基于误差限制的神经网络混合精度量化方法 (特邀)

李奕铎，郭子博，刘　凯，孙逍遥

(西安电子科技大学 计算机科学与技术学院，陕西 西安 710071)

摘　要：基于卷积神经网络的深度学习算法展现出卓越性能的同时也带来了冗杂的数据量和计算量，大量的存储与计算开销也成了该类算法在硬件平台部署过程中的最大阻碍。而神经网络模型量化使用低精度定点数代替原始模型中的高精度浮点数，在损失较小精度的前提下可有效压缩模型大小，减少硬件资源开销，提高模型推理速度。现有的量化方法大多将模型各层数据量化至相同精度，混合精度量化则根据不同层的数据分布设置不同的量化精度，旨在相同压缩比下达到更高的模型准确率，但寻找合适的混合精度量化策略仍十分困难。因此，提出一种基于误差限制的混合精度量化策略，通过对神经网络卷积层中的放缩因子进行统一等比限制，确定各层的量化精度，并使用截断方法线性量化权重和激活至低精度定点数，在相同压缩比下，相比统一精度量化方法有更高的准确率。其次，将卷积神经网络的经典目标检测算法 YOLOV5s 作为基准模型，测试了方法的效果。在 COCO 数据集和 VOC 数据集上，该方法与统一精度量化相比，压缩到 5 位的模型平均精度均值 (mean Average Precision, mAP) 分别提高了 6% 和 24.9%。

关键词：深度学习；　混合精度；　截断量化；　YOLOV5

中图分类号：TP391.4　　**文献标志码：**A　　**DOI：**10.3788/IRLA20220166

Mixed-precision quantization for neural networks based on error limit (*Invited*)

Li Yiduo，Guo Zibo，Liu Kai，Sun Xiaoyao

(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

Abstract: The deep learning algorithm based on convolutional neural network exhibits excellent performance, but also brings a complex amount of data and calculation. A large amout of storage and computing overhead has alse become the biggest obstacle to the deployment of such algorithms in hardware platforms.The neural network model quantization uses low-precision fixed-point numbers instead of high-precision floating-point numbers in the original model, which can effectively compress the model size, reduce hardware resource overhead, and improve model inference speed on the premise of losing less precision. Most of the existing quantization methods quantize the data of each layer to the same accuracy, while mixed-precision quantization sets different quantization accuracy according to the data distribution of different layers, aiming to achieve a higher model accuracy under the same compression ratio, but finding a suitable mixed-precision quantization strategy is still very difficult. Therefore, a mixed-precision quantization strategy based on error limitation was proposed. By uniformly and proportionally limiting the scaling factors in each layer of the neural network, the quantization accuracy of each layer was determined, and the truncation method was used to linearly quantize the weights and

收稿日期:2022−03−10；　**修订日期:**2022−04−11

作者简介:李奕铎，男，硕士生，主要从事深度学习神经网络模型压缩等方面的研究。

导师简介:刘凯，男，教授，博士生导师，博士，主要从事数据编码设计等方面的研究。

activate to low-precision fixed-point numbers. Under the same compression radio, this method had higher accuracy than the unified precision quantization method. Secondly, the classical object detection algorithm YOLOV5s based on convolutional neural network was used as the benchmark model to test the effect of the method. On the COCO data set and VOC data set, compared with the unified precision quantization, the mean average precision (mAP) of the model compressed to 5 bits was improved by 6% and 24.9%.

Key words: deep learning; mixed precision; truncated quantization; YOLOV5

0 引 言

近年来, 基于卷积神经网络 (Convolutional Neural Networks, CNN) 的深度学习方法受到了学术界和工业界的关注, 在目标分类、目标检测等方面得到了广泛的应用, 已经成为各种计算机视觉任务的主流方法。随着数据集规模的扩大、硬件设备的发展, 可处理卷积神经网络模型的层数也在不断增加, 例如, 从 AlexNet[1]、VGGNet[2]、GoogleNet 到 ResNet[3], ILSVRC 挑战赛的冠军模型已经从 8 层提升到 100 多层。随着层数的加深, 模型参数规模与计算强度也随之增大增强。一个 50 层的残差网络有超过 2 550 万个参数, 模型推理时计算量更是达到 4 GFLOP 以上。如此庞杂的参数量与数据量为其在带宽与存储受限的边缘设备上部署平添了诸多问题。主要包括:

