一种基于数据标准差的卷积神经网络量化方法

黄 赟[1],张 帆[2],郭 威[2],陈 立[1],羊 光[3]
(1. 信息工程大学,河南郑州 450001; 2. 国家数字交换系统工程技术研究中心,河南郑州 450002;
3. 河南省广播电视监测中心,河南郑州 450002)

摘 要: 当前卷积神经网络模型存在规模过大且运算复杂的问题,难以应用部署在资源受限的计算平台. 针对此问题,本文基于数据标准差提出了一种适合部署在现场可编程门阵列(Field Programmable Gate Array, FPGA)上的对数量化方法. 首先,依据FPGA的特性提出对数量化方法,将32 bit浮点乘法运算转换为整数乘法及移位运算,提高了运算效率. 然后通过研究数据分布特点,提出基于数据标准差的输入量化及权值混合bit量化方法,能够有效减少量化损失. 通过对RepVGG、EfficientNet等网络进行效率与精度对比实验,8 bit量化使得大型神经网络精度仅下降1%左右;输入量化为8 bit,权重量化为10 bit场景下,模型精度损失小于0.2%,达到浮点模型几乎相同的准确率. 实验表明,所提量化方法能够使得模型大小减少75%左右,在基本保持原有模型准确率的同时有效地降低功耗损失、提高运算效率.

关键词: 卷积神经网络;现场可编程门阵列;对数量化;数据标准差;混合bit
基金项目: 国家自然科学基金创新研究群体项目(No.61521003)
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2023)03-0639-09
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210691

A Quantification Method of Convolutional Neural Network Based on Data Standard Deviation

HUANG Yun[1], ZHANG Fan[2], GUO Wei[2], CHEN Li[1], YANG Guang[3]
(1. *Information Engineering University*, *Zhengzhou*, *Hennan* 450001, *China*;
2. *National Digital Switching System Engineering Technology Research Center*, *Zhengzhou*, *Hennan* 450002, *China*;
3. *Henan Administration of Radio and Television Monitoring Center*, *Zhengzhou*, *Hennan* 450002, *China*)

Abstract: Due to the large scale of the current convolutional neural network model and complex calculations, it is not suitable for deployment on resource-constrained computing platforms. In order to solve this problem, this paper proposes a logarithmic quantization method based on data standard deviation, which is suitable for deployment on FPGA (Field Programmable Gate Array). According to the characteristics of FPGA, this paper proposes a logarithmic quantization method to convert the 32 bit floating point multiplication operation into integer multiplication and shift operation, which improves the efficiency of the operation. By studying the characteristics of data distribution, the input quantization and mixed bit weight quantization methods based on data standard deviation are proposed, which can effectively reduce the quantization loss. The experimental results show that the accuracy of large-scale neural network is only reduced by about 1% due to 8-bit quantization. When the input is quantized to 8 bits and the weight is quantized to 10 bits, the accuracy loss of the model is less than 0.2%, which is almost the same as that of the floating-point model. Experimental results show that the proposed method can reduce the size of the model by about 75%, and effectively reduce the power loss and improve the computing efficiency while maintaining the accuracy of the original model.

Key words: convolutional neural networks; field programmable gate array(FPGA); logarithmic quantization; standard deviation of the data; mixed bit number
Foundation Item(s): Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No.61521003)

收稿日期:2021-05-29;修回日期:2021-10-18;责任编辑:梅志强

1 概述

卷积神经网络(Convolutional Neural Networks, CNN)近年来成功地应用在图像分类[1,2]与目标检测[3,4]等计算机视觉领域,但因模型太大、运算复杂及任务的实时性要求等问题使其在智能手机与无人机等资源受限平台难以部署应用,极大地限制了CNN模型的部署应用. 如VGG16[5]网络参数量有1亿3千多万个,完成一次图片识别任务需要占用500多MB内存空间以及进行300多亿次浮点运算.

针对上述问题,近年来提出了一系列模型压缩方法来解决,其中可以大致归纳为如下几类:模型剪枝[6]、知识蒸馏[7]、模型量化[8]和低秩分解[9]. 其中模型量化指损失部分模型准确率,将原始模型中32位浮点表示的数据用低精度数来表示. 模型量化能够有效地减少内存消耗、加快推理速度及降低功耗损失,成为当前实际应用中最常用的模型压缩方式. 然而将模型参数从高精度数转换为低精度数通常会带来较大的模型准确率损失,对此近年有大量的研究来解决这个问题. 其中一类是量化感知训练[10],量化感知训练指在模型训练的过程中对模型进行量化,其能够在损失较小的准确率情况下获得较好的量化效果,但量化感知训练的量化过程复杂耗时,需要大量的训练数据集,而对于某些存在数据集敏感性问题及需要量化实时性的场景并不适用. 另一类是训练后量化[11],训练后量化指将训练好的模型直接转换为低精度模型. 训练后量化有实现简单高效、量化效果较好和无需大量训练数据集等优点.

