软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(8):2391−2407 [doi: 10.13328/j.cnki.jos.006189]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种超低损失的深度神经网络量化压缩方法*

龚 成[1,2], 卢 冶[1,2], 代素蓉[1,2], 刘方鑫[1,2], 陈新伟[3], 李 涛[1,2,4]

[1](南开大学 计算机学院,天津 300350)

[2](天津市网络和数据安全技术重点实验室(南开大学),天津 300350)

[3](工业机器人应用福建省高校工程研究中心(闽江学院),福建 福州 350121)

[4](计算机体系结构国家重点实验室(中国科学院 计算技术研究所),北京 100190)

通讯作者: 卢冶, E-mail: luye@nankai.edu.cn

摘 要: 深度神经网络(deep neural network,简称 DNN)量化是一种高效的模型压缩方法,使用少量位宽表示模型计算过程中的参数和中间结果数据.数据位宽会直接影响内存占用、计算效率和能耗.以往的模型量化研究缺乏有效的定量分析,这导致量化损失难以预测.提出了一种超低损失的 DNN 量化方法(ultra-low loss quantization,简称 μL2Q),以揭示量化位宽与量化损失之间的内在联系,指导量化位宽选择并降低量化损失.首先,将原始数据映射为标准正态分布的数据;然后,在等宽的量化区间中搜索最优量化参数;最后,将μL2Q 方法融合进 DNN 的训练过程,并嵌入到主流的机器学习框架 Caffe 及 Keras 中,以支撑端到端模型压缩的设计和训练.实验结果表明,与最新的研究方法相比,在相同的位宽条件下,μL2Q 方法能够保证更高的模型精度,在典型的神经网络模型上精度分别提高了 1.94%,3.73%和 8.24%.显著性物体检测实验结果表明,μL2Q 方法能够胜任复杂的计算机视觉任务.

关键词: 神经网络压缩;神经网络量化;权值分布;均匀量化;量化损失最优解

中图法分类号: TP181

中文引用格式: 龚成,卢冶,代素蓉,刘方鑫,陈新伟,李涛.一种超低损失的深度神经网络量化压缩方法.软件学报,2021,32(8):2391−2407. http://www.jos.org.cn/1000-9825/6189.htm

英文引用格式: Gong C, Lu Y, Dai SR, Liu FX, Chen XW, Li T. Ultra-low loss quantization method for deep neural network compression. Ruan Jian Xue Bao/Journal of Software, 2021,32(8):2391−2407 (in Chinese). http://www.jos.org.cn/1000-9825/6189.htm

Ultra-low Loss Quantization Method for Deep Neural Network Compression

GONG Cheng[1,2], LU Ye[1,2], DAI Su-Rong[1,2], LIU Fang-Xin[1,2], CHEN Xin-Wei[3], LI Tao[1,2,4]

[1](College of Computer Science, Nankai University, Tianjin 300350, China)

[2](Tianjin Key Laboratory of Network and Data Security Technology (Nankai University), Tianjin 300350, China)

[3](Industrial Robot Application of Fujian University Engineering Research Center (Minjiang University), Fujian 350121, China)

* 基金项目: 国家重点研发计划(2018YFB2100300); 国家自然科学基金(62002175, 61872200); 天津自然科学基金(19JCZDJC31600, 19JCQNJC00600); 计算机体系结构国家重点实验室(中国科学院计算技术研究所)开放课题(CARCHB202016, CARCH201905); 中国高校产学研创新基金(2020HYA01003); 工业机器人应用福建省高校工程研究中心(闽江学院)开放基金(MJUKF-IRA1902)

Foundation item: National Key Research and Development Program of China (2018YFB2100300); National Natural Science Foundation of China (62002175, 61872200); Natural Science Foundation of Tianjin Municipality (19JCZDJC31600, 19JCQNJC00600); Open Fund of State Key Laboratory of Computer Architecture (Institute of Computing Technology, Chinese Academy of Sciences) (CARCHB202016, CARCH201905); Innovation Fund of Chinese Universities Industry-University-Research (2020HYA01003); Open Fund of Industrial Robot Application of Fujian University Engineering Research Center (Minjiang University) (MJUKF-IRA1902)

本文由“泛在嵌入式智能系统”专题特约编辑郭兵教授、王泉教授、邓庆绪教授、陈铭松教授、张凯龙副教授推荐.

收稿时间: 2020-07-21; 修改时间: 2020-09-07; 采用时间: 2020-11-02; jos 在线出版时间: 2021-02-07

© 中国科学院软件研究所 http://www.jos.org.cn

[4](State Key Laboratory of Computer Architecture (Institute of Computing Technology, Chinese Academy of Sciences), Beijing 100190, China)

Abstract: Deep neural network (DNN) quantization is an efficient model compression method, in which parameters and intermediate results are expressed by low bit width. The bit width of data will directly affect the memory footprint, computing power and energy consumption. Previous researches on model quantization lack effective quantitative analysis, which leads to unpredictable quantization loss of these methods. This study proposes an ultra-low loss quantization (μL2Q) method for DNN compression, which reveals the internal relationship between quantization bit width and quantization loss, effectively guiding the selection of quantization bit width and reducing quantization loss. First, the original data is mapped to the data with standard normal distribution and then the optimal parameter configuration is sought to reduce the quantization loss under the target bit width. Finally, μL2Q has been encapsulated and integrated into two popular deep learning training frameworks, including Caffe and Keras, to support the design and training of end-to-end model compression. The experimental results show that compared with the state-of-the-art three clusters of quantization solutions, μL2Q can still guarantee the accuracy and deliver 1.94%, 3.73%, and 8.24% of accuracy improvements under the typical neural networks with the same quantization bit width, respectively. In addition, it is also verified that μL2Q can be competent for more complex computer vision tasks through salient object detection experiments.

Key words: neural network compression; neural network quantization; weight distribution; uniform quantization; extremum of quantizationloss

随着深度神经网络(DNN)在多个研究领域取得实质性突破,边缘智能场景下 DNN 模型的应用和部署,吸引了研究人员的广泛关注.为了追求更高的推理精度,近年来,DNN 模型的计算规模变得愈加庞大、网络结构愈加复杂且不规则[12]、参数量巨大[3−13],其运行时需要强大的计算能力支持并且极其耗能[3].然而,边缘智能设备的计算资源与存储资源有限,并且对能耗及延迟具有严格的约束.因此,在资源受限的边缘设备上部署庞大而复杂的 DNN 模型极具挑战[11].将 DNN 模型进行压缩,可以有效减少模型的复杂度,使 DNN 模型得以应用于边缘智能计算场景.

DNN 模型压缩的主要目的是:在确保 DNN 模型推理精度的前提下,消除冗余的模型参数,减少中间结果并降低网络结构的复杂度,从而得到满足精度要求的精简模型.DNN 量化是 DNN 模型压缩的一种重要方法[14],它利用低位宽的参数来表示原始的全精度模型,显著降低 DNN 模型的计算复杂度和内存占用,使得 DNN 模型能够直接在资源受限的边缘设备上进行部署.当原始的模型被量化到极低的位宽时,模型压缩效果尤为明显[15−18].例如,二值神经网络 BNN[15]和三值神经网络 TWNs[16]可分别将 32 位全精度模型的尺寸压缩 32 倍和 16 倍.

但是,现有的 DNN 量化方法存在诸多问题,通常依赖于经验猜测和实验尝试[15−17,19],缺乏有效的理论支撑.具体表现如下:

- 第一,现有量化方法难以在数据位宽和模型精度之间进行有效权衡.通过极少量的比特位表示 DNN 模型,将导致模型精度明显下降而失去应用价值.例如,利用二值量化或三值量化的DNN可以获得极高的压缩比,但在实际应用中无法满足精度要求[15−18,20];而保留较多数量的比特位[19,21],虽然能够防止精度显著下降,但量化后的 DNN 模型仍然过于庞大,难以直接部署[22,23].
- 第二,启发式的参数选择需要大量的人工尝试,而现有的量化方法在将全精度模型数据转换成低位宽数据之前,需要寻找恰当的参数来限制中间结果数据的表示范围.该步骤通常需要引入缩放参数[18],而现有的量化方法在确定缩放参数时,往往是启发式的[17,22].
- 第三,对权值数据分布的拟合程度会直接影响量化后的模型精度,而现有量化方法通常会忽视权值数据的分布规律.例如,Dorefa-Net[17]和 FP[19]等工作由于量化后的权值不拟合原始的权值分布,从而导致精度显著下降.尽管 DNN 模型中各层数据的分布规律不同,但均可通过数学变换而近似满足标准正态分布.本文例举了 4 种典型的 DNN 模型的各层的权值数据,如图 1 所示,其数据皆近似服从标准正态分布.量化方法的设计应该充分利用权值数据分布规律,以有效提高 DNN 模型量化后的精度.

为了解决上述问题,本文提出一种超低损失的模型量化方法(ultra-low loss quantization,简称μL2Q).该方法在模型量化时充分考虑了原始权值数据的分布规律并进行了定量分析,并在确定缩放参数时有效地减少了人

© 中国科学院软件研究所 http://www.jos.org.cn

工实验次数,为模型量化提供了新的实现途径.