(1) 模型大小的限制: 深度学习卷积网络模型中数百万个可训练参数成就了其强大的检测能力。例如, 存储一个经过训练的语义分割 DeepLabV3+模型会占用超过 100 MB 内存空间, 这对硬件设备是一个巨大的资源负担。

(2) 运行内存: 在推理过程中, CNN 的卷积操作中间变量会占用更多的内存空间, 普通硬件设备难以支撑。

(3) 计算用时: 在对高分辨率图像进行密集卷积操作时, 硬件设备往往延时较高, 难以在较低功耗下实现实时的结果预测。

参数量化通过减少神经网络参数值所需的数据位宽来压缩原始网络。且量化后, 低精度定点乘累加运算 (Multiply Accumulate, MAC) 可以替代浮点运算以降低资源开销与硬件能耗。Vanhoucke 等人[4] 的研究表明, 8 位推理运算能够在提升速度的同时保证最小的模型精度损失。目前主流的量化方法大多数采用统一精度量化, 如统一 16 位量化方法[5]、谷歌 8 位量化方法[6], 将网络模型各层的权重和激活数据量化至相同的位宽。但不同卷积层中参数的分布范围和冗余度都不相同, 如图 1(b) 中展示了 YOLOV5 s 网络前 20 层中的权重参数分布。可以看出, 各层的权重最值分布有较大的差异, 第 1 层的权重最大值约为第 3 层的 25 倍。因此一些研究者提出了差异位宽分布的混合精度量化方法。如零样本量化框架 (a novel zero shot quantization framework, ZeroQ) 方法[7]、强化学习量化策略搜索方法[8−9] 等, 通过奖励函数设置不同卷积层的量化位宽, 训练得到最优分配策略。

文中对比了不同量化方法在 VOC2007 数据集上的表现, 选择合适的量化方法, 通过对各层的放缩因子进行统一等比限制, 得到了合适的网络模型分层量化方法, 并依据该方法对权重和激活值进行混合截断量化, 取得了更高的准确率。文中的工作包含以下方面:

(1) 通过对比移位量化形式与乘积量化形式、无截断量化方法与基于均方误差的截断量化方法, 选择最优方法作为文中的实现方法;

(2) 提出基于误差限制的量化搜索策略, 不同于强化学习或策略搜索等方法, 文中方法具有更低的复杂度与更高的普遍性, 可快速获取分层量化策略;

(3) 根据混合精度分层量化策略对 YOLOV5 s 网络中不同卷积层数据进行混合截断量化, 并使用 VOC2007 数据集进行对比实验;

(4) 在 COCO 数据集和 VOC2011 数据集上使用 YOLOV5 s 网络对混合精度量化方法进行测试。

1 神经网络压缩方法

目前, 深度学习神经网络的主流压缩和加速方法主要有两类: 模型剪枝、参数量化[10]。

1.1 模型剪枝

模型剪枝是深度学习神经网络模型中应用最为广泛的压缩与加速方法, 基本思想是通过对训练后的 CNN 模型进行稀疏化操作, 剪除冗余的、信息量较少的参数值, 达到压缩模型大小的目的。在 2015 年,

Chen 等人[11] 提出了 HashedNets 模型，引入哈希函数并根据参数间汉明距离重组权重，实现参数共享，是典型的非结构化剪枝方法。然而，非结构化剪枝在计算过程中引入稀疏矩阵造成内存获取的不规则性，影响硬件工作效率，进而降低运算速度。2017 年，Liu 等人[12] 提出一种通道级别结构化剪枝方法，向存在于每个卷积层的批量标准化 (Batch Normalization) 中的缩放因子添加稀疏正则限制，通过批量标准化中超参数 γ 的大小对卷积通道进行剪除，再通过微调恢复精度，有效降低了网络复杂度。2020 年，徐等人[13] 为所有卷积层中的卷积核数量乘以缩小因子 α 以压缩网络，有效地降低了冗余参数，同时压缩后的计算量减少了 α 倍。Lin 等人[14] 提出一种基于特征图矩阵秩的滤波器剪枝方法，剪除低秩特征图的滤波器，在 ResNet-110 上缩小了 58.2% 的计算量，精度仅损失 0.14%。He 等人[15] 提出了一种滤波器剪枝准则 (Learning Filter Pruning Criteria, LFPC) 指导剪枝，该方法能够考虑到网络中所有层的协同作用，自适应为不同卷积层选择适合自己的剪枝规则。