目前,基于现场可编程逻辑门阵列(Field Programmable Gate Array, FPGA)的CNN模型部署研究因其擅长定点运算、具有可编程性、运算高速及低功耗等特点成为了一个研究热点. 但FPGA存在存储资源受限且不适合浮点运算,在FPGA中浮点运算相较于定点运算要耗费数倍的资源,同时当前基于FPGA的卷积神经网络量化加速研究面临着模型精度损失过大、资源利用率低及软硬件设计不合理等问题. 对此本文聚焦CNN模型训练后量化算法,采用对硬件友好的对数量化方案,结合FPGA的运算特性[12,13],提出一种使CNN模型适合部署在FPGA芯片上的模型量化策略. 本文的主要工作如下:

(1) 提出一种使CNN量化模型适合部署在FPGA上的对数量化方法;

(2) 提出一种基于数据标准差的量化因子调整方法;

(3) 提出一种基于数据标准差的权值混合bit量化方法.

2 现状分析

CNN模型的参数基本为卷积运算中权重及偏置值[1],同时绝大多数运算集中在卷积计算过程,因而CNN量化指的是对卷积层的权重、输入和偏置进行量化,由此减小模型大小,提高运算效率. 卷积基本计算公式如下:

$$z=\omega * x+b \tag{1}$$

$$z'=f(z) \tag{2}$$

式(1)中ω为权重,x为输入特征,b为偏置值;式(2)中$f(\cdot)$为激活函数,z'为输出结果.

CNN模型量化指的是将原来使用单精度数据表示的模型用低bit数据来代替,因而量化是一个数据值与类型转换的仿射变换过程. 其基本原理[14]如下所示:

$$x_{\text{int}}=\text{round}(\Delta x)+o \tag{3}$$

$$x_Q=\text{clamp}(x_{q\_\min},x_{q\_\max},x_{\text{int}}) \tag{4}$$

其中:

$$\text{clamp}(a,b,x)=\begin{cases}a, & x\le a\\ x, & a<x<b\\ b, & x\ge b\end{cases} \tag{5}$$

式(3)中round(·)为取整操作;x为输入的浮点数据;x_{int}为量化后输出的整数值;Δ为量化缩放因子;o为量化零点,常取0值,因此可直接将o舍去. 式(4)中$x_{q_\min},x_{q_\max}$为量化区间的最小值及最大值;clamp(·)为截断函数.

在量化之后,通常需要通过反量化还原之前的缩放因子来评估量化效果,其基本公式如下:

$$x_{D\_Q}=(x_Q-o)\Delta \tag{6}$$

如图1所示,将原始模型32位浮点权重及输入通过量化函数$Q(\cdot)$转换为8 bit整型数据,再进行卷积操作. 反量化DQ(·)操作为一个可选操作,在实际部署中可将其与下一层卷积的量化操作相合并.

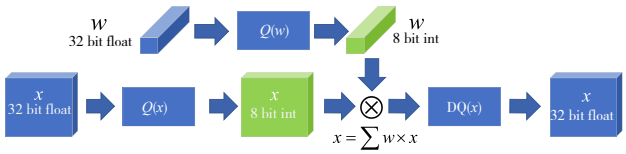


图1 量化过程示意图

由式(3)可以看到,模型量化主要难点集中在量化缩放因子Δ的选取上. 对此,文献[15]利用模型权重的近似高斯分布特性,提出了分段线性量化(PieceWise Linear Quantization, PWLQ)方法,有效地提高了量化模型的精度. PWLQ将区间分为中心区域与尾部区域两部分,再给每个部位分配相同的量化级别,对每部分分别进行量化. 但其需要通过寻找最优切分点对数据进行分割,同时将同一层的数据使用不同的量化参数对

其进行量化及反量化操作增加了硬件实现的难度和运算开销. 文献[16]对权重利用了对数形式的量化,使用逐点移位及标志位翻转操作代理了乘法运算,使用移位操作代替绝对值乘法操作,有效地缓解了计算成本过高的问题,有利于硬件的实现. 但其量化为量化感知训练,因而需要大量的训练数据集及时间来训练模型. 文献[17]通过利用ReLU激活函数的尺寸等价缩放特性,提出了一种无需数据集的训练后量化方法,针对逐层量化会使得值较小的通道直接全部被置为0而导致精度下降的问题,利用激活函数ReLU的数学特性,均衡相邻两层权重各通道的数据范围,在不增加硬件开销的情况下达到了逐通道量化的效果,同时还可以纠正量化过程中引入的量化误差. 但文献[17]仅考虑了层与层之间单层连接的模型量化,而对于含有多输入多输出层的模型无法量化,其量化虽然减少了模型参数,但是量化过程仍然含有浮点运算,计算复杂度并没有得到显著降低.

当前对CNN量化方法研究主要集中在减小量化损失方面进行. 如上述文献[15,16]分别通过研究分段线性量化、基于对数形式的非均匀量化等方法来细化量化尺度,有效地减小了量化损失. 但其通常仅考虑获得更小的量化损失而忽略了硬件设备的运算特性等问题,导致硬件实现效率不高. 还有些文献,如文献[18]使用1 bit数等超低bit位宽来量化模型,此类方法虽能够极大减小模型大小,具有高效的推理效率,但其通常会导致严重的模型准确率下降,因而一般需要从头开始训练模型,模型的普适性会大打折扣. 综上,本文使用基于对数的训练后量化方法,通过对缩放因子取对数使得将量化及反量化过程中,被量化数据与缩放因子的浮点乘法运算转换为移位操作,能够更有效地提高模型在FPGA上的运算效率. 本文采用8 bit位宽来量化模型,在尽可能高效快速压缩模型的同时确保模型精度损失在1%以内. 下文对其具体内容进行介绍.

