第47卷 第5期 计 算 机 工 程 2021年5月
Vol.47 No.5 Computer Engineering May 2021

·人工智能与模式识别· 文章编号：1000-3428(2021)05-0080-08 文献标志码：A 中图分类号：TP183

结合半波高斯量化与交替更新的神经网络压缩方法

张红梅，严海兵，张向利

（桂林电子科技大学 广西高校云计算与复杂系统重点实验室，广西 桂林 541004）

摘 要：为使神经网络模型能在实时性要求较高且内存容量受限的边缘设备上部署使用，提出一种基于半波高斯量化与交替更新的混合压缩方法。对神经网络模型输入部分进行2 bit均匀半波高斯量化，将量化值输入带有缩放因子的二值网络通过训练得到初始二值模型，利用交替更新方法对已训练的二值模型进行逐层微调以提高模型测试精度。在CIFAR-10和ImageNet数据集上的实验结果表明，该方法能有效降低参数和结构冗余所导致的内存和时间开销，在神经网络模型压缩比接近30的前提下，测试精度相比HWGQ-Net方法提高0.8和2.0个百分点且实现了10倍的训练加速。

关键词：卷积神经网络；量化；模型压缩；半波高斯量化；交替更新

开放科学（资源服务）标志码（OSID）：

中文引用格式：张红梅，严海兵，张向利.结合半波高斯量化与交替更新的神经网络压缩方法[J].计算机工程，2021，47(5)：80-87.

英文引用格式：ZHANG Hongmei，YAN Haibing，ZHANG Xiangli. Neural network compression method combining half-wave Gaussian quantization and alternate update[J].Computer Engineering，2021，47(5)：80-87.

Neural Network Compression Method Combining Half-Wave Gaussian Quantization and Alternate Update

ZHANG Hongmei，YAN Haibing，ZHANG Xiangli

（Guangxi Colleges and Universities Key Laboratory of Cloud Computing and Complex Systems，Guilin University of Electronic Technology，Guilin，Guangxi 541004，China）

【Abstract】To enable the deployment of neural network models on edge devices with a limited memory size and high real-time performance requirements，this paper proposes a hybrid compression method combining Half-Wave Gaussian Quantization（HWGQ）and alternate update.By performing the 2 bit uniform HWGQ on the input of the neural network model，the quantized value is input into a binary network with a scaling factor，which is trained to obtain the initial binary model.Then the trained binary model is fine-tuned layer by layer using the alternating update method to improve the accuracy of the model.Experimental results on the CIFAR-10 and ImageNet datasets show that the proposed method significantly reduces the memory consumption and time consumption caused by parameter redundancy and structural redundancy. When the model compression ratio is about 30，the accuracy of the model is increased by 0.8 and 2.0 percentage points compared with that of the HWGQ-Net method，and its training speed is increased by 10 times.

【Key words】Convolutional Neural Network（CNN）；quantization；model compression；Half-Wave Gaussian Quantization（HWGQ）；alternate update

DOI：10. 19678/j. issn. 1000-3428. 0057842

0 概述

近年来，边缘计算技术发展迅速，而体积普遍庞大且计算复杂的卷积神经网络（Convolution Neural Network，CNN）模型仍难以在实时性要求较高但内存容量受限的边缘设备上部署使用，因此卷积神经网络模型压缩与加速成为了学术界和工业界均重点关注的研究领域。随着卷积神经网络模型压缩与加

基金项目：国家自然科学基金（61461010）；认知无线电与信息处理省部共建教育部重点实验室基金（CRKL170103，CRKL170104）；广西密码学与信息安全重点实验室基金（GCIS201626）。

作者简介：张红梅（1970—），女，教授、博士，主研方向为网络信息安全、嵌入式系统、智能信息处理；严海兵，硕士研究生；张向利，教授、博士。

收稿日期：2020-03-24 **修回日期：**2020-04-26 **E-mail：**yhb_qwer1234@qq.com

速研究的不断深入，其中的网络量化方法得到了广泛应用。网络量化的核心思想是使用较少的位（bit）代替原始浮点型（32 bit）参数，进而减少模型存储空间。文献[1]将全精度浮点型参数量化到16 bit固定长度表示，并在训练过程中使用随机约束技术，从而缩减网络存储和浮点计算次数，但压缩程度不高且浮点计算依旧复杂。文献[2]在模型训练过程中直接将全精度权值量化为+1或−1并用1 bit表示，理论上能把模型压缩至原有的1/32，同时将卷积计算中的乘加运算转换为加减运算，达到加速的目的，但因激活值为全精度，无法大幅度加速网络计算。文献[3]提出BNN网络，该网络通过把权值和激活值量化为+1和−1，将原始的卷积计算变成同或和位计数运算，大幅压缩和加速深度网络，但此类简单量化的方式导致了较严重的精度损失。为此，文献[4]提出XNOR-Net和BWN两个网络，对权值和激活值分别引入缩放因子，减少量化误差并提高训练精度，但在训练过程中会出现梯度不匹配问题，影响精度的进一步提升。针对该问题，文献[5]提出HWGQ-Net，有效地解决了训练过程中的梯度不匹配问题，但加速效果不明显。为减少BWN网络的量化误差，文献[6]提出TWN网络，将权值量化到三元网络，即$-w$、0、$+w$，相比BWN网络具有更强的表达能力以及更高的训练精度，文献[7]在TWN网络基础上引入不同的缩放因子，相比TWN网络精度得到进一步提升。文献[6-7]通过引入量化值0，减少了精度损失，但模型压缩比仅为BWN网络的一半。文献[8]通过对梯度值进行量化，达到训练加速的目的，却导致训练精度的下降。文献[9]提出渐进式量化方法，减少了量化损失，但分组、量化和再训练方式导致了较高的计算复杂度。