1.2 参数量化

量化是卷积神经网络模型压缩移植的主要实现方法。2015 年，Han 等人[16] 引用一个三阶段的管道流程—网络权重剪枝、训练量化网络、哈夫曼编码，将神经网络的存储需求压缩到原来的$\frac{1}{35}$ ~ $\frac{1}{49}$。2017 年，谷歌提出一种非对称量化方法[6]，如图 1(a) 中所示，将权重和激活都量化到 8 位整数，并在训练阶段添加量化训练框架以模拟量化运算带来的误差，能够最大限度地减少真实模型上量化带来的精度损失。2019 年，Gong 等人[17] 提出了 Differentia-ble Soft Quantization (DSQ) 方法构建可微分的软量化函数，使得量化参数也可以跟随神经网络一起训练学习。2020 年，Zhu 等人[18] 在将网络参数量化到 8 bit 的同时，引入了误差敏感的学习率调节方法，并且对量化造成的精度损失问题提出了方向自适应的梯度阶段处理方法。

统一精度量化将神经网络中所有层量化到固定的位宽，而混合精度量化为不同层分配不同的位宽。事实上，卷积神经网络各层的权重最大值分布是不同的，因此，2019 年，Wang 等人[8] 提出基于硬件感知的混合精度自动量化方法方法，将寻找目标网络混合量

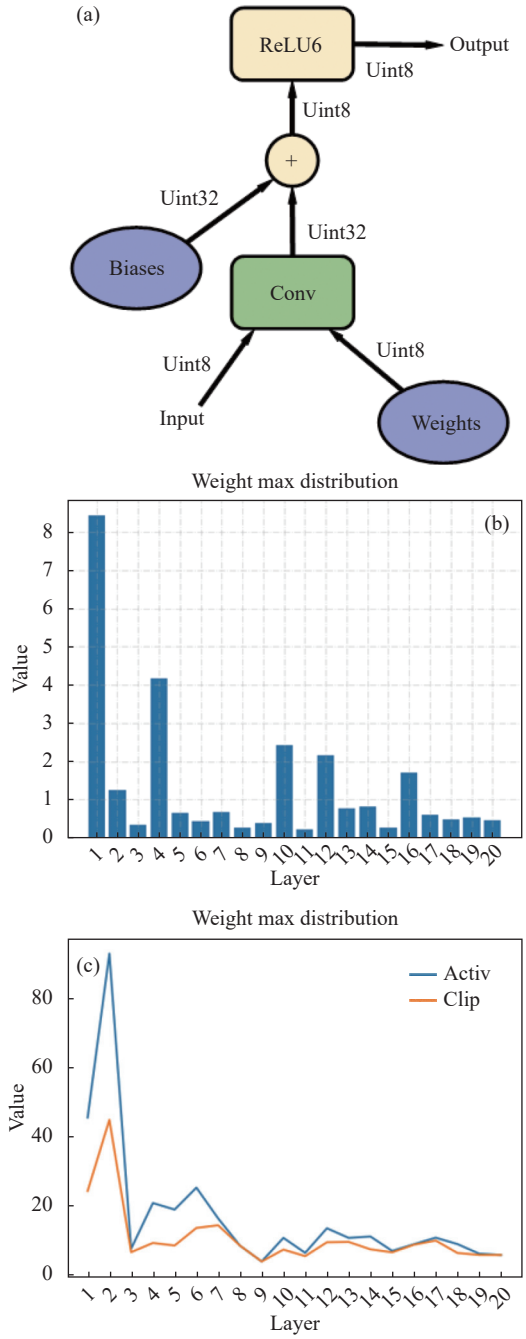


图 1 (a) 深度学习卷积 8 位量化过程[6]; (b)YOLOV5 s 网络前 20 层权重最值分布趋势; (c)YOLOV5 s 网络量化过程中的激活最大值与截断值分布

