doi:10.11887/j.issn.1001-2486.25010015 http://journal.nudt.edu.cn

考虑层敏感性的卷积神经网络混合精度量化方法

刘海军,张晨曦,王析羽,陈长林,陈 军,李智炜*
(国防科技大学 电子科学学院,湖南 长沙 410073)

摘 要:针对如何将神经网络保真映射到资源受限的嵌入式设备这一问题,提出基于层敏感性分析的卷积神经网络混合精度量化方法。通过计算Hessian矩阵平均迹衡量卷积层参数的敏感性,为位宽分配提供依据;使用逐层升降方法进行位宽分配,最终完成网络模型的混合精度量化。实验结果表明,与DoReFa和LSQ+两种固定精度量化方法相比,所提出的混合精度量化方法在平均位宽为3 bit的情况下将识别准确率提高了10.2%和1.7%;与其他混合精度量化方法相比,所提方法识别准确率提高了1%以上。此外,加噪训练能够有效提高混合精度量化方法的鲁棒性,在噪声标准差为0.5的情况下,将识别准确率提高了16%。

关键词:卷积神经网络;模型量化;人工智能;混合精度
中图分类号:TP183 **文献标志码**:A **文章编号**:1001-2486(2025)04-143-08

论文拓展

Convolutional neural network mixed-precision quantization method considering layer sensitivity

*LIU Haijun, ZHANG Chenxi, WANG Xiyu, CHEN Changlin, CHEN Jun, LI Zhiwei**
(College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China)

Abstract: To address the problem of how to faithfully map neural networks to resource-constrained embedded devices, a mixed-precision quantization method for convolutional neural networks based on layer sensitivity analysis was proposed. The sensitivity of convolutional layer parameters was measured by calculating the average trace of the Hessian matrix, providing a basis for bit-width allocation. A layer-wise ascending-descending approach was employed for bit-width allocation, ultimately achieving mixed-precision quantization of the network model. Experimental results demonstrate that compared to the fixed-precision quantization methods DoReFa and LSQ+, the proposed mixed-precision quantization method improves recognition accuracy by 10.2% and 1.7%, respectively, at an average bit-width of 3 bit. When compared to other mixed-precision quantization methods, the proposed approach achieves over 1% higher recognition accuracy. Additionally, noise-injected training effectively enhances the robustness of the mixed-precision quantization method, improving recognition accuracy by 16% under a noise standard deviation of 0.5.

Keywords: convolutional neural network; model quantization; artificial intelligence; mixed precision

随着智能计算技术的不断发展,神经网络被广泛应用于各种任务中,如图像分类[1-4]、目标检测[5-7]和图像分割[8]等,并取得了非常好的效果。出于实时性、隐私性和安全性的考虑,神经网络越来越多地被部署于各式各样的边缘计算设备中。相较于计算机,这些边缘计算设备的存储空间和计算能力有限,直接承载神经网络往往存在一定的困难,因此,迫切需要开展神经网络压缩研究[9-13]。

量化作为一种高效的神经网络压缩方法,能够通过降低位宽来减少使用的计算和存储资源[14-16],从而允许在硬件实现中使用更低精度的计算单元。很多研究学者采用相同的位宽量化网络的权值与激活值[17-19],即采取固定精度量化方法。这种方法硬件实现简单,但由于没有考虑到网络模型各层量化产生的误差差异,无法充分利

收稿日期:2025-01-10
基金项目:国家自然科学基金资助项目(62074166,62304254,62104256,62404253,U23A20322)
第一作者:刘海军(1982—),男,山东德州人,副教授,博士,硕士生导师,E-mail:liuhaijun@nudt.edu.cn
*通信作者:李智炜(1989—),男,江西南昌人,副研究员,博士,硕士生导师,E-mail:lizhiwei@nudt.edu.cn
引用格式:刘海军,张晨曦,王析羽,等. 考虑层敏感性的卷积神经网络混合精度量化方法[J]. 国防科技大学学报,2025,47(4):143-150.
Citation: LIU H J, ZHANG C X, WANG X Y, et al. Convolutional neural network mixed-precision quantization method considering layer sensitivity[J]. Journal of National University of Defense Technology, 2025, 47(4): 143-150.

用有限的硬件资源。此外,还有混合精度量化方法,该方法能为各层分配不同的位宽,能够充分利用存储与计算资源,获得更好的性能表现。当前,混合精度量化方法面临的主要问题是如何为网络各层分配合适的位宽。基于搜索的方法通常通过拓扑结构搜索、网络单元搜索等方式来确定混合精度量化的策略。Wang 等提出在全精度模型和量化模型之间保持输入图像的属性排名,从而搜索可迁移的混合精度量化策略[20]。然而,该方法忽略了丰富的类别级特征。为了解决这个问题,Tang 等提出了分离正则化方法,以搜索量化策略,使类别级特征对量化噪声更具鲁棒性[21]。尽管这些方法提高了搜索效率,但将基于搜索的方法推广到嵌入式设备部署仍然面临着搜索时间过长等挑战。基于度量的方法则是通过采用特定的度量策略来确定每一层的重要性或敏感性,进而相应地分配位宽。Ma 等使用层正交性构造线性规划问题,并推导出位宽分配策略[16]。Tang 等通过一次量化感知训练来构建整数线性规划,进而分配位宽,显著减少了搜索时间[22]。Liu 等对低锐度的层使用了更低的位宽[15],在较低平均位宽时取得了较好的性能表现。上述方法在模型量化方面取得了很好的效果,但在识别准确率、位宽搜索效率等方面还有较大的改进空间。