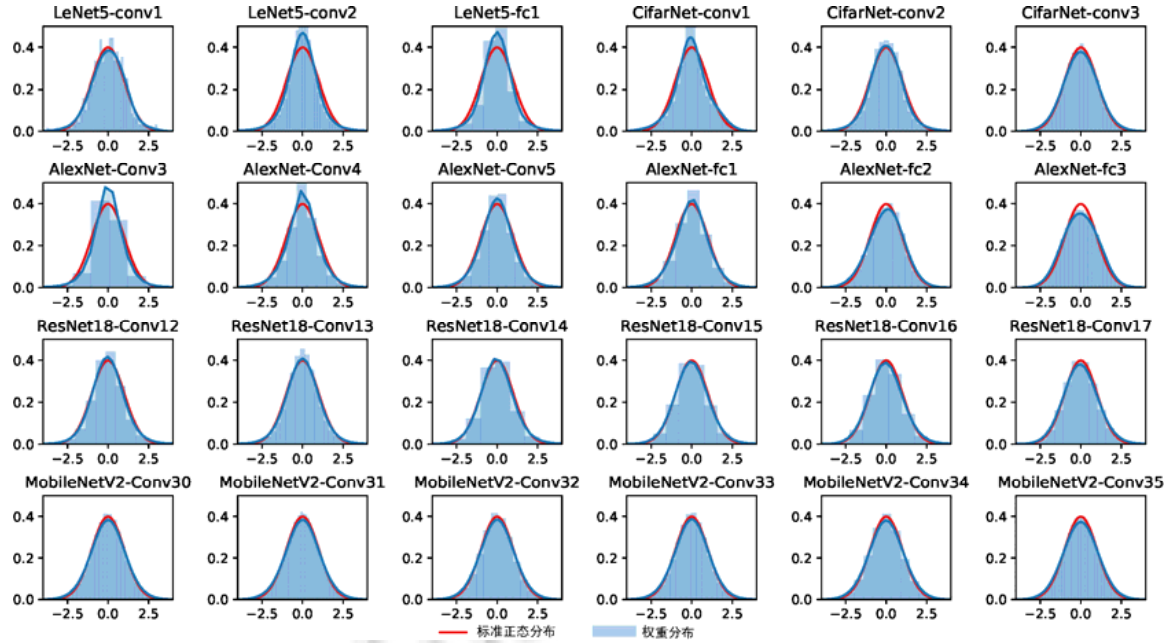


Fig.1 Distributions of layer weights of four DNN models

图 1 4 种 DNN 模型中各层的权值数据分布

本文的主要贡献包括 3 点:

- 提出了一种有效的超低损失量化方法,该方法可在给定位宽下,根据模型输入数据的分布规律,通过求解解析式的极值来获得最优量化参数配置,有效降低数据量化损失,以保证量化后的 DNN 模型推理精度;
- 设计了最优量化参数表,通过呈现各种位宽条件下的最低量化损失方案,为量化服从正态分布的权值数据提供了参考依据,有效地减少了启发式参数搜索的实验次数;
- 实现了基于μL2Q 的 DNN 模型训练方法,将μL2Q 嵌入到主流机器学习框架(如 Caffe,Keras)中进行推理精度的实验验证,以便捷的调用方式加速了 DNN 模型量化的端到端设计与训练流程.

1 相关工作与研究动机

首先,本节阐明了 DNN 模型权值数据的分布规律;然后,总结现有典型的模型量化方法的特点;最后讨论了本文的研究动机.为表达上简洁明了,本文采用 $w_f \in \mathbb{R}^d$ 表示权值向量,即全精度模型的权值数据;采用 $w_q \in \mathbb{R}^d$ 来表示经过量化后的权值向量,其中,d 代表权值数据的维度.

1.1 DNN权值数据分布

在 DNN 研究领域的权威著作《Pattern recognition and machine learning》[24]和《Machine learning: A probabilistic perspective》[25]中,皆假设 DNN 权值数据服从正态分布来进行相关研究.具体来讲,对于任意给定的 DNN 模型,设其输入特征为 X,分类结果为 y,w 为模型的权值,根据贝叶斯后验概率理论[24,25],则有:

$$p(w|X,y) \propto p(X,y|w)p(w) \tag{1}$$

其中,$p(w|X,y)$是在数据集$\{X,y\}$上计算得到 w 的最大后验概率,$p(X,y|w)$是似然函数,$p(w)$是 w 的先验概率.假设 w 满足正态分布,即 $p(w_i)=N(w_i|\mu,\sigma^2)$,则式(1)中的最大似然对数函数为

© 中国科学院软件研究所 http://www.jos.org.cn

$$l(w)=\log(p(X,y\mid w)p(w))=\log(p(X,y\mid w))-\lambda\sum(w_i)^2,\mu=0,\sigma=\sqrt{\frac{1}{\lambda}} \tag{2}$$

其中,$\lambda\sum(w_i)^2$ 是 *L2* 正则项,通常用于防止模型产生过拟合现象,增强模型泛化能力.在 w 满足正态分布的假设前提下,在模型训练过程中可利用 *L2* 正则化进行约束,以此得到最终结果.

1.2 典型量化方法

现有的典型量化方法可归纳为 3 种类型,即二值量化、三值量化和定点数量化.

- 二值量化[15,17,18]

该方法用 1 个比特来代替原始的全精度浮点数.经典的二值量化方法如式(3)所示[18].

$$w_q=E(|w_f|)\times sign(w_f) \tag{3}$$

其中,$E(|w_f|)$表示全精度权值 w_f的绝对值的平均值,它可被用作缩放参数. $sign(w_f)=2I_{(w_f\geq 0)}-1$,该函数的返回值可能为 1 或−1.二值量化方法能够达到 32 倍[15]的模型压缩率,并且可通过二进制操作完全代替乘法计算[18],显著降低存储空间并提高计算效率.但是,二值量化方法往往需要依靠工程经验来选取缩放参数,需要大量的人工尝试.此外,仅使用 1 比特表示权值将严重降低模型精度.

- 三值量化

受二值量化的启发,三值量化方法利用 3 个值,即{−1,0,+1},并结合缩放参数[16,26,27]表示网络模型,即

$$\begin{cases}\alpha, & w_f>\Delta\\ 0, & |w_f|\leqslant\Delta\\ -\alpha, & w_f<-\Delta\end{cases} \tag{4}$$

其中缩放参数为

$$\alpha=E(|w_f(i)|),\ i\in\{i\,||w_f(i)|>\Delta\} \tag{5}$$

$\Delta=0.7E(|w_f|)$[16]为阈值.三值量化使用 2 个比特来表示原始 DNN 模型的数据,并通过计算 w_f和 w_q之间的最小 *L2* 距离来获得数据量化后的结果.但是,三值量化模型的推理精度下降仍然十分明显.

- 定点数量化

定点数量化由于实现过程较为简单,因此成为最常用的量化方法[17,19,28].该方法的总体思路是,按一定的比特位数来保留数据中的整数部分和小数部分.将 w_f量化为 k 比特的数据 w_q可表示为

$$\begin{cases}p=\left\lfloor\log_2(\max(|w_f|))\right\rfloor-(k-2)\\ w_q=\left(\dfrac{w_f}{2^p}\right)\times 2^p\end{cases} \tag{6}$$

其中,w_f是待量化的输入数据;max(·)函数用来计算最大值;p 代表需要进行移位的比特数量,计算 p 的作用在于有效防止数据溢出.定点数量化方法中,高比特位具有高优先级,所以需要优先保留$|w_f|$中的最大值 max($|w_f|$),由此计算相应的 p 值,以确保最大的绝对值 max($|w_f|$)的高位 $n=\max\{i:\max(|w_f|)>2^i\}$不会被舍去而导致较大的舍入误差.如式(6)所示,$n$ 可由 $\log_2(\cdot)$函数计算.$R(\cdot)$代表舍入操作,即丢弃浮点数的小数部分.由于符号位占用 1 比特,对 w_f应用 k 比特的定点数量化方法,则需要保留从第 n 比特开始的连续 k−1 个比特位,即从 n−(k−2)至 n 位的总共 k−1 个比特,并舍去其他比特位,因此定点数的总比特数量为 k−1 个.

1.3 研究动机

从上文所述可知,以往的量化方法依赖人工选择或经验猜测,需要通过反复实验才能确定相关参数,缺乏行之有效的理论分析和实践指导,从而导致这些方法难以兼顾量化位宽和模型精度.如上文所述,二值量化和三值量化会导致模型精度急剧下降;而定点数量化虽在高位宽时表现良好,但针对低位宽量化时也会失效[17,19].这些方法忽视了权值数据的分布规律,缺乏对量化损失的准确评估.由此可见,仅靠减少数据位宽来进行模型量化将很难保证模型精度.对此,本文在模型量化时充分考虑权值数据的分布规律,并建立相应的定量评估方法来衡量

© 中国科学院软件研究所 http://www.jos.org.cn

模型量化的损失;然后,结合对结果值的分析和择优,通过不同数据位宽下的最优量化结果来揭示数据位宽和量化损失之间的关系.

此外,一些方法尽管能够量化模型并得到较好的模型精度,但是量化后的权值由于是浮点数类型,所以难以加速计算过程.例如,文献[29]所提出的权值共享并结合知识蒸馏的手段,在其利用 *k*-means 算法进行实现时,需要处理复杂的聚类迭代,而聚类中心是非均匀的浮点型数值,其计算过程仍然是典型的浮点数计算.需要说明的是:知识蒸馏方法虽与量化方法并不冲突,但是知识蒸馏往往需要更庞大的教师网络进行指导,将极大增加训练时间,因而许多量化工作并未主动使用知识蒸馏来提升模型量化的精度.为了与其他量化方法进行客观公正的比较,本文暂不引入知识蒸馏来提升量化模型的精度.