Fig.1 (a) Photograph of deep learning convolutional 8-bit quantization procession[6]; (b) The distribution trend of the most valued weights in the first 20 layers of the YOLOV5 s network; (c) Distribution of activation maximum and cutoff value during network quantization in YOLOV5 s

化策略的问题建模为强化学习问题，通过深度确定性策略梯度 (deep deterministic policy gradient, DDPG)

算法自动搜索最佳量化策略。Huang 等人[9]优化了强化学习中的奖励函数设置,使得搜索出的量化策略够在更小的模型下仍有较高的准确率。

人工搜索合适的分层量化策略是非常困难的。如经典的 YOLOV3_TINY[19]算法,拥有 13 个卷积层,可能的量化策略就有8^{13}种,仅靠人工搜索最佳策略是几乎不可能完成的任务。目前的主流的剪枝方法和量化搜索策略依赖强化学习 (reinforcement learning) 或进化算法 (evolutionary algorithm-m),需要较长的搜索阶段才能收敛,且当对不同的数据集以及压缩率进行实验时,强化学习方法都需要进行重新调参和训练以获得最终结果。文中通过对量化舍入误差分析,提出基于误差限制的混合精度量化方法,相比于强化学习方法,文中拥有更低的时间复杂度与更高的普遍性;相比于全层统一精度量化方法,文中在同样压缩率下拥有更高的精度。

2 基于误差限制的混合精度量化方法

2.1 卷积截断量化方法

2.1.1 量化方法

量化将浮点数转换为定点数进行运算,在推理过程中只使用整数,因此对于硬件实现更加友好高效。文中在表 1 所示的两种量化方法中选择,其中,$round(x)$函数为四舍五入,乘积运算因子s定义为$s = w_{max}/(2^{bi-1}-1)$,代表着量化运算前后的缩放尺度;$fl = b_i - 1 - \log_2 w_{max}$为量化的移位因子,代表量化运算的移位步长。移位量化方法缩小了量化运算带来的功耗损失,在提高运算效率的同时增大了精度损失。使用 YOLOV5 s 网络在 VOC2007 数据集上分别对两种方法进行了测试,结果如表 2 所示。乘积量化方法在低位宽量化时相较于移位量化方法整体拥有更高的精度,随着量化位宽的降低,两种方法的精度损失逐步增大,移位量化方法的精度损失下降更快,乘积量化方法在 7 位量化时仍有不错的精度。

表 1 乘积量化方法与移位量化方法

Tab.1 Product quantization method and shift quantization method

Quantitative method	Operation
$q(w,b_i) = round(w/s)$	Multiplication
$q(w,b_i) = round\left(w2^{fl}\right)$	Displacement

表 2 不同量化方法在 VOC2007 数据集上的表现

Tab.2 The performance of different quantification methods on the VOC2007 dataset

Network model	Dataset	bit	mAP.5-.95	
			Displacement	Multiplication
YOLOV5 s	VOC	8	63.4%	77.9%
		7	26.5%	68.8%
		6	4.6%	39.5%
		32	81.8%	

2.1.2 截断方法

硬件设备一般存储卷积神经网络中的权重和偏置等信息,在观察网络中各层权重分布时,有一些离散值的存在,数量很小,但影响该层权重的最大值,增加了权重分布范围,进而影响了量化后网络的准确率。因此,对权重数据先截断再量化,可以减少量化带来的精度损失。如图 1(c) 中所示,YOLOV5 s 前 20 层中实际采用的截断值约为原始最大值的 1/2。文中采用截断操作优化乘积运算的量化方法,具体来说,该方法逐层将权值截断至$[-c,c]$的范围,后将其量化至b_i位。量化方法为:

$$q(w,b_i,c) = round(clamp(w,c)/s) \tag{1}$$

式中:clamp(w,c)表示将权值w截断到$[-c,c]$的范围;b_i为第 i 层量化位宽。截断值 c 的选择如下:

$$c = {}_{x}^{argmin} d_{MSE}(w \| w_q(w,x,s)) \tag{2}$$

式中:d_{MSE}表示原始权值分布和量化模拟后的权重分布之间的均方误差。

文中在 YOLOV5 s 网络对 VOC2007 数据集使用截断方法前后的性能进行了测试对比,结果如表 3 所示,其中,MAX 代表最大值量化,MSE 表示基于均方误差 (mean squared error, MSE) 的截断量化方法。可