针对上述问题,提出基于层敏感性分析的卷积神经网络混合精度量化方法,力图通过制定更合理的位宽分配策略,获得更好的性能表现,同时提高位宽搜索效率。

1 基于层敏感性分析的混合精度量化方法步骤

基于层敏感性分析的混合精度量化方法主要包括以下 3 个步骤。

步骤 1:基于 Hessian 矩阵平均迹的层敏感性分析。计算卷积神经网络各个卷积层参数的 Hessian 矩阵平均迹,将其作为各个卷积层敏感性的衡量指标,为位宽分配提供依据。

步骤 2:基于逐层升降的位宽分配策略。根据前一步得到的各卷积层的敏感性,使用逐层升降的方法进行位宽分配,确定各个卷积层的位宽。

步骤 3:基于量化感知训练的混合精度量化方法。根据前一步确定的各卷积层的位宽,基于量化感知训练完成网络模型的混合精度量化。

1.1 基于 Hessian 矩阵平均迹的层敏感性分析

1.1.1 理论依据

文献[23]论证了利用权值的 Hessian 矩阵平均迹作为层敏感性衡量指标的可行性,在量化方面取得了非常好的效果。本节在此基础上,将 Hessian 矩阵平均迹拓展为同时包容权值、激活值的情况。具体为:将卷积神经网络各层的权值 $\boldsymbol{W}_i(i=1,2,\cdots,m)$、激活值 $\boldsymbol{A}_i(i=1,2,\cdots,n)$用参数 $\boldsymbol{\theta}_i(i=1,2,\cdots,m+n)$统一表示。层敏感性大小定义为:参数受到同等幅度的改变后损失函数值的大小,即

$$L(\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2^*,\cdots,\boldsymbol{\theta}_i^*+\Delta\boldsymbol{\theta}_i^*,\cdots,\boldsymbol{\theta}_N^*) \tag{1}$$

式中,L 为损失函数,$\boldsymbol{\theta}_i^*$ 为网络收敛时第 i 组参数,$\Delta\boldsymbol{\theta}_i^*$ 为参数变化量。为了便于说明,后续用 $L(\boldsymbol{\theta}_i^*+\Delta\boldsymbol{\theta}_i^*)$表示式(1)。采用损失函数泰勒级数展开的方式计算损失函数,如式(2)所示。

$$L(\boldsymbol{\theta}_i^*+\Delta\boldsymbol{\theta}_i^*)=L(\boldsymbol{\theta}_i^*)+\boldsymbol{g}_i^{\mathrm{T}}\Delta\boldsymbol{\theta}_i^*+\frac{1}{2}\Delta\boldsymbol{\theta}_i^{*\mathrm{T}}\boldsymbol{H}_i\Delta\boldsymbol{\theta}_i^* \tag{2}$$

式中:$\boldsymbol{g}_i^{\mathrm{T}}$ 是第 i 组参数的梯度,由于网络收敛,所以存在 $\boldsymbol{g}_i^{\mathrm{T}}=\boldsymbol{0}$;$L(\boldsymbol{\theta}_i^*)$为网络收敛时的损失函数,即 $L(\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2^*,\cdots,\boldsymbol{\theta}_i^*,\cdots,\boldsymbol{\theta}_N^*)$;$\boldsymbol{H}_i$ 是第 i 组参数的 Hessian 矩阵。为了便于比较第 i、第 j 层参数的层敏感性,对二者受到同等幅度改变后的损失函数作差,需要注意的是,在参数未改变的情况下,存在 $L(\boldsymbol{\theta}_i^*)=L(\boldsymbol{\theta}_1^*,\boldsymbol{\theta}_2^*,\cdots,\boldsymbol{\theta}_i^*,\cdots,\boldsymbol{\theta}_N^*)=L(\boldsymbol{\theta}_j^*)$,则

$$\begin{aligned}&L(\boldsymbol{\theta}_i^*+\Delta\boldsymbol{\theta}_i^*)-L(\boldsymbol{\theta}_j^*+\Delta\boldsymbol{\theta}_j^*)\\&=\frac{1}{2}(\Delta\boldsymbol{\theta}_i^{*\mathrm{T}}\boldsymbol{H}_i\Delta\boldsymbol{\theta}_i^*-\Delta\boldsymbol{\theta}_j^{*\mathrm{T}}\boldsymbol{H}_j\Delta\boldsymbol{\theta}_j^*)\end{aligned} \tag{3}$$

$\Delta\boldsymbol{\theta}_i^*$ 可以表示为:

$$\Delta\boldsymbol{\theta}_i^*=\alpha_{\mathrm{bit},i}\sum_{k=1}^{n_i}\boldsymbol{v}_k^i \tag{4}$$

式中,n_i 是 $\boldsymbol{\theta}_i$ 的维数,$\alpha_{\mathrm{bit},i}$是一个与位宽 bit 及量化区间有关的常量,$\boldsymbol{v}_k^i$ 是第 i 组 Hessian 矩阵的第 k 维特征正交单位向量。则

$$\Delta\boldsymbol{\theta}_i^{*\mathrm{T}}\boldsymbol{H}_i\Delta\boldsymbol{\theta}_i^*=\sum_{k=1}^{n_i}\alpha_{\mathrm{bit},i}^2\boldsymbol{v}_k^{i\mathrm{T}}\boldsymbol{H}_i\boldsymbol{v}_k^i=\alpha_{\mathrm{bit},i}^2\sum_{k=1}^{n_i}\lambda_k^i \tag{5}$$