本文设计一种结合半波高斯量化（Half-Wave Gaussian Quantization，HWGQ）和交替更新的神经网络模型压缩方法，改进2 bit均匀半波高斯量化器，使量化后的值分解为带有缩放因子的+1、0和−1的组合值，当与采用BWN量化的权值进行卷积运算时，可将浮点型卷积运算转化为仅有+1和−1参与的同或和位计数运算（数值0可看作没有参与运算）加速训练过程，并使用交替更新方法[10]对已训练的二值模型进行逐层微调进一步提高模型测试精度。

1　混合压缩方法

本文提出的混合压缩框架如图1所示，首先对模型输入部分进行2 bit均匀半波高斯量化，然后将值输入到带有缩放因子的二值网络中进行训练得到一个初始的二值模型，再使用交替更新方法对模型进行微调，最终得到优化后的二值模型。在图1中，$\boldsymbol{X}$是上一层经过卷积运算（卷积层）或者矩阵运算（全连接层）的输出，$\tilde{\boldsymbol{X}}$是$\boldsymbol{X}$经过2 bit均匀半波高斯量化器的激活量化值，其中q_1、q_2、q_3分别代表3个量化值，且满足等式$q_2-q_1=q_3-q_2$，t_1和t_2分别代表量化值q_1和q_2对应的量化间隔点，$\boldsymbol{A}$和$\boldsymbol{B}$分别是缩放因子矩阵和二元权值矩阵，若$\boldsymbol{W}\in\mathbb{R}^{(c_{\text{in}}\times w\times h)\times c_{\text{out}}}$是经过维度变换后的卷积核，则$\boldsymbol{W}=\boldsymbol{BA}$，其中，$\boldsymbol{B}\in\{+1,-1\}^{(c_{\text{in}}\times w\times h)\times c_{\text{out}}}$，$\boldsymbol{A}\in\mathbb{R}^{c_{\text{out}}\times c_{\text{out}}}$是对角矩阵，且每个对角元素$\alpha_i$与$\boldsymbol{B}_i\in\mathbb{R}^{c_{\text{in}}\times w\times h}$一一对应，$\boldsymbol{B}_i$是$\boldsymbol{B}$的列向量，$i=1,2,\cdots,c_{\text{out}}$。

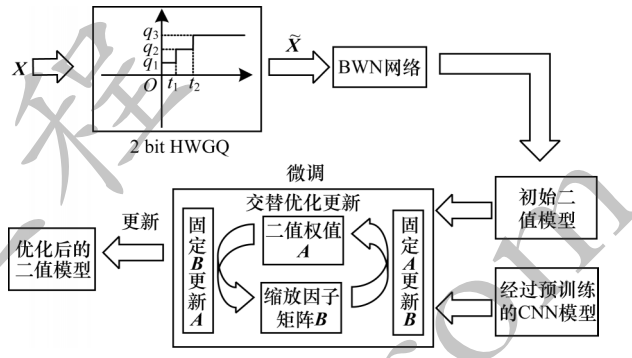

图1　混合压缩框架

Fig.1　Hybrid compression framework

1.1　半波高斯量化

在BNN和XNOR网络中，在前向传播阶段采用sign作为激活值量化函数，在反向传播阶段采用$\widetilde{\text{sign}}$替代sign，以避免梯度全为0的情况发生，影响梯度下降算法的更新，其中，sign和$\widetilde{\text{sign}}$函数定义如图2所示。

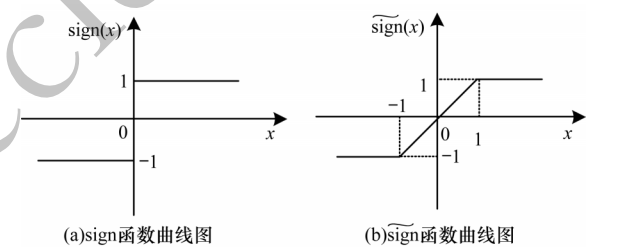

图2　sign和$\widetilde{\text{sign}}$的函数曲线图

Fig.2　sign and $\widetilde{\text{sign}}$ function curve graph

sign和$\widetilde{\text{sign}}$可分别看作前向传播阶段和反向传播阶段中非线性激活函数tanh的近似，其中tanh是双曲正切函数，但该近似并不能产生很好的效果，因为tanh对非线性部分进行挤压，具有明显的饱和效应，容易产生梯度消失问题，进而影响其在反向传播中的有效性，而且sign和$\widetilde{\text{sign}}$近似tanh时差异较大，会导致前馈模型与使用梯度更新所得模型之间不匹配，即存在梯度不匹配问题[11]。

为解决上述问题，本文在前向传播阶段采用近似ReLU[12]的$Q(x)$作为量化函数，在反向传播阶段为解决梯度全为0的问题，采用近似ReLU的$\tilde{Q}(x)$作为

$Q(x)$的替代函数，其中ReLU也称为半波整流器，定义为：

$$h(x)=\max(0,x) \tag{1}$$

其中：当自变量x小于0时，$h(x)$值等于0；当自变量x大于等于0时，$h(x)$值等于自变量x。

该方案相比sign和$\widetilde{\text{sign}}$近似方案具有以下优势：1）ReLU函数是非饱和函数，有效解决了梯度消失问题，提高了反向传播效率；2）ReLU函数使一部分神经元输出为0，在一定程度上可缓解过拟合现象的发生；3）解决了梯度不匹配问题，能有效减少训练过程中的精度损失。

