以及所提激活函数在模型训练过程中的性能。

2.1 实验设置

本文实验分为2个部分:根据MobileNet V1模型结构调整后的实验和根据改进MobileNet V1模型成型后的模型压缩实验。

针对MobileNet V1网络结构的改进,根据1.2.1节ReLU函数存在信息丢失的缺点,为避免信息丢失情况的发生,本文对MobileNet V1的卷积单元进行改进,如图4所示。



(a)原始MobileNet V1卷积单元 (b)改进后MobileNet V1卷积单元

图4 MobileNet V1卷积单元结构

Fig.4 Structure of MobileNet V1 convolution unit

从图4(b)可以看出,改进后的卷积单元将原结构中最后一个影响精度的ReLU函数层去掉,并且将深度卷积之后的ReLU函数改为本文提出的全新激活函数ReLU-h-swish函数。本文将改进后的MobileNet V1模型称为MobileNet-rhs,以下实验与应用设计都用该名称来区别MobileNet V1。

2.2 基于改进算法的实验分析

为验证ReLU-h-swish算法的有效性,本文分别对swish、hard-swish、RReLU和ReLU-h-swish算法的性能进行对比。

2.2.1 CIFAR-10数据集

CIFAR-10数据集共包含60 000张彩色三通道图片,含有10个分类,每个分类包含6 000张图片。该数据集含有50 000张适用于神经网络训练的图片,10 000张适用于神经网络训练过程中测试验证的图片,两者构成测试集。

基于MobileNet V1模型,本文分别引入swish函数、hard-swish函数、RReLU函数和ReLU-h-swish激活函数构建MobileNet-swish、MobileNet-hswish、MobileNet-optirelu和MobileNet-rhs模型。在CIFAR-10数据集上,MobileNet V1模型与其他模型的分类准确率对比如表1所示,表中加粗数字为最优数据。

表1　在CIFAR-10数据集上不同模型的分类准确率对比

Table 1　Classification accuracy comparison among different models on CIFAR-10 dataset　%

模型	分类准确率
MobileNet-swish模型	76.38
MobileNet-hswish模型	77.69
MobileNet-optirelu模型	78.20
MobileNet-rhs模型	**80.38**

本文采用数据打点的方式对表1中的模型的收敛性进行绘图分析。在每个epoch内执行迭代第100的整数倍时，从测试批的同个标签下的测试图片中，随机抽取一张图片进行损失值和精度值统计，然后把统计结果输出到文件中，最后将统计出来的数据打点绘制成折线图的形式。在测试集上4种模型的分类准确率对比如图5所示（彩色效果见《计算机工程》官网HTML版）。

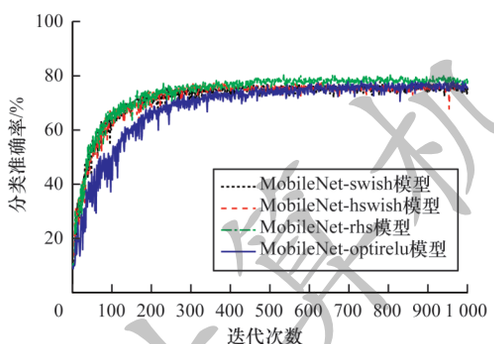


图5　在测试集上不同模型的分类准确率对比

Fig.5　Classification accuracy comparison among different models on testing set

从图5可以看出，MobileNet-rhs模型具有最优的分类准确率，其他3种激活函数的分类准确率较相近。hard-swish函数相对于swish函数适当降低了两种情况下的计算成本，且没有影响原swish函数的性能，因此hard-swish函数更适合移动终端平台。4种模型的损失值对比如图6所示（彩色效果见《计算机工程》官网HTML版）。