式中,λ_k^i 表示与 $\boldsymbol{v}_k^i$ 对应的特征值。假设 $\|\Delta\boldsymbol{\theta}_i^*\|=\|\Delta\boldsymbol{\theta}_j^*\|$,即$\sqrt{n_i}\alpha_{\mathrm{bit},i}=\sqrt{n_j}\alpha_{\mathrm{bit},j}$,所以

$$\begin{aligned}&L(\boldsymbol{\theta}_i^*+\Delta\boldsymbol{\theta}_i^*)-L(\boldsymbol{\theta}_j^*+\Delta\boldsymbol{\theta}_j^*)=\\&\frac{1}{2}\alpha_{\mathrm{bit},j}^2n_j\left(\frac{1}{n_i}\sum_{k=1}^{n_i}\lambda_k^i-\frac{1}{n_j}\sum_{k=1}^{n_j}\lambda_k^j\right)<0\end{aligned} \tag{6}$$

而

$$\frac{1}{n_i}\sum_{k=1}^{n_i}\lambda_k^i=\frac{1}{n_i}\mathrm{tr}(\boldsymbol{H}_i) \tag{7}$$

式中,$\mathrm{tr}(\boldsymbol{H}_i)$表示 Hessian 矩阵 $\boldsymbol{H}_i$ 的迹。因此,可

以利用 Hessian 矩阵的平均迹作为层敏感性的衡量指标。

1.1.2　Hessian 矩阵平均迹的计算方法

由于 Hessian 矩阵的元素数量是参数量的平方,通过直接计算每一层的 Hessian 矩阵来获取 Hessian 矩阵的平均迹,计算代价和存储代价较大。可以通过计算 Hessian 矩阵与某一向量乘积的方式来实现,具体计算过程为:

$$\boldsymbol{H}_i\boldsymbol{z}=\frac{\mathrm{d}\boldsymbol{g}_i^{\mathrm{T}}}{\mathrm{d}\boldsymbol{\theta}_i^*}\boldsymbol{z}=\frac{\mathrm{d}\boldsymbol{g}_i^{\mathrm{T}}}{\mathrm{d}\boldsymbol{\theta}_i^*}\boldsymbol{z}+\boldsymbol{g}_i^{\mathrm{T}}\frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}\boldsymbol{\theta}_i^*}=\frac{\mathrm{d}\boldsymbol{g}_i^{\mathrm{T}}\boldsymbol{z}}{\mathrm{d}\boldsymbol{\theta}_i^*} \tag{8}$$

式中,$\boldsymbol{z}$ 是与 $\boldsymbol{\theta}_i^*$ 无关的随机向量,满足高斯分布 $N(0,1)$。根据 Hutchinson 算法[24],则

$$\begin{aligned}\mathrm{tr}(\boldsymbol{H}_i)&=\mathrm{tr}(\boldsymbol{H}_i\boldsymbol{I})=\mathrm{tr}(\boldsymbol{H}_iE[\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}])\\&=E[\mathrm{tr}(\boldsymbol{H}_i\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}})]=E[\boldsymbol{z}^{\mathrm{T}}\boldsymbol{H}_i\boldsymbol{z}]\end{aligned} \tag{9}$$

式中,$E[\cdot]$表示数学期望。如果给定 n 个向量,那么就能够用统计的方法估计迹的值,即

$$\mathrm{tr}(\boldsymbol{H}_i)\approx\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{z}_j^{\mathrm{T}}\boldsymbol{H}\boldsymbol{z}_j \tag{10}$$

而 Hessian 矩阵平均迹的计算公式为:

$$\overline{\mathrm{tr}}(\boldsymbol{H}_i)=\frac{1}{m_i}\mathrm{tr}(\boldsymbol{H}_i) \tag{11}$$

其中,$\overline{\mathrm{tr}}(\boldsymbol{H}_i)$表示 Hessian 矩阵 $\boldsymbol{H}_i$ 的平均迹,m_i 表示每层权值矩阵的元素总数。

1.2　基于逐层升降的位宽分配策略

1.2.1　位宽分配问题建模

在实施量化之前,首先需要解决的主要问题为:如何为每一层分配合适的位宽,使得在特定平均位宽情况下损失函数值最小。位宽分配问题模型如图 1 所示。

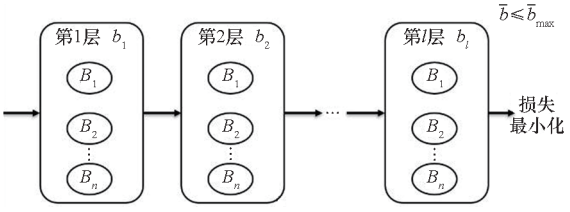


图 1　位宽分配问题模型

Fig. 1　Bit-width allocation problem model

图 1 中,卷积神经网络所有层位宽组成的集合为 $b=\{b_1,b_2,\cdots,b_i,\cdots,b_l\}$,$b_i$ 表示第 i 层的位宽,$1\leqslant i\leqslant l$,l 表示卷积神经网络的总层数;每一层的位宽选择空间均为 $B=\{B_1,B_2,\cdots,B_n\}$,n 表示位宽的种类数。用 $\bar{b}$ 表示平均位宽,假设平均位宽的上限值为 $\bar{b}_{\max}$,则 $\bar{b}\leqslant\bar{b}_{\max}$。通常情况下,$\bar{b}$ 越接近于 $\bar{b}_{\max}$,量化带来的损失越小。位宽分配问题可以表示为:

$$b=\mathrm{argmin}(\bar{b}_{\max}-\bar{b})\quad b_i\in B,i\in\{1,2,\cdots,l\} \tag{12}$$

$$\bar{b}=\frac{\sum_{i=1}^{l}m_ib_i}{\sum_{i=1}^{l}m_i} \tag{13}$$

其中,m_i 表示每层权值矩阵的元素总数。因为每一层的位宽搜索空间大小为 n 种,完成整个卷积神经网络量化所需的搜索空间大小就有 n^l 种,直接对式(12)进行求解需要花费大量的时间。为此,提出了基于逐层升降的位宽分配方法。

