卷积神经网络的损失最小训练后参数量化方法

张帆[1]，黄赟[2,3]，方子茁[3,4]，郭威[1]

（1. 国家数字交换系统工程技术研究中心，河南 郑州 450002；
2. 信息工程大学，河南 郑州 450001；3. 紫金山实验室，江苏 南京 211111；
4. 东南大学网络空间安全学院，江苏 南京 211189）

摘 要：针对数据敏感性场景下模型量化存在数据集不可用的问题，提出了一种不需要使用数据集的模型量化方法。首先，依据批归一化层参数及图像数据分布特性，通过误差最小化方法获得模拟输入数据；然后，通过研究数据舍入特性，提出基于损失最小化的因子动态舍入方法。通过对 GhostNet 等分类模型及 M2Det 等目标检测模型进行量化实验，验证了所提量化方法对图像分类及目标检测模型的有效性。实验结果表明，所提量化方法能够使模型大小减少 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；批归一化；模拟输入数据；动态舍入

中图分类号：TP391

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2022068

Lost-minimum post-training parameter quantization method for convolutional neural network

ZHANG Fan[1], HUANG Yun[2,3], FANG Zizhuo[3,4], GUO Wei[1]

1. National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China
2. Information Engineering University, Zhengzhou 450001, China
3. Purple Mountain Laboratories, Nanjing 211111, China
4. School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China

Abstract: To solve the problem that that no dataset is available for model quantization in data-sensitive scenarios, a model quantization method without using data sets was proposed. Firstly, according to the parameters of batch normalized layer and the distribution characteristics of image data, the simulated input data was obtained by error minimization method. Then, by studying the characteristics of data rounding, a factor dynamic rounding method based on loss minimization was proposed. Through quantitative experiments on classification models such as GhostNet and target detection models such as M2Det, the effectiveness of the proposed quantification method for image classification and target detection models was verified. The experimental results show that the proposed quantization method can reduce the model size by about 75%, effectively reduce the power loss and improve the computing efficiency while basically maintaining the accuracy of the original model.

Keywords: convolutional neural network, batch normalization, simulation input data, dynamic rounding

0 引言

卷积神经网络（CNN，convolutional neural network）模型在计算机视觉[1]、无人驾驶[2]等领域的高速发展和模型推理过程中所需巨大的内存占用及高能耗问题引起了人们的关注。CNN 量化指的

收稿日期：2021−12−18；**修回日期**：2022−03−18

通信作者：黄赟，yyhuangz@163.com

基金项目：国家自然科学基金资助项目（No.61521003）

Foundation Item: The National Natural Science Foundation of China (No.61521003)

是将原始模型中使用 32 位浮点表示的参数用更低比特位宽的参数来表示，由此来减小模型大小、提高运算效率，能够有效地解决 CNN 模型部署难的问题，成为神经网络加速领域的一个研究热点。模型量化面临着精度损失的难题，对此常用的量化方法通常使用数据集进行量化训练或微调来降低精度损失，达到了较好的量化效果。而对于一些存在数据敏感性问题或需要量化实时性的应用场景，如医疗、商业等领域，此时量化过程需要用到数据集的方法并不适用。

为解决数据敏感性场景下数据集不可用问题，本文利用神经网络批归一化（BN, batch normalization）[3]层参数来生成模拟输入数据，通过损失最小化量化方法动态调节量化缩放因子进行量化微调，不需要使用数据集也能获得较好的量化效果。

当前 CNN 模型大小、运算效率及能耗问题等成为制约其应用部署的主要因素，而随着存储设备的发展，模型的运算效率及能耗相比较而言显得更加突出。对于现场可编程门阵列（FPGA, field programmable gate array），相比于 32 位浮点运算，8 位整数运算的功耗仅为 $\frac{1}{30}$，面积仅为 $\frac{1}{116}$ [4]，从而显著提高计算吞吐量。对此本文采用 8 bit 位宽来量化 CNN 模型，能够有效减小模型大小；使用对数量化方法，使量化模型只含有整数乘法、加法及移位运算，能够有效提高硬件设备的运算效率、降低功耗损失，有利于神经网络模型实际部署应用。本文的主要工作如下。

1) 针对模型运算效率低及功耗大的问题，提出一种基于损失最小的对数量化方法。

2) 针对量化过程需要使用数据集的问题，提出一种模拟数据生成方法及激活无数据量化方法。

3) 针对所提量化方法，使用常用的图像分类模型及目标检测模型验证量化效果。

1　相关工作

根据量化发生的阶段不同，模型量化可分为两类：训练中参数量化和训练后参数量化。训练中参数量化指在模型训练过程中对其进行量化，其主要面临模型量化在反向传播过程中的梯度消失问题，对此可使用直通估计器来解决。文献[5]针对二值量化导致模型前向和反向传播过程产生严重的信息丢失问题，提出了一种信息保留网络（IR-Net, information retention network），通过保留前向传播和反向传播中的信息来训练二值量化模型。IR-Net 在反向传播过程中采用误差衰减估计器（EDE, error decay estimator）来计算梯度，通过更好地逼近符号函数来最小化信息损失。文献[6]提出了学习步长量化（LSQ, learned step size quantization）方法，将量化缩放因子设置为可训练的参数，使其在网络反向传播过程中进行学习调整；该研究通过一种简单的启发式方式，将缩放因子的更新和权重的更新保持平衡，得到更好的收敛精度。上述训练中参数量化方法都能有效降低量化损失，但实际中出于对隐私及数据安全性等问题考虑，训练数据存在无法访问使用的情况，此时训练中参数量化训练并不适用。虽然参数量化过程发生在模型训练好之后，但当前常用的训练后参数量化方法需要使用少量数据集来估算激活量化因子值或进行量化微调，同样面临着数据不可用问题。

