卷积神经网络混合截断量化

黄钲喆, 杜慧敏, 常立博*

(西安邮电大学电子工程学院 西安 710121)
(changlibo@xupt.edu.cn)

摘要: 量化是压缩卷积神经网络、加速卷积神经网络推理的主要方法。现有的量化方法大多将所有层量化至相同的位宽, 混合精度量化则可以在相同的压缩比下获得更高的准确率, 但寻找混合精度量化策略是很困难的。为解决这种问题, 提出了一种基于强化学习的卷积神经网络混合截断量化方法, 使用强化学习的方法搜索混合精度量化策略, 并根据搜索得到的量化策略混合截断权重数据后再进行量化, 进一步提高了量化后网络的准确率。在 ImageNet 数据集上测试了 ResNet18/50 以及 MobileNet-V2 使用此方法量化前后的 Top-1 准确率, 在 COCO 数据集上测试了 YOLOV3 网络量化前后的 mAP。与 HAQ, ZeroQ 相比, MobileNet-V2 网络量化至 4 位的 Top-1 准确率分别提高了 2.7% 和 0.3%; 与分层量化相比, YOLOV3 网络量化至 6 位的 mAP 提高了 2.6%。

关键词: 卷积神经网络; 混合精度量化; 强化学习; 混合截断
中图分类号: TP391.41 **DOI:** 10.3724/SPJ.1089.2021.18509

Mixed-Clipping Quantization for Convolutional Neural Networks

Huang Zhengzhe, Du Huimin, and Chang Libo*

(School of Electronic Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710121)

Abstract: Quantization is the main method to compress convolutional neural networks and accelerate convolutional neural network inference. Most existing quantization methods quantize all layers to the same bit width. Mixed-precision quantization can obtain higher precision under the same compression ratio, but it is difficult to find a mixed-precision quantization strategy. To solve this problem, a mixed-clipping quantization method based on reinforcement learning is proposed. It uses reinforcement learning to search for a mixed-precision quantization strategy, and uses a mixed-clipping method to clip weight data according to the searched quantization strategy before quantization. This method further improves the accuracy of the quantized network. We extensively test this method on a diverse set of models, including ResNet18/50, MobileNet-V2 on ImageNet, as well as YOLOV3 on the Microsoft COCO dataset. The experimental results show that our method can achieve 2.7% and 0.3% higher Top-1 accuracy on MobileNet-V2 (4 bit), as compared to the HAQ and ZeroQ method. And our method can achieve 2.6% higher mAP on YOLOV3 (6 bit), as compared to per-layer quantization method.

Key words: convolutional neural networks; mixed-precision quantization; reinforcement learning; mixed-clipping

收稿日期: 2020-07-22; 修回日期: 2020-12-09. 基金项目: 西安市科技计划(201805040YD18CG24(5)). 黄钲喆(1995—), 男, 硕士研究生, CCF 学生会员, 主要研究方向为计算机视觉、计算机体系结构; 杜慧敏(1966—), 女, 博士, 教授, 硕士生导师, CCF 会员, 主要研究方向为计算机视觉、计算机体系结构; 常立博(1985—), 男, 硕士研究生, 工程师, CCF 会员, 论文通讯作者, 主要研究方向为计算机视觉、计算机体系结构。

近年来,卷积神经网络在目标分类、目标检测、图像语义分割等方面得到广泛的应用,但是由于卷积神经网络的推理需要大算力和高带宽的支持,很难在嵌入式平台部署,限制了它们在无人机、机器人、自动驾驶和移动端 AR 等嵌入式系统中的应用.因此,如何使卷积神经网络满足这些系统的实时性、准确性和功耗等要求是工业界和学术界关注热点问题之一.目前最常用的一种方法是通过网络的参数进行量化,以满足嵌入式系统应用的需求.

目前的量化方法大多将不同层的权重数据都量化至相同的位宽^[1],但网络不同层中权重数据的分布范围与冗余度都不相同,因此有学者采用混合精度量化方法^[2],混合精度量化后卷积神经网络可以在相同的压缩比下获得更高的准确率.但是,找到卷积神经网络的分层量化策略是非常困难的,如经典的 ResNet18 网络^[3],它拥有 21 层网络,每一层的权重都可能量化至 1~8 位,那么可能的量化策略就有 8^{21} 种.这是一个巨大的搜索空间,在搜索过程中需要对比大量的数据,很难人工完成.