1.2.2　位宽确定算法

根据上一节的层敏感性分析方法可以得到卷积神经网络所有卷积层的敏感性。对于敏感性较强的卷积层,位宽应该设置一个较大的值,避免损失过大;对于敏感性较弱的卷积层,位宽应该设置一个较小的值,以增加模型的压缩率。利用式(11)计算获得各层 Hessian 矩阵的平均迹$\overline{\mathrm{tr}}(\boldsymbol{H}_i)$,作为层敏感性的衡量指标,Hessian 矩阵平均迹越大,则层敏感性越强,对应层的位宽排序越靠后;反之,Hessian 矩阵平均迹越小,则层敏感性越弱,对应层的位宽排序越靠前。按照 Hessian 矩阵平均迹的大小将对应的层位宽重新排列为:

$$b=\{b_k\in B,k=1,2,\cdots,l\,|\,b_i\leqslant b_{i+1},i=1,2,\cdots,l-1\} \tag{14}$$

具体如算法 1 所示。首先,将所有层的位宽都设置为位宽搜索空间的平均数。当 $\bar{b}$ 不超过 $\bar{b}_{\max}$时,使敏感性最强的层位宽加 1;当 $\bar{b}$ 超过 $\bar{b}_{\max}$时,使敏感性最弱的层位宽减 1。当敏感性最强的层位宽达到位宽搜索空间 B 的最大值 B_n 时,将其排除,转而对敏感性次强的层位宽进行调整;当敏感性最弱的层位宽达到位宽搜索空间 B 的最小值 B_1 时,将其排除,转而对敏感性次弱的层位宽进行调整。当所有层位宽都调整完毕后,迭代次数 e 加 1,直到迭代次数 e 达到最大值,则迭代结束。位宽搜索期间,将持续记录 $\bar{b}$ 与 $\bar{b}_{\max}$相差最小时 b 的结果。

随着迭代次数的增加,b 将会趋于两极分化,即一部分值为位宽搜索空间 B 的最小值 B_1,另一部分值为位宽搜索空间 B 的最大值 B_n,此时若继续进行迭代,则会在某个中间层的位宽产生来回震荡,但因为记录的是 $\bar{b}$ 与 $\bar{b}_{\max}$相差最小时 b 的结果,所以震荡并不会影响算法的收敛性。

算法 1 逐层升降的位宽确定算法

Alg. 1 Layer-by-layer bit width adjustment algorithm

输入:位宽选择空间 B

输出:卷积神经网络各层位宽集合 b

初始化 $b=\text{floor}(\bar{B})$
初始化误差 $\delta=\bar{b}_{\max}-\bar{b}$
迭代次数 $e=0$
升指示器 $i=0$
降指示器 $j=0$
while $e<10$
  if $\bar{b}\leqslant\bar{b}_{\max}$
    if $b_{l-i}<\max(B)$
      $b_{l-i}=b_{l-i}+1$
    end if
    $i=i+1$
  else
    if $b_{j+1}>\min(B)$
      $b_{j+1}=b_{j+1}-1$
    end if
    $j=j+1$
  end if
  if $\bar{b}\leqslant\bar{b}_{\max}$
    if $\delta>\bar{b}_{\max}-\bar{b}$
      $\delta=\bar{b}_{\max}-\bar{b}$
    end if
  end if
if $i+j==l$
  $e=e+1$
  $i=0$
  $j=0$
end if

迭代结束后,就确定了基于层敏感性分析的卷积神经网络所有卷积层的位宽。

1.3 基于量化感知训练的混合精度量化方法

固定精度量化主要实现层内权值和激活值的量化,而混合精度量化在此基础上,还要实现层间位宽分配功能。因此,混合精度量化方法可以在固定精度量化方法的基础上实现。

基于量化感知训练的混合精度量化方法流程主要包括 6 个步骤。

步骤 1:加载预训练模型。

步骤 2:计算 Hessian 矩阵平均迹 $\overline{\text{tr}}(\boldsymbol{H}_i)$,获得网络各层敏感性。

步骤 3:依据敏感性排序,使用逐层升降的方法确定各层位宽,获得位宽集合 b。

步骤 4:各层按照确定的位宽进行权值与激活值量化。

步骤 5:进行量化感知训练,在训练中将误差逐渐降低。

步骤 6:训练结束,得到混合精度量化的网络模型。需要说明的是:第五步的量化感知训练中,在预训练权值的基础上,计算各卷积层 Hessian 矩阵的平均迹 $\overline{\text{tr}}(\boldsymbol{H}_i)$ 以衡量敏感性,基于层敏感性使用前面的位宽分配方法迭代 e 次以获得各层位宽,对各个卷积层进行量化与反量化,使模型在训练过程中逐渐感知并适应量化带来的误差,逐步降低量化产生的性能损失,提高识别准确率。