为解决上述问题，面向无数据的训练后参数量化方法由此提出。无数据量化针对数据无法访问的情况及量化激活时需要使用输入数据这个矛盾，利用模型本身的一些参数特性来生成模拟数据，以此来量化激活。无数据量化的难点主要集中在模拟数据的生成上，其量化性能很大程度取决于模拟数据的质量，因而生成有意义的模拟输入数据至关重要。文献[7]首先生成符合高斯分布的随机输入数据，然后利用 BN 层中均值与方差参数来调节输入数据，再利用激活函数 ReLU 的线性变换缩放不变特性，将量化因子进行等价缩放来调节不同通道的权重范围，均衡同层间不同通道数据值，使逐层量化达到了逐通道量化的效果。文献[8]引入知识蒸馏的思想，通过输入蒸馏数据计算得来的 BN 层均值与方差与原始模型对应参数的差值误差最小化，不断在反向传播过程中调整蒸馏数据来得到模拟输入数据；该研究设计了一种基于帕累托边界的混合比特量化方法，进一步减小了量化误差。文献[9]使用对抗生成网络的思想来产生模拟输入数据，提出了一种零样本对抗量化（ZAQ, zero-shot adversarial quantization）框架，综合考虑了最终输出层差异及中间层通道间的量化差异，设计了一个基于两级差异的结构建模策略来衡量量化模型与原始模型之间的误差。ZAQ 中的生成器基于极大极小博弈优化思想，以对抗性学习的方式生成信息丰富且多样化的模拟输入数据，实现了有效的差异估计和知识转移。

上述方法虽然达到了无数据量化的目的，但并没

有充分研究利用真实的图像数据分布特性。同时上述方法使用参数最大值来直接计算得到浮点量化缩放因子，获得了较好的量化效果，但其量化及反量化过程中参数与量化因子的乘法操作依然为浮点运算，并没有达到完全消除浮点乘法的效果，存在改进的空间。对此本文基于 BN 层参数及图像数据分布特性来生成模拟输入数据，使用对数量化方法，使量化及反量化过程仅包含整数乘法、加法及移位运算，能够进一步提高硬件设备的运算效率、降低能耗损失。

2 损失最小化无数据量化策略

相较于采用浮点缩放因子的线性量化方法，对缩放因子取对数的线性量化方法能够解决线性量化含有浮点乘法算子的问题。但由于对数量化需对量化因子的指数部分进行取整操作，其舍入过程会产生较大的误差，因此本文使用损失最小化方法来动态调节取整值。

浮点模型各 BN 层参数在模型训练过程中由卷积层输出数据计算得到，与输入数据相关。因而使用逆向思维方式，通过输入随机分布数据得到的 BN 层参数，将其与原始模型参数相比较，构建损失函数，在反向传播过程中通过最小化损失调整模拟数据得到最优模拟输入数据，用来量化激活值。下面对其具体实现进行介绍。

2.1 对数量化方法

线性量化运算简单，有利于硬件设备实现，故常被各种量化方法[6-9]所采用。本文在线性量化的基础上，求量化因子时采用取对数的策略，能够消除量化过程中的浮点乘法运算，使模型在推理过程中更加高效。

$$x_{\mathrm{q}} = C(R(\Delta(x-z)),c) \tag{1}$$

其中，x 为被量化数据，x_{q} 为量化后的整数值；z 为量化零点值，本文取 $z=0$；Δ 为量化缩放因子；$R(\cdot)$为取整函数；c 为截断参数，$C(\cdot)$为截断函数，其计算方式为

$$C(x,c)=\begin{cases}-c, & x<-c\\ x, & -c\leqslant x\leqslant c\\ c, & x>c\end{cases} \tag{2}$$