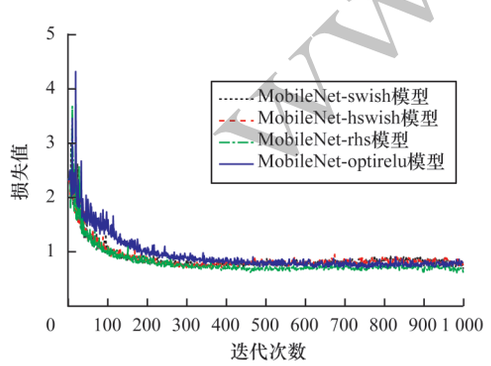


图6　在测试集上不同模型的损失值对比

Fig.6　Loss values comparison among different models on testing set

从图6可以看出，引入ReLU-h-swish函数构建的MobileNet-rhs模型的损失值最低，其次是MobileNet-optirelu模型，但是其收敛速度较慢。损失值最高的是引入swish函数和hard-swish函数分别构建的MobileNet-swish和MobileNet-hswish模型。图5、图6说明ReLU-h-swish激活函数均获得最高分类准确率与最低的损失值，在训练过程中既没有较大的波动又能保持较快的收敛速度。ReLU-h-swish激活函数缓解了ReLU函数神经元静默的问题，相比swish函数，其计算成本较低，更适合移动终端平台。

2.2.2　CIFAR-100数据集

CIFAR-100有100个类别，并且每个类别包含600张图像，每类别各有500张训练图像和100张测试图像。如无特殊说明，本文以下实验均在CIFAR-100上进行，采用不同的训练策略，将epoch数由10上调至200，并将每个epoch中输入图像的数量由16上调至128；学习率设置衰减策略，由CIFAR-10实验中的0.001调整为0.1，并在第60、120、160个epoch时衰减至0.02、0.004、0.000 8。因此，在MobileNet V1上引入6种激活函数的Top-1、Top-5测试分类准确率如表2所示。

表2　在MobileNetV1模型上引入不同激活函数的分类准确率对比

Table 2　Classification accuracy comparison of MobileNetV1 model with different activation functions　%

模型	Top-1分类准确率	Top-5分类准确率
MobileNet-relu	64.98	87.97
MobileNet-leakyrelu	68.46	88.73
MobileNet-optirelu	66.54	87.04
MobileNet-swish	68.28	90.03
MobileNet-hswish	67.61	89.88
MobileNet-rhs	70.10	88.62

从表2可以看出，改进后的模型分类准确率较原模型都有提升，而在CIFAR-100数据集上本文提出的MobileNet-rhs模型相比MobileNet-relu模型的Top-1分类准确率上提升了约5个百分点，相对其他模型也均有提升。

除MobileNet V1模型以外，本文还对其他轻量级卷积神经网络ShuffleNet[7]、ShuffleNet V2[8]进行实验，并将这2个神经网络中使用的ReLU函数直接替换为ReLU-h-swish激活函数，训练超参数与原文的超参数设置保持一致，最终结果得到小幅度提升，如表3所示。

表3　在ShuffleNet模型上引入不同激活函数的分类准确率对比

Table 3　Classification accuracy comparison of ShuffleNet models with different activation functions　%

模型	Top-1分类准确率	Top-5分类准确率
ShuffleNet	69.84	91.68
ShuffleNet-rhs	70.23	91.92
ShuffleNet V2	69.44	91.49
ShuffleNet V2-rhs	69.85	91.57

2.3 模型压缩实验结果分析

MobileNet V1设定宽度乘数因子α和分辨率乘数因子β 2个超参数。宽度乘数因子具有降低激活空间维度的作用,分辨率乘数因子可以改变输入数据的分辨率,也能减少参数。在压缩MobileNet V1模型之前,本文首先测试模型的精度,以此为基准衡量MobileNet V1模型的压缩效果。MobileNet-rhs模型的性能指标如表4所示。

表4 MobileNet-rhs模型的性能指标

Table 4 Performance indexs of MobileNet-rhs model

模型	Top-1分类准确率/%	Top-5分类准确率/%	参数量
MobileNet-rhs	70.10	88.62	3 315 428

根据1.4节的压缩流程,本节超参数的设置与CIFAR-100实验保持一致。MobileNet V1中的逐点卷积结构如表5所示。其中,s1表示步长为1,即stride=1,层表示第几层逐点卷积[6]。为快速得到较理想的压缩结果,本文从表5中第13层逐点卷积开始执行压缩流程,即卷积核大小为1×1×1 024×1 024的逐点卷积。