2 实验结果

2.1 实验设置

2.1.1 参数设置

本实验基于 PyTorch 开源库进行,使用的卷积神经网络模型为 ResNet18。数据集为 CIFAR-10,该数据集包含 10 个类别共 60 000 张 32 像素×32 像素的图片,其中 50 000 张用于训练,10 000张用于测试。量化感知训练初始学习率设置为 0.005,并随着训练轮次增加逐渐减小;训练轮次设置为 120;权重衰减系数设置为 5×10^{-4}。

2.1.2 指标定义

1)准确率。假设预测正确的目标数量为 R_{N},测试集所有目标的数量为 T_{N},则准确率 A_{c} 为

$$A_{\text{c}}=\frac{R_{\text{N}}}{T_{\text{N}}} \tag{15}$$

2)搜索效率。采用位宽搜索过程中的加法次数和乘法次数来衡量搜索效率,在保证识别准确率的情况下,加法次数和乘法次数越少,则方法的搜索效率越高。假设卷积神经网络有 l 个卷积层,每个卷积层的备选位宽有 m 种,第 i 个卷积层的参数个数为 N_i,通道数为 c_i,总训练轮次为 e,训练集数据个数为 t。则不同算法的搜索效率如下:

海森感知量化[25](Hessian aware quantization, HAWQ)在位宽搜索部分的总加法次数 A_{n} 和总乘法次数 M_{n} 分别为

$$\begin{cases}A_{\text{n}}=\sum\limits_{j=1}^{m}\text{C}_m^j\text{C}_{l-1}^{j-1}\left[l-1+\sum\limits_{i=1}^{l}(N_i-1)\right]\\ M_{\text{n}}=\sum\limits_{j=1}^{m}\text{C}_m^j\text{C}_{l-1}^{j-1}\sum\limits_{i=1}^{l}(N_i+1)\end{cases} \tag{16}$$

其中,C_*^* 表示组合数。

数据质量感知混合精度量化[26](data quality-aware mixed-precision quantization, DQMQ)在位宽

搜索部分的总加法次数 A_n 和总乘法次数 M_n 分别为

$$\begin{cases} A_n = et\sum_{i=1}^{l}(N_i + 9c_i + 90) \\ M_n = et\sum_{i=1}^{l}(12c_i + 100) \end{cases} \tag{17}$$

所提方法在位宽搜索部分的总加法次数 A_n 和总乘法次数 M_n 分别为

$$\begin{cases} A_n = 20l(l-1)f_{\text{round}}\left(\dfrac{m-1}{2}\right) \\ M_n = 10l(l+1)f_{\text{round}}\left(\dfrac{m-1}{2}\right) \end{cases} \tag{18}$$

其中,$f_{\text{round}}(\cdot)$为四舍五入函数。

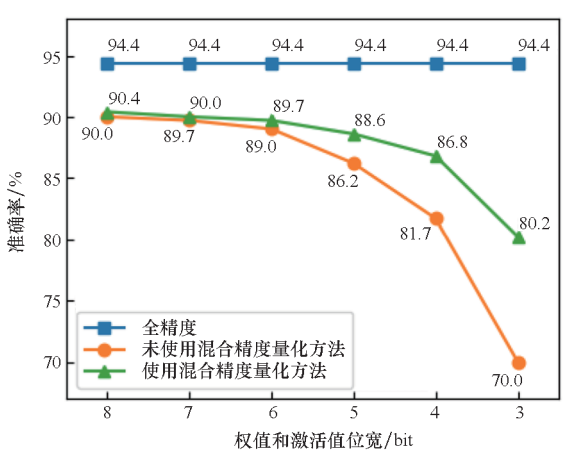
2.2　性能实验

本小节主要包括 4 个实验:实验一,在 DoReFa[27] 和可学习步长量化改进版[28](learned step size quantization +,LSQ +)两种固定精度量化方法的基础上,验证了所提混合精度量化方法的普适性;实验二,将所提混合精度量化方法与其他混合精度量化方法进行对比,体现了所提方法的性能优势;实验三,采用加噪训练的方式提升混合精度量化方法的鲁棒性,验证了其噪声适应能力;实验四,验证了混合精度量化方法在其他数据集上的性能,并分析了其位宽分配过程与训练过程。

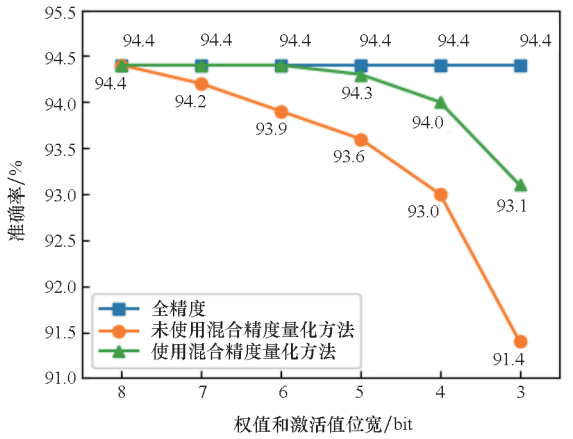
2.2.1　混合精度量化方法的普适性

在两种固定精度量化方法 DoReFa[27] 和 LSQ +[28] 的基础上,应用所提出的层敏感性分析、位宽分配策略等实现混合精度量化方法,并与未使用混合精度量化的情况进行对比。