量化缩放因子 Δ 的计算方式为

$$s = R\left(\mathrm{lb}(|x|_{\max})\right) \tag{3}$$

$$\Delta = 2^{n-s-1} \tag{4}$$

其中，n 为量化位宽，s 为 Δ 的指数部分值。因为 Δ 为 2 的指数形式，所以量化及反量化过程中 Δ 与 x 的浮点乘法运算可用移位运算来代替，而移位操作几乎不会带来额外的推理时间和存储消耗。量化之后通常需要反量化来还原之前的缩放尺寸，而对数量化能够将权重及激活的量化因子与下一层的量化因子相结合，简化了反量化操作。反量化的计算方式为

$$\begin{cases}x_{\mathrm{dq}} = \dfrac{x_{\mathrm{q}}}{\Delta_x} = \dfrac{x_{\mathrm{q}}}{2^{n_x-s_x-1}}\\ w_{\mathrm{dq}} = \dfrac{w_{\mathrm{q}}}{\Delta_w} = \dfrac{w_{\mathrm{q}}}{2^{n_w-s_w-1}}\end{cases} \Rightarrow x_{\mathrm{dq}} * w_{\mathrm{dq}} = \frac{x_{\mathrm{q}} * w_{\mathrm{q}}}{2^{n_x+n_w-s_x-s_w-2}} = 2^{s_x+s_w+2-n_x-n_w}\, x_{\mathrm{q}} * w_{\mathrm{q}} \tag{5}$$

其中，x_{dq} 和 w_{dq} 分别为反量化后的激活及权重值，因 $x_{\mathrm{dq}} * w_{\mathrm{dq}}$ 为整数值与 2 的指数形式参数相乘，可将其与下一卷积层量化因子相结合。因而本文所用方法能够使量化及反量化过程中只包含整数乘法、移位及加法等简单运算，能够进一步提高运算效率。

2.2 损失最小化量化

对数量化能够提高运算效率、降低能耗损失，但由式(3)可知，在计算量化缩放因子 Δ 时，对所求的因子 s 需要进行取整操作，而 s 为 Δ 的指数部分值，对其取整会导致 Δ 与原始最大值相差较大，最坏情况下会使其与原始最大值相差 $2^{\pm 0.5}$，这给量化操作带来更大的量化误差。文献[10]认为对量化缩放因子取整时，最优的缩放因子并不一定是离其最近的整数值。本文借用上述观点，在对 s 进行四舍五入取整的基础上动态对其进行微调，对比不同因子得到的量化前后数值，选取误差最小所对应因子 s 取值。

本文使用 L2 范数来衡量量化值与原始值之间的误差，其计算方式为

$$\mathrm{loss} = f_{\mathrm{L2}}\left(x, x_{\mathrm{dq}}\right) = \left\| x - x_{\mathrm{dq}} \right\|_2 \tag{6}$$

其中，x 为原始浮点数值；x_{dq} 为反量化后得到的数值，其为整数。

如式(7)~式(9)所示，首先通过四舍五入取整得到 s_0，然后得到量化缩放因子 Δ_0，代入式(1)得到量化值 x_{q} 及反量化值 x_{dq}，代入式(6)得到对应的损失值。

$$s_0 = \mathrm{round}(\mathrm{lb}(|x|_{\max})) \tag{7}$$

$$\Delta_0 = 2^{n-s_0-1} \tag{8}$$

$$\mathrm{loss}_0 = f_{\mathrm{L2}}\left(x, x_{\mathrm{dq}}\right) \tag{9}$$

构造相邻舍入因子，令

$$\begin{cases} s_1 = s_0 - 1 \\ s_2 = s_0 + 1 \end{cases} \tag{10}$$

同理可求得 s_1、s_2 对应的损失值 loss_1、loss_2，比较 loss_0、loss_1、loss_2 的大小，则最小的损失所对应的缩放因子为所求值，即

$$\text{loss}_p = \min\left(\text{loss}_0, \text{loss}_1, \text{loss}_2\right), p \in \{0,1,2\} \Rightarrow s = s_p \tag{11}$$

将 s 代入式(4)即可求得量化缩放因子 Δ。

2.3　无数据量化

当前 CNN 模型在卷积运算后通过使用 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，加快网络收敛速度。其计算方式为

$$y = \frac{\gamma(x-\mu)}{\sqrt{\sigma^2+\varepsilon}} + \beta \tag{12}$$

其中，γ 和 β 为模型训练过程中学习得到的参数，u 和 σ^2 分别为

$$\begin{aligned} \mu &= \frac{1}{m}\sum_{i=1}^{m} x_i \\ \sigma^2 &= \frac{1}{m}\sum_{i=1}^{m}\left(x_i-\mu\right)^2 \end{aligned} \tag{13}$$

由式(13)可以看出，BN 层的均值 μ 及方差 σ^2 皆由输入的激活值 x 计算得到。因而可以利用 BN 层的参数，通过对原始模型输入一组随机数据 X_0，在模型前向传播过程中算得相应新的参数值和 $\sigma^{2\prime}$，将其与原始模型的 μ 和 σ^2 相比较构建损失函数，在反向传播过程中使其损失最小化，不断调整初始输入数据 X_0 值，得到与真实数据相似的最优模拟输入数据。

对于均值 μ 和方差 σ^2 的实际值与模拟数据计算值之间的误差，构建相应的损失函数为

$$\begin{aligned} L &= \text{Loss}_{\mu}(X) + \text{Loss}_{\sigma^2}(X) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\mu_i - \mu_i'(X)\right)^2 + \frac{1}{n}\sum_{i=1}^{n}\left(\sigma^2{}_i - \sigma^2{}_i'(X)\right)^2 \end{aligned} \tag{14}$$