3　本文策略

针对当前模型量化方法存在着不利于硬件设备的部署实现及量化模型仍然含有浮点运算等问题,对此本文采用对数量化策略,将原始CNN模型中卷积层的浮点乘法运算转换为整数乘法及移位运算,能够有效加快卷积层的计算速度、降低功耗损失、减小模型大小. 数据的标准差能够很好的表现数据的分散程度,而数据的分散程度与量化位宽的敏感度息息相关,在量化过程中引入标准差能够有效地减少量化损失,提高量化模型的精度. 本文使用对数量化方法,结合数据标准差,提出一种使CNN量化模型适合部署在FPGA芯片上的量化方法.

3.1　对数量化策略

针对式(3)中量化缩放因子取值,本文提出基于对数的取值方案,如下所示:

$$s=\mathrm{ceil}(\log_2(|x|_{\max})) \tag{7}$$

$$\Delta=2^{n-s-1} \tag{8}$$

式(7)中ceil(·)为向上取整函数,x为被量化数值;式(8)中Δ为量化缩放因子,n为需要量化的低bit位宽. 将式(7)带入式(3)中可以得到:

$$x_{\mathrm{int}}=\mathrm{round}(2^{n-s-1}x) \tag{9}$$

令式(9)中$r=n-s-1$,而$2^r x$可用简单的移位器来实现,如下式所示:

$$2^r x=\begin{cases} x & ,r=0 \\ x<<r & ,r>0 \\ x>>r & ,r<0 \end{cases} \tag{10}$$

由此,对数量化将原始卷积浮点计算转换为定点整数乘法及移位运算. 而对于卷积操作,当同时将输入与权重使用对数方法进行量化时,可将乘法操作转换为整数乘法与加法操作,如下所示:

$$z=x_{\mathrm{int}}*w_{\mathrm{int}} \tag{11}$$

$$r_z=r_x+r_w \tag{12}$$

其中,z为卷积计算的结果,其为整数;r_z为卷积计算后的移位量,在运算过程中可直接与下一层卷积运算中的输入量化因子的移位量相结合. 将原始卷积操作中的浮点乘法运算转换为整数乘法,能够有效降低计算复杂度. 如相比于32位浮点运算,8位整数运算可以节省高达30倍的功耗及116倍的面积[11],从而显著提高计算吞吐量.

下面以8 bit整数量化为例,展示对数量化大致的运算过程. 如图2所示,首先选取量化输入数据绝对值的最大值,带入式(7)和式(8)求出量化缩放因子,然后带入式(9)求量化后的整数值. 对数量化虽然能够加快运算速度,但因为所求缩放因子为2的指数形式,对指数取整的过程中会给缩放因子带来较大的误差,使得量化最大值和量化区间的最大值相差较大,由此导致了有一部分区间被浪费,如图3所示. 因而需要在对数量化的基础上进行改进优化.

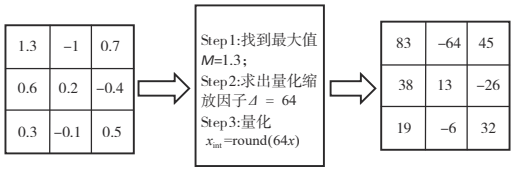


图2　对数量化计算示意图

3.2　基于标准差的输入量化

对于输入的量化,我们使用少量数据集(取500张

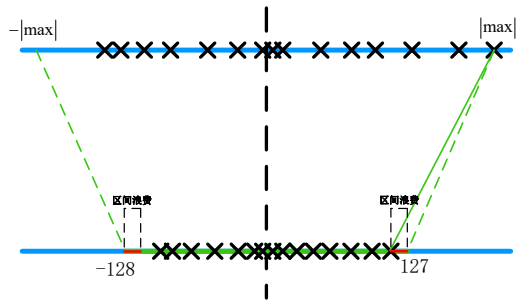


图3 对数量化舍入误差示意图

左右图片)来预估每一个卷积层输入值绝对值的最大值,从而会带来一定的预估误差. 对此,通过分析不同层输入数据的分布特点,本文认为对于标准差较大的数据,其数据分布会较为分散而导致最大值会偏离数据中心较多,对此需要将其最大值适当地缩小,使其量化宽度更小;而对于标准差较小的数据,本文认为其数据分布较为集中因而需要将其最大值适当地放大,使得量化数据分布更加的分散. 如图4所示,根据数据标准差来调节量化数据的离散程度,对于标准差较大的数将其适当地缩小,一定程度上可以抵消对数量化中缩放因子因取整所带来的舍入误差;对于标准差较小的数将其适当地放大,能够使得量化数据更加的分散,使得量化区间的选取更加精准.