1.1.1 前向近似

考虑到ReLU的半波整流性，前向近似ReLU的量化函数$Q(x)$定义如下：

$$Q(x)=\begin{cases}q_i, x\in(t_{i-1},t_i]\\0, x\leqslant 0\end{cases} \tag{2}$$

其中，$q_i\in\mathbb{R}^+$，$t_i\in\mathbb{R}^+$，$i=1,2,\cdots,m$，$t_0=0$，$t_m=\infty$。本文采用最小化均方误差的方法得到最优解$Q^*(x)$，$p(x)$是x的概率密度函数，假设x的数学期望为$E(x)$、$y=g(x)$，且$\int_{-\infty}^{+\infty}g(x)p(x)\mathrm{d}x$绝对收敛，因此有$E(y)=E(g(x))=\int_{-\infty}^{+\infty}g(x)p(x)\mathrm{d}x$，在$(-\infty,0)$区间内$Q(x)$与$h(x)$均为0，在$(0,+\infty)$区间内$h(x)=x$，于是得到式（3）：

$$\begin{aligned}Q^*(x)=&\mathop{\text{argmin}}_{Q}E_x\left[(Q(x)-x)^2\right]=\\&\mathop{\text{argmin}}_{Q}\int_{t_0}^{t_m}p(x)(Q(x)-x)^2\mathrm{d}x=\\&\mathop{\text{argmin}}_{Q}\int_{t_0}^{t_m}p(x)(Q(x)-h(x))^2\mathrm{d}x=\\&\mathop{\text{argmin}}_{Q}E_x\left[(Q(x)-h(x))^2\right]\end{aligned} \tag{3}$$

本文采用文献[13]中提出的Lloyd算法对$Q^*(x)$进行求解。虽然Lloyd算法是一种迭代算法，但输入分布一般没有规律，导致概率密度函数$p(x)$难以确定，并且不同层输入分布一般不同，会随着反向传播参数的迭代更新而不断改变。上述情况使得Lloyd算法很难得到最优解$Q^*(x)$。通过在量化器$Q(x)$前加入批量标准化（Batch Normalization，BN）[14]操作解决上述问题，批量标准化使得每层输入变成均值为0、方差为1的标准高斯分布。此时，每层具有相同的输入分布，概率密度函数能唯一确定，并且只需要应用一次Lloyd算法，加入批量标准化操作的量化器$Q(x)$称为半波高斯量化器。

1.1.2 反向近似

为解决$Q(x)$在反向传播过程中的梯度消失问题，需要寻找一个近似ReLU的连续函数$\tilde{Q}(x)$，考虑到量化函数$Q(x)$前面加入批标准化后的输入分布变为标准高斯分布，输入越接近0出现的概率越高，假设大于q_m的x值出现的概率很低，因此超出q_m的部分x值实际上是离群值。本文选择Clipped ReLU作为$\tilde{Q}(x)$，定义如下：

$$\tilde{Q}(x)=\begin{cases}q_m, x>q_m\\x, x\in(0,q_m]\\0, \text{其他}\end{cases} \tag{4}$$

本文选择Clipped ReLU作为ReLU的反向近似，主要原因为：1）避免在尾部出现与$Q(x)$不匹配的现象，减少了两者之间的误差；2）大部分输入值集中于小于q_m的部分，因此截断的ReLU不仅能很好地近似ReLU，而且易于梯度计算；3）Clipped ReLU能够保证稳定优化，与文献[15]中裁剪的梯度能够增强深层网络的学习性能类似。

1.2 BWN方法

本文采用文献[4]中的BWN方法对网络权重部分进行量化。假设网络有L层，第l层的卷积核个数为K^l，其中，$1\leqslant l\leqslant L$，$1\leqslant k\leqslant K^l$，令第$l$层的输入$\boldsymbol{X}\in\mathbb{R}^{c_{\text{in}}\times w_{\text{in}}\times h_{\text{in}}}$，第$l$层的第$k$个卷积核$\boldsymbol{W}\in\mathbb{R}^{c_{\text{in}}\times w\times h}$，$c_{\text{in}}$为输入通道数，$w_{\text{in}}$和$h_{\text{in}}$均为输入特征图，$w$为卷积核宽度，$h$为卷积核高度，且$w\leqslant w_{\text{in}}$、$h\leqslant h_{\text{in}}$。第$l$层的卷积运算如式（5）所示：

$$\boldsymbol{X}*\boldsymbol{W}\overset{\text{BWN}}{\approx}\boldsymbol{X}*(\alpha\boldsymbol{B})=\alpha\cdot(\boldsymbol{X}\oplus\boldsymbol{B}) \tag{5}$$

其中，$\alpha=\frac{1}{n}\cdot\|\boldsymbol{W}\|_{L1}=\frac{1}{n}\cdot\sum_{i=1}^{n}|W_i|$，$n=c_{\text{in}}\times w\times h$，$\boldsymbol{B}=\text{sign}(\boldsymbol{W})$，$\oplus$表示只有加减的卷积运算。通过式（5）BWN理论上能将权值用1 bit表示，模型压缩至原有的1/32并通过大幅度移除卷积运算中的乘法操作达到加速目的。若$\boldsymbol{W}=[W_1,W_2,\cdots,W_n]$、$\widetilde{\boldsymbol{W}}=[\widetilde{W}_1,\widetilde{W}_2,\cdots,\widetilde{W}_n]$，则$\widetilde{W}_i=\alpha\cdot\text{sign}(W_i)$，$i\in\{1,2,\cdots,n\}$，其中，$n=c_{\text{in}}\times w\times h$，于是有：

$$\begin{aligned}\frac{\partial C}{\partial W_i}=&\sum_{j=1}^{n}\left(\frac{\partial C}{\partial\widetilde{W}_j}\cdot\frac{\partial\widetilde{W}_j}{\partial W_i}\right)=\sum_{j=1}^{n}\left[\frac{\partial C}{\partial\widetilde{W}_j}\cdot\frac{\partial(\alpha\cdot\text{sign}(W_j))}{\partial W_i}\right]=\\&\sum_{j=1}^{n}\left[\frac{\partial C}{\partial\widetilde{W}_j}\cdot\text{sign}(W_j)\cdot\frac{\partial\alpha}{\partial W_i}\right]+\frac{\partial C}{\partial\widetilde{W}_i}\cdot\frac{\partial\text{sign}(W_i)}{\partial W_i}\cdot\alpha=\\&\frac{1}{n}\cdot\text{sign}(W_i)\cdot\sum_{j=1}^{n}\left[\frac{\partial C}{\partial\widetilde{W}_j}\cdot\text{sign}(W_j)\right]+\\&\frac{\partial C}{\partial\widetilde{W}_i}\cdot\frac{\partial\text{sign}(W_i)}{\partial W_i}\cdot\alpha\end{aligned} \tag{6}$$