综上所述,本文在模型量化时除了考虑数据分布对模型精度的影响外,还需兼顾量化后权值是否有利于模型的计算加速.即在量化过程中,本文充分考虑 DNN 模型各层权值的数据分布规律,减少模型量化造成的精度损失,并采用均匀量化的方式,通过简洁的舍入操作,将原始权值量化为间隔均匀的定点数,来加速量化模型的乘法计算.

2 μL2Q 量化设计

本节阐述模型量化的精度损失评估方法.首先定义了量化损失,然后论述了μL2Q 量化思想的依据,并据此提出基于权值数据分布分析的μL2Q 量化过程的形式化表达.

2.1 损失评估

经过模型量化后的权值数据由 w_q 表示,则量化的一般过程可由式(7)表示.

$$w_q = Quantize(w_f) \text{ s.t. } w_q \in Q^d, Q = \{q_1, q_2, ..., q_n\} \tag{7}$$

其中,$Quantize(\cdot)$是量化函数,它将输入数据 $w_f \in \mathbb{R}^d$ 的空间映射到仅由有限的且间隔均匀的离散数据所构成的数据空间.在式(7)中,$n=2^k$,k 是量化位宽.基于前人的研究工作[16,18,21,30],本文将量化损失定义为 w_f 和 w_q 的欧式距离的平方,即 *L2* 距离:

$$J = \| w_q - w_f \|_2^2 \tag{8}$$

考虑到 DNN 模型自身的复杂性和不确定性,本文使用距离 J 作为量化损失评估依据而非模型的最终推理精度,这样能够更为准确地反映量化方法所产生的直接结果.

2.2 数据空间转换

对于给定的满足正态分布的权值 $w_f=N(\mu,\sigma^2)$,将其转换成标准正态分布的过程可表示为

$$\varphi = \frac{w_f - \mu}{\sigma} \sim N(0,1) \tag{9}$$

μL2Q 的量化结合了数据标准正态分布的规律,则缩放参数α和偏移参数β可表示为

$$\begin{cases} \alpha = \lambda\sigma \\ \beta = \mu \end{cases} \tag{10}$$

其中,λ是量化参数,通过λ将数据缩放到恰当范围.则式(9)可转换为

$$\frac{w_f - \beta}{\alpha} = \frac{\varphi}{\lambda} \sim N\left(0, \frac{1}{\lambda^2}\right) \tag{11}$$

由此,μL2Q 的量化思路可归纳为:利用μ和σ将一组服从或近似服从一般正态分布的数据转换为服从标准正态分布的数据,再利用λ将其缩放到恰当的给定范围,最后通过截断和舍入操作将其映射到整数值空间.

2.3 量化过程

μL2Q 量化思路的关键环节是将满足标准正态分布的权值分割为等宽的量化区域,即均匀的量化间隔.实际上,第 2.2 节中的λ即可表示量化区域的分割宽度,即量化间隔.如图 2 所示,量化过程是:首先,假设量化目标的

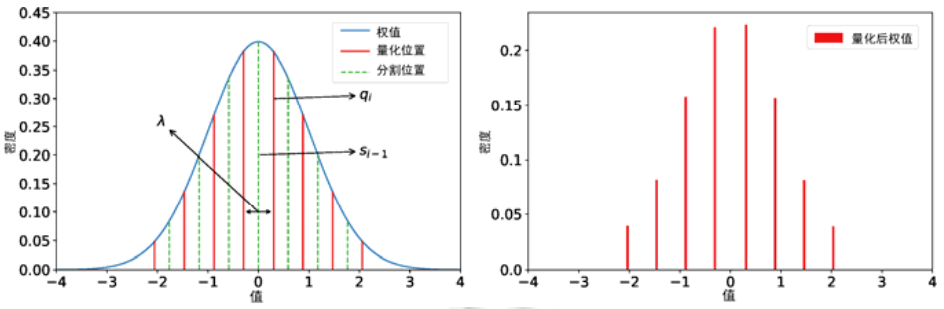
© 中国科学院软件研究所 http://www.jos.org.cn

位宽是 k,式(9)中的φ空间被分割成宽度为λ的 2^k 个量化区域;然后将每个量化区域中所有的权值替换为该区域中的某个最优值.经过标准正态分布的数据空间转换之后,μL2Q 的量化过程可总结为以下 3 个步骤.

(1) 如图 2(a)所示,将原始全精度浮点数据 w_f 量化为 k 个比特位所表示的离散数据 w_q,即被量化为 $n=2^k$ 个数.φ被分割为 n 个区域$\{R_1,R_2,\ldots,R_n\}$,R_i和 R_{i+1}区域的分割边界为 s_i.为了方便表示,令 $s_0=-\infty,s_n=+\infty$,S 表示分割边界的集合:$S=\{s_0,s_1,\ldots,s_n\}$.

(2) 对于每个量化区域 $R_i(i=1,2,\ldots,n)$,在其中搜索最优的量化值 q_i.q_i 的计算方法将在第 3 节具体阐述.

(3) 利用所选取的最优量化值 q_i 来代替区域 R_i 中的所有值,最终得到量化后的数据空间,如图 2(b)所示.



(a) 通过分割边界 S 将φ分割为 n 个量化区域,并在量化区域 R_i 内选择最优的量化值 $q_i\in Q$

(b) 用 q_i 替换 R_i 中的所有值

Fig.2 Quantization process of μL2Q

图 2 μL2Q 的量化过程

至此,DNN 模型的权值数据被量化成了 n 个值,可表示为 $Q=\{q_1,q_2,\ldots,q_n\}$.利用μL2Q 具有均匀的量化间隔λ的特征,即$|q_2-q_1|=\ldots=|q_n-q_{n-1}|=\lambda$,可以进行相应的求解.如前文所述,尽管可以通过复杂的精细化的非均匀划分来得到较高的量化模型精度,但是这样的设计思路难以在硬件上真正实现,仅仅只能起到压缩模型的效果,而本文采用均匀划分的方式,能够在压缩率、模型精度和硬件计算加速上达到很好的平衡.

二值、三值和定点数量化时往往需要特定的缩放参数,但其缩放参数无法直接与量化损失建立关联.相比之下,μL2Q 所需的缩放参数和偏移参数与量化损失直接相关,其关联关系如式(12)所示.

$$\begin{cases} w'_q = C\left(R\left(\dfrac{w_f-\beta}{\alpha}\right)-\dfrac{1}{2},-2^{k-1},2^{k-1}-1\right)+\dfrac{1}{2} \\ w_q = \alpha w'_q+\beta \end{cases} \tag{12}$$

其中,$C(\cdot)$是截断操作,它有 3 个参数,若其第 1 个参数的值超出由第 2 个参数和第 3 个参数所表示的范围,即$[-2^{k-1},2^{k-1},-1]$,则将第 1 个参数转换为离其最近的边界值(-2^{k-1} 或 $2^{k-1}-1$).$R(\cdot)$是舍入操作,其作用是舍弃浮点数的小数部分.偏移量 1/2 的作用是可将缩放($1/\lambda$倍)后的量化位置移动 1/2 个区域,借此防止形成以 0 为中心的对称量化值,从而保证能够获得 2^k 个有效的量化值,减少模型量化带来的直接损失.值得注意的是:由于式(12)中使用舍入操作 $R(\cdot)$来实现量化,而 $R(\cdot)$是不可导的,所以式(12)不可导.这将导致在模型训练时无法收敛,因此还需关注如何实现可导的量化过程,本文利用直通量估计来解决这一问题,具体步骤将在第 3.3 节中阐释.

3 μL2Q 量化实现

本节讨论如何对量化参数λ进行最优分析和最优值计算,并结合算法伪代码来阐述μL2Q 量化方法的实现步骤;最后,本节阐明将μL2Q 融合到典型 DNN 框架 Caffe 及 Keras 来进行模型训练的关键环节.

3.1 最优值分析

由第 2.2 节可知,λ是μL2Q 量化方法中的关键参数,与量化损失评估直接相关.它将标准正态分布的数据缩放到恰当的范围,以确保目标数据范围能够覆盖指定位宽所表示的量化范围$[-2^{k-1}+1/2,2^{k-1}-1/2]$.如图 3 所示:当

© 中国科学院软件研究所 http://www.jos.org.cn

λ=2 时,数据只能被量化为 4 个定点数{−1.5,−0.5,0.5,1.5}.当 k>2,将出现位宽空闲,未能表示有效数据的比特位将会造成位宽浪费.因此,需要更紧致的量化间隔,比如λ=1 或者λ=0.5.

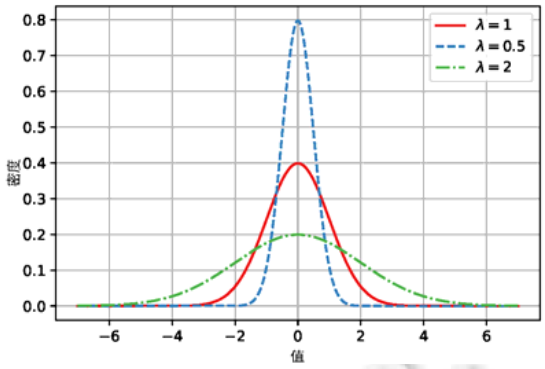


Fig.3 λ can scale the standard normal distribution to different ranges

图 3 标准正态分布由λ放缩到不同范围

μL2Q 的量化值择优目标是获得式(8)中量化损失为最小的值,可表示为

$$\lambda^* = \arg\min_{\lambda}(J) \tag{13}$$

其中,λ^*是使得量化损失最小的缩放参数 与穷举求解的方法[31]不同,本文探索在已知的权值数据分布下,求解λ^*的最优解析解.量化损失可进一步表示为

$$J = \| w_q - w_f \|_2^2 \propto \| \varphi - w_q' \|_2^2 = d\sum_{i=1}^{n}\int_{t\in R_i}(t-q_i)^2 p(t)\mathrm{d}t \propto \sum_{i=1}^{n}\int_{s_{i-1}}^{s_i}(t-q_i)^2 p(t)\mathrm{d}t \tag{14}$$

其中,d 是φ的维度,s_i 是分割边界,q_i 是每个量化区域 R_i 中的量化值,$p(t)$=$N(t;0,1)$.为了求解式(13),还需要定义式(14)中 s_i 和 q_i 的具体值,它们只依赖于λ.

$$s_i = \begin{cases} -\infty, & i=0 \\ \left(i-\dfrac{n}{2}\right)\lambda, & \text{others} \\ +\infty, & i=n \end{cases},\quad q_i = \left(i-\frac{n}{2}+\frac{1}{2}\right)\lambda, i=1,2,...,n \tag{15}$$

其中,n=2^k,k 为量化位宽.