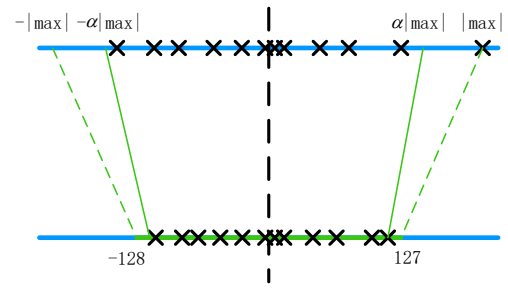


图4 基于标准差的对数量化示意图

对于每一张输入图片,获取每一层卷积层输入数据绝对值的最大值,然后依据输入数据的标准差对其最大值进行适当地缩放,由此来调节量化因子Δ取值,达到更好的量化效果. 本文选取90%分位点与10%分位点数据标准差作为基准,分别对最大值进行缩放. 具体过程如下:对于输入的第k张图片,获得第i层卷积层的输入特征绝对值的最大值$x_{\max}^{k_i}$及其标准差σ_k^i. 令$\sigma_k=(\sigma_k^1,\sigma_k^2,\cdots,\sigma_k^M)$($M$为卷积层个数),则将其中大于其90%分位点标准差$p_{0.9}$所对应的特征最大值缩放$a$倍,将其中小于其10%分位点标准差$p_{0.1}$所对应的特征最大值缩放$b$倍. 如$\sigma_k^l$大于数列$\sigma_k$的90%分位点的值,则将其输入最大值更新为:$x_{\max}^{k\_l}=ax_{\max}^{k\_l}$. 而对于第$i$层,将每张图片输入所得最大值更新后得到最大值数列$x^i=(x_{\max}^{1_i},x_{\max}^{2_i},\cdots,x_{\max}^{N_i})$($N$为数据集中图片数量),取其最大值$x_{\max}^i$带入式(8)中得到量化缩放因子$\Delta$. 具体计算如下:

$$x_{\max}^{k\_i}=\max(|x^{k\_i}|) \tag{13}$$

$$x_{\max}^{k\_i}=\alpha_i x_{\max}^{k\_i} \tag{14}$$

$$x_{\max}^{i}=\max\{x_{\max}^{1\_i},x_{\max}^{2\_i},\cdots,x_{\max}^{N\_i}\} \tag{15}$$

其中:

$$\sigma_k^i=\sqrt{\frac{1}{N}\sum_{j=1}^{N}(x_j^{k\_i}-\overline{x^{k\_i}})^2} \tag{16}$$

$$\alpha_i=\begin{cases}a, & \sigma_k^i>p_{0.9}\\ b, & \sigma_k^i<p_{0.1}\\ 1, & \text{others}\end{cases} \tag{17}$$

3.3 基于标准差的权值混合bit量化

使用单一bit的量化方法,存在忽略不同卷积层对量化位宽敏感度不同的问题而造成一定的量化损失. 对此,提出了基于数据标准差的权值混合bit量化策略,利用不同bit位宽敏感度不同的特性使用不同的bit位宽,来改进对数量化效果. 通过对每一个卷积层权重值标准差与所有卷积层权重标准差所组成数列的四分位点相比较,自动选择量化位数,使得能够根据不同卷积层的数据分布特点动态量化权重值,获得更好的模型量化效果. 如对第i层数据,其标准差为σ_i,$\sigma=(\sigma_1,\sigma_2,\cdots,\sigma_N)$为所有层标准差组成的数列,$p_{0.25}$和$p_{0.75}$分别为$\sigma$的下四分位点与上四分位点,$m_1$、$m_2$、$m_3$为不同bit位宽,则量化位宽取值为:

$$n_i=\begin{cases}m_1, & \sigma_i>p_{0.75}\\ m_2, & p_{0.25}\le\sigma_i\le p_{0.75}\\ m_3, & \sigma_i<p_{0.25}\end{cases} \tag{18}$$

式(18)中n_i为第i层量化bit位,将其带入式(8)可得量化缩放因子Δ.

4 实现设计

CNN模型量化主要针对卷积运算来设计,针对卷积计算中的输入、权重及偏置值,本文使用基于标准差的对数量化方法对其进行量化. 当前CNN模型中通常包含批归一化(Batch Normalization, BN)层[19],但其会影响模型推理速度,增加量化难度,对此本文采取的策略是将BN层与卷积层相合并. 下面对其具体实现进行介绍.

4.1 BN层合并

当前CNN模型大多在卷积层后面增加了一个BN层将数据归一化,能够有效解决过拟合和梯度爆炸等问题,加快训练过程中网络收敛速度. 其计算公式为:

$$y=\frac{\gamma(z-\mu)}{\sqrt{\sigma^2+\varepsilon}}+\beta \tag{19}$$

其中z为卷积层输出的计算结果,γ为缩放因子,μ为均值,σ^2为方差,ε为一个很小的正数,β为偏置系数,y为BN层运算的结果. 但在模型推理过程中,增加BN层需要占用更多的计算资源,会减慢推理速度. 同时BN层使用浮点计算,从而也要将其量化,从而增加了量化的难度. 因此,本文将BN层合并到卷积层的权重及偏置参数中,来提升模型推理速度,减少量化工作. 其过程如下.