1.3 基于HWGQ+BWN的二值模型训练

对于半波高斯量化器，本文令$m=3$、$q_{i+1}-q_i=\Delta$。由于此时其量化值只能取0、$\beta-\Delta$、β、$\beta+\Delta$这4个值并用2 bit进行表示，因此也可称为2 bit均匀半波高斯量化器。图3为对输入部分和权重部分分别采用改进后的2 bit均匀半波高斯量化器和BWN方法量化后的卷积计算过程，其中，*表示卷积

运算，⊛表示只有同或和位计数操作的卷积运算。可以看出，经过HWGQ量化后的值被分解为两部分，每部分均是带有缩放因子−1、0和+1的组合，最终浮点型的卷积运算转化为仅有−1和+1参与的同或和位计数操作的卷积运算（数值0可看作没有参与运算），从而实现模型的训练加速。

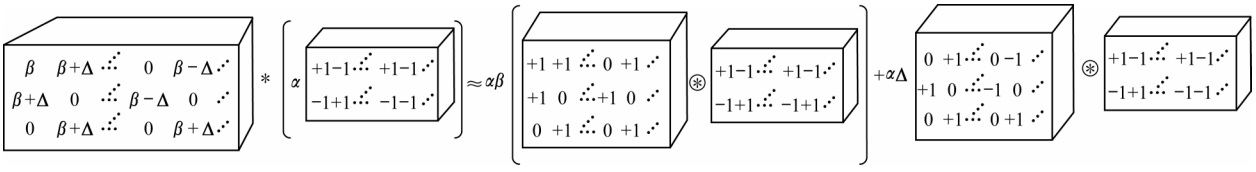

图3　具有加速作用的HWGQ+BWN卷积计算过程

Fig.3　Accelerated HWGQ+BWN convolution calculation process

本文对第一层和最后一层保留全精度，只对中间层进行量化处理，在前向阶段和反向阶段使用量化的权值，在权值更新阶段使用全精度值。

算法1　基于HWGQ+BWN的二值模型训练算法

输入　批量输入$\boldsymbol{X}$、目标输出$\boldsymbol{Y}$、批量归一化参数θ和初始学习率η^t

输出　二值模型、二值模型训练精度和更新后的学习率η^{t+1}

1.构建网络时随机初始化$\mathrm{W}^1\mathrm{W}^2\cdots\mathrm{W}^{\mathrm{L}}$

2.{第一层}

3.$\mathrm{Y}^1=\mathrm{Conv}(\mathrm{X},\mathrm{W}^1)$

4.{中间层}

5.for l = 2 to L−1 do

6.$\mathrm{X}^{\mathrm{l}}=\mathrm{Y}^{\mathrm{l}-1}$

7.$\mathrm{X}^{\mathrm{l}}=\mathrm{BatchNorm}(\mathrm{X}^{\mathrm{l}},\theta^{\mathrm{l}})$

8.$\widetilde{\mathrm{X}}^{\mathrm{l}}=\mathrm{Q}(\mathrm{X}^{\mathrm{l}})$

9.for k filter in l layer do

10.$\widetilde{\mathrm{W}}^{\mathrm{l}}_{\mathrm{k}}=\alpha_{\mathrm{lk}}\mathrm{B}_{\mathrm{lk}}=\left(\frac{1}{\mathrm{n}}\times\left\|\mathrm{W}^{\mathrm{l}}_{\mathrm{k}}\right\|_{\mathrm{L1}}\right)\times\mathrm{sign}(\mathrm{W}^{\mathrm{l}}_{\mathrm{k}})$

11.$\mathrm{Y}^{\mathrm{l}}=\mathrm{BinConv}(\widetilde{\mathrm{X}}^{\mathrm{l}},\widetilde{\mathrm{W}}^{\mathrm{l}})$

12.{最后一层}

13.$\mathrm{X}^{\mathrm{L}}=\mathrm{Y}^{\mathrm{L}-1}$

14.$\mathrm{Y}^{\mathrm{L}}=\mathrm{Conv}(\mathrm{X}^{\mathrm{L}},\mathrm{W}^{\mathrm{L}})$

15.$\mathrm{C}=\mathrm{Loss}(\mathrm{Y},\mathrm{Y}^{(\mathrm{L})})$

16.$\frac{\partial\mathrm{C}}{\partial\mathrm{W}}\leftarrow\frac{\partial\mathrm{C}}{\partial\widetilde{\mathrm{W}}}$

17.$\mathrm{W}\leftarrow\mathrm{W}-\eta^{\mathrm{t}}\frac{\partial\mathrm{C}}{\partial\mathrm{W}}$

18.$\eta^{\mathrm{t}+1}=\mathrm{UpdateLearningrate}(\eta^{\mathrm{t}},\mathrm{t})$

1.4　二值模型微调

针对输入部分和权重部分同时量化而导致精度损失较大的问题，本文采用文献[10]中的交替更新方法对二值模型进行微调。交替更新方法主要是对已经训练好的模型进行微调，并且考虑了对输入部分和权重部分同时进行量化的情况，而文献[10]仅考虑了对权重部分的量化。