研究表明,对网络模型各层数据进行截断后再进行量化,可以提升量化后网络的准确率^[2-5].但不同位宽上的最佳截断方法是不同的,基于搜索得到的混合精度量化策略,灵活地选择不同的截断方法可以进一步提升量化后的网络准确率.

本文将寻找量化策略问题建模为强化学习问题,使用深度强化学习的方法来搜索混合精度量化的最佳策略,并针对混合精度量化的特点优化了线性量化方法,取得了更高的准确率.本文的工作包含以下几个方面:

- (1) 通过优化强化学习中的奖励函数,更好地搜索混合精度量化策略;
- (2) 根据混合精度量化策略对不同层网络中的权重数据进行混合截断,以减少量化带来的准确率损失;
- (3) 根据搜索得到的混合精度量化策略,分层线性量化卷积神经网络;
- (4) 在 ImageNet 数据集^[6]上对量化前后的目标分类网络进行了测试;在 COCO2014 数据集^[7]上对量化前后的目标检测网络进行测试.

1 相关工作

量化是卷积神经网络模型压缩的主要实现方法.2015 年, Han 等^[8]使用聚类的方法对网络中的

参数进行量化,并获得较高的准确率,开创了模型量化的先河.2017 年,谷歌使用仅使整数运算推理的量化方法^[9],使得神经网络在推理过程中只使用整数运算,降低了推理过程的存储资源需求与计算资源需求,便于卷积神经网络的硬件实现.因此,人们对于卷积神经网络模型量化的研究都主要集中在线性量化上.

线性量化将网络推理过程中所用到的权重与激活都从 32 位浮点数量化至低精度的整数,大幅降低保存参数的存储以及运算过程中参数交换的带宽需求;运算操作从浮点运算变为整数运算,减少了计算复杂度.线性量化分为分通道量化与分层量化 2 种方法,分通道量化通常能得到更高的准确率,如 4 位分通道量化方法^[10],但其在推理过程中需要更多的硬件开销^[11].分层量化的方法将各层统一地量化至 8 位或更低的位宽,如无数据量化(data-free quantization, DFQ)^[11]、离群值通道分离量化(outlier channel splitting, OCS)^[12]和参数化裁剪激活函数(parameterized clipping activation, PACT)^[13]等.

事实上,卷积神经网络各层的权重参数分布是不同的,如图 1 展示了 ResNet18 网络^[3]中不同层的权重数据分布状况.其中,横坐标为网络的各层,纵坐标为参数的数值大小.可以看出,各层权重数值分布范围有比较大的差异,那么,可以将数值分布范围大的层中的参数用较大的位宽表示,分布范围较小的层中的参数用较小的位宽表示.

图 1 ResNet18 网络不同层的权重分布

基于此,使用混合精度量化的硬件感知混合精度自动量化(hardware-aware automated quantization with mixed precision, HAQ)^[2]和零样本量化框架(a novel zero shot quantization framework, ZeroQ)^[14]方法,进一步提高了量化后网络的准确率.其中,ZeroQ^[14]采用帕累托边界的思想,通过对比各个量化策略对网络输出结果的影响,找出各位宽下的最优量化策略;这种方法需要统计大量的

量化策略对推理结果的影响, 才能找出最优策略. HAQ^[2]使用强化学习搜索量化策略的方法, 通过智能体自动搜索最佳量化策略, 但其奖励函数的设置只与准确率有关. 这导致搜索出的量化策略对应的模型大小会尽量接近设置的模型大小约束, 而量化的目标是在保证准确率的情况下尽量降低模型大小. 因此, 本文优化了奖励函数的设置, 使搜索出的量化策略能够在更小的模型大小下仍有较高的准确率.

2 基于强化学习的混合截断量化方法

2.1 基于强化学习的混合精度量化策略搜索

本文将寻找目标网络混合量化最佳策略的问题建模为强化学习问题, 使用强化学习的方法寻找最佳混合精度量化策略, 整体框架如图2所示. 根据智能体给出的量化策略对网络进行混合截断量化, 智能体则根据由准确率和压缩率组成的反馈进行学习, 找到最优混合精度量化策略.