至此,式(14)中的量化损失 J 只与参数λ相关,并且分析可知,J 是关于λ的可导凸函数.因此,通过求解量化损失的极值即可得到量化损失的最小值.μL2Q 的量化损失 J,在不同量化位宽 k 下关于λ的变化曲线如图 4 所示,红色五角星代表相应曲线的极值点.

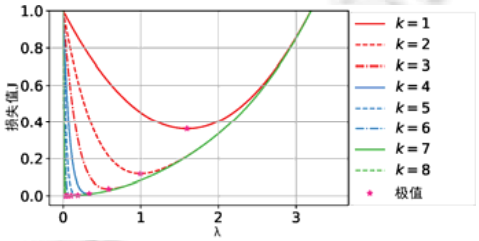


Fig.4 Curve of quantization loss J with different λ

图 4 量化损失 J 关于λ值的曲线

由图 4 可知,在各种位宽条件下,J 都存在最小值,即λ都存在最优解.通过上述方法可计算 1~8 比特位宽下的λ的最优解,并能够获得相应的最小量化损失,其结果见表 1.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 1 Optimal value of λ for different bit widths

表 1 不同位宽对应的最佳λ值

k(位宽)	1	2	3	4	5	6	7	8
λ	1.595 8	0.995 7	0.586 0	0.335 2	0.188 1	0.104 1	0.056 9	0.030 8
Loss	0.363 4	0.118 8	0.037 4	0.011 5	0.003 5	0.001 0	0.000 3	0.000 1

3.2 算法实现

μL2Q 的实现流程如算法 1 所示:首先,将满足正态分布的输入数据转换为标准正态分布,其中,需要记录偏移量$\beta=\mu$和标准差σ,然后,基于该标准正态分布,分析和计算最优参数λ的值,得到缩放参数$\alpha=\lambda\sigma$.

算法 1. μL2Q 算法.

输入: $w_f\sim N(\mu,\sigma^2),k\geqslant 1$.

输出: $w'_q,\alpha,\beta,w_q=\alpha w'_q+\beta$.

1: **if** k>8 **then**

2: $\lambda_k=\dfrac{\max(w_f)-\min(w_f)}{2^k-1}$

3: **else**

4: 从表 1 中获取λ_k

5: **end if**

6: $\alpha=\lambda_k\sigma,\beta=\mu$

7: $\varphi=\dfrac{w_f-\beta}{\alpha}-\dfrac{1}{2}$

8: $w'_q=C(R(\varphi),-2^{k-1},2^{k-1}-1)+\dfrac{1}{2}$

3.3 模型训练

模型训练的过程往往十分耗时,本文将μL2Q 量化方法融合进主流机器学习训练框架如 Caffe[32]和 Keras[33]来加速模型量化的实现速度,借此来满足广泛的 DNN 模型量化需求.具体来讲,对于 DNN 模型中的每一层 l,在其前向传播时,首先使用μL2Q 量化方法将 $w_f(l)$量化到 $w_q(l)$,然后使用 $w_q(l)$计算每一层在前向传播中的输出,得到前向传播的最终 DNN 损失 $L(w_q)$.在反向传播时,由于式(12)和算法 1 中的量化过程不可导,本文采用直通量估计(straight through estimation,简称 STE)来实现μL2Q 量化.STE 在许多量化工作中[16,17,19]被广泛应用于可训练的量化实现,其将 $w_q(l)$对 $w_f(l)$的梯度设定为固定数值 1,然后利用链式求导法则计算网络损失 $L(w_q)$对 w_f的梯度,以此来进行量化模型的训练.基于 STE 和链式求导法则,可以计算 $L(w_q)$对 w_f的梯度如下:

$$g(w_f(l))=\frac{\partial L(w_q)}{\partial w_f(l)}=\frac{\partial L(w_q)}{\partial w_q(l)}\times 1 \tag{16}$$

4 实验评估

本节从数据仿真实验和真实的 DNN 模型精度实验两个方面来评估μL2Q.其中,数据仿真用于对比现有的典型量化方法与μL2Q 的量化损失情况.DNN 模型精度实验选取了图像分类和显著性物体检测两类真实应用.在精度评估实验中,本文将μL2Q 融合进 DNN 训练框架 Caffe 和 Keras 中,加速 DNN 模型量化实现,利用模型的实际推理精度来观测μL2Q 的真实效果.除了权值量化的实现,本文还在 Keras 上实现了激活值量化,以验证μL2Q 在激活值量化方面的性能.Caffe 框架和 Keras 框架分别采用 1.0.0 版本和 2.2.4 版本.μL2Q 量化方法对于框架融合并无挑剔,从其实现过程和实验结果上来讲,使用 Caffe 和 Keras 框架,两者并无显著差异.

μL2Q 和定点数量化可以实现任意位宽的数据量化,μL2Q-x 和 Fixed-x 分别表示两种方法在位宽为 x 时,数据仿真的量化效果.在 DNN 模型精度评估中,μL2Q-x 和 Fixed-x 表示将模型的权值数据(包括卷积层和全连接层

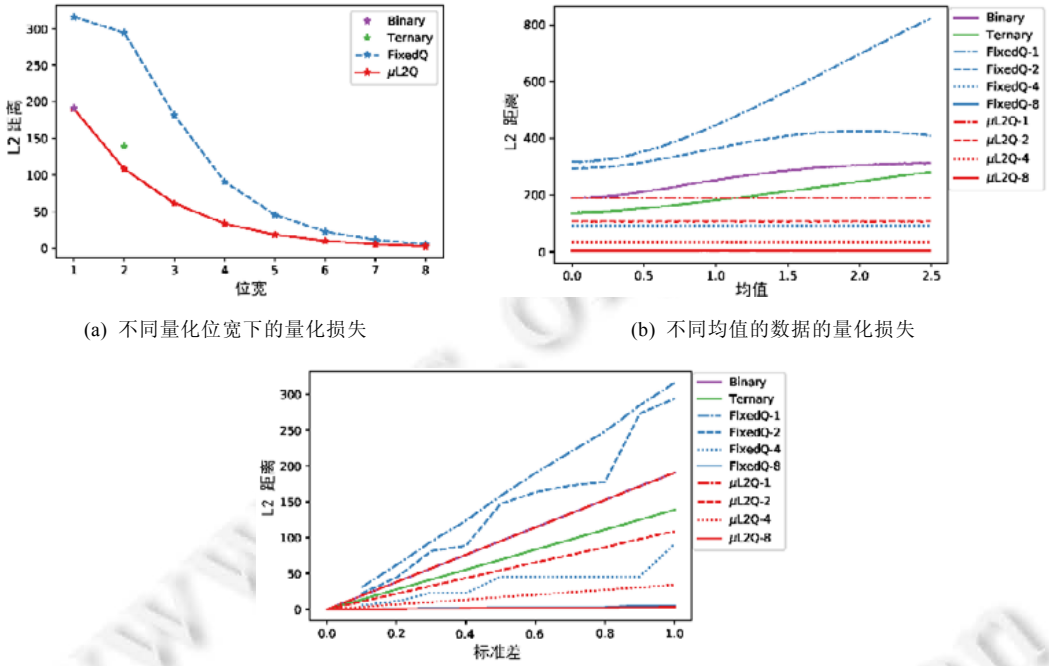
© 中国科学院软件研究所 http://www.jos.org.cn

的权值数据)量化到 x 比特.

4.1 数据仿真评估

- 实验设置

首先,随机生成满足正态分布 $N(\mu,\sigma^2)$的 100 000 个数据.分别考量条件参数的变化对量化损失所产生的影响:(1) 不同的量化位宽;(2) 满足标准差σ=1.0 但均值μ不同的正态分布;(3) 满足均值μ=0,但标准差σ不同的正态分布.图 5 展示了μL2Q 在仿真数据上的结果.



(a) 不同量化位宽下的量化损失

(b) 不同均值的数据的量化损失

(c) 不同标准差的数据的量化损失

Fig.5 Quantization losses for data with different means and standard deviations

图 5 具有不同均值和标准差的数据的量化损失

- 不同的量化位宽

如图 5(a)所示,μL2Q 能在任何给定位宽下保证最低的量化损失.在二值量化时,量化后的数据仅占用 1 比特,而μL2Q 能和文献[18]保持相同的量化损失.三值量化的数据只占用 2 比特,μL2Q 的量化损失比三值量化方法[16]的损失更低.这是因为三值量化的 2 个比特只存储了 3 个值{−1,0,1},而μL2Q 使用 2 比特存储 4 个值{−2,−1,0,1},能够表示更多数据信息.由图 5(a)可知,μL2Q 的量化损失在任意位宽下都比定点数量化的结果要低.其原因是:定点量化没有考虑数据分布规律,而仅仅针对数据本身进行了处理;而μL2Q 利用数据正态分布的特性,并将其贯穿于整个量化实现过程.