其中，n 为原始模型中 BN 层的数量；μ_i'、$\sigma^2{}_i'$ 为第 i 层 BN 层计算的参数值，是输入数据 X 的函数，通过将模拟数据 X=X_0 输入原始模型，经式(13)计算得到；μ_i、$\sigma^2{}_i$ 为其原始模型中的参数值。将损失 $L(X)$通过 Adam 算法在反向传播过程中不断更新优化初始 X_0，Adam 算法如算法 1 所示。

算法 1　Adam 算法

输入　α, β_1, β_2

初始化　$X_0, m_0 \to 0, v_0 \to 0, t \to 0$

1) while　X_t 未收敛　do

2)　$t \leftarrow t+1$

3)　$g_t \leftarrow \nabla_X L(X_t - 1)$

4)　$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$

5)　$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$

6)　$\hat{m}_t \leftarrow \dfrac{m_t}{(1-\beta_1^t)}$

7)　$\hat{v}_t \leftarrow \dfrac{v_t}{(1-\beta_2^t)}$

8)　$X_t \leftarrow X_{t-1} - \dfrac{\alpha \hat{m}_t}{(\sqrt{\hat{v}_t} + \varepsilon)}$

9) end while

算法 1 中，g_t 为目标损失函数 L 的梯度；m_t 和 v_t 为其偏一阶矩和偏二阶矩估计；β_1（取 $\beta_1 = 0.9$）和 β_2（取 $\beta_2 = 0.999$）为矩估计的指数衰减率；X_t 为待更新输入数据值；α（取 $\alpha = 0.001$）为学习率；ε（取 $\varepsilon = 1\times10^{-8}$）为极小的常数。通过 Adam 算法得到总模拟输入数据 X=X_t，然后使用得到的模拟输入数据及 2.2 节中提到的损失最小化量化方法就能够量化激活值，达到无数据量化的效果。

3　设计和实现

卷积神经网络模型量化主要针对卷积计算过程的激活和权重进行量化操作。对于激活的量化，本文利用 BN 层的参数，使用误差最小方法得到模拟输入数据，然后使用损失最小化量化方法对其进行量化。对于权重的量化，则直接使用损失最小化量化方法对其进行量化。下面对其具体实现进行介绍。

3.1　数据生成

任取两张 ImageNet 数据集中的图片，将其进行标准化、裁剪等数据预处理操作（数据维度为 3×224×224），之后将其按通道展平为一维数据（3 个一维数据，单个维度为 1×50 176），再将每一个数据组按步长 500 等距离取样得到相同位置的散点分布。图 1 展示了两张图片在 3 个不同通道的取样数据分布，实线为将离散数据采用 10 次多项式插值得到的拟合曲线。从图 1 中

可以看到，不同通道数据的拟合曲线有很强的相似性，本文认为这是因为每张图片的通道间相同位置数据分布有很强的相似性。因而在使用 2.3 节提出的模拟数据生成方法时，首先仅生成单一通道的随机分布数据，然后将其扩充为三通道数据，使之更加符合实际的数据分布特点。

$$x_0 = \alpha \frac{\mathrm{ri}(m,s) - 127}{128} \tag{15}$$

其中，$\mathrm{ri}(m,s)$ 为随机取整数函数；m 为最大值，对图像数据取 $m=255$；$s=(n,n)$ 为生成数据维度大小，如 Resnet[11]模型取 $n=224$。本文在对图像进行预处理时，标准化过程中使用均值 $\mathrm{mean}=[0.485,0.456,0.406]$，方差 $\mathrm{std}=[0.229,0.224,0.225]$，经过归一化及标准化处理后各维数据的最大值约为 $[2.249,2.429,2.624]$，因而在式(15)中将归一化的数据折中乘以系数 $\alpha=2.5$。把 x_0 扩充为三通道图像数据 $X_0=(x_0,x_0,x_0)$，将其输入原始模型计算得到 BN 层的参数 μ' 及 $\sigma^{2\prime}$，代入式(5)计算其相应的损失值，依据损失最小化在反向传播过程中不断更新初始值 X_0，得到模拟输入数据。

3.2　激活量化

使用 3.1 节生成的模拟输入数据，能够解决量化激活需要使用数据集的问题。通过将模拟数据 X_0 输入原始模型得到每层卷积层的激活数据值，使用 2.2 节提出的损失最小化量化方法量化激活值，可得到相应的激活量化参数。

图 2 展示了激活量化的大致过程。图 2 中，步骤 1 表示将一组随机分布的初始数据输入原始浮点模型 P，使用 3.1 节提出的数据生成方法得到模拟数据，利用 BN 层参数及损失最小化原理，通过 Adam(X_0,L)优化在反向传播不断调整模拟数据，得到最优的模拟输入数据 X_0。步骤 2 表示将 X_0 输入浮点模型得到激活值，使用 2.2 节提出的损失最小化量化方法通过量化函数 $Q(x_i)$量化激活值，得到相应的激活量化模型 Q。