图2 本文方法框架

智能体选用深度确定性策略梯度(deep deterministic policy gradient, DDPG)算法^[15], 其基于Actor-Critic 算法^[16], 结合深度Q网络(deep Q networks, DQN)^[17]的思想和确定性策略梯度(deterministic policy gradient, DPG)算法^[18], 解决了Actor-Critic 算法^[16]难以收敛的问题. 每一次动作为网络的一层分配位宽, 一个回合由多个动作组成, 完成网络中各层的位宽分配. 探索中的Q函数定义为 $\hat{Q}_i = r_i + \gamma Q(S_{i+1}, \mu(S_{i+1} | \theta^\mu) | \theta^Q)$. 损失函数定义为 $L = \frac{1}{N_S} \sum_{i=1}^{N_S} (\hat{Q}_i - Q(S_i, a_i | \theta^Q))^2$. 其中, N_S 为一个回合中动作的总数; S 为当前状态; μ 为当前采用的策略; r 为本次动作的奖励; θ 为Actor-Critic 网络中的参数; γ 为折扣因子. 本文认为, 每一层对最终结果的影响都是相同的, 因此在

实验中设置 $\gamma=1$.

DDPG 的动作空间是连续的, 量化位宽的分布是离散的, 使 $b = \text{round}(b_{\min} - 0.5 + a \times (b_{\max} - b_{\min}))$ 将连续的动作转换为离散的量化位宽. 其中, b 为位宽; a 为智能体做出的动作, 取值范围为 $[0,1]$; $b_{\min}$ 为最小位宽 1 位; $b_{\max}$ 为最大位宽 8 位. 为了保证模型大小可以控制, 本文根据量化策略计算量化后的模型的压缩率, 并对动作加以约束, 保证量化后的模型大小在约束范围内; 如平均量化至 4 位时, 探索中的模型最大压缩率被设置为 0.125.

采用分层量化的方法对卷积神经网络进行量化处理, 则智能体逐层处理网络中的权重数据. 定义第 i 层网络的状态空间 $S_i = (i, s_{\text{params}}, a_{i-1})$. 其中, i 为当前层数, s_{params} 为第 i 层的参数数量, a_{i-1} 为智能体对前一层网络给出的动作.

在强化学习中, 奖励函数的设置一直是影响强化学习收敛精度和训练速度的关键因素. 量化后模型的准确率与大小是评判强化学习探索结果好坏最直接且有效的 2 个参数. 因此, 本文使用准确率和压缩率 2 个参数构建奖励函数, 希望在模型较小的情况下仍获得较高的准确率. 奖励函数定义为 $R = \lambda_1 \times (A_q - A_{\text{ori}}) + \lambda_2 \times (r_q - r_c)$. 其中, A_q 为量化后的模型准确率; A_{ori} 为量化前模型的原始准确率; λ_1 为准确率的尺度因子; r_q 为量化后模型的压缩率; r_c 为压缩率的约束; λ_2 为压缩率的尺度因子.

为确定尺度因子 λ_1 与 λ_2 的取值, 在 ResNet50^[3] 量化至 4 位混合精度量化策略的探索中, 测试了不同比例的尺度因子对探索结果的影响, 如表 1 所示. 可以看出, 随着准确率尺度因子 λ_1 的减小与压缩率尺度因子 λ_2 的增加, 量化后网络的模型大

表1 尺度因子取值对比

λ_1 / λ_2	模型大小/MB	Top-1 准确率/%
1.0/0.0	12.186	73.5
0.9/0.1	12.157	73.4
0.8/0.2	12.118	73.5
0.7/0.3	12.177	73.3
0.6/0.4	12.167	72.4
0.5/0.5	12.108	74.2
0.4/0.6	12.030	73.2
0.3/0.7	12.028	71.2
0.2/0.8	11.231	72.6
0.1/0.9	10.324	72.6

小逐渐降低,而量化后模型的准确率并未随之降低.与使用未加入压缩率的奖励函数探索得到的量化策略相比,当 $\lambda_1=\lambda_2=0.5$ 时,量化后模型大小更小,但取得了更高的准确率.因此,在后续实验中,均设置 $\lambda_1=\lambda_2=0.5$.