- 不同均值和标准差

如图 5(b)所示:当均值偏离原始位置时,二值量化和三值量化的损失都会增加.对于定点数量化,由于符号位的设置是不支持 1 比特的量化,但仍把这个值放在图中便于比较.由图 5(b)可知,定点数量化在位宽为 2 时波动很大,但是在 4 和 8 位时表现稳定.这是因为拥有足够的比特数可以表示更多的数据.而相比之下,μL2Q 的量化损失不随均值的变化而变化.此外,μL2Q 的 1 比特量化甚至可以达到定点数 4 比特量化的效果.如图 5(c)所示:相同位宽下,μL2Q 在不同标准差下的量化损失仍要比其他方法要低.由此可知,不同均值和不同标准差条件下,μL2Q 量化方法与其他方法相比,性能表现最为稳定.

© 中国科学院软件研究所 http://www.jos.org.cn

4.2 图像分类实验

图像分类任务能够体现模型量化方法的性能,可用于评估μL2Q 的有效性和优势.实验中,DNN 模型中所有卷积层和全连接层(包括第 1 层和最后一层)的权值数据都被量化到相同的位宽.

- 数据集与模型

本文实验选取代表性数据集 MNIST[34]、Cifar10[35]和 ImageNet[36],详细信息见表 2.选取几种应用广泛的 DNN 模型来进行量化效果的评估,包括应用于 MNIST 的模型 Lenet-5[34]、应用于 Cifar-10 上的模型 CifaNet[35], VGG-like[3]以及应用在 ImageNet 上的模型 AlexNe[37],Resnet-18[4]和轻量级深度模型 MobileNetV2[38].表 3 中展示了实验中用于评估的 DNN 规模(参数量)和具体训练参数.

Table 2 Dataset attributes

表 2 数据集信息

	数据集		
	MNIST	Cifar10	ImageNet
图像尺寸	28×28×1	32×32×1	224×224×3
分类数量	10	10	1000
图片数量	60 000	50 000	1 281 167
总像素数量($\log_{10}(\cdot)$)	7.67	8.19	11.29

Table 3 Model size and training parameter setting

表 3 模型规模及训练参数

	模型					
	Lenet5	CifarNet	VGG-like	AlexNet	Resnet-18	MobileNetV2
参数量(M)	1.67	0.279	5.36	50.88	11.69	3.54
权值衰减	0.000 4	0.000 1	0.000 4	0.000 1	0.000 1	0.000 1
批大小	100	100	100	256×4	256×2	256×2
初始学习率	0.1	0.1	0.2	1.0	1.0	1.0
学习率衰减率	0.1	0.1	0.2	0.1	0.1	0.1
学习率衰减时间	32,48	120,130	250,290	50,60,65	50,60,65	50,60,65
动量	0.9	0.9	0.9	0.9	0.9	0.9

为了与最新的方法进行公平对比,实验时,对 CifarNet 使用 TWN[16]中应用的数据增强方法,对 ImageNet 使用 Tensorflow 标准库中使用的数据增强方法.本文基于 Keras 框架进行权值量化和激活值量化等实验,由于 Caffe 自身难以实现复杂数据集的数据增强,所以本文仅在 Caffe 上测试了 LeNet5 的权值量化的结果.

- 评估

MNIST 和 Cifar10 数据集由于分类较少,仅有 10 个类别,因此只用 Top1 的分类精度作为 DNN 模型评估指标.ImageNet 数据集有 1 000 个分类类别,因此可使用 Top1 和 Top5 的分类精度作为评估指标.

- 权值量化

不同于二值量化、三值量化等方法,μL2Q 支持灵活的量化位宽.实验结果见表 4.在 1 比特位宽下进行权值量化,模型的分类精度严重下降.这是因为μL2Q 将权值量化为{−1,1},存在极大的信息损失,进而导致较大的精度下降.但是从 2 比特开始,μL2Q 量化模型的精度显著提升,能够在除 MobileNetV2 外的模型上达到低于 1.58%的平均精度损失(与全精度模型相比).对于轻量级模型 MobileNetV2,尽管其对低位宽量化更为敏感,但在 4 比特的 MobileNetV2 量化模型上,μL2Q 甚至能达到高于全精度模型的推理精度.随着给定位宽的增加,量化模型的精度也随之提升.Lenet5、CifarNet 和 VGG-like 的分类精度分别在 4 比特、8 比特和 8 比特时达到 99.51%,81.66%和 93.53%,其中,LeNet 和 VGG-like 的结果甚至比全精度模型还提升了 0.11%和 0.04%.在 ImageNet 数据集上,μL2Q 能使 AlexNet、ResNet18 和 MobileNetV2 在 8 比特量化位宽下达到 61.4%,70.23%和 72.23%(Top1)的分类精度,与全精度模型的结果相比分别提升了 1.39%,0.63%和 0.93%.以上的实验结果说明,利用μL2Q 进行权值量化有助于提升模型的泛化能力,即有助于提升模型的分类精度.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 4 Results of μL2Q across various bit widths

表 4 μL2Q 在不同比特位宽下的结果

位宽 (W/A)	Lenet5	CifarNet	VGG-like	AlexNet		ResNet18		MobileNetV2	
				Top1	Top5	Top1	Top5	Top1	Top5
32/32	99.40	**81.82**	93.49	60.01	81.90	69.60	**89.24**	71.30	90.10
1/32	99.45	79.28	90.06	54.47	77.41	63.40	84.73	49.65	74.00
2/32	99.47	80.26	92.18	59.05	81.04	68.11	88.13	63.90	84.91
4/32	**99.51**	81.32	93.35	59.88	81.59	69.15	88.66	71.49	90.25
8/32	99.47	81.66	**93.53**	**61.40**	**82.87**	**70.23**	89.22	**72.23**	**90.42**
2/8	99.32	80.53	92.31	58.95	81.00	67.62	87.70	64.39	85.23
4/8	99.31	81.34	93.06	59.87	81.75	69.01	88.53	71.70	90.24
8/8	99.30	81.98	93.38	61.13	82.77	70.08	89.12	72.28	90.61

- 激活值量化

与权值不同,DNN 模型中每层的激活值会随着输入变化而变化.这意味着如果进行激活值量化,则必须在每次推理时进行实时的计算.为了减少实时计算量,本文对量化参数(α,β)进行离线计算,以提升推理效率.本文利用指数移动平均算法,在模型训练过程中来估计每层的激活值在整个训练集上的量化参数,并离线应用到模型推理的过程中.表 4 给出了基于μL2Q 进行 8 比特激活值量化的实验结果.在表 4 中,与具有全精度激活值的模型相比,具有 2/4/8 比特权值和 8 比特激活值的量化模型的推理精度仅仅下降不到 0.49%(在 ResNet18 上),0.29%(在 VGG-like 上),0.27%(在 AlexNet 上).在 ImageNet 数据集和 MobileNetV2 上,基于 8 比特μL2Q 的激活值量化的结果比全精度激活值的量化模型的结果分别高了 0.49%(2 比特权值量化),0.21%(4 比特权值量化)和 0.05%(8 比特权值量化).实验结果表明,μL2Q 可应用于激活值量化并且不会造成额外的精度损失,还能降低推理过程中的内存占用和计算消耗.

- 模型存储容量与运行时内存占用

表 5 中给出了基于μL2Q 的不同量化位宽下,不同量化模型的存储容量和运行时内存占用.模型存储容量主要包括模型权值的存储,运行时内存占用是指模型推理一张图片时所需耗费的权值和激活值的内存占用.在 1 比特权值量化下,模型存储容量压缩率可达 32 倍,比如 CifarNet 的模型存储容量甚至只有 10.64KB.全精度模型 AlexNet 的存储容量高达 203.38MB,在经过μL2Q 压缩后,其模型存储容量最小可至 6.36MB,被压缩了 96.87%,极大地降低了模型的存储代价.此外,LeNet5,CifarNet 和 VGG-like 的运行时内存占用最低可达 360.54KB(1/32),64.39KB(2/8)和 1.57MB(2/8),其运行时内存占用量比全精度的模型(32/32)分别降低了 94.71%,87.44%和 92.97%.在基于 ImageNet 训练的模型 AlexNet,ResNet18 和 MobileNetV2 上,其运行时内存占用降至 8.79MB,5.61MB 和 7.55MB,比全精度模型(32/32)分别减少了 95.73%,90.24%和 81.40%.此外,低位宽的权值和激活表示也将有效地降低计算资源消耗.μL2Q 可在极低的模型存储容量和运行时内存占用的情况下,保证极低的量化损失,提升量化模型的精度.比如表 4 和表 5 中,具有 2 比特权值和 8 比特激活值的 MobileNetV2 规模只有 867.69KB,运行时内存占用只有 7.55MB,但其在 ImageNet 上的 top1 精度可达 64.39%.极低的存储容量和运行时内存占用,确保了基于μL2Q 量化的 DNN 模型可广泛部署到存储和计算资源受限的边缘设备上,而且模型精度能够提供可靠的推理服务.