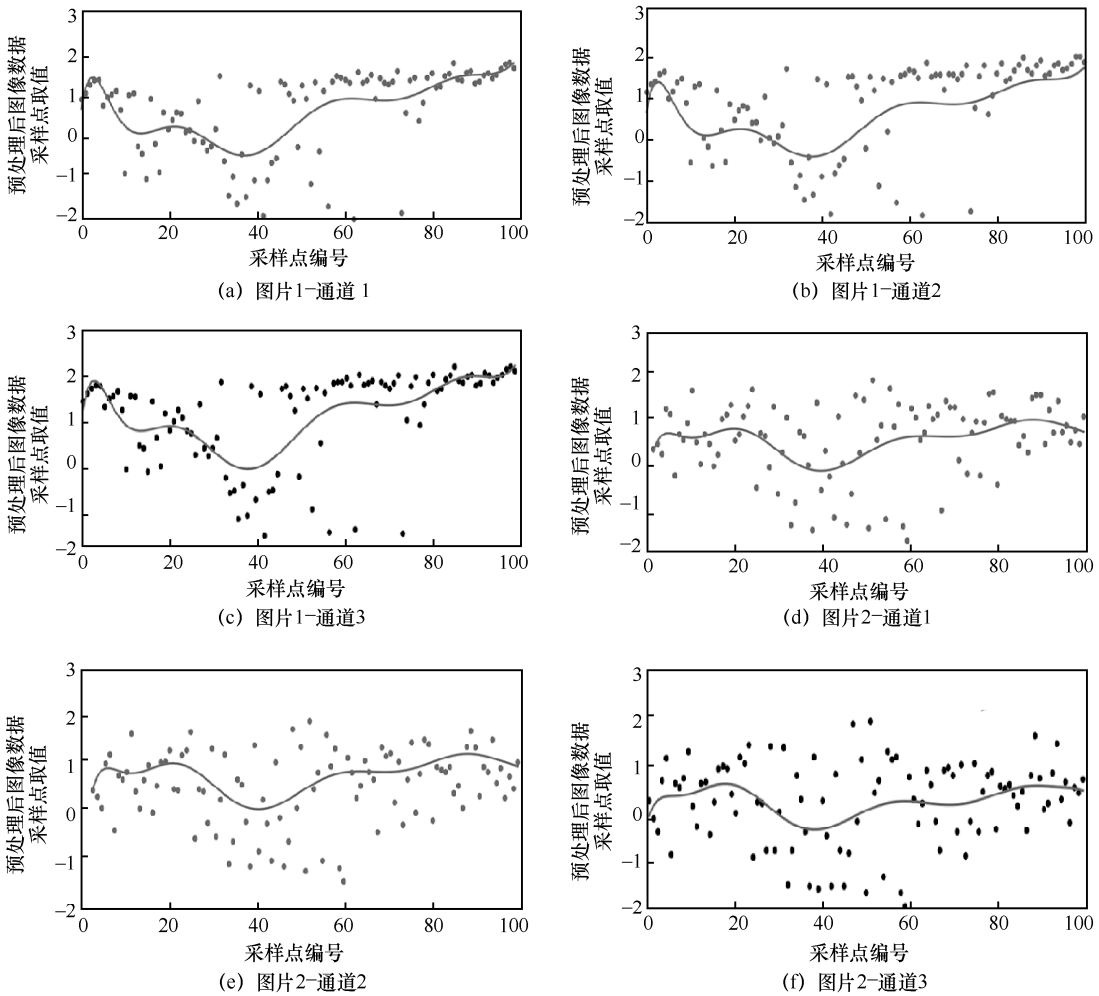


图 1　两张图片在 3 个不同通道的取样数据分布

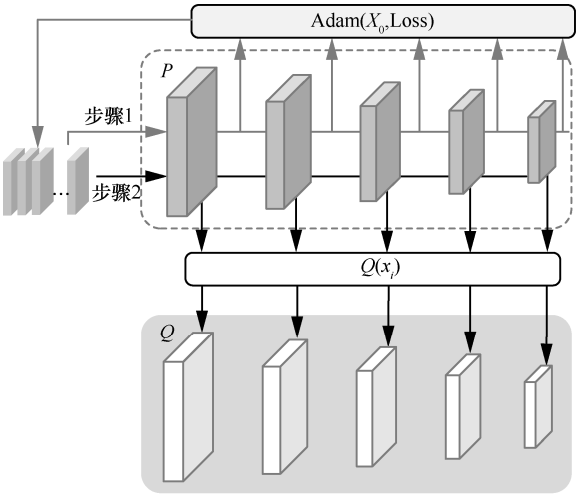


图 2　激活量化的大致过程

3.3　权重量化

本节采用2.2节提出的损失最小化量化方法进行权重量化。首先，对于原始浮点模型给出的权重，分别计算每层卷积权重对应的 s_0；然后，根据反量化后的值与原始权重误差最小化原理动态调整 s_0 得到更优的因子 s；最后，计算量化缩放因子 Δ，对权重进行量化。图 3 展示了损失最小化量化方法对 ResNet50 网络权重进行量化得到的初始因子 s_0 与调整后的因子 s 取值对比。从图 3 可以看出，对于 ResNet50 网络的权重量化，大概有 35%的点取值被调整。