2.2 混合截断量化方法

2.2.1 量化方法

线性量化^[9]在推理过程中只使用了整数,因此对于硬件实现更加友好和高效.本文根据智能体给出的策略,分层地对卷积神经网络进行线性量化.对第 i 层中的每一个权重数据 w ,都线性地量化到 b_i 位,量化方法为

$$w_q(w, c, s) = R(C(w, c) / s) \times s \quad (1)$$

其中, $R(x)$ 函数为四舍五入;

$$C(w, c) = \begin{cases} w, & \text{if } -c \leq x \leq c \\ -c, & \text{if } x < -c \\ c, & \text{if } x > c \end{cases};$$

c 为截断值; $s = 2c / (2^{b_i} - 1)$ 为量化的尺度因子.由于大部分网络的激活层都使用 ReLU 函数,因此每一层输出的函数的分布范围都为 $[0, c]$.对于激活的量化尺度因子 s ,本文定义 $s = c / (2^{b_i} - 1)$.YOLOV3 网络^[19]的激活函数使用了 Leaky ReLU,因此,对于该网络的激活采用与权重相同的量化方法.

2.2.2 混合截断

在观察网络中各层权重分布时,发现有一些离群值的存在,它们的数量很少,但影响了该层权重的最大值和最小值,增加了权重分布的范围,进而影响了量化后网络的准确率.因此在数据位宽较低时,对权重数据先截断再进行量化,可以减少量化带来的准确率损失.

本文在具有代表性的目标分类网络 ResNet50^[3]与 MobileNet-V2^[20]上测试了使用各种截断方法对量化后网络 Top-1 准确率的影响,对比结果如表 2 所示.可以看出,不使用截断的最大值-最小值量化在位宽较高时准确率最高,当位宽降至 7 位以下时,先截断再量化的方法由于减小了数据分布,准确率比不使用截断的量化方法更高.其中,基于均方误差(mean squared error, MSE)的截断方法在位宽为 5 位与 6 位时表现优于基于相对熵(Kullback-Leibler divergence, KLD)的截断方法,在位宽为 4 位时基于 KLD 的截断方法则更为有效.

表 2 不同截断方法对 2 种模型量化后网络准确率

模型	量化位宽	Top-1 准确率/%			最佳截断方法
		无截断	MSE	KLD	
ResNet50 ^[3]	8	74.5	72.8	67.2	无截断
	7	74.2	72.3	60.5	无截断
	6	70.0	71.5	63.3	MSE
	5	30.1	67.2	61.8	MSE
	4	0.2	52.3	59.4	KLD
MobileNet-V2 ^[20]	8	70.2	69.4	68.8	无截断
	7	67.3	65.3	63.5	无截断
	6	56.9	60.9	57.1	MSE
	5	5.1	35.4	30.6	MSE
	4	0.1	0.2	1.0	KLD

因此,根据分析提出混合截断方法对网络不同层中的参数进行处理:

(1) 当量化位宽为 7 位或 8 位时,不对数据进行截断处理.

(2) 当量化位宽为 5 位或 6 位时,采用基于 MSE 的截断方法

$$c = \arg \min_x d_{\text{MSE}}(w \| w_q(w, x, s)) \quad (2)$$

其中, $d_{\text{MSE}}(\cdot \| \cdot)$ 为计算 2 组数据分布之间的均方误差; w 为量化前的权重; w_q 为量化后的权重; c 为截断值,将式(1)代入式(2)确定 c 的取值.

(3) 当量化位宽为 4 位及以下时,采用基于 KLD 的截断方法

$$c = \arg \min_x d_{\text{KLD}}(w \| w_q(w, x, s)) \quad (3)$$

其中, $d_{\text{KLD}}(\cdot \| \cdot)$ 为计算 2 组数据分布之间的 KL 散度; w 为量化前的权重; w_q 为量化后的权重; c 为截断值,将式(1)代入式(3)确定 c 的取值.

3 实验结果与分析

3.1 实验环境

为了验证本文方法,在搭载显存为 22 GB 的 NVIDIA Tesla P40 的服务器上进行实验,系统环境为 Ubuntu16.04, Pytorch 版本为 1.3.1, Torchvision 版本为 0.4.2, Python 版本为 3.7. 目标分类微调训练中使用随机梯度下降(stochastic gradient descent, SGD)^[21]优化函数,学习率为 10^{-4} ,动量为 0.9;目标分类的网络结构与预训练模型使用 Torchvision 中提供的版本, YOLOV3 网络^[19]结构使用已有的 Pytorch 实现方法^①与 DarkNet 官网提供的预训练

① <https://github.com/eriklindernoren/PyTorch-YOLOv3>

模型. 目标检测微调训练中使用自适应矩估计 (adaptive moment estimation, Adam)^[22]优化方法, 学习率为 10^{-3} .