Table 5 Model size and runtime memory of μL2Q across various bit widths

表 5 μL2Q 在不同比特位宽的模型大小和运行时内存占用

位宽 (W/A)	Lenet5		CifarNet		VGG-like		AlexNet		ResNet18		MobileNetV2	
	规模	内存	规模	内存	规模	内存	规模	内存	规模	内存	规模	内存
32/32	6.65M	6.81M	340.39K	512.85K	21.40M	22.33M	203.38M	205.81M	46.74M	57.48M	13.88M	40.60M
1/32	**207.92K**	**360.54K**	**10.64K**	183.09K	**668.87K**	1.59M	**6.36M**	**8.79M**	**1.46M**	12.20M	**433.85K**	27.15M
2/32	415.84K	568.46K	21.27K	193.73K	1.34M	2.26M	12.71M	15.15M	2.92M	13.66M	867.69K	27.58M
4/32	831.68K	984.30K	42.55K	215.00K	2.68M	3.60M	25.42M	27.86M	5.84M	16.58M	1.74M	28.45M
8/32	1.66M	1.82M	85.10K	257.55K	5.35M	6.27M	50.84M	53.28M	11.68M	22.43M	3.47M	30.19M
2/8	415.84K	454.00K	21.27K	**64.39K**	1.34M	**1.57M**	12.71M	13.32M	2.92M	**5.61M**	867.69K	**7.55M**
4/8	831.68K	869.84K	42.55K	85.66K	2.68M	2.91M	25.42M	26.03M	5.84M	8.53M	1.74M	8.41M
8/8	1.66M	1.70M	85.10K	128.21K	5.35M	5.58M	50.84M	51.45M	11.68M	14.37M	3.47M	10.15M

© 中国科学院软件研究所 http://www.jos.org.cn

- 方法对比

选择当前最为先进且典型的量化方法来进行比较,这些量化方法见表 6.

Table 6 State-of-the-art quantization methods

表 6 最新的量化方法

类别	方法
二值	RebNet[39],BNN[20],BPWN[16],BWN[18],BC[40]
三值	TNN[41],TTQ[26],TN[42],TWN[16],STC[27],ENN[43],TSQ[30],TC[44]
定点数量化	FP[19],Dorefa-Net[17],QAT[22],TQT[23]
本文方法	μL2Q

表 7 展示了在相同位宽条件下,μL2Q 和其他量化方法在几种 DNN 模型上的图像分类精度.其中,对于最新的量化方法,本文直接引用其文献中的结果进行比较.本文关于结果分析中精度平均提升值的计算方式如下.

(1) μL2Q 与二值量化的对比:在 VGG-like 上,μL2Q 相对 ReBNet(*M*=3)精度提升了 3.08%,相对 BC 精度提升了 1.62%.因此μL2Q 与最新的二值量化方法相比,量化后模型精度平均提升了(3.08%+1.62%)/2=2.35%.同理可得μL2Q 与其他三值量化方法的对比结果.

(2) μL2Q 与定点数量化的对比:在 CifarNet 上,2 比特位宽下μL2Q 相对 FP 提升了 61.16%,4 比特位宽下μL2Q 相对 FP 提升了 5.42%,8 比特位宽下μL2Q 相对 FP 提升了 0.26%.因此,μL2Q 相对 FP,精度平均提升了(61.16%+5.42%+0.26%)/3=22.28%.

Table 7 Comparison with state-of-the-art methods

表 7 与最新方法的比较

方法	位宽	精度	方法	位宽	精度	方法	位宽	精度
LeNet5			VGG-like			MobileNetV2(Top1/Top5)		
Float	32	99.40	Float	32	93.49	Float	32	71.30/90.10
BNN	1	98.67	ReBNet(*M*=3)	1	86.98	QAT-t	8	70.09/-
μL2Q	1	**99.45**	BC	1	88.44	QAT-c	8	71.10/-
TNN	2	98.33	μL2Q	1	**90.06**	TQT-wt	8	68.20/89.00
TN	2	82.70	TNN	2	87.89	TQT-wt-th	8	71.80/90.60
FP	2	98.90	TN	2	83.41	μL2Q	8	**72.23**/90.42
μL2Q	2	**99.47**	STC	2	88.58	AlexNet(Top1/Top5)		
FP	4	99.10	TC	2	89.07	Float	32	60.01/81.90
μL2Q	4	**99.51**	μL2Q	2	**92.18**	BWN	1	**56.80/79.40**
FP	8	99.10	ResNet18(Top1/Top5)			μL2Q	1	54.47/77.41
μL2Q	8	**99.47**	Float	32	69.60/89.24	TWN	2	57.50/79.80
CifarNet			BWN	1	60.80/83.00	TTQ	2	57.50/79.90
Float	32	81.82	BPWN	1	57.50/81.20	ENN	2	58.20/80.60
FP	2	19.10	μL2Q	1	**63.40/84.73**	TSQ	2	58.00/80.50
μL2Q	2	**80.26**	TWN	2	61.80/84.20	μL2Q	2	**59.05/81.04**
FP	4	75.90	TTQ	2	66.60/87.20	Dorefa-Net	8	53.00/-
μL2Q	4	**81.32**	ENN	2	67.00/87.50	μL2Q	8	**61.4/82.87**
FP	8	81.40	μL2Q	2	**68.11/88.13**	–	–	–
μL2Q	8	**81.66**	–	–	–	–	–	–

- μL2Q 与二值量化

二值量化只能将权值量化为{−1,1}两个值,如表 7 所示,在 1 比特量化时,μL2Q 在 LeNet5、VGG-like 和 ResNet18 上与二值量化方法的结果相比,平均提升了 0.78%,2.35%和 4.25%(Top1).这是因为二值量化对权值的均值变化和标准差变化比较敏感,导致了额外的精度下降.与之相比,由于存在平移参数和缩放参数,μL2Q 对不同均值和标准差的权值是鲁棒的,能够有效保证量化模型的精度.在 AlexNet 上,μL2Q 与 BWN 相比存在一定程度的精度下降,这是因为 BWN 使用的模型具有 61MB 参数量,而μL2Q 使用了较小的模型,只有 50.88MB,而且量化了模型中的所有层的权值到 1 比特,包括第 1 层和最后一层.在 1 比特时,μL2Q 量化模型的精度比最新的二值量化方法的结果平均高 1.94%.

© 中国科学院软件研究所 http://www.jos.org.cn

- μL2Q 与三值量化

三值量化将权值量化为{−1,0,1},其位宽最少为 2 比特.在位宽为 2 比特时,与最新的方法相比,μL2Q 在 LeNet5、VGG-like、AlexNet 和 ResNet18 上分别平均提升了 6.16%,4.94%,1.25%和 2.98%.显著优于其他量化方法的原因是:首先,因为与启发式的量化方法相比,μL2Q 对权值的均值和标准差的变化是更鲁棒的;其次,μL2Q 还充分利用 2 比特的位宽来表示 4 个数{−2,−1,0,1},从而可以得到较低的数据量化损失和较高的模型精度.在位宽为 2 比特时,μL2Q 量化模型的推理精度比最新的三值量化模型的推理精度平均高 3.73%.

- μL2Q 与定点数量化

定点数量化可以实现灵活的量化位宽.与最新方法相比,μL2Q 在 LeNet5、CifarNet、AlexNet 和 MobileNetV2 上分别平均提升了 0.39%,22.28%,8.34%和 1.93%.在 2 比特量化时,FP 只实现了 19.10%的分类精度,而μL2Q 却能实现 80.26%的精度,甚至在 1 比特量化时,μL2Q 也能实现 79.28%的精度.主要原因在于:FP 使用定点数量化,而μL2Q 则引入了平移参数和缩放参数,在几乎不增加额外计算量的情况下,实现了极大的精度提升.Dorefa-Net 和 QAT/TQT 方法则因忽略了权值数据的分布,没有对量化损失和量化参数的内在联系进行定量分析,仅使用启发式的量化参数设置,造成了额外的精度下降.值得注意的是:QAT 保留了 MobileNetV2 的第 1 层和最后一层数据为全精度的浮点数,而 TQT 使用全局损失感知来学习量化参数的方法.QAT 和 TQT 没有相应的低比特位宽的量化结果,与之对比,μL2Q 在低比特位宽时更具优势,即使在 8 比特的权值量化结果上也能得到出具有竞争力的结果.在相同的位宽下,μL2Q 的模型精度比最新的定点数量化模型的精度平均提高 8.24%.

4.3 显著性物体检测实验

显著性物体检测旨在突出图像中显著的目标区域[45],它提供了可供观测的方法依据和重要的评价指标.前述实验结果表明,μL2Q 在 2 比特量化时能够在精度和量化位宽之间取得较好的平衡.因此,本实验选取 2 比特的 μL2Q 权值量化进行验证.

- 数据集与模型

训练集采用 MSRA10K[47]数据集中的训练数据(占整个数据集的 80%).训练后,在多个数据集上进行评估,包括 MSRA10K 的测试集(整个数据集的 20%)、ECSSD[47]、HKU-IS[48]、DUTS[49]和 DUT-OMRRON[50]中包含目标对象和现有的真值图.所选数据集的详细信息见表 8.为了便于训练和测试,实验中将所有图像的大小都调整为 224×224.实验中选择了 3 个著名的端到端语义分割模型 U-Net[51]、LinkNet[52]和 UNet++[53]进行综合比较,其详细信息见表 9.这些模型都以 ResNet50[4]作为骨干网,并使用 ImageNet 数据集上训练的权值进行初始化.