将式(1)带入式(19),整理可得:

$$y=\frac{\gamma w}{\sqrt{\sigma^2+\varepsilon}}*x+\frac{\gamma(b-\mu)}{\sqrt{\sigma^2+\varepsilon}}+\beta \tag{20}$$

令:

$$w'=\frac{\gamma w}{\sqrt{\sigma^2+\varepsilon}} \tag{21}$$

$$b'=\frac{\gamma(b-\mu)}{\sqrt{\sigma^2+\varepsilon}}+\beta \tag{22}$$

可得:

$$y=\omega'*x+b' \tag{23}$$

如图5所示,将卷积层与BN层相合并后得到新的权重w'及偏置值b',能够有效地减小模型推理过程的计算,加快模型推理速度.

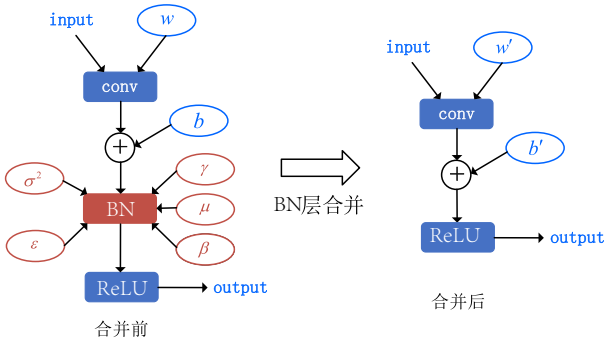


图5　BN层合并示意图

4.2　输入、权重及偏置量化

本文首先将输入、权重及偏置值采用对数量化方法,将其分别量化为不同的bit位数进行对比分析. 然后在8 bit对数量化方案的基础上,使用基于标准差的输入量化方法来调整输入的量化值,使用基于标准差的权值混合bit量化方案调整权重量化值.

对输入量化需要使用少量数据集(取500张左右图片)来获得每一层卷积输入的预估最大值,来求量化缩放因子Δ. 本文使用基于标准差的输入量化方法来量化输入值,通过比较卷积输入数据的标准差来动态缩放其最大值,由此获得更精准的最大值,在实验中选取$a=0.6, b=1.4$,带入式(17)可求得量化缩放因子.

在使用混合bit量化权重过程中根据权重数据的标准差来动态选择量化位数,实验中发现10 bit量化模型几乎没有量化损失,8 bit量化能够获得较好的量化效果,而6 bit量化会有较大的量化损失. 对此,在本文折中选取{7,8,9}三种bit来量化权重值,使其量化压缩效果与8 bit量化基本相同时能够获得更高的准确率. 综上,本文量化大致实验过程如图6所示.

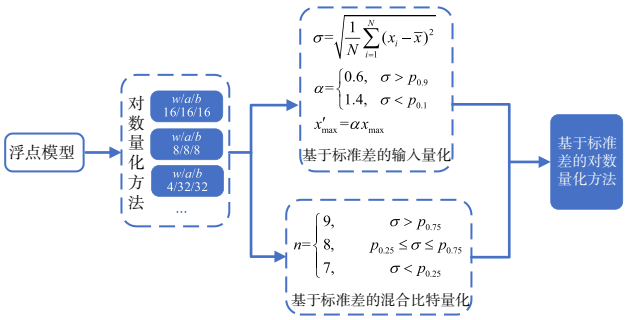


图6　量化方法实验过程图

5　实验评估

本文在ImageNet[20]数据集上验证所提及的量化策略,对比了Resnet50[21]、Inception V3[22]、MobileNet V2[23]、RepVGG[24]、EfficientNet[25]和RexNet[26]等常用的图像分类模型的量化效果. 本文实验分为如下几个部分:(1)使用对数量化策略将CNN模型的输入、权重及偏置量化为16、8、4等不同bit位宽进行对比研究;(2)采用8 bit对数量化,在第一部分的实验基础上使用基于标准差的输入量化方法,对输入值进行量化分析;(3)在第一部分实验的基础上使用基于标准差的权值混合bit量化方法对权重进行量化分析;(4)结合第二和第三部分实验结果,验证基于标准差的对数量化方法效果.

本文实验在Centos7.6操作系统下,使用两块Nvidia Tesla V100 16 GB GPU,在tensorflow1.5框架下进行,浮点模型使用赛灵思Vitis AI提供的图像分类模型,实验中对于模型准确率使用top1准确率和top5准确率[20]来评估效果. 本文在评估量化性能时使用反量化操作将卷积计算结果转换为单精度类型数据,使得量化模型能够在tensorflow框架下正常运行. 在实际项目中使用xilinx alveo u280设备验证了所提方法的有效性.