位宽范围设置为 3 ~ 8 bit,权值位宽与激活值位宽相同。需要说明的是,对于混合精度量化方法而言,此处的位宽指的是平均位宽。实验结果如图 2 所示。



(a) DoReFa



(b) LSQ +

图 2　DoReFa 和 LSQ + 使用混合精度量化前后对比

Fig. 2　DoReFa and LSQ + comparison before and after using mixed-precision quantization

对于 DoReFa 方法,在权值和激活值位宽逐渐降低时,混合精度量化方法的识别准确率下降较缓。在权值和激活值位宽为 3 bit 时,相较于未使用混合精度量化方法,使用混合精度量化方法后,识别准确率提高了 10.2%。对于 LSQ + 方法,在权值和激活值位宽较大时,所提混合精度量化方法的识别准确率与全精度类似;在权值和激活值位宽逐渐减小时,混合精度量化方法同样能够抑制识别准确率的下降;在位宽为 3 bit 时,相较于未使用混合精度量化方法,使用混合精度量化方法后,准确率提高了 1.7%。

上述两种方法中,DoReFa 方法在反向传播过程中采用的是直通估计,未使用其他梯度调整方法,而 LSQ + 方法在反向传播过程中,使用了截断与可学习步长和零点的梯度调整方法。梯度调整方法能够在训练过程中将梯度控制在合理的范围内,使权值收敛更加精确迅速。从图 2 也可以看出,LSQ + 方法的性能表现优于 DoReFa 方法,说明梯度调整方法能够提升模型的性能。

此外,为了进一步验证混合精度量化方法的普适性,在轻量二元卷积神经网络[29](light binary convolutional neural networks, LB-CNN)固定精度量化方法的基础上也应用所提出的层敏感性分析、位宽分配策略等实现混合精度量化,在权值平均位宽为 2 bit、激活值平均位宽为 4 bit 时,混合精度量化后的识别准确率为 89.0%,相较于 LB-CNN 方法提升了 7.8%。并且,在 LSQ + 方法的基础上应用所提出的混合精度量化方法,在权值平均位宽为 2.4 bit、激活值平均位宽为 4 bit 时[30],识别准确率为 93.7%,比文献[30]使用的混合精度量化方法高出约 1.3%。

2.2.2 混合精度量化方法的有效性

为了验证所提混合精度量化方法在性能上的优势,将所提方法与 HAWQ[25] 和 DQMQ[26] 两种混合精度量化方法进行对比。位宽设置与文献[25-26]相同,权值平均位宽为 2 bit,激活值平均位宽为 4 bit。实验结果对比情况如表 1 所示,其中 HAWQ 和 DQMQ 的数据均来自文献[26]。

表 1 各种混合精度量化方法性能对比

Tab. 1 Performance comparison of various mixed-precision quantization methods

方法	准确率/%	平均位宽/bit (权值, 激活值)	搜索效率 (加法,乘法)
HAWQ[25]	81.90	2, 4	1.4×10^8, 1.4×10^8
DQMQ[26]	82.03	2, 4	2.1×10^{12}, 1.4×10^{11}
本文	83.16	2, 4	7.6×10^3, 4.2×10^3

从表 1 可以看出,所提方法的识别准确率高于其余两种混合精度量化方法。在权值平均位宽为 2 bit,激活值平均位宽为 4 bit 的情况下,所提方法的识别准确率比 HAWQ 方法高了 1.26%,比 DQMQ 方法高了 1.13%;在搜索效率上,所提方法完成位宽搜索任务所需的加法次数和乘法次数分别为 7.6×10^3 和 4.2×10^3,均远低于其他两种方法。

2.2.3 混合精度量化方法的鲁棒性

为了增强采用混合精度量化方法的卷积神经网络的鲁棒性,本小节采用了与文献[31]类似的加噪训练方式,降低含噪声数据带来的影响。向输入数据注入噪声的过程为:

$$\tilde{x}=x+n \quad n\sim N(0,\sigma^2) \tag{19}$$

式中,x 表示原始数据,$\tilde{x}$ 表示注入噪声后的数据,n 表示注入的噪声,满足均值为 0、标准差为 σ 的正态分布。

将加噪训练后的混合精度量化方法、加噪训练后的固定精度量化方法,以及未经过加噪训练的混合精度量化方法、未经过加噪训练的固定精度量化方法进行对比,对比结果如图 3 所示。此处的噪声标准差、位宽设置等与文献[31]相同,其中,$0.1\leqslant\sigma\leqslant0.5$,权值、激活值位宽均为4 bit。固定精度量化方法采用 DoReFa[27] 方法,混合精度量化方法在 DoReFa[27] 方法基础上实现。