1.4.1　维度变换

若要运用交替更新方法，则需对卷积层的输入和输出以及卷积核作维度变换。从文献[16]得到启发，假定卷积层输入$\boldsymbol{X}\in\mathbb{R}^{c_{\mathrm{in}}\times w_{\mathrm{in}}\times h_{\mathrm{in}}}$，卷积核$\boldsymbol{W}\in\mathbb{R}^{c_{\mathrm{out}}\times c_{\mathrm{in}}\times w\times h}$，那么卷积层输出$\boldsymbol{Y}\in\mathbb{R}^{c_{\mathrm{out}}\times w_{\mathrm{out}}\times h_{\mathrm{out}}}$，若对卷积层的输入$\boldsymbol{X}$、卷积核$\boldsymbol{W}$和输出$\boldsymbol{Y}$进行维度变换转换为二维矩阵$\boldsymbol{X}_r\in\mathbb{R}^{(c_{\mathrm{in}}\times w\times h)\times(w_{\mathrm{out}}\times h_{\mathrm{out}})}$、$\boldsymbol{W}_r\in\mathbb{R}^{(c_{\mathrm{in}}\times w\times h)\times c_{\mathrm{out}}}$和$\boldsymbol{Y}_r\in\mathbb{R}^{(w_{\mathrm{out}}\times h_{\mathrm{out}})\times c_{\mathrm{out}}}$，其中下标$\boldsymbol{r}$表示张量经过维度变换后由多维变成二维，其中$w_{\mathrm{out}}=(w_{\mathrm{in}}+2\times p-w)/s+1$，$h_{\mathrm{out}}=(h_{\mathrm{in}}+2\times p-h)/s+1$，$p$和$s$分别表示填充值（padding）和步长（stride），此时可将卷积运算转变为矩阵运算$\boldsymbol{Y}_r=(\boldsymbol{X}_r)^{\mathrm{T}}\boldsymbol{W}_r$，具体过程如图4所示。

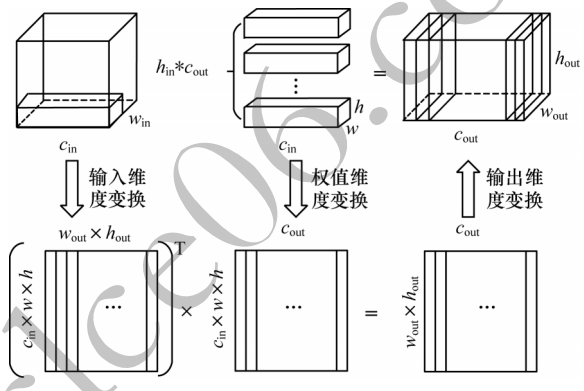

图4　卷积层上的卷积运算转变为矩阵运算的过程

Fig.4　The process of transforming convolution operation into matrix operation on convolution layer

1.4.2　逐层微调

由于对输入部分和权重部分同时进行量化会产生一定的误差，该误差会逐层进行积累，因此本文采用交替更新方法对二值模型进行逐层微调解决以上问题。受文献[17]启发，假设一个CNN网络有L层，记未对输入部分和权重部分进行量化和训练的模型为全精度模型，若由维度变换得到的第$l(1\leqslant l\leqslant L)$层全精度模型和二值模型的输入分别为$\boldsymbol{X}^l$和$\tilde{\boldsymbol{X}}^l$，为使得量化误差最小，需要优化的目标函数为：

$$\min L(\boldsymbol{A},\boldsymbol{B})=\left\|(\boldsymbol{X}^l)^{\mathrm{T}}\boldsymbol{W}^l-(\tilde{\boldsymbol{X}}^l)^{\mathrm{T}}\tilde{\boldsymbol{W}}^l\right\|_{\mathrm{F}}^2=\left\|\boldsymbol{S}^l-(\tilde{\boldsymbol{X}}^l)^{\mathrm{T}}\boldsymbol{B}^l\boldsymbol{A}^l\right\|_{\mathrm{F}}^2\tag{7}$$

对式(7)进一步展开，目标函数变为：

$$\min L(\alpha_i^l,\boldsymbol{B}_i^l)=\sum_{i=1}^{c_{\mathrm{out}}}\left\|\boldsymbol{S}_i^l-\alpha_i^l(\tilde{\boldsymbol{X}}^l)^{\mathrm{T}}\boldsymbol{B}_i^l\right\|_{\mathrm{F}}^2\tag{8}$$

其中，$\boldsymbol{S}_i^l\in\mathbb{R}^{w_{\mathrm{out}}\times h_{\mathrm{out}}}$，$\boldsymbol{B}^l\in\{+1,-1\}^{(c_{\mathrm{in}}\times w\times h)\times c_{\mathrm{out}}}$且$\boldsymbol{B}_i^l$是$\boldsymbol{B}^l$的

列向量，$\boldsymbol{A}^l \in \mathbb{R}^{c_{out} \times c_{out}}$是对角矩阵，$\alpha_i^l = A_{ii}^l$是对应于$\boldsymbol{B}_i^l$的缩放因子，F表示Frobenius范数，简称F-范数。

式(8)的求解过程具体如下：

1)初始化α_i^l和$\boldsymbol{B}_i^l$，记二值模型权重为$\{\tilde{\boldsymbol{W}}^l\}_{l=1}^L$，得到$\boldsymbol{B}_i^l = \text{sign}(\tilde{\boldsymbol{W}}_i^l)$，$\alpha_i^l = \frac{1}{T} \cdot \left\| \tilde{\boldsymbol{W}}_i^l \right\|_{L1} = \frac{1}{T} \cdot \sum_{j=1}^{c_{out}} w_{ij}^l$，其中$\tilde{\boldsymbol{W}}_i^l$是$\tilde{\boldsymbol{W}}^l$的列向量，$T = c_{in} \times w \times h$。