5.1　对数量化

本节使用对数量化方法对Resnet50及Inception v3网络的权重、输入和偏置值进行量化处理,并比较分析了将其量化成16 bit、8 bit和4 bit等不同bit位模型压缩准确率变化情况. 表1列出了Resnet50网络和Inception v3网络量化前后的结果. 其中*w/a/b*表示网络中的权重、输入值和偏置值的bit位数. 从表中可以看到,当直接使用对数量化将权值、输入和偏置值量化至8 bit

时,模型的Top1准确率降低了1%~2%左右,Top5准确率降低1%左右. 当单独量化输入时,可以看到模型的准确率会有较大的下降,本文认为这是由于输入量化使用的是少量数据集获得的近似最大值来得到量化参数,从而会带来预估误差,对此提出基于标准差的输入量化来解决. 当单独将权重值量化为10 bit时会造成极小的模型准确率下降,将其量化为6 bit时有较大的量化误差,而将其量化为4 bit会使模型彻底失效,由此可以看到权重量化对量化宽度较为敏感,对此提出基于标准差的权值混合bit量化来解决.

5.2 基于标准差的输入量化

本小节在4.1实验基础上,对输入采用2.2节中提出的基于标准差的输入量化方法,单独将输入量化为8 bit整数,对比了Resnet50网络和Inception v3网络量化前后的结果. 从表2可以看到,基于标准差的输入量化能够减小模型Top1准确率75%以上的对数方法量化损失,而对Resnet50网络的对数方法减小的量化损失达到了82%左右,使得输入的对数量化模型达到了和原始模型几乎相同的准确率.

5.3 基于标准差的权值混合bit量化

本小节在4.1实验基础上,权重量化使用2.3节提出的基于标准差的权值混合bit量化方法,对比了Resnet50网络与Inception v3网络量化前后的结果. 表3使用对数量化指将权重量化为8 bit整数,表3中带*号的结果为将权重值量化为{6,8,10}混合bit的量化结果,不带*号的为将权重量化为{7,8,9}混合bit的量化结果. 可以看到将模型权重量化为{6,8,10}bit范围时模型准确率相对对数量化均有所下降,而将其量化为{7,8,9}bit范围是准确率有所提高. 本文认为基于标准差的权值混合bit量化能够改善权值对数量化的量化效果,但当量化bit减小到一定大小时,此时低bit影响更大,会造成更大的量化误差. 本文选用{7,8,9}bit范围量化,使得模型压缩大小保持和8 bit量化大致相同的情况下能够有效地减小量化损失.

表1 Resnet50和Inception v3网络对数量化结果

模型	位宽(*w*/*a*/*b*)/bit	准确率/%	
		Top1	Top5
Resnet50	32/32/32	75.202	92.194
	16/32/32	75.212	92.196
	16/16/16	74.912	91.91
	10/32/32	75.204	92.196
	8/32/32	74.62	91.886
	6/32/32	60.084	82.55
	8/32/8	74.526	91.866
	32/8/32	74.842	92.03
	8/8/8	74.242	91.588
	4/32/32	0.094	0.512
Inception V3	32/32/32	77.978	93.942
	16/32/32	77.980	93.944
	16/16/16	77.982	93.944
	10/32/32	77.96	93.88
	8/32/32	77.052	93.38
	6/32/32	50.302	73.446
	8/32/8	76.798	93.25
	32/8/32	77.728	93.812
	8/8/8	76.584	93.204
	4/32/32	0.144	0.544

表2 基于标准差的输入量化结果

Model	原始模型准确率/%		对数量化准确率/%		基于标准差的输入量化准确率/%		降低的量化损失/%	
	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
Resnet50	75.202	92.194	74.842	92.03	75.164	92.038	82.56	4.88
Inception V3	77.978	93.942	77.728	93.812	77.914	93.838	75	2

表3 基于标准差的权值混合bit量化结果

Model	原始模型准确率/%		对数量化准确率/%		基于标准差的混合bit量化准确率/%	
	Top1	Top5	Top1	Top5	Top1	Top5
Resnet50	75.202	92.194	74.62	91.886	73.804*	91.726*
					74.866	92.088
Inception V3	77.978	93.942	77.052	93.38	76.274*	92.993*
					77.284	93.566

5.4 基于标准差的对数量化方法

本小节在对数量化的基础上,结合使用基于标准差的输入量化及基于标准差的权重混合bit量化方法,对Resnet50网络、Inception V3网络、VGG16网络、RexNet3网络及RepVGG A0网络进行量化实验,对比分析实验效果. 表4对各实验模型的偏置采用对数量化将其转换为8 bit整数,输入采用基于标准差的输入量化将其转换为8 bit整数,权重选取{7,8,9}bit范围内的

混合bit量化将其量化为整数. 实验结果表明基于标准差的对数量化方法能够使得量化模型相比于原始模型Top1准确率基本降低到1%以内,Top5准确率降低到0.5%以内,同时8 bit整数量化能够减小75%左右的模型大小,有效地改善了对数量化效果. 表5将偏置采用对数量化将其量为32 bit整数,输入采用基于标准差的输入量化将其量化为8 bit整数,权重选取{9,10,11}bit范围的混合bit量化. 此时量化模型Top1误差基本都减小到0.15%左右,达到了和原始模型几乎相同的准确率.