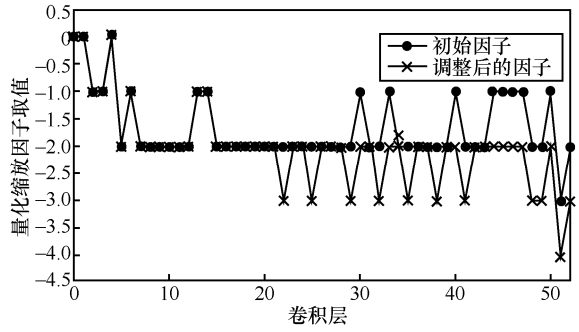


图 3　因子调整前后对比

4　实验与分析

本节使用本文所提量化方法，在 ImageNet[12] 数据集上验证了 ResNet50、Inception-v3[13]、RexNet3[14]等大型图像分类模型及 MobileNetV2[15]、RegNet[16]、GhostNet[17]等轻型图像分类模型的量化效果，在 COCO[18]数据集上验证了 RefineDet[19]和 M2Det[20]目标检测模型的量化效果。本节实验分为如下几个部分：1) 将本文所提模拟数据生成方法与 ZeroQ 框架模拟数据生成方法进行对比分析实验；2) 在 ImageNet 数据集上使用常用的图像分类模型进行量化实验，分析实验结果；3) 在 COCO 数据集上对 RefineDet 及 M2Det 目标检测模型进行量化实验，分析实验结果。

本文实验是在 Centos7.6 操作系统下进行的，使用的 GPU 为 NVIDIA Tesla V100 16 GB，实验运行的 Python 版本为 3.6.13，Pytorch 版本为 1.8.1，torchvision 版本为 0.8.2，pytorchcv 版本为 0.0.66。实验中对图像分类模型使用 top1 及 top5 准确率[13]来评估模型效果，对目标检测模型使用平均精准度（AP, average precision）[19]来评估模型效果。

4.1　数据生成对比实验

本节使用 ResNet50 图像分类网络，对 3.2 节所提的无数据激活量化方法与文献[9]中 ZeroQ 框架的激活量化方法进行对比实验。本节实验主要对比模拟数据生成效率，因而实验中对 ZeroQ 同样采用对数量化策略。因为实验结果依赖生成的模拟数据集，具有一定的随机性，所以使用多次实验取平均值的方式进行，实验结果如表 1 所示。从表 1 中可以看出，本文所提无数据激活量化方法相比于 ZeroQ 框架的激活量化方法对激活的量化效果，在模拟数据生成速率提高 2.33 倍的情况下，量化模型的准确率几乎相同。本节实验验证了本文所提方法在有效减少数据生成时间的同时达到了与 ZeroQ 相同的效果。

上述实验结果验证了 3.1 节所提依据数据分布特性所提出的模拟数据生成方式能够有效地加快模拟数据的生成，同时生成的模拟数据通过 Adam 优化调整而不影响其准确率。

4.2　图像分类模型量化实验

本节使用常用的图像分类模型，对激活采用 3.2 节所提的无数据激活量化方法，对权重采用 3.3 节所提的权重量化方法对网络进行量化实验。表 2 展示了 ResNet50、Inception-v3 及 RexNet3 等相对较大的图像分类模型，将其激活和权重量化为 8 bit 整数的实验结果，本节实验采用 3 次实验取平均值的方式展示实验结果。从表 2 中可以看到，本文所提方法将模型的激活和权重量化为 8 bit 整数后，其 top1 准确率下降在 0.5%左右，top5 准确率下降 0.2%左右，量化模型准确率和原始模型几乎差不多。将模型量化至 8 bit 能够减小 75%

表 1 模拟数据生成效率对比

实验次序	无数据激活量化方法			ZeroQ 框架的激活量化方法		
	模拟数据生成时间/μs	准确率		模拟数据生成时间/μs	准确率	
		top1	top5		top1	top5
1	160 935	76.02%	92.88%	362 118	75.94%	92.86%
2	110 693	75.98%	92.84%	364 473	76.02%	92.89%
3	103 880	76.00%	92.84%	382 389	76.01%	92.8%
4	99 332	75.98%	92.80%	393 344	76.01%	92.8%
5	153 992	75.99%	92.84%	359 681	76.00%	92.8%
6	82 687	76.00%	92.83%	398 909	76.03%	92.83%
7	90 528	76.03%	92.88%	382 489	76.04%	92.86%
8	89 355	75.98%	92.84%	407 976	76.01%	92.85%
9	157 200	76.06%	92.87%	386 953	75.95%	92.82%
10	95 509	76.02%	92.88%	369 122	76.00%	92.86%
平均	114 411.1	76.01%	92.85%	380 745.4	76.00%	92.85%