Table 8 Datasets for salient object detection

表 8 显著性物体检测数据集

数据集	图像数	难度
MSRA10K	10 000	*
ECSSD	1 000	*
HKU-IS	4 000	**
DUTs	15 572	**
DUT-OMRON	5 168	**

Table 9 Models for salient object detection

表 9 显著性物体检测模型

	模型		
	U-Net	LinkNet	UNet++
骨干网	ResNet50	ResNet50	ResNet50
卷积层	64	69	76
参数量(M)	36.54	28.78	37.7
模型大小(M)	139.37	109.80	143.81
量化大小(M)	9.05	7.24	9.35

- 评价指标

选择 4 个广泛使用的度量指标进行综合评价,包括平均绝对误差(MAE)[54]、最大 *F*-measure(*MaxF*)[55]、结

© 中国科学院软件研究所 http://www.jos.org.cn

构度量(*S*-measure)[56]和增强对齐度量(*E*-measure)[57].MAE 越低,*MaxF*、*S*-measure 和 *E*-measure 越高,代表结果越理想.

- 实验结果

如表 10 所示,UNet++、U-Net 和 LinkNet 是 3 个端到端的显著性物体检测模型,用*标记的表示量化后的模型.偏差记录量化模型与全精度的模型在不同数据集的不同指标上的差异;规模表示模型的大小,单位为兆字节(MB).*M*,*F*,*S*,*E* 分别表示 4 个指标:MAE、*MaxF*、*S*-measure、*E*-measure,箭头表示指标的评价趋势,↑表示指标越高越好,↓表示指标越低越好.表格中,指标降低在 0.01 以内的所有结果用粗体表示,表示较低的性能下降.从表 10 的实验结果可知,在数据集 MSRA10K 上,μL2Q 的精度下降最少.与全精度的模型相比,μL2Q 的精度在几乎所有指标上的下降都低于 0.01.这是因为该实验中所使用的模型,包括全精度的模型和量化模型,都是基于 MSRA10K 的,其训练集和验证集基于同一数据集进行划分,数据偏差小,可以达到较好的验证效果.而其余的几个数据集只作为验证集,由于不同的数据集之间存在偏差,因此在这些数据集上出现了一定的性能波动.但模型的尺寸通过μL2Q 量化可减少 93%以上,实现了极高的压缩率.显著性物体检测的实验表明了:μL2Q 可以应用到实际的计算机视觉任务中,且能有效保证性能.

Table 10 Quantization results on salient object detection models

表 10 在显著性目标检测模型上的量化结果

模型	规模	数据集											
		MSRA-10K				ECSSD				HKU-IS			
		M↓	*F*↑	*S*↑	*E*↑	*M*↓	*F*↑	*S*↑	*E*↑	*M*↓	*F*↑	*S*↑	*E*↑
U-Net	143.81	0.030	0.945	0.931	0.962	0.057	0.909	0.886	0.914	0.045	0.907	0.884	0.930
U-Net*	9.35	0.033	0.937	0.923	0.958	0.073	0.878	0.849	0.889	0.058	0.868	0.845	0.907
偏差	**93.50%**	**–0.002**	**0.007**	**0.008**	**0.003**	–0.016	0.031	0.038	0.025	–0.013	0.039	0.039	0.023
LinkNet	139.37	0.032	0.942	0.928	0.959	0.060	0.905	0.882	0.911	0.048	0.900	0.878	0.927
LinkNet*	9.05	0.039	0.926	0.910	0.951	0.087	0.847	0.819	0.869	0.074	0.828	0.811	0.883
偏差	**93.51%**	**–0.007**	0.016	0.017	**0.009**	–0.027	0.058	0.062	0.041	–0.026	0.072	0.066	0.044
UNet++	109.80	0.029	0.948	0.933	0.964	0.056	0.910	0.888	0.915	0.044	0.909	0.887	0.93
UNet++*	7.24	0.032	0.938	0.924	0.959	0.071	0.879	0.851	0.893	0.059	0.865	0.845	0.904
偏差	**93.40%**	**–0.003**	**0.009**	**0.01**	**0.005**	–0.015	0.031	0.037	0.022	–0.015	0.043	0.042	0.026

Table 10 Quantization results on salient object detection models (Continued)

表 10 在显著性目标检测模型上的量化结果(续)

模型	规模	数据集							
		DUTs				DUT-OMRON			
		M↓	*F*↑	*S*↑	*E*↑	*M*↓	*F*↑	*S*↑	*E*↑
U-Net	143.81	0.060	0.896	0.865	0.874	0.070	0.804	0.803	0.829
U-Net*	9.35	0.071	0.869	0.836	0.858	0.079	0.764	0.772	0.817
偏差	**93.50%**	–0.011	0.027	0.029	0.016	**–0.009**	0.040	0.031	0.012
LinkNet	139.37	0.062	0.892	0.861	0.871	0.071	0.801	0.799	0.825
LinkNet*	9.05	0.085	0.843	0.812	0.842	0.092	0.729	0.748	0.795
偏差	**93.51%**	–0.022	0.049	0.049	0.029	–0.021	0.073	0.051	0.030
UNet++	109.80	0.059	0.897	0.867	0.876	0.070	0.805	0.805	0.829
UNet++*	7.24	0.072	0.868	0.836	0.856	0.080	0.769	0.775	0.817
偏差	**93.40%**	–0.013	0.029	0.031	0.020	**–0.010**	0.035	0.030	0.012

5 总 结

本文提出了一种超低损失的 DNN 模型量化方法μL2Q,该方法的设计考虑了 DNN 权值数据的分布规律,通过定量分析量化位宽与量化损失间的关系,呈现了不同量化位宽条件下的最低损失所对应的量化参数,实现了极低的量化损失,从而能对 DNN 模型有效压缩并能保证模型推理精度.此外,本文还将μL2Q 融合进主流机器学习训练框架中如 Keras,为 DNN 模型压缩的设计和实现提供了便捷途径,并显著降低了工程实现的难度,也有效减少了人工尝试等重复性劳动.实验评估结果表明,本文提出的μL2Q 实现了极低的量化损失,显著优于其他

© 中国科学院软件研究所 http://www.jos.org.cn

对比方法,其优秀的量化模型精度能够满足边缘计算场景中的 DNN 应用需求.

References:

[1] Peng YL, Zhang L, Zhang Y, Liu SG, Guo M. Deep deconvolution neural network for image super-resolution. Ruan Jian Xue Bao/Journal of Software, 2018,29(4):926−934 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5407.htm [doi: 10.13328/j.cnki.jos.005407]

[2] Ge DH, Li HS, Zhang L, Liu RY, Shen PY, Miao QG. Survey of lightweight neural network. Ruan Jian Xue Bao/Journal of Software, 2020,31(9):2627−2653 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5942.htm [doi: 10.13328/j.cnki.jos.005942]

[3] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Bengio Y, Le Cun Y, eds. Proc. of the ICLR. San Diego, 2015. [doi: 10.13328/j.cnki.jos.005428]

[4] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proc. of the CVPR. Las Vegas: IEEE Computer Society, 2016. 770−778. [doi: 10.1109/CVPR.2016.90]

[5] Fan DP, Wang W, Cheng MM, Shen J. Shifting more attention to video salient object detection. In: Proc. of the CVPR. Long Beach: Computer Vision Foundation IEEE, 2019. 8554−8564. [doi: 10.1109/CVPR.2019.00875]

[6] Girshick RB. Fast R-CNN. In: Proc. of the ICCV. Santiago: IEEE Computer Society, 2015. 1440−1448. [doi: 10.1109/ICCV.2015.169]

[7] Liu W, Anguelov D, Erhan D, Szegedy C, Reed SE, Fu CY, Berg AC. SSD: Single shot MultiBox detector. In: Proc. of the ECCV. Cham: Springer-Verlag, 2016. 21−37. [doi: 10.1007/978-3-319-46448-0_2]

[8] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proc. of the AAAI. San Francisco: AAAI Press, 2017. 4278−4284.

[9] Fan D, Cheng M, Liu J, Gao S, Hou Q, Borji A. Salient objects in clutter: Bringing salient object detection to the foreground. In: Proc. of the ECCV. Munich: Springer-Verlag, 2018. 196−212. [doi: 10.1007/978-3-030-01267-0_12]

[10] Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation. In: Proc. of the ICCV. Santiago: IEEE Computer Society, 2015. 1520−1528. [doi: 10.1109/ICCV.2015.178]

[11] Pohlen T, Alex, Hermans E, Mathias M, Leibe B. Full-Resolution residual networks for semantic segmentation in street scenes. In: Proc. of the CVPR. Honolulu: IEEE Computer Society, 2017. 3309−3318. [doi: 10.1109/CVPR.2017.353]

[12] Girshick RB, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc. of the CVPR. Columbus: IEEE Computer Society, 2014. 580−587. [doi: 10.1109/CVPR.2014.81]

[13] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proc. of the CVPR. Boston: IEEE Computer Society, 2015. 3431−3440. [doi: 10.1109/CVPR.2015.7298965]

[14] Lei J, Gao X, Song J, Wang XL, Song ML. Survey of deep neural network model compression. Ruan Jian Xue Bao/Journal of Software, 2018,29(2):251−266 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5428.htm [doi: 10.13328/j.cnki.jos.005428]

[15] Hubara I, Courbariaux M, Soudry D, El-Yaniv R, Bengio Y. Binarized neural networks. In: Lee DD, ed. Proc. of the NIPS. 2016. 4107−4115.

[16] Li F, Zhang B, Liu B. Ternary weight networks. arXiv preprint arXiv:1605.04711, 2016.

[17] Zhou S, Ni Z, Zhou X, Wen H, Wu Y, Zou Y. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160, 2016.

[18] Rastegari M, Ordonez V, Redmon J, Farhadi A. XNOR-Net: ImageNet classification using binary convolutional neural networks. In: Leibe B, ed. Proc. of the ECCV. Springer-Verlag, 2016. 525−542. [doi: 10.1007/978-3-319-46493-0_32]

[19] Gysel P, Motamedi M, Ghiasi S. Hardware-Oriented approximation of convolutional neural networks. arXiv preprint arXiv:1604.03168, 2016.