表4　基于标准差的对数量化结果(1)

Model	原始模型准确率/%		基于标准差的对数量化准确率/%		量化误差/%		模型大小		
	Top1	Top5	Top1	Top5	Top1	Top5	量化前/MB	量化后/MB	压缩率/%
Resnet50	75.202	92.194	74.406	91.862	0.796	0.332	97.84	24.51	74.95
Inception V3	77.978	93.942	76.984	93.484	0.994	0.458	91.23	22.82	74.98
VGG16	70.894	89.848	70.528	89.75	0.366	0.098	527.81	131.96	74.99
RexNet3	82.63	96.25	82.09	96.01	0.54	0.24	132.95	33.30	74.95
RepVGG A0	72.42	90.49	71.40	90.08	1.02	0.41	34.89	8.75	74.92

表5　基于标准差的对数量化结果(2)

Model	原始模型准确率/%		基于标准差的对数量化准确率/%		量化误差/%	
	Top1	Top5	Top1	Top5	Top1	Top5
Resnet50	75.202	92.194	75.052	92.1	0.15	0.094
Inception V3	77.978	93.942	77.814	93.812	0.164	0.13
VGG16	70.894	89.848	70.742	89.778	0.152	0.07
RexNet3	82.63	96.25	82.49	96.16	0.14	0.11
RepVGG A0	72.42	90.49	77.20	90.34	0.22	0.15

表6将MobileNet V2网络的偏置采用对数量化将其转换为32 bit整数,输入采用基于标准差的输入量化将其转换为8 bit整数,权重采用对数量化将其转换为8 bit、10bit和16bit等不同位宽的整数. 结果显示虽然将MobileNet V2网络的权重量化为8bit会使得量化模型失效,但将其量化为12bit时Top1量化误差下降到2%以内,将其量化为16bit时量化模型达到原始模型几乎相同的准确率. 当把EfficientNet-b0网络的偏置采用对数量化将其转换为32 bit整数,输入采用基于标准差的输入量化将其量化为16 bit整数,权重采用对数量化将其转换为8 bit、10 bit和16 bit等不同位宽的整数. 结果显示将EfficientNet-b0网络的权重量化为12 bit时Top1量化误差下降到0.576%,当权重量化为16 bit时Top1量化误差下降到0.328%,Top5量化误差下降到0.143%,达到与原始模型几乎相同的准确率. 实验结果表明本文所提出的量化方法对轻量型图像分类网络依然有效.

表6　轻量型网络量化结果

Model	权重位宽/bit	原始模型准确率/%		基于标准差的对数量化准确率/%		量化误差/%		模型大小		
		Top1	Top5	Top1	Top5	Top1	Top5	量化前/MB	量化后/MB	压缩率/%
MobileNet v2	8	70.126	89.532	0.266	1.184	/	/	13.6	/	/
	12			68.298	88.44	1.828	1.092		/	/
	16			69.84	89.31	0.286	0.222		6.74	50.44
EfficientNet-b0	8	76.63	93.025	24.854	44.864	/	/	20.3	8.99	55.714
	12			76.054	92.83	0.576	0.195		/	/
	16			76.302	92.882	0.328	0.143		12.76	37.143

6　结束语

由于当前CNN模型存在的模型过大、消耗的计算资源过多及能耗高等问题,因而并不适合部署在资源受限的计算平台,极大地限制了其应用部署. 对此,本文提出了一种适合部署在FPGA的CNN模型量化策略. 本文提出的对数量化方法,使得卷积操作中的浮点数乘法运算转换为低bit整数乘法及移位运算,有效地提高了运算速度. 通过研究数据分布的特点,在对数量化的基础上提出了基于标准差的输入量化和基于标准差的权值混合bit量化策略,使得对数量化模型达到与原始模型几乎相同的准确率. 在实验中通过对常用图像分类模型进行量化实验验证了所提方法的有效性,且所验证的分类模型常被用来做图像目标检测算法的骨干网络,因而所提方法对目标检测模型依然有效. 本文通过聚焦FPGA的运算特性使用对数量化方法,改变了当前研究主要针对减小量化损失而忽略实际部署设

备特性的做法,更有利于将神经网络模型应用在实际设备中. 本文提及的方法能够有效地解决当前神经网络模型难以走出实验室的难题,使得CNN模型真正地落地部署,走向实际应用. 下一步考虑将对数量化的计算优势与量化感知训练相结合,进一步减小量化损失.

参考文献

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]//Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. Red Hook: Curran Associates Inc, 2012: 1097-1105.

[2] 江泽涛, 秦嘉奇, 张少钦. 参数池化卷积神经网络图像分类方法[J]. 电子学报, 2020, 48(9): 1729-1734.
JIANG Z T, QIN J Q, ZHANG S Q. Parameterized pooling convolution neural network for image classification[J]. Acta Electronica Sinica, 2020, 48(9): 1729-1734. (in Chinese)

[3] 李宝奇, 贺昱曜, 强伟, 等. 基于并行附加特征提取网络的SSD地面小目标检测模型[J]. 电子学报, 2020, 48(1): 84-91.
LI B Q, HE Y Y, QIANG W, et al. SSD with parallel additional feature extraction network for ground small target detection[J]. Acta Electronica Sinica, 2020, 48(1): 84-91. (in Chinese)

[4] 罗会兰, 陈鸿坤. 基于深度学习的目标检测研究综述[J]. 电子学报, 2020, 48(6): 1230-1239.
LUO H L, CHEN H K. Survey of object detection based on deep learning[J]. Acta Electronica Sinica, 2020, 48(6): 1230-1239. (in Chinese)

[5] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. (2014-09-04). https://arxiv.org/abs/1409.1556.