表 2 大型图像分类模型量化结果

模型	原始模型准确率		基于损失最小的无数据量化准确率			量化误差		模型大小		
	top1	top5	实验次序	top1	top5	top1	top5	量化前/MB	量化后/MB	压缩率
ResNet50	76.13%	92.86%	1	75.67%	92.73%	0.46%	0.13%	97.8	24.48	74.96%
			2	75.66%	92.73%	0.47%	0.13%			
			3	75.63%	92.74%	0.50%	0.12%			
			平均	75.653%	92.733%	0.477%	0.127%			
Inception-v3	78.83%	94.42%	1	78.49%	94.22%	0.34%	0.20%	91.3	22.87	74.95%
			2	78.51%	94.22%	0.32%	0.20%			
			3	78.55%	94.21%	0.28%	0.21%			
			平均	78.517%	94.217%	0.313%	0.203%			
RexNet3	82.63%	96.25%	1	82.28%	96.15%	0.35%	0.10%	132	33.07	74.95%
			2	82.21%	96.14%	0.42%	0.11%			
			3	82.21%	96.14%	0.42%	0.11%			
			平均	82.233%	96.143%	0.397%	0.107%			

左右模型大小，有效减少内存消耗、提高运算效率。

表 3 展示了对 MobileNetV2、RegNet 及 GhostNet 等轻量型图像分类模型的量化结果。表 3 中 W/A 分别表示权重及激活量化位宽，如 W8/A8 表示将权重和激活分别量化为 8 bit 整数。当将权重和激活量化至 8 bit 整数时，网络 top1 准确率下降了 1%～2%；当将其量化至 16 bit 整数，此时 top1 准确率最大仅减少了 0.32%，top5 准确率减少量小于 0.1%，达到了和原始浮点模型几乎相同的精度。将权重和激活量化至 16 bit 能够减小模型大小 50% 左右，因而本文所提方法对轻量型图像分类模型依然有压缩效果。

本节通过量化实验验证了所提方法对图像分类模型压缩加速的有效性。图像分类任务作为计算机视觉领域的基础性任务之一，分类模型不仅用于图像分类任务，而且常用于图像目标检测、实例分割等模型的骨干网络使用，因而所提方法对目标检测等模型依然有效。

4.3 图像目标检测模型量化实验

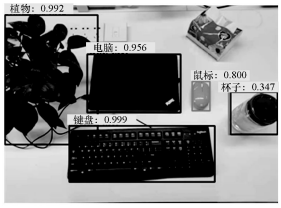
本节使用所提方法，对目标检测模型 RefineDet

表 3　　**轻量型图像分类模型的量化结果**

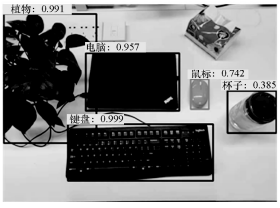
模型	原始模型准确率		基于损失最小的无数据量化准确率			量化误差	
	top1	top5	权重/位宽	top1	top5	top1	top5
MobileNetV2	73.03%	91.13%	W8/A8	71.96%	90.57%	1.07%	0.56%
			W16/A16	73.03%	91.14%	0%	−0.01%
RegNet	69.85%	89.37%	W8/A8	67.61%	88.30%	2.24%	1.07%
			W16/A16	69.81%	89.35%	0.04%	0.02%
GhostNet	73.98%	91.46%	W8/A8	73.02%	91.14%	0.96%	0.32%
			W16/A16	73.66%	91.39%	0.32%	0.07%

和 M2Det 在 COCO 2014 验证集进行了量化对比实验。表 4 展示了将模型的激活采用 3.2 节所提的无数据激活量化方法，权重采用 3.3 节所提的权重量化方法，分别将其量化至 8 bit 整数的量化效果。从表 4 中可以看出，相较于原始模型，量化模型的 RefineDet 的 $AP_{50:95}$ 量化误差约为 1.0%，M2Det 模型的 $AP_{50:95}$ 误差约为 0.6%，量化模型的误差值都控制在 1.0%左右的可接受范围内。图 4 展示了 M2Det 模型对任意一张图片量化前后目标检测实际效果对比情况。图 4(b)和图 4(c)分别为原始浮点模型及 8 bit 量化模型检测效果，从图 4 中可以看出，相对而言，仅鼠标检测的置信度有较大下降，但这几乎不影响对物体的检测框，对于大部分物体其置信度并不受影响。

(a) 原始图片



(b) 原始浮点模型检测效果



(c) 8 bit量化模型检测效果

图 4　M2Det 模型量化效果对比

对于上述实验结果，本文认为首先所提模拟数据生成方法生成的模拟输入数据达到了与真实数据相近似的分布。其次，虽然对数量化会引入较大的量化误差，但所提损失最小化量化方法能够有效减小量化误差，进一步提高量化精度。

5　结束语

针对数据集不可用场景下 CNN 量化问题，本文基于 BN 层参数提出一种基于损失最小的 CNN 无数据量化方法。本文依据图像各通道数据分布特性，利用 BN 层参数及最小化误差的方法，生成模拟数据用来量化激活值。在对量化因子取整的操作中，提出基于最小化不同取整值与原始浮点数值间误差进行选择性取整的方法，选取最接近原始值的舍入值，有效地降低了量化损失。通过对常用图像分类及目标检测模型进行量化对比实验，尤其对轻量型分类模型进行了大量实验，验证了所提方法的有效性。本文使用的对数量化方法能够消除量化及反量化过程中的浮点乘法运算，进一步提高模型运算效率、降低功耗损失，有助于 CNN 模型在手机等运算受限设备及云端部署应用。下一步考虑将所提方法与模型剪枝等压缩方式相结合，进一步减小模型大小。