3.2 实验结果

3.2.1 目标分类网络

ResNet^[3]与 MobileNet-V2 网络^[20]是经典的目标分类网络, 并且是很多目标检测网络的主干网络. 因此本文在 ImageNet 数据集^[6]上测试了 ResNet18^[3], ResNet50^[3]和 MobileNet-V2 网络^[20]量化前后的准确率, 并与其他多种量化方法进行了对比. 实验结果表明, 混合精度量化可以在相同的网络大小下获得更高的准确率, 而本文的混合截断量化方法可以进一步提升混合精度量化的效果.

表 3 展示了对 ResNet18^[3]使用各种量化方法后在 ImageNet 数据集上的测试结果, 其中 MP 为混合精度量化. 可以看到, 与 DFQ^[11]和 RQ^[23]方法相比, 在平均位宽均为 6 位(模型大小为 8.34 MB)时, 本文方法可以得到更高的准确率, 分别提高了 3.1%和 0.7%, 而模型大小下降了 0.02 MB. 与精度损失较小的分通道量化方法^[8]相比, 在平均位宽均为 4 位(模型大小为 5.57 MB)时, 本文方法准确率提高了 0.1%, 并且在平均位宽为 3 位(模型大小为 4.42 MB)的条件下, 可以获得与 4 位分通道量化相当的准确率.

表 3 不同量化方法 ResNet18^[3]测试结果

量化方法	量化位宽	Top-1 准确率/%	模型大小/MB
DFQ ^[11]	6	66.3	8.36
RQ ^[23]	6	68.7	8.36
分通道量化 ^[10]	4	67.0	5.57
本文	MP(6)	69.4	8.34
	MP(4)	67.1	5.57
	MP(3)	67.0	4.42
原始模型	32	69.7	44.59

注. MP 后括号中数字为混合精度量化的平均位宽.

表 4 展示了对 ResNet50^[3]使用各种量化方法后在 ImageNet 数据集上的测试结果, 其中 MP 为混合精度量化. 可以看到, 与 OCS^[12]方法相比, 本文方法在模型大小降低 6.1 MB 的情况下, Top-1 准确率仅降低了 0.5%, 在平均位宽为 6 位(模型大小为 18.07 MB)时, 在模型大小降低 0.21 MB 的同时准确率提升了 0.4%; 而在平均位宽降低到 4 位(模型大小为 12.15 MB)时, 使用 OCS^[12]方法对网络进行量化得到的准确率会大幅降低, 而本文方法量化得到的准确率比 OCS^[12]方法提高了 8.0%, 并且

在平均位宽为 3 位(模型大小为 9.69 MB)时, 在模型大小更小(降低了 2.5 MB)的情况下取得了更高的准确率(提高了 7.7%).

表 4 不同量化方法 ResNet50^[3]测试结果

量化方法	量化位宽	Top-1 准确率/%	模型大小/MB
OCS ^[12]	8	75.7	24.37
	6	74.8	18.28
	4	66.2	12.19
本文	MP(6)	75.2	18.07
	MP(4)	74.2	12.15
	MP(3)	73.9	9.69
原始模型	32	76.1	97.49

注. MP 后括号中数字为混合精度量化的平均位宽.

表 5 展示了对 MobileNet-V2^[20]使用各种量化方法后在 ImageNet 数据集上的测试结果, 其中 MP 为混合精度量化. 当平均位宽为 6 位(模型大小 2.47 MB)时, 本文与 PACT 方法取得了相同的 Top-1 准确率, 但网络大小与之相比下降了 0.04 MB. 对于本身就是精简网络的 MobileNet-V2^[20], 将平均位宽降低到 6 位以下是非常困难的, 在平均位宽为 4 位(模型大小为 1.67 MB)时, 使用 PACT^[13]量化得到的 Top-1 准确率从 71.9%降至 61.4%, 降低了 10.5%. 而混合精度量化在相同位宽的条件下获得了更高的准确率, 使用 HAQ^[2]和 ZeroQ^[14]进行量化的准确率分别为 67.0%和 69.4%, 本文方法提高到了 69.7%. 而在平均位宽为 3 位(模型大小为 1.25 MB)时, 取得了 67.9%的准确率, 与 HAQ^[2]相比, 在模型大小更小(降低了 0.42 MB)的情况下准确率提高了 0.9%. 而在平均位宽为 6 位(模型大小为 2.47 MB)及以上的条件下, 各种方法准确率差距不大, 本文方法则以最小的模型大小(2.47 MB)获得