[20] Kim M, Smaragdis P. Bitwise neural networks. arXiv preprint arXiv:1601.06071, 2016.

[21] Han S, Mao H, Dally WJ. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In: Proc. of the ICLR. Puerto Rico, 2015.

© 中国科学院软件研究所 http://www.jos.org.cn

[22] Jacob B, Kligys S, Chen B, Zhu M, Tang M, Howard AG, Adam H, Kalenichenko D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proc. of the CVPR. Salt Lake City: IEEE Computer Society, 2018. 2704−2713.

[23] Jain SR, Gural A, Wu M, Dick C. Trained uniform quantization for accurate and efficient neural network inference on fixed-point hardware. arXiv preprint arXiv:1903.08066, 2016.

[24] Bishop CM. Pattern Recognition and Machine Learning. Springer-Verlag, 2006.

[25] Murphy KP. Machine Learning: A Probabilistic Perspective. MIT Press, 2012.

[26] Zhu C, Han S, Mao H, Dally WJ. Trained ternary quantization. In: Proc. of the ICLR. 2017. https://openreview.net/pdf?id=S1_pAu9xl

[27] Jin C, Sun H, Kimura S. Sparse ternary connect: Convolutional neural networks using ternarized weights with enhanced sparsity. In: Shin Y, ed. Proc. of the ASP-DAC. IEEE, 2018. 190−195. [doi: 10.1109/ASPDAC.2018.8297304]

[28] Lin DD, Talathi SS, Annapureddy VS. Fixed point quantization of deep convolutional networks. In: Balcan M, Weinberger KQ, eds. Proc. of the ICML. New York, 2016. 2849−2858.

[29] Polino A, Pascanu R, Alistarh D. Model compression via distillation and quantization. In: Proc. of the ICLR. 2018. https://openreview.net/pdf?id=S1XolQbRW

[30] Wang P, Hu Q, Zhang Y, Zhang C, Liu Y, Cheng J. Two-Step quantization for low-bit neural networks. In: Proc. of the CVPR. IEEE Computer Society, 2018. 4376−4384. [doi: 10.1109/CVPR.2018.00460]

[31] Gong C, Li T, Lu Y, Hao C, Zhang X, Chen D, Chen Y. μL2Q: An ultra-low loss quantization method for DNN compression. In: Proc. of the IJCNN. IEEE, 2019. 1−8. [doi: 10.1109/IJCNN.2019.8851699]

[32] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick RB, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. In: Hua KA, ed. Proc. of the 22nd ACM Int'l Conf. on Multimedia. ACM, 2014. 675−678. [doi: 10.1145/2647868.2654889]

[33] Chollet F, *et al*. In GitHub repository. 2015. https://github.com/keras-team/keras

[34] Le Cun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based learning applied to document recognition. Proc. of the IEEE, 1998, 86(11):2278−2324. [doi: 10.1109/5.726791]

[35] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. 2009. http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[36] Deng J, Dong W, Socher R, Li L, Li K, Li F. Imagenet: A large-scale hierarchical image database. In: Proc. of the CVPR. IEEE Computer Society, 2009. 248−255. [doi: 10.1109/CVPR.2009.5206848]

[37] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2017,60(6):84−90. [doi: 10.1145/3065386]

[38] Sandler M, Howard A, Zhu ML, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proc. of the CVPR. IEEE Computer Society, 2018. 4510−4520. [doi: 10.1109/CVPR.2018.00474]

[39] Ghasemzadeh M, Samragh M, Koushanfar F. ReBNet: Residual binarized neural network. In: Proc. of the FCCM. IEEE Computer Society, 2018. 57−64. [doi: 10.1109/FCCM.2018.00018]

[40] Courbariaux M, Bengio Y, David JP. Binaryconnect: Training deep neural networks with binary weights during propagations. In: Proc. of the NIPS 2015. 2015. 3123−3131.

[41] Alemdar H, Leroy V, Prost-Boucle A, Petro F. Ternary neural networks for resource-efficient AI applications. In: Proc. of the IJCNN. IEEE, 2017. 2547−2554. [doi: 10.1109/IJCNN.2017.7966166]

[42] Esser SK, Appuswamy R, Merolla P, Arthur JV, Modha DS. Backpropagation for energy-efficient neuromorphic computing. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, eds. Proc. of the NIPS. 2015. 1117−1125.

[43] Leng C, Dou Z, Li H, Zhu S, Jin R. Extremely low bit neural network: squeeze the last bit out with ADMM. In: McIlraith SA, Weinberger KQ, eds. Proc. of the AAAI. AAAI Press, 2018. 3466−3473.

[44] Lin ZH, Courbariaux M, Memisevic R, Bengio Y. Neural networks with few multiplications. arXiv preprint arXiv:1510.03009, 2015.

[45] Wang W, Lai Q, Fu H, Shen J, Ling H. Salient object detection in the deep learning era: An in-depth survey. arXiv preprint arXiv:1904.09146, 2016.

[46] Cheng M, Mitra NJ, Huang X, Torr PHS, Hu S. Global contrast based salient region detection. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2015,37(3):569−582. [doi: 10.1109/TPAMI.2014.2345401]

© 中国科学院软件研究所 http://www.jos.org.cn

[47] Yan Q, Xu L, Shi J, Jia J. Hierarchical saliency detection. In: Proc. of the CVPR. IEEE Computer Society, 2013. 1155−1162.

[48] Wang L, Lu H, Wang Y, Feng M, Wang D, Yin B, Ruan X. Learning to detect salient objects with image-level supervision. In: Proc. of the CVPR. IEEE Computer Society, 2017. 3796−3805. [doi: 10.1109/CVPR.2017.404]

[49] Movahedi V, Elder JH. Design and perceptual validation of performance measures for salient object segmentation. In: Proc. of the CVPR. IEEE Computer Society, 2010. 49−56. [doi: 10.1109/CVPRW.2010.5543739]

[50] Cheng M, Mitra NJ, Huang X, Hu S. Salientshape: Group saliency in image collections. The Visual Computer, 2014,30(4): 443−453.

[51] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: Navab N, *et al*., eds. Proc. of the MICCAI. Cham: Springer-Verlag, 2015. 234−241. [doi: 10.1007/978-3-319-24574-4_ 28]

[52] Chaurasia A, Culurciello E. Linknet: Exploiting encoder representations for efficient semantic segmentation. In: Proc. of the VCIP. IEEE, 2017. 1−4. [doi: 10.1109/VCIP.2017.8305148]

[53] Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J. Unet++: A nested u-net architecture for medical image segmentation. In: Stoyanov D, Taylor Z, eds. Proc. of the MICCAI. Cham: Springer-Verlag, 2018. 3−11. [doi: 10.1007/978-3-030-00889-5_1]

[54] Perazzi F, Krähenbühl P, Pritch Y, Alex, Hornung E. Saliency filters: Contrast based filtering for salient region detection. In: Proc. of the ICCV. IEEE, 2012. 733−740. [doi: 10.1109/CVPR.2012.6247743]

[55] Achanta R, Hemami S, Estrada F, Susstrunk S. Frequency-Tuned salient region detection. In: Proc. of the CVPR. IEEE Computer Society, 2009. 1597−1604. [doi: 10.1109/CVPR.2009.5206596]

[56] Fan D, Cheng M, Liu Y, Li T, Borji A. Structure-Measure: A new way to evaluate foreground maps. In: Proc. of the ICCV. IEEE Computer Society, 2017. 4558−4567. [doi: 10.1109/ICCV.2017.487]

[57] Fan D, Gong C, Cao Y, Ren B, Cheng M, Borji A. Enhanced-Alignment measure for binary foreground map evaluation. In: Lang J, ed. Proc. of the PIJCAI. Stockholm, 2018. 698−704. [doi: 10.24963/ijcai.2018/97]

附中文参考文献:

[1] 彭亚丽,张鲁,张钰,刘侍刚,郭敏.基于深度反卷积神经网络的图像超分辨率算法.软件学报,2018,29(4):926−934. http://www.jos.org.cn/1000-9825/5407.htm [doi: 10.13328/j.cnki.jos.005407]

[2] 葛道辉,李洪升,张亮,刘如意,沈沛意,苗启广.轻量级神经网络架构综述.软件学报,2020,31(9):2627−2653. http://www.jos.org.cn/1000-9825/5942.htm [doi: 10.13328/j.cnki.jos.005942]

[14] 雷杰,高鑫,宋杰,王兴路,宋明黎.深度网络模型压缩综述.软件学报,2018,29(2):251−266. http://www.jos.org.cn/1000-9825/5428.htm [doi: 10.13328/j.cnki.jos.005428]

龚成(1993－),男,博士生,CCF 学生会员,主要研究领域为神经网络压缩,高性能嵌入式系统,异构计算,人工智能.

刘方鑫(1996－),男,硕士,主要研究领域为神经网络压缩,异构计算,人工智能.

卢冶(1986－),男,博士,副教授,CCF 专业会员,主要研究领域为神经网络压缩,高性能嵌入式系统,异构计算,人工智能.

陈新伟(1984－),男,博士,副教授,主要研究领域为机器人控制技术,工业视觉系统,移动机器人系统.

代素蓉(1997－),女,硕士生,CCF 学生会员,主要研究领域为神经网络压缩,机器学习,异构计算.

李涛(1977－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为异构计算,机器学习,物联网.

© 中国科学院软件研究所 http://www.jos.org.cn