2)保持B_i^l不变，更新α_i^l值。对式(8)进行展开得到$\min L_i(\alpha_i^l) = \text{const} + \alpha_i^l \left\| (\tilde{\boldsymbol{X}}^l)^{\mathrm{T}} \boldsymbol{B}_i \right\|_{\mathrm{F}}^2 - 2\alpha_i^l (\boldsymbol{S}_i^l)^{\mathrm{T}} (\tilde{\boldsymbol{X}}^l)^{\mathrm{T}} \boldsymbol{B}_i^l$，其中$\text{const} = (\boldsymbol{S}_i^l)^{\mathrm{T}} \boldsymbol{S}_i^l$，然后求$\alpha_i^l$的导数并令导数为0，得到：

$$\alpha_i^l = \frac{(\boldsymbol{S}_i^l)^{\mathrm{T}} (\tilde{\boldsymbol{X}}^l)^{\mathrm{T}} \boldsymbol{B}_i^l}{\left\| (\tilde{\boldsymbol{X}}^l)^{\mathrm{T}} \boldsymbol{B}_i^l \right\|_{\mathrm{F}}^2} \tag{9}$$

3)保持α_i^l不变，更新$\boldsymbol{B}_i^l$值。对式(8)进行展开得到$\min L_i(\boldsymbol{B}_i^l) = \text{const} + \left\| (\alpha_i^l \tilde{\boldsymbol{X}}^l)^{\mathrm{T}} \boldsymbol{B}_i^l \right\|_{\mathrm{F}}^2 - 2(\alpha_i^l \tilde{\boldsymbol{X}}^l \boldsymbol{S}_i^l)^{\mathrm{T}} \boldsymbol{B}_i^l$，令$\boldsymbol{Z}^l = \alpha_i^l \tilde{\boldsymbol{X}}^l$、$\boldsymbol{q}^l = \alpha_i^l \tilde{\boldsymbol{X}}^l \boldsymbol{S}_i^l$，得到：

$$\min L_i(\boldsymbol{B}_i^l) = \text{const} + \left\| (\boldsymbol{Z}^l)^{\mathrm{T}} \boldsymbol{B}_i^l \right\|_{\mathrm{F}}^2 - 2\text{tr}[(\boldsymbol{B}_i^l)^{\mathrm{T}} \boldsymbol{q}^l] \tag{10}$$

其中：tr( )表示迹范数；令b为$\boldsymbol{B}_i^l$的第j个元素，$(\boldsymbol{B}_i^l)'$是$\boldsymbol{B}_i^l$中除去元素b的列向量；q_j^l是$\boldsymbol{q}^l$中的第j个元素，$(\boldsymbol{q}^l)'$是$\boldsymbol{q}^l$中除去元素q_j^l的列向量；$(\boldsymbol{v}_j^l)^{\mathrm{T}}$是$\boldsymbol{Z}^l$的第$j$个行向量，$(\boldsymbol{Z}^l)'$是$\boldsymbol{Z}^l$中除去行向量$(\boldsymbol{v}_j^l)^{\mathrm{T}}$的二维张量。

通过文献[18]中提出的离散循环坐标下降法，式(10)可优化为$\min[((\boldsymbol{B}_i^l)')^{\mathrm{T}} (\boldsymbol{Z}^l)' \boldsymbol{v}_j^l - q_j^l]b$, s.t. $b \in \{+1, -1\}$，得到：

$$b = \text{sign}[q_j^l - ((\boldsymbol{B}_i^l)')^{\mathrm{T}} (\boldsymbol{Z}^l)' \boldsymbol{v}_j^l] \tag{11}$$

利用式(11)可迭代求出$\boldsymbol{B}_i^l$中其他元素的值，最终求出$\boldsymbol{B}_i^l$的值。

1.4.3 整体微调

算法2 基于交替更新方法的二值模型微调算法

输入 预训练模型$\{\boldsymbol{W}^l\}_{l=1}^L$、二值模型$\{\widetilde{\boldsymbol{W}}^l\}_{l=1}^L$和最大迭代次数Max_Iter

输出 微调后的二值模型$\{\widetilde{\boldsymbol{W}}^l\}_{l=1}^L$

1.for l = 2； l ≤ L − 1 do
2.从数据集中抽样得到小批量数据集
3.前向传播得到X^l，X^l通过HWGQ量化得到$\widetilde{X}^l$
4.计算S^l，$S^l = (X^l)^{\mathrm{T}} W^l$
5.for i = 1； i ≤ c_{out} do
6.B_i^l初始化为sign($\widetilde{W}_i^l$)
7.α_i^l初始化为$\widetilde{W}_i^l$的L1范数的平均值
8.while iter ≤ Max_Iter do
9.使用式(9)更新α_i^l
10.for j = 1； j ≤ c_{in} × w × h do
11.使用式(11)更新B_i^l
12.end
13.end
14.end
15.end
16.得到微调后的二值模型$\{\widetilde{W}^l = \alpha_i^l B_i^l\}_{l=1}^L$

2 实验与结果分析

本文使用CIFAR-10和ImageNet[19]这两种经典数据集验证混合压缩方法的有效性。CIFAR-10数据集对应的网络结构为VGG14，共有10个类的60 000张RGB三通道图片，其中，训练集有50 000张，测试集有10 000张。ImageNet对应的网络结构为AlexNet[20]，共有1 000个类的1.25×10^6张GRB三通道图片，其中，训练集有1.2×10^6张，验证集有5×10^4张。

VGG14的网络结构为(2×64C3)-MP2-(2×128C3)-MP2-(3×256C3)-MP2-(3×256C3)-MP2-(3×512C3)-MP2-10FC-Softmax，其中："64C3"代表64个大小为3×3的卷积核，步长和填充值都为1；"MP2"代表采样核为2×2，步长为2的最大池化层。AlexNet包括5个卷积层和3个全连接层。VGG14和AlexNet的网络结构如图5和如图6所示，其中：C3=3×3 filter，s=p=1，C代表卷积(Convolution)操作，filter代表卷积核(滤波器)；MP3=2×2，s=2，MP代表最大池化(Max Pooling)；FC代表全卷积(Fully Convolution)。