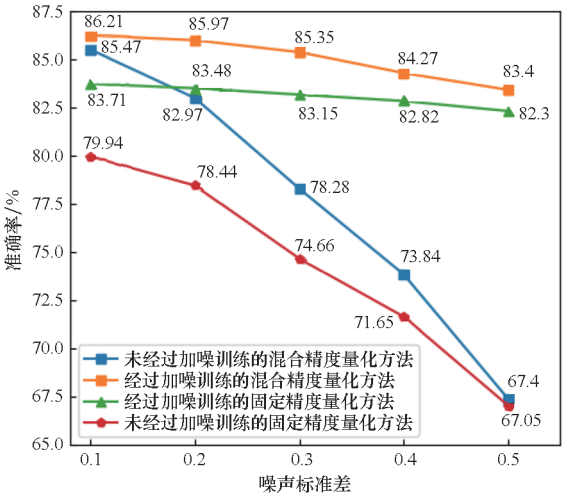


图 3 加噪训练前后方法性能对比

Fig. 3 Comparison of model performance before and after noise-robust training

从图 3 可以看出,随着噪声标准差逐渐增大,未经过加噪训练的混合精度量化方法、未经过加噪训练的固定精度量化方法识别准确率大幅下降,而经过加噪训练的混合精度量化方法、经过加噪训练的固定精度量化方法识别准确率能够维持在较高水平。与未经过加噪训练的量化方法相比,无论是固定精度量化方法,还是混合精度量化方法,加噪训练能够提高不同量化方法对噪声的适应性。在噪声标准差为 0.5 时,经过加噪训练的混合精度量化方法识别准确率高出未经过加噪训练的混合精度量化方法 16%。与固定精度量化方法相比,在噪声标准差 $0.1\leqslant\sigma\leqslant0.5$ 时,混合精度量化方法具有更高的准确率,表明所提出的混合精度量化方法对噪声具有更好的适应能力。

2.2.4 混合精度量化方法的位宽分配与训练过程

为了验证所提方法在其他数据集上的性能,并对混合精度量化方法的位宽分配与训练过程进行分析观察了位宽分配算法迭代过程中平均位宽的变化趋势与量化感知训练过程中准确率的变化趋势。数据集采用 CIFAR-100,该数据集包含 100 个类别共 60 000 张 32 像素 ×32 像素的图片,其中 50 000 张用于训练,10 000 张用于测试。权值、激活值的平均位宽设置与文献[32]相同,均为 3.42 bit。实验结果如图 4 ~5 所示。

从图 4 可以看出,随着迭代次数的增加,平均位宽逐渐逼近 3.42 bit 并保持不变,验证了所提方法在位宽分配过程中的收敛性。需要说明的是,其收敛速度存在差异是因为卷积层各层

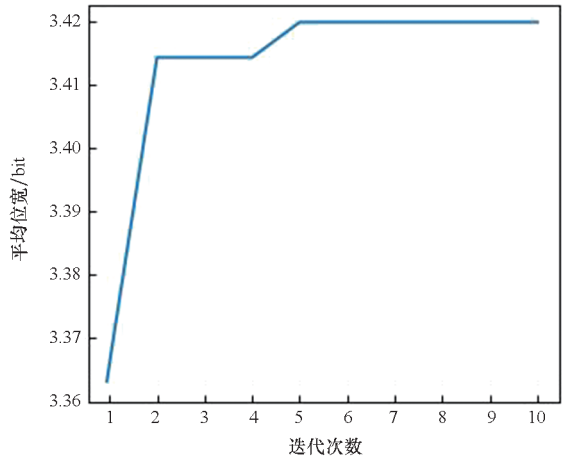


图 4　位宽分配算法过程中平均位宽的变化趋势

Fig. 4　Trend of the average bit width change during the bit width allocation algorithm process

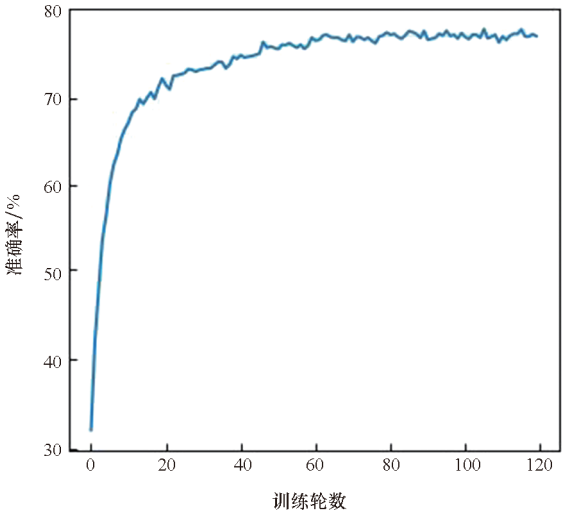


图 5　量化感知训练过程中准确率的变化趋势

Fig. 5　Trend of accuracy change during the quantization-aware training process

参数量不同造成的。从图 5 可以看出,在量化感知训练中,刚开始准确率提升较快,随着训练轮次的增加,准确率提升减慢,虽然出现波动但总体性能仍然在缓慢提升,最终准确率达到 76.78% 左右,比文献[32](同一平均位宽下)的准确率高 0.8%。

3　结论

提出了一种基于层敏感性分析的卷积神经网络混合精度量化方法,通过计算 Hessian 矩阵的平均迹来衡量卷积层参数的敏感性,并根据敏感性使用逐层升降的方法进行位宽分配,最终完成网络模型的混合精度量化。基于 ResNet18 网络模型在 CIFAR-10 数据集和 CIFAR-100 数据集上开展实验验证,结果表明:与固定精度量化方法相比,所提混合精度量化方法能够显著提高识别的准确率;与其他混合精度量化方法相比,所提混合精度量化方法具有较好的识别准确率和搜索效率优势。此外,加噪训练能够显著增强量化方法的鲁棒性。

参考文献(References)

[1] SAFA ALDIN S, ALDIN N B, AYKAÇ M. Enhanced image classification using edge CNN (E-CNN)[J]. The Visual Computer, 2024, 40: 319 – 332.

[2] SHAO R, BI X J, CHEN Z. Hybrid ViT-CNN network for fine-grained image classification[J]. IEEE Signal Processing Letters, 2024, 31: 1109 – 1113.