表 3 不同方法量化前后的网络准确率

Tab.3 Network accuracy before and after quantization with different truncation methods

bit		8	7	6	5	32
mAP	MAX	78.9%	67.4%	46.7%	4.0%	82.6%
	MSE	82.7%	76.0%	69.0%	31.7%	

以看出，截断方法能够有效控制量化误差，在各个量化位宽，相较于无截断方法均有一定的精度提升，且位宽越低，性能对比越明显，当量化位宽为 5 bit 和 6 bit 时，采用 MSE 截断量化方法比无截断量化方法性能分别提升了 27.7% 和 22.3%，说明基于均方误差的量化方法能有效恢复检测精度。

2.2 基于误差限制的量化搜索策略

量化过程中的误差主要来自于两个方面：取整损失与截断损失。截断操作的引入可以缩小量化区间，减少量化过程中的取整损失。在此分析软件端模拟量化过程中的取整损失。

输入量化操作：

$$q_{in} = round(in/s_1) \tag{3}$$

权重量化操作：

$$q_w = round(w/s_2) \tag{4}$$

偏置量化操作：

$$q_b = round(bias/s_1 \times s_2) \tag{5}$$

激活量化操作为：

$$q_{activ} = round(activ/s_3) \tag{6}$$

$$activ = (q_{in} \otimes q_w + q_b) \times s_1 \times s_2 \tag{7}$$

由于四舍五入取整操作带来的损失在 0.5 以内，因此输入量化、权重量化以及激活量化操作带来的误差均在0.5×scale范围内。且与scale大小成正相关。文中认为，每一层对最终结果的影响都是相同的，因此，笔者课题组可以选择一种误差限制的量化策略，通过对卷积层的放缩因子scale的大小进行限制，得到不同卷积层的量化精度。使用 γ 作为卷积层的误差限制参数，初始时设置所有卷积层的量化位宽为 8，如果放缩因子scale小于 γ，则该层位宽减 1,直至该层放缩因子scale超过 γ。整体流程如图 2 所示。

文中设计循环计算获得最佳 γ，初始位宽设置为 8，逐层推理并调整位宽大小，从而确定最佳位宽策略。由于卷积层中最大值出现在激活部分，因此使用激活部分放缩因子scale与 γ 进行对比，如果scale小于 γ，则该层位宽减 1，直至该层放缩因子超过 γ，进入下一卷积层。

笔者课题组在 YOLOV5 s 训练 VOC2007 数据集的量化方法探索中，测试了不同 γ 值对于量化结果的影响，如表 4 所示。γ 可设置在[0.08,0.25]之间，在最

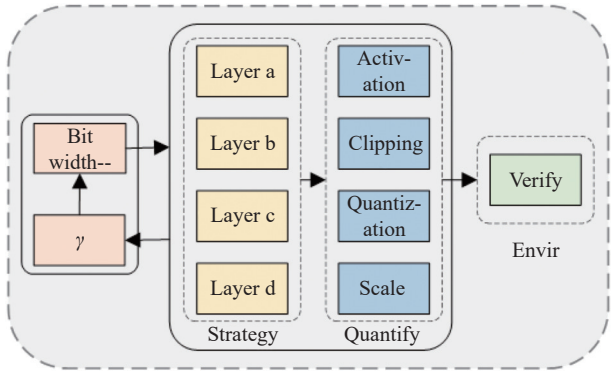


图 2 网络分层策略方法框架

Fig.2 Framework of network hierarchical policy methodology

表 4 误差限制参数 γ 取值对比

Tab.4 Error limit parameter γ value comparison

γ	Compression radio	Average bit	mAP
0.08	4.93	6.49	79.6%
0.10	5.13	6.23	77.8%
0.125	5.74	5.57	72.3%
0.142	6.11	5.23	62.8%
0.166	6.31	5.07	63.3%
0.20	7.14	4.48	21.0%