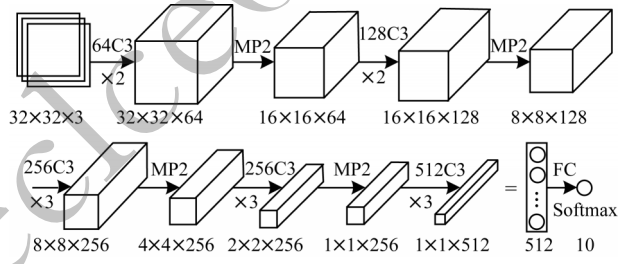

图5 VGG14网络结构

Fig.5 VGG14 network structure

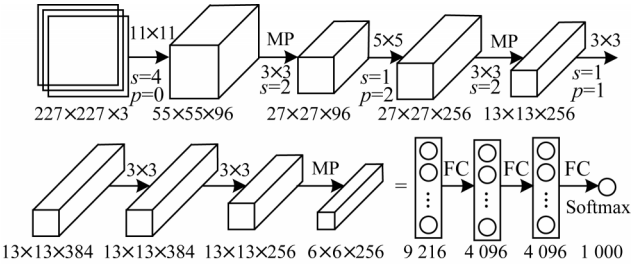

图6 AlexNet网络结构

Fig.6 AlexNet network structure

本文实验硬件环境为8核Intel® Xeon™ CPU E5-2620 V4@2.10 GHz，磁盘容量为3.7 TB，总内存为31 GB，可用内存为22 GB；GPU型号为GeForce GTX 1080Ti的工作站1个，专用GPU内存为11 GB，共享GPU内存为16 GB。软件环境为64位的

Ubuntu 16.04 LTS，CUDA10.0，Pytorch0.3.1，Python 3.5和gcc 5.4。

2.1　数据预处理

2.1.1　半波高斯量化器参数设置

本文利用Lloyd算法[13]可以得到半波高斯量化器的参数值，2 bit均匀半波高斯量化器参数设置如表1所示。

表1　2 bit均匀半波高斯量化器参数设置

Table 1　Parameters setting of 2 bit uniform half-wave Gaussian quantizer

i	q_i	t_i
1	0.377 935	0.797 113
2	1.216 290	1.635 468
3	2.054 645	$+\infty$

2.1.2　ImageNet数据集预处理

针对ImageNet数据集大、分辨率高和训练占用内存大的特点，为提高训练速度和方便实验调试，本文对原始的ImageNet数据集进行预处理。在对训练集进行Resize、随机裁剪和随机翻转后，数据规模由原来的167 GB变成13.5 GB；在对验证集进行Resize和中心裁剪之后，数据规模由原来的6.7 GB变成543.8 MB。预处理后每张图片分辨率为227像素×227像素。表2为ImageNet数据集预处理前后数据规模对比结果，可以看出预处理前后的数据规模压缩比约为12.5，通过预处理加快了训练和测试的速度。

表2　ImageNet数据集预处理前后的数据规模对比

Table 2　Comparison of data scale before and after ImageNet dataset preprocessing

数据集	预处理前/GB	预处理后	预处理前后压缩比
训练集	167.0	13.5 GB	12.37
验证集	6.7	543.8 MB	12.60

2.2　压缩比分析

本文使用HWGQ+BWN方法的压缩效果较明显，表3为压缩前和压缩后的模型规模对比结果，可以看出，本文提出的混合压缩方法在VGG14和AlexNet网络结构上的压缩比分别为29.5和30.8，接近理论值32。

表3　网络压缩前后模型规模对比

Table 3　Comparison of model scale before and after network compression

网络结构	压缩前/MB	压缩后/MB	压缩比
VGG14	59.0	2.0	29.5
AlexNet	249.6	8.1	30.8

2.3　测试精度分析

对于小型数据集CIFAR-10以及对应的网络结构VGG14，超参数设置具体如下：L2正则化的权重衰减系数为1×10^{-5}，迭代次数（epoch）为300，初始学习率为0.1，epoch从150开始，每隔50个epoch学习率降低10倍，batch-size为128，使用带有momentum的SGD作为参数优化器，其中momentum值为0.9，采用L2正则化防止训练时产生过拟合现象，提高网络泛化能力，选择交叉熵作为损失函数。

对于VGG14网络结构，HWGQ+BWN方法得到的二值模型测试精度为91.3%，如图7所示，其中Full-Precision表示未使用量化方法的原始网络。在此基础上，对二值模型进行微调（HWGQ+BWN+Fine-tune），微调结果如图8所示。可以看出，当最大迭代次数为10时，测试精度约稳定于92.1%。