表5 MobileNetV1模型的逐点卷积结构

Table 5 Pointwise convolution structure of MobileNetV1 model

卷积类型/步长/层	卷积核大小	输入尺寸
Conv/s1/1	1×1×32×64	112×112×32
Conv/s1/2	1×1×64×128	56×56×64
Conv/s1/3	1×1×128×128	56×56×128
Conv/s1/4	1×1×128×256	28×28×128
Conv/s1/5	1×1×256×256	28×28×256
Conv/s1/6	1×1×256×512	14×14×256
Conv/s1/7	1×1×512×512	14×14×512
Conv/s1/8	1×1×512×512	14×14×512
Conv/s1/9	1×1×512×512	14×14×512
Conv/s1/10	1×1×512×512	14×14×512
Conv/s1/11	1×1×512×512	14×14×512
Conv/s1/12	1×1×512×1 024	7×7×512
Conv/s1/13	1×1×1 024×1 024	7×7×1 024

经过一轮遍历搜索之后发现,当n=8时,即第13层逐点卷积剔除256个卷积核后满足压缩流程的要求,压缩第13层后MobileNetV1模型的逐点卷积结构信息如表6所示,压缩第13层后的MobileNet-rhs性能如表7所示。

表6 压缩第13层后MoblieNetV1模型的逐点卷积结构

Table 6 Pointwise convolution structure of MobileNetV1 model after 13th layer's compression

卷积类型/步长/层	卷积核大小	输入尺寸
Conv/s1/13	1×1×1 024×768	7×7×1 024

表7 压缩第13层后的MobileNet-rhs模型性能指标

Table 7 Performance indexs of MobileNet-rhs model after 13th layer's compression

模型	Top-1分类准确率/%	Top-5分类准确率/%	参数量
MobileNet-rhs	69.19	88.24	3 026 916

相比MobileNet-rhs模型的原始参数,压缩第13层后MobileNet-rhs模型的参数量减少了288 512,即减少约8.7%的参数量。在模型精度方面,Top-1和Top-5的精度减少0.91和0.38个百分点。因此,压缩模型具有高精度。

本文对第11层和第12层逐点卷积进行压缩,当n=6时,即第11层逐点卷积剔除64个卷积核,当n=7时,即第12层逐点卷积剔除128个卷积核之后满足压缩流程,压缩后MobileNetV1模型的逐点卷积结构如表8所示。

表8 压缩第11和12层后MobileNetV1模型的逐点卷积结构

Table 8 Pointwise convolution structure of MobileNetV1 model after 11th, 12th layer's compression

卷积类型/步长/层	卷积核大小	输入尺寸
Conv/s1/11	1×1×512×448	14×14×512
Conv/s1/12	1×1×448×896	7×7×448
Conv/s1/13	1×1×896×768	7×7×896

因剔除了第12层逐点卷积核的数量,导致第2层输出通道减少,因此第13层的输入通道数也要相应减少至896,之后压缩实验卷积核大小变化依此类推。压缩第11和12层后的MobileNet-rhs模型Top-1和Top-5的分类准确率分别为68.89%和87.86%,参数量为2 770 276。相比原模型MobileNet-rhs的性能指标,压缩第11、12层后模型的参数量减少了545 152,即减少了约16.4%的参数量,Top-1和Top-5分类准确度分别减少1.21和0.76个百分点。

最终停止在第9层逐点卷积上,再向上剔除卷积核之后的效果可忽略不计。若再从第13层开始进一步剔除卷积核会导致模型精度大幅降低,因此压缩流程结束,最终压缩后MobileNetV1模型的逐点卷积压缩结构如表9所示。