佳点之前，网络精度随着压缩率的增加在不断波动，达到最佳点之后，随着压缩率的增加，网络精度整体呈现下降趋势，相较于表 3，VOC 数据集统一量化到 7 位和 5 位的精度分别为 76% 和 31.7%，混合量化到 6.49 位和 5.07 位的精度分别为 79.6% 和 63.3%。混合精度量化方法均能在更小的模型中达到更高的检测精度。

2.3 相关处理

2.3.1 激活函数处理

由于大部分网络的激活函数都使用 ReLU 函数，因此每一层的输出函数的分布范围为[0,c]。对于 ReLU 激活函数的缩放因子 s，文中定义为$s = c/(2^{bi} - 1)$，对于 Leaky ReLU、SiLU 等值域涉及负数的激活函数，文中使用$s = c/(2^{bi-1} - 1)$作为量化放缩因子。

2.3.2 链接模块与残差模块处理

部分卷积神经网络拥有链接模块与残差模块。链接模块用于在指定维度链接两个张量，残差模块拥有张量相加操作。

链接操作：

$$r_3 = concat[q_{activ1}, q_{activ2}] \tag{8}$$

相加操作：

$$r_3 = add[q_{activ1}, q_{activ2}] \tag{9}$$

文中在神经网络中涉及多个张量之间的操作时，对多个张量采用统一的缩放因子与量化位宽。即：

$$b_i = max(b_{i1}, b_{i2}) \tag{10}$$

$$\text{scale} = max(scale_1, scale_2) \tag{11}$$

3 实验结果与分析

3.1 实验环境

为了验证文中的方法，笔者课题组在搭载显存为 24 GB 的 Geforce RTX 3090 的服务器上进行实验，系统环境为 Ubuntu16.04，Pytorch 版本为 1.10.1，Torchvision 版本为 0.4.2，Python 版本为 3.6.0，网络学习率为10^{-4}，YOLOV5 网络结构使用已有的 Pytorch 实现方法与提供的 YOLOV5 s.pt 预训练模型。

3.2 实验结果

YOLOV5 网络是目前单阶段目标检测网络中性能最好的网络之一，文中在 COCO 数据集和 VOC2011 数据集上对统一量化方法与基于误差限制的混合截断量化方法进行了对比测试，如表 5 所示。

在 COCO 数据集，与统一 6 位量化 (模型为 5.45 MB) 相比，文中的方法在 5.05 MB 时拥有更高的精度，性能分别提升了 3.3% (mAP@0.5) 和 2.1% (mAP@0.5-0.95)，与统一 5 位量化相比，文中在相似的模型大小下性能分别提升了 6% (mAP@0.5) 和 4.5% (mAP@0.5-0.95)。

在 VOC2011 数据集，相比全精度模型，文中的方法将模型压缩到 5.05 MB 的同时 mAP@0.5 提升了 3.1%，部分图片的检测效果有所提升；与统一精度量化方法相比，文中的方法在相似的模型大小 (统一 5 位量化与平均 5 位量化) 下性能分别提升了 24.9% (mAP@0.5) 和 16.1% (mAP@0.5-0.95)。

整体上看，随着量化位宽的降低，统一精度量化方法和混合精度量化方法带来的精度损失都在逐步增加，但文中的方法能够将量化误差平均分配在各个卷积层，减少了部分层误差较大情况的出现，因此精度下降较缓，整体上相对于统一精度量化方法拥有更高的精度。表 6 展示了 VOC2011 数据集采用不同方法进行检测的不同类别的精度，可以看出，混合精度量化方法与统一精度量化方法相比，Dog 类别精度有所下降，Bird、Chair、Sheep、Train 类别精度相同，Aeroplane、Bicycle、Boat、Bottle、Person、Tvmonitor 类别精度均有所上升。图 3 展示了 YOLOV5 s 对 COCO 数据集采用不同量化方法的检测结果，其中，图 3(a) 为训练真值，标识出了全部待检测目标；图 3(b) 为全精度检测结果；图 3(c) 为采用文中的方法进行混合