表 5 不同量化方法 MobileNet-V2^[20]测试结果

量化方法	量化位宽	Top-1 准确率/%	模型大小/MB
DFQ ^[11]	8	71.2	3.34
分层量化 ^[11]	8	70.9	3.34
PACT ^[13]	6	71.3	2.51
	4	61.4	1.67
HAQ ^[2]	MP(4)	67.0	1.67
ZeroQ ^[14]	MP(4)	69.4	1.67
本文	MP(6)	71.3	2.47
	MP(4)	69.7	1.67
	MP(3)	67.9	1.25
原始模型	32	71.9	13.37

注. MP 后括号中数字为混合精度量化的平均位宽.

了与其他方法相当的准确率 71.3%.

3.2.2 目标检测网络

YOLOV3 网络^[19]是目前一阶段目标检测网络中性能最好的网络之一, 本文在 COCO2014 数据集^[7]上测试了其量化前后的 mAP, 实验结果表明, 混合精度量化可以在相同的网络大小下获得更高的 mAP.

表 6 展示了对 YOLOV3^[19]使用各种量化方法后在 COCO2014 数据集^[7]上的测试结果, 其中 MP 为混合精度量化. 可以看到, 与统一分层量化相比, 平均位宽为 7 位和 6 位时, 本文方法 mAP 分别提高了 1.1%和 2.6%; 当模型平均位宽为 4 位时, 统一一位宽分层量化的 mAP 降为 9.7%, 本文方法仍取得 31.6%的 mAP.

图 3 展示了 YOLOV3^[19]量化至不同位宽的检测结果, 其中 MP 为混合精度量化, 从图中可以看出, 将网络量化至 6 位时, 混合精度量化的检测效

果与原始网络相当, 而 6 位统一量化的检测结果中则丢失了小目标与遮挡物体的检测框; 如第 1 张图片中未检测出交通灯, 第 2 张图片中未检测出被遮挡的马, 并出现了误检的情况. 将网络量化至 4 位时, 与统一量化相比, 混合精度量化则能检测到更多的目标; 如第 1 张图片中检测到了更多的车辆, 第 3 张图片中多检测出了狗, 检测效果更好.

表 6 不同量化方法 YOLOV3^[19]测试结果

量化方法	量化位宽	mAP/%	模型大小/MB
分层量化	7	49.0	51.6
	6	46.8	44.3
	4	9.7	29.5
本文	MP(7)	50.1	51.6
	MP(6)	49.4	44.1
	MP(4)	31.6	29.4
原始模型	32	51.5	236.1

注. MP 后括号中数字为混合精度量化的平均位宽.

图 3 量化至不同位宽检测结果示例

4 结 语

本文使用深度强化学习的方法搜索最优混合精度量化策略, 据此对网络参数进行混合截断并线性量化. 在多个卷积神经网络包括目标分类网络 ResNet18^[3], ResNet50^[3]和 MobileNet-V2^[20]上使用本文方法进行混合精度量化, 使用 ImageNet 数据集对量化后的网络进行测试验证; 采用目标检测网络 YOLOV3^[19], 在 COCO2014 数据集^[7]上对量化后的网络进行测试验证. 分别对上述网络进行 6 位、4 位、3 位的量化, ResNet18^[3]量化后的 Top-1 准确率分别为 69.4%, 67.1%和 67.0%; ResNet50^[3]量化后的 Top-1 准确率分别为 75.2%, 74.2%

和 73.9%; MobileNet-V2^[20]量化后的 Top-1 准确率分别为 71.3%, 69.7%和 67.9%. YOLOV3^[20]量化到 7 位、6 位、4 位的 mAP 分别为 50.1%, 49.4%和 31.6%, 与其他方法相比均能在相同或更小的网络大小下获得更高的准确率. 下一步工作将考虑优化分层线性量化方法, 进一步提升量化后网络的准确率, 并在深度学习处理器上进行量化后网络的推理.