参考文献：

[1] 郭璠, 张泳祥, 唐琎, 等. YOLOv3-A: 基于注意力机制的交通标志检测网络[J]. 通信学报, 2021, 42(1): 87-99.
GUO F, ZHANG Y X, TANG J, et al. YOLOv3-A: a traffic sign detection network based on attention mechanism[J]. Journal on Communi-

表 4　　**RefineDet 和 M2Det 模型量化结果**

模型	原始模型平均准确率			量化模型平均准确率			量化误差		
	$AP_{50:95}$	AP_{50}	AP_{75}	$AP_{50:95}$	AP_{50}	AP_{75}	$AP_{50:95}$	AP_{50}	AP_{75}
RefineDet_ResNet101_512	44.9%	67.1%	50.1%	43.9%	66.2%	49.0%	1.0%	0.9%	1.1%
M2Det_Vgg16_512	48.9%	68.2%	55.0%	48.3%	68.0%	54.5%	0.6%	0.2%	0.5%

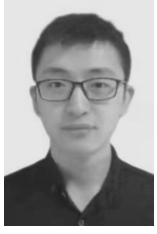
cations, 2021, 42(1): 87-99.
[2] 黄志清, 曲志伟, 张吉, 等. 基于深度强化学习的端到端无人驾驶决策[J]. 电子学报, 2020, 48(9): 1711-1719.
HUANG Z Q, QU Z W, ZHANG J, et al. End-to-end autonomous driving decision based on deep reinforcement learning[J]. Acta Electronica Sinica, 2020, 48(9): 1711-1719.
[3] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift[C]//Proceedings of the 32nd International Conference on International Conference on Machine Learning. [S.l.]: JMLR.org, 2015: 448-456.
[4] CHOUKROUN Y, KRAVCHIK E, YANG F, et al. Low-bit quantization of neural networks for efficient inference[C]//Proceedings of 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). Piscataway: IEEE Press, 2019: 3009-3018.
[5] QIN H T, GONG R H, LIU X L, et al. Forward and backward information retention for accurate binary neural networks[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 2247-2256.
[6] ESSER S K, MCKINSTRY J L, BABLANI D, et al. Learned step size quantization[J]. arxiv Preprint, arxiv: 1902.08153, 2019.
[7] NAGEL M, BAALEN M V, BLANKEVOORT T, et al. Data-free quantization through weight equalization and bias correction[C]//Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2019: 1325-1334.
[8] LIU Y A, ZHANG W, WANG J. Zero-shot adversarial quantization[C]//Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2021: 1512-1521.
[9] CAI Y H, YAO Z W, DONG Z, et al. ZeroQ: a novel zero shot quantization framework[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 13166-13175.
[10] NAGEL M, AMJAD R A, BAALEN M V, et al. Up or down? adaptive rounding for post-training quantization[C]//Proceedings of 2020 International Conference on Machine Learning. New York: ACM Press, 2020: 7197-7206.
[11] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2016: 770-778.
[12] DENG J, DONG W, SOCHER R, et al. ImageNet: a large-scale hierarchical image database[C]//Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2009: 248-255.
[13] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2016: 2818-2826.
[14] HAN D, YUN S, HEO B, et al. Rethinking channel dimensions for efficient model design[C]//Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2021: 732-741.
[15] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 4510-4520.
[16] RADOSAVOVIC I, KOSARAJU R P, GIRSHICK R, et al. Designing network design spaces[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 10425-10433.
[17] HAN K, WANG Y H, TIAN Q, et al. GhostNet: more features from cheap operations[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 1577-1586.
[18] LIN T Y, MAIRE M, BELONGIE S J, et al. Microsoft COCO: common objects in context[C]//Proceedings of 2014 European Conference on Computer Vision. Berlin: Springer, 2014: 740-755.
[19] ZHANG S F, WEN L Y, BIAN X, et al. Single-shot refinement neural network for object detection[C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 4203-4212.
[20] ZHAO Q J, SHENG T, WANG Y T, et al. M2Det: a single-shot object detector based on multi-level feature pyramid network[C]//Proceedings of the AAAI Conference on Artificial Intelligence. New York: ACM Press, 2019: 9259-9266.

[作者简介]

张帆（1981− ），男，河南郑州人，博士，国家数字交换系统工程技术研究中心副研究员，主要研究方向为主动防御、人工智能等。

黄赟（1993− ），男，江西新余人，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、网络内生安全等。

方子茁（1997− ），男，河南郑州人，东南大学硕士生，主要研究方向为网络内生安全、数据库安全、人工智能安全等。



郭威（1990− ），男，北京人，博士，国家数字交换系统工程技术研究中心副研究员，主要研究方向为主动防御、人工智能安全等。