表 5 不同量化方法对 COCO 数据集和 VOC2011 数据集的测试结果

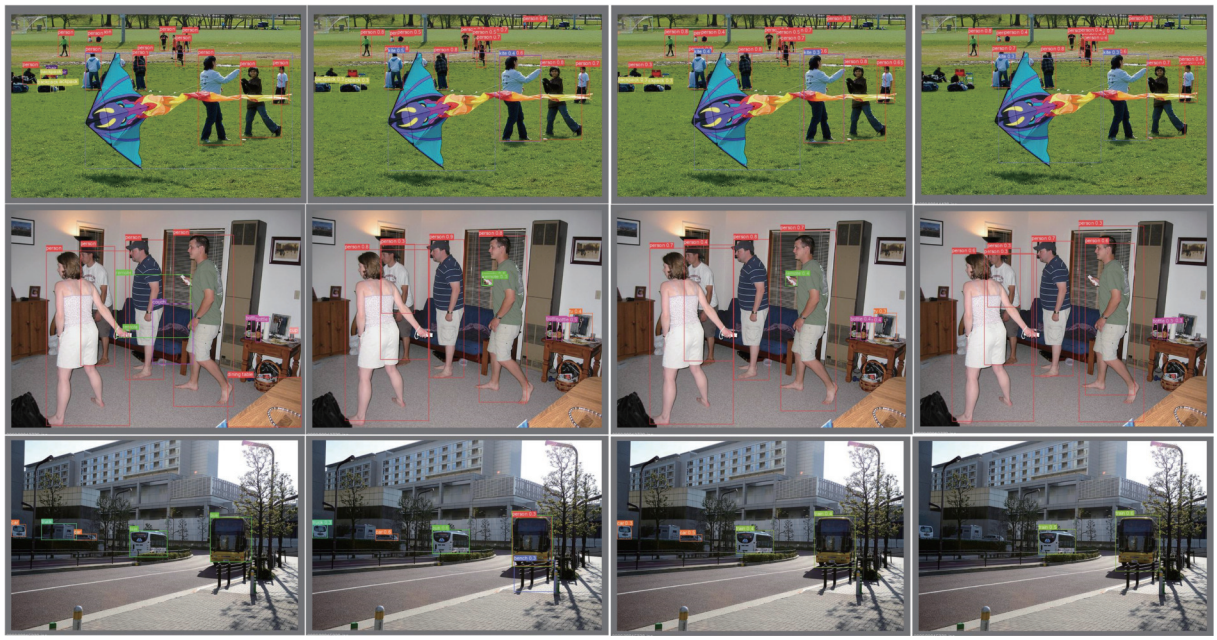
Tab.5 Test results of different quantification methods on COCO dataset and VOC2011 dataset

Dataset	Method	bit	γ	mAP@0.5	mAP@0.5-0.95	Model size
COCO	Unified bit	7		0.567	0.345	6.35
		6		0.503	0.301	5.45
		5		0.386	0.215	4.54
	Mixed bit	6.49	0.08	0.602	0.368	5.89
		5.57	0.125	0.546	0.322	5.05
		5.07	0.166	0.446	0.260	4.60
	Ori model	32		0.636	0.411	29.07
VOC2011	Unified bit	7		0.950	0.732	6.35
		6		0.925	0.643	5.45
		5		0.533	0.295	4.54
	Mixed bit	6.49	0.08	0.950	0.706	5.89
		5.57	0.125	0.981	0.669	5.05
		5.07	0.166	0.782	0.456	4.60
	Ori model	32		0.950	0.786	29.07

表 6 VOC2011 数据集类别精度检测表

Tab.6 VOC2011 dataset category accuracy detection table

Dataset	Method	bit	mAP@0.5	Aeroplane	Bicycle	Bird	Boat	Bottle	Chair	Dog	Person	Sheep	Train	Tvmonitor
VOC2011	Unite	5	0.782	0.753	0.435	0.497	0.995	0.801	0.995	0.249	0.897	0.995	0.995	0.995
	Mixed		0.533	0.232	0.324	0.497	0.484	0.209	0.995	0.332	0.455	0.995	0.995	0.34



(a) 真值 (a) Ground truth
(b) 全精度 (b) Full precision
(c) 平均 6 位量化 (c) Average 6-bit quantization
(d) 统一 6 位量化 (d) Unified 6-bit quantization