表9 最终压缩后MobileNetV1模型的逐点卷积结构

Table 9 Pointwise convolution structure of final compressed MobileNetV1 model

卷积类型/步长/层	卷积核大小	输入尺寸
Conv/s1/9	1×1×512×496	14×14×512
Conv/s1/10	1×1×496×480	14×14×496
Conv/s1/11	1×1×480×448	14×14×480
Conv/s1/12	1×1×448×896	7×7×448
Conv/s1/13	1×1×896×768	7×7×896

当压缩进行至第9层逐点卷积时,本文训练的模型精度和参数量减少比例相对理想,训练模型的超参

数基于2.2.2节未改变。相比原模型MobileNet-rhs，压缩后的MobileNet-rhs模型的参数量减少了592 416，即减少17.9%的参数量，如表10所示。

表10　压缩结束后的MobileNet-rhs性能指标

Table 10　Performance index of MobileNet-rhs after final compression

模型	分类准确率/%		参数量
	Top-1	Top-5	
压缩MobileNet-rhs	67.82	87.23	2 723 012
MobileNet-rhs	70.10	88.62	3 315 428

2.4　基于Android的轻量级卷积网络应用

根据上述理论和实验分析，在Android平台上，本文将以图像分类为基础进行app开发，实现基于图像分类的手机相册图片自动归类。

本文app应用借助Tensorflow[25]平台实现移动终端的模型部署，整个项目使用多模块开发，其中除了Tensorflow核心功能实现的模块，还有app的核心功能实现的模块。开发环境操作系统为macOS Catalina 10.15.3、集成开发环境为Android Studio 3.6.1、语言为Kotlin 1.3+、深度学习平台为Tensorflow 2.1.0。

本文采用Kotlin协程处理数据加载，与线程的区别在于，它是运行在单线程中的并发程序，省去了传统多线程并发机制中线程切换时带来的线程上下文切换、线程状态切换、线程初始化时的性能损耗，能够大幅提高并发性能。

在本实例协程主要调用Tensorflow模块函数对获取到的图片进行分类，处理完成后将结果传递搭配LiveData中，UI层刷新界面，最终效果如图7所示。

图7　图像分类相册应用

Fig.7　Application of image classifcation album

3　结束语

本文提出一种改进的激活函数和压缩神经网络模型。以MobileNet V1作为骨干网络，结合ReLU函数和swish函数的优点，设计激活函数ReLU-h-swish，通过优化卷积单元结构，以减少特征信息丢失。为减少轻量级卷积网络的计算开销，采用剔除卷积核的方法对模型进行压缩，从而减少逐点卷积的参数量。在此基础上，将训练好的模型部署到安卓平台，实现图像分类相册的应用。在CIFAR-10和CIFAR-100数据集上的实验结果表明，相比swish、hard-swish等函数，引入ReLU-h-swish函数构建MobileNet-rhs模型的Top-1分类准确率为80.38%，相比MobileNet-rhs模型，压缩后MobileNet-rhs模型的参数量减少了17.9%，其Top-1分类准确度仅减少2.28个百分点。后续将结合生物神经学，构建符合类脑神经元的信号稀疏激活模型，并通过结构重参数化技术研究模型推理加速的方法，进一步提高轻量级卷积神经网络的性能。

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]. [2021-04-03]. https://arxiv.org/pdf/1409.1556.pdf.

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

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

[5] IANDOLA F N, HAN S, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size[EB/OL]. [2021-04-03]. https://arxiv.org/pdf/1602.07360.pdf.

[6] HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[EB/OL]. [2021-04-03]. https://arxiv.org/pdf/1704.04861.pdf.

[7] ZHANG X Y, ZHOU X Y, LIN M X, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. Washington D.C., USA: IEEE Press, 2018: 6848-6856.

[8] FERRARI V, HEBERT M, SMINCHISESCU C, et al. Unsupervised class-specific deblurring[C]//Proceedings of European Conference on Computer Vision. Berlin, Germany: Spring, 2018: 258-374.