[3] DAS N, BORTIEW A, PATRA S, et al. Dual-branch CNN incorporating multiscale SVD profile for PolSAR image classification[J]. IEEE Transactions on Geoscience and Remote Sensing, 2024, 62: 5223612.

[4] LIU Z S, SHEN L. CECT: controllable ensemble CNN and transformer for COVID-19 image classification[J]. Computers in Biology and Medicine, 2024, 173: 108388.

[5] LI R D, WANG Y, LIANG F, et al. Fully quantized network for object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[6] PANG J M, CHEN K, SHI J P, et al. Libra R-CNN: towards balanced learning for object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[7] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137 – 1149.

[8] EVERINGHAM M, ALI ESLAMI S M, VAN GOOL L, et al. The pascal visual object classes challenge: a retrospective[J]. International Journal of Computer Vision, 2015, 111: 98 – 136.

[9] GUO Z C, ZHANG X Y, MU H Y, et al. Single path one-shot neural architecture search with uniform sampling[EB/OL]. (2020 – 07 – 08)[2024 – 12 – 30]. https://arxiv.org/abs/1904.00420v4.

[10] HAN S, MAO H, DALLY W J. Deep compression: compressing deep neural network with pruning, trained quantization and huffman coding [C]//International Conference on Learning Representations (ICLR), 2016.

[11] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[J]. Advances in Neural Information Processing Systems, 2015, 1: 1135 – 1143.

[12] WU B C, WANG Y H, ZHANG P Z, et al. Mixed precision quantization of ConvNets via differentiable neural architecture search[EB/OL]. (2018 – 11 – 30)[2024 – 12 – 30]. https://arxiv.org/abs/1812.00090v1.

[13] YU H B, HAN Q, LI J B, et al. Search what you want: barrier panelty NAS for mixed precision quantization[C]//Proceedings of the European Conference on Computer Vision, 2020.

[14] CAI Y H, YAO Z W, DONG Z, et al. ZeroQ: a novel zero shot quantization framework[C]//Proceedings of the IEEE/

CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[15] LIU J, CAI J F, ZHUANG B H. Sharpness-aware quantization for deep neural networks[EB/OL]. (2023 - 03 - 21)[2024 - 12 - 30]. https://arxiv.org/abs/2111.12273v5.

[16] MA Y X, JIN T S, ZHENG X W, et al. OMPQ: orthogonal mixed precision quantization[EB/OL]. (2022 - 11 - 23)[2024 - 12 - 30]. https://arxiv.org/abs/2109.07865v3.

[17] CHOI J, WANG Z, VENKATARAMANI S, et al. PACT: parameterized clipping activation for quantized neural networks[EB/OL]. (2018 - 07 - 17)[2024 - 12 - 30]. https://arxiv.org/abs/1805.06085.

[18] ESSER S K, MCKINSTRY J L, BABLANI D, et al. Learned step size quantization[C]// International Conference on Learning Representations (ICLR), IEEE, 2019.

[19] ZHANG D Q, YANG J L, YE D Q, et al. LQ-nets: learned quantization for highly accurate and compact deep neural networks[C]// Proceedings of the European Conference on Computer Vision, 2018.

[20] WANG Z W, XIAO H, LU J W, et al. Generalizable mixed-precision quantization via attribution rank preservation[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[21] TANG C, OUYANG K, CHAI Z H, et al. SEAM: searching transferable mixed-precision quantization policy through large margin regularization[C]//Proceedings of the 31st ACM International Conference on Multimedia, 2023: 7971 - 7980.

[22] TANG C, OUYANG K, WANG Z, et al. Mixed-precision neural network quantization via learned layer-wise importance[EB/OL]. (2023 - 03 - 05)[2024 - 12 - 30]. https://arxiv.org/abs/2203.08368v5.

[23] DONG Z, YAO Z W, CAI Y H, et al. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks[EB/OL]. (2019 - 11 - 10)[2024 - 12 - 30]. https://arxiv.org/abs/1911.03852v1.

[24] AVRON H, TOLEDO S. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix[J]. Journal of the ACM, 2011, 58(2): 1 - 34.

[25] DONG Z, YAO Z W, GHOLAMI A, et al. HAWQ: Hessian aware quantization of neural networks with mixed-precision[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[26] WANG Y C, GUO S, GUO J C, et al. Data quality-aware mixed-precision quantization via hybrid reinforcement learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2025, 36(5): 9018 - 9031.

[27] ZHOU S C, WU Y X, NI Z K, et al. DoReFa-net: training low bitwidth convolutional neural networks with low bitwidth gradients[EB/OL]. (2018 - 02 - 02)[2024 - 12 - 30]. https://arxiv.org/abs/1606.06160v3.

[28] BHALGAT Y, LEE J, NAGEL M, et al. LSQ +: improving low-bit quantization through learnable offsets and better initialization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020.

[29] CAI Y, TANG T Q, XIA L X, et al. Low bit-width convolutional neural network on RRAM[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 39(7): 1414 - 1427.

[30] PENG J, LIU H J, ZHAO Z J, et al. CMQ: crossbar-aware neural network mixed-precision quantization via differentiable architecture search[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41(11): 4124 - 4133.

[31] ZHANG C X, ZHAO Z J, LUO H F, et al. Network quantization with gradient adjustment for rounding difference[C]// Proceedings of the IEEE 5th International Conference on Pattern Recognition and Machine Learning (PRML), 2024.

[32] KUNDU S, WANG S K, SUN Q R, et al. BMPQ: bit-gradient sensitivity-driven mixed-precision quantization of DNNs from scratch[C]//Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2022.