[6] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[C]//Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. Cambridge, MA, USA: MIT Press, 2015. 1135 – 1143.

[7] YE H J, LU S, ZHAN D C. Distilling cross-task knowledge *via* relationship matching[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 12393-12402.

[8] CAI Y H, YAO Z W, DONG Z, et al. ZeroQ: A novel zero shot quantization framework[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 13166-13175.

[9] DENTON E L, ZAREMBA W, Bruna J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]//Proceedings of the Advances in Neural Information Processing Systems. Cambridge, MA, USA: MIT Press, 2014: 1269-1277.

[10] ZHANG D Q, YANG J L, YE D, et al. LQ-Nets: Learned quantization for highly accurate and compact deep neural networks[C]//European Conference on Computer Vision. Cham: Springer, 2018: 373-390.

[11] CHOUKROUN Y, KRAVCHIK E, YANG F, et al. Low-bit quantization of neural networks for efficient inference [C]//2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). Piscataway: IEEE, 2019: 3009-3018.

[12] 刘杰, 葛一凡, 田明, 等. 基于ZYNQ的可重构卷积神经网络加速器[J]. 电子学报, 2021, 49(4): 729-735.
LIU J, GE Y F, TIAN M, et al. Reconfigurable convolutional network accelerator based on ZYNQ[J]. Acta Electronica Sinica, 2021, 49(4): 729-735. (in Chinese)

[13] 蹇强, 张培勇, 王雪洁. 一种可配置的CNN协加速器的FPGA实现方法[J]. 电子学报, 2019, 47(7): 1525-1531.
JIAN Q, ZHANG P Y, WANG X J. An FPGA implementation method for configurable CNN co-accelerator[J]. Acta Electronica Sinica, 2019, 47(7): 1525-1531. (in Chinese)

[14] KRISHNAMOORTHI R. Quantizing deep convolutional networks for efficient inference: A whitepaper[EB/OL]. (2018-06-21). https://arxiv.org/abs/1806.08342.

[15] FANG J, SHAFIEE A, ABDEL-AZIZ H, et al. Post-training Piecewise Linear Quantization for Deep Neural Networks[C]//European Conference on Computer Vision. Cham: Springer, 2020: 69-86.

[16] ELHOUSHI M, CHEN Z H, SHAFIQ F, et al. DeepShift: towards multiplication-less neural networks[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE, 2021: 2359-2368.

[17] NAGEL M, BAALEN M V, BLANKEVOORT T, et al. Data-free quantization through weight equalization and bias correction[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2019: 1325-1334.

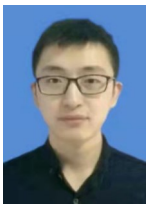
[18] QIN H T, GONG R H, LIU X L, et al. Forward and backward information retention for accurate binary neural networks[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 2247-2256.

[19] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift[C]//Proceedings of the 32nd International Conference on International Conference on Machine Learning. Lille, France: JMLR.org, 2015: 448-456.
[20] DENG J, DONG W, SOCHER R, et al. ImageNet: A large-scale hierarchical image database[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2009: 248-255.
[21] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2016: 770-778.
[22] 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. Piscataway: IEEE, 2016: 2818-2826.
[23] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 4510-4520.
[24] DING X H, ZHANG X Y, MA N N, et al. RepVGG: making VGG-style ConvNets great again[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2021: 13728-13737.
[25] TAN Mingxing, LE Quoc V. EfficientNet: Rethinking model scaling for convolutional neural networks[EB/OL]. (2019-05-28).https://arxiv.org/abs/1905.11946V4.
[26] HAN D, YUN S, HEO B, et al. Rethinking channel dimensions for efficient model design[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2021: 732-741.

作者简介

黄　赟　男,1993年9月出生于江西省新余市. 信息工程大学硕士生. 主要研究方向为神经网络模型量化压缩、网络内生安全.
E-mail: yyhuangz@163.com

张　帆(通讯作者)　男,1981年9月出生. 博士. 现为国家数字交换系统工程技术研究中心副研究员、硕士生导师. 主要研究方向为主动防御、人工智能、高性能计算. 中国电子学会会员编号:E190013697M.
E-mail: 17034203@qq.com



郭　威　男,1990年8月出生. 博士. 现为国家数字交换系统工程技术研究中心助理研究员. 主要研究方向为主动防御、人工智能、高性能计算. 中国电子学会会员编号:E190029991M.
E-mail: guowjss@126.com

陈　立　男,1997年2月出生于浙江省义乌市. 信息工程大学硕士生. 主要研究方向为计算机视觉.
E-mail: 2464863136@qq.com

羊　光　女,1986年11月出生于河南省驻马店市. 学士. 主要研究方向为网络流量分类、入侵检测、人工智能.
E-mail: flyingaki@126.com