[9] 崔婷婷，唐林波，衡勇，等. 基于轻量化卷积神经网络的空中红外目标识别[C]//第十二届全国信号和智能信息处理与应用学术会议论文集. 北京，中国：中国学术期刊(光盘版)电子杂志社，2018：5.
CUI T T, TANG L B, HENG Y, et al. Airborne infrared target recognition based on lightweight convolutional neural network[C]//Proceedings of the 12th National Conference on Signal and Intelligent Information Processing and Application. Beijing, China: China Academic Journal (CD

version) Electronic Magazine, 2018: 5. (in Chinese)
[10] 曹昭睿,白帆,刘凤丽,等. 基于轻量化神经网络的目标识别跟踪算法研究[J]. 弹箭与制导学报,2020,40(1):19-23.
CAO Z R, BAI F, LIU F L, et al. Design of target recognizing and tracking algorithm based on tiny convolution neural network[J]. Journal of Projectiles, Rockets, Missiles and Guidance, 2020, 40(1): 19-23. (in Chinese)
[11] 刘俊,姜涛,徐小康,等. 基于轻量化深度网络的舰船目标识别技术研究[J]. 无线电工程,2019,49(12):1025-1030.
LIU J, JIANG T, XU X K, et al. Research on ship target recognition technology based on lightweight deep network[J]. Radio Engineering, 2019, 49(12): 1025-1030. (in Chinese)
[12] 付佐毅,周世杰,李顶根. 轻量级目标识别深度神经网络及其应用[J]. 计算机工程与应用,2020,56(18):131-136.
FU Z Y, ZHOU S J, LI D G. Lightweight target recognition deep neural network and its application[J]. Computer Engineering and Applications, 2020, 56(18): 131-136. (in Chinese)
[13] 李亚辉. 面向舰船目标识别应用的关键技术研究[D]. 杭州:杭州电子科技大学,2019.
LI Y H. Research on key technologies for ship target recognition application[D]. Hangzhou: Hangzhou Dianzi University, 2019. (in Chinese)
[14] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2018: 4510-4520.
[15] 徐梦婕. 轻量化地标感知算法及认知地图中的应用[D]. 成都:电子科技大学,2020.
XU M J. Lightweight landmark perception algorithm and its application in cognitive MAP[D]. Chengdu: University of Electronic Science and Technology of China, 2020. (in Chinese)
[16] YANG Y T, YANG R Z, PAN L H, et al. A lightweight deep learning algorithm for inspection of laser welding defects on safety vent of power battery[J]. Computers in Industry, 2020, 123: 103306.
[17] QASAIMEH M, AL-QASSAS R S, MOHAMMAD F, et al. A novel simplified AES algorithm for lightweight real-time applications: testing and discussion[J]. Recent Advances in Computer Science and Communications, 2020, 13(3): 435-445.
[18] RAJAKUMAR M P, RAMYA J, MAHESWARI B U. Health monitoring and fault prediction using a lightweight deep convolutional neural network optimized by Levy flight optimization algorithm[J]. Neural Computing and Applications, 2021, 33(19): 12513-12534.
[19] HE K M, ZHANG X Y, REN S Q, et al. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification[C]//Proceedings of IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2015: 1026-1034.
[20] LI C L, RAVANBAKHSH S, POCZOS B. Annealing Gaussian into ReLU: a new sampling strategy for leaky-ReLU RBM[EB/OL]. [2021-04-03]. https://arxiv.org/pdf/1611.03879.pdf.
[21] RAMACHANDRAN P, ZOPH B, LE Q V. Searching for activation functions[EB/OL]. [2021-04-03]. https://arxiv.org/pdf/1710.05941.pdf.
[22] HOWARD A, SANDLER M, CHEN B, et al. Searching for MobileNetV3[C]//Proceedings of IEEE/CVF International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2019: 1314-1324.
[23] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. Cambridge, USA: MIT Press, 2016.
[24] GLOROT X, BORDES A, BENGIO Y. Deep sparse rectifier neural networks[C]//Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. [S.l.]: AAAI Press, 2011: 315-323.
[25] ABADI M, AGARWAL A, BARHAM P, et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems[EB/OL]. [2021-04-03]. https://arxiv.org/abs/1603.04467.

编辑 薛晋栋