参考文献(References):

- [1] Krishnamoorthi R. Quantizing deep convolutional networks for efficient inference: a whitepaper[OL]. [2020-07-22]. <https://arxiv.org/abs/1806.08342>
- [2] Wang K, Liu Z J, Lin Y J, et al. HAQ: hardware-aware auto-

- mated quantization with mixed precision[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 8612-8620
- [3] He K M, Zhang X Y, Ren S Q, *et al.* Deep residual learning for image recognition[C] //Proceedings of Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 770-778
- [4] Migacz S. 8-bit inference with tensorrt[OL]. [2020-07-22]. <https://on-demand.gputechconf.com/gtc/2017/presentation/s7310-8-bit-inference-with-tensorrt.pdf>
- [5] Banner R, Nahshan Y, Hoffer E, *et al.* ACIQ: analytical Clipping for Integer Quantization of neural networks[OL]. [2020-07-22]. <https://arxiv.org/abs/1810.05723v1>
- [6] Deng J, Dong W, Socher R, *et al.* ImageNet: a large-scale hierarchical image database[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2009: 248-255
- [7] Chen X L, Fang H, Lin T Y, *et al.* Microsoft COCO captions: data collection and evaluation server[OL]. [2020-07-22]. <https://arxiv.org/abs/1504.00325>
- [8] Han S, Mao H, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding[OL]. [2020-07-22]. <https://arxiv.org/abs/1510.00149>
- [9] Jacob B, Kligys S, Chen B, *et al.* Quantization and training of neural networks for efficient integer-arithmetic-only inference[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 2704-2713
- [10] Banner R, Nahshan Y, Hoffer E, *et al.* Post-training 4-bit quantization of convolution networks for rapid-deployment[OL]. [2020-07-22]. <https://arxiv.org/abs/1810.05723>
- [11] Nagel M, van Baalen M, Blankevoort T, *et al.* Data-free quantization through weight equalization and bias correction[C] // Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 1325-1334
- [12] Zhao R, Hu Y W, Dotzel J, *et al.* Improving neural network quantization without retraining using outlier channel splitting[OL]. [2020-07-22]. <https://arxiv.org/abs/1901.09504>
- [13] Choi J, Wang Z, Venkataramani S, *et al.* PACT: parameterized clipping activation for quantized neural networks[OL]. [2020-07-22]. <https://arxiv.org/abs/1805.06085>
- [14] 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. Los Alamitos: IEEE Computer Society Press, 2020: 13169-13178
- [15] Lillicrap T P, Hunt J J, Pritzel A, *et al.* Continuous control with deep reinforcement learning[OL]. [2020-07-22]. <https://arxiv.org/abs/1509.02971>
- [16] Degris T, White M, Sutton R S. Off-policy actor-critic[OL]. [2020-07-22]. <https://arxiv.org/abs/1205.4839>
- [17] Mnih V, Kavukcuoglu K, Silver D, *et al.* Playing Atari with deep reinforcement learning[OL]. [2013-12-19]. <https://arxiv.org/abs/1312.5602>
- [18] Silver D, Lever G, Heess N, *et al.* Deterministic policy gradient algorithms[C] //Proceedings of the 31st International Conference on International Conference on Machine Learning. New York: ACM Press, 2014: 387-395
- [19] Redmon J, Farhadi A. YOLOV3: an incremental improvement[OL]. [2020-07-22]. <https://arxiv.org/abs/1804.02767v1>
- [20] Sandler M, Howard A, Zhu M L, *et al.* MobileNetV2: inverted residuals and linear bottlenecks[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 4510-4520
- [21] Bottou L. Stochastic gradient descent tricks[M] //Neural networks: Tricks of the Trade. Heidelberg: Springer, 2012: 421-436
- [22] Kingma D P, Ba J. Adam: a method for stochastic optimization[OL]. [2020-07-22]. <https://arxiv.org/abs/1412.6980>
- [23] Louizos C, Reisser M, Blankevoort T, *et al.* Relaxed quantization for discretized neural networks[OL]. [2020-07-22]. <https://arxiv.org/abs/1810.01875>