图 3 COCO 数据集检测结果示例

Fig.3 Example of COCO dataset detection results

6 位量化的检测结果；图 3(d) 为统一 6 位量化方法的检测结果。从图中可以看出，当 γ 为 0.09 时，模型压缩到 5.43 MB，即平均 6 位量化，对目标图像的检测效果与原始网络相当，而统一 6 位量化方法会丢失一些小目标与背景模糊物体的检测框。如第一张图片丢失了小目标背包的检测框，第二张图片丢失了桌子上的部分小目标检测框，对被遮挡人物的检测也有所损失。第三张图片中则损失了两辆背景模糊的车辆检测框。在第二张图片中，原始网络对于中间人物的检测出现精度损失，而 6 位混合量化方法恢复了人物的检测损失。

4 结束语

文中深入探讨了不同量化形式与量化方法对于卷积神经网络结果的影响，最终选择基于均方误差的截断方法作为文中的量化方法，同时，通过对网络量化过程中的舍入误差分析，提出基于误差限制的深度学习分层量化策略，采用误差限制因子 γ 对卷积层误差参数进行等比限制，得到不同卷积层的量化精度，并据此对网络参数进行混合截断量化。最终使用 YOLOV5 s 网络在 COCO 数据集和 VOC 数据集上进行测试并验证，与统一精度量化相比，YOLOV5 s 网络混合量化到 5 位精度分别提升了 6% 和 24.9%。目前，已通过算法实现并验证了混合精度量化方法在目标检测领域的应用，下一步工作将考虑对比尝试更多的量化方法、优化分层策略并在硬件端实现目标检测网络的混合位宽推理。

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 [J]. ***arXiv preprint***, 2014: 1409.1556.

[3] He K, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[4] Vanhoucke V, Senior A, Mao M Z. Improving the speed of neural networks on CPUs[C]//Advances in Neural Information Processing Systems, 2011.

[5] Gupta S, Agrawal A, Gopalakrishnan K, et al. Deep learning with limited numerical precision[C]//International Conference on Machine Learning, 2015, 37: 1737-1746.

[6] Jacob B, Kligys S, Chen B, et al. Quantization and training of neual networks for efficient integer-arithmetic-only inference[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 2704-2713.

[7] Cai Y H, Yao Z W, Dong Z, et al. ZeroQ: A novel zero shot quantization framework[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020: 13169-13178.

[8] Wang K, Liu Z J, Lin Y J, et al. HAQ: Hardware-aware automated quantization with mixed precision[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019: 8612-8620.

[9] Huang Z Z, Du H M, Chang L B. Mixed-clipping quantization for convolutional neural networks [J]. ***Journal of Computer-Aided Design & Computer Graphics***, 2021, 33(4): 553-559. (in Chinese)

[10] Zeng H Q, Hu H L, Lin X W, et al. Deep neural network compression and acceleration: An overview [J]. ***Journal of Signal Processing***, 2022, 38(1): 183-194. (in Chinese)

[11] Chen W L, Wilson J T, Tyree S, et al. Compressing neural networks with the hashing trick[C]//32nd International Conference on Machine Learning, 2015, 37: 2285-2294.

[12] Liu Z, Li J G, Shen Z Q, et al. Learning efficient convolutional networks through network sliming[C]//2017 IEEE International Conference on Computer Vision (ICCV), 2017: 2775-2763.

[13] Xu Y F, Zhang D Z, Wang L, et al. Lightweight feature fusion network design for local feature recognition of non-cooperative target [J]. ***Infrared and Laser Engineering***, 2020, 49(7): 20200170. (in Chinese)

[14] Lin M, Ji R, Wang Y, et al. HRank: Filter pruning using high-rank feature map[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020: 1529-1538

[15] He Y, Ding Y, Liu P, et al. Learning filter pruning criteria for deep convolutional neural networks acceleration[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 2020: 2006-2015.

[16] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning trained quantization and huffman coding[C]//Conference on Computer Vision and Pattern Recognition, 2016.

[17] Gong R, Liu X, Jiang S, et al. Differentiable soft quantization: Bridging full-precision and low-bit neural networks[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV), 2019: 4852-4861.

[18] Zhu F, Gong R, Yu F, et al. Towards unified int8 training for convolutional neural network[C]//Proceeding of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 2020: 1969-1979.

[19] Redmon J, Farhadi A. YOLOV3: An incremental improvement [J]. ***arXiv***, 2018: 1804.02767.