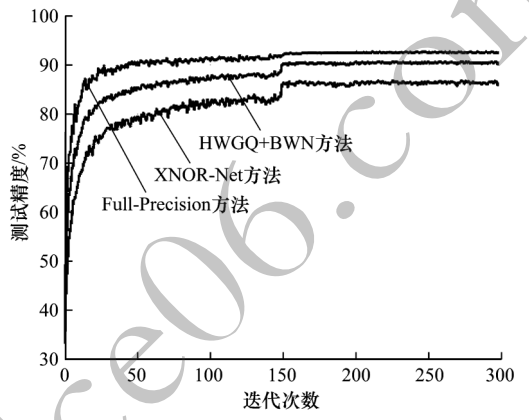

图7　3种压缩方法在VGG14上的测试精度

Fig.7　Test accuracy of three compression methods on VGG14

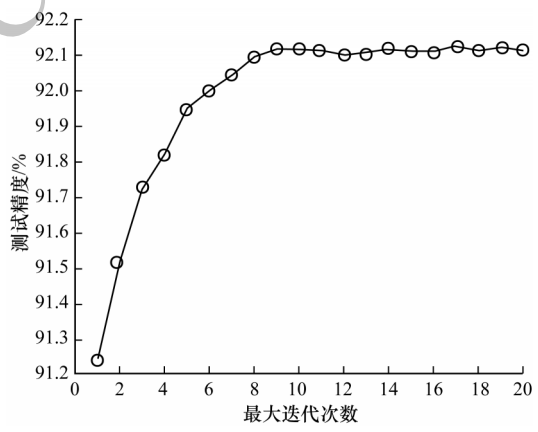

图8　HWGQ+BWN方法的二值模型在VGG14上的微调结果

Fig.8　Fine-tune results of binary model of HWGQ+BWN method on VGG14

基于CIFAR-10数据集的4种压缩方法在VGG14中的测试精度对比结果如表4所示，可以看出本文所提的HWGQ+BWN+Fine-tune方法相比

HWGQ-Net方法在压缩模型规模保持不变的前提下,测试精度提高了0.8个百分点。

表4 CIFAR-10数据集在VGG14中的测试精度对比

Table 4 Comparison of test accuracy of CIFAR-10 dataset in VGG14

方法	测试精度/%	模型规模/MB
Full-Precision	92.6	59
XNOR-Net	88.5	2
HWGQ-Net	91.3	2
HWGQ+BWN+Fine-tune	92.1	2

对于大型数据集ImageNet以及对应的网络结构AlexNet,超参数设置具体如下:L2正则化的权重衰减系数为1×10^{-5},epoch为20,初始学习率为0.001,每隔5个epoch学习率降低10倍,batch-size为512,使用Adam[21]作为参数优化器,选择交叉熵作为损失函数。对于AlexNet网络结构,HWGQ+BWN最终训练得到的二值模型Top-1测试精度和Top-5测试精度为50.7%和74.8%,如图9所示。在此基础上,对二值模型进行微调(HWGQ+BWN+Fine-tune),微调结果如图10和图11所示,可以看出在最大迭代次数约为20时,Top-1测试精度和Top-5测试精度约稳定于52.7%和76.8%。基于ImageNet数据集的4种压缩方法在AlexNet中的测试精度对比结果如表5所示,可以看出对于ImageNet数据集,本文所提的HWGQ+BWN+Fine-tune方法与相比HWGQ-Net方法在压缩模型规模保持不变的前提下,Top-1测试精度和Top-5测试精度分别提高了2.0和1.6个百分点。

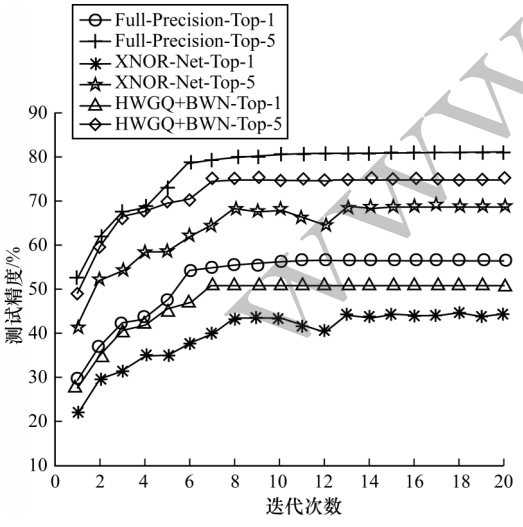

图9 3种压缩方法在AlexNet上的测试精度

Fig.9 Test accuracy of three compression methods on AlexNet

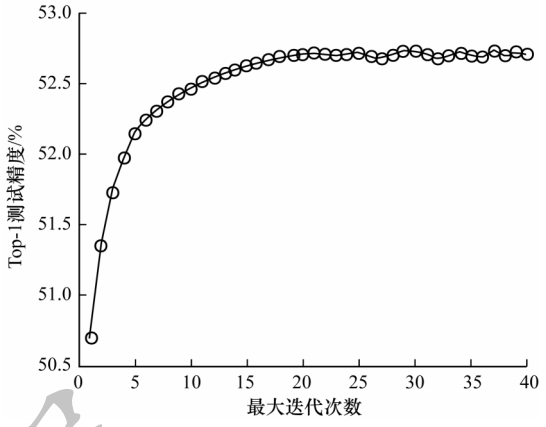

图10 HWGQ+BWN方法的二值模型在AlexNet上的Top-1微调结果

Fig.10 Top-1 fine-tune results of binary model of HWGQ+BWN method on AlexNet

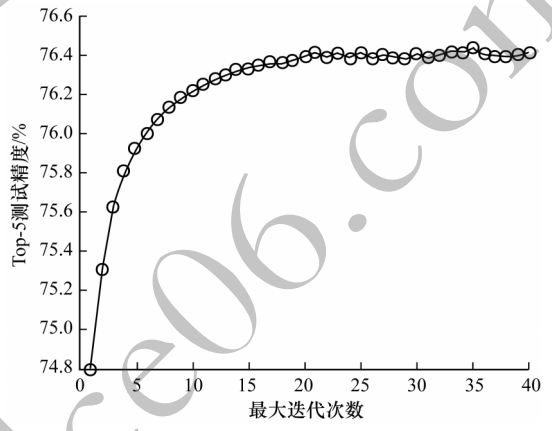

图11 HWGQ+BWN方法的二值模型在AlexNet上的Top-5微调结果

Fig.11 Top-5 fine-tune results of binary model of HWGQ+BWN method on AlexNet

表5 ImageNet数据集在AlexNet中的测试精度对比

Table 5 Comparison of test accuracy of ImageNet dataset in AlexNet

方法	Top-1测试精度/%	Top-5测试精度/%	模型规模/MB
Full-Precision	56.5	80.8	249.6
XNOR-Net	44.4	68.6	8.1
HWGQ-Net	50.7	74.8	8.1
HWGQ+BWN+Fine-tune	52.7	76.4	8.1

2.4 加速效果分析

本文设计了一个具有加速作用的2 bit均匀量化半波高斯量化器,能将浮点型卷积运算转化为简单的位运算和同或运算。如表6所示,本文所提的HWGQ+BWN方法通过对半波高斯量化器的改进,相比HWGQ-Net方法实现了10倍的训练加速,相比Full-Precision方法实现了30倍的训练加速。

表6　3种压缩方法在训练过程中的加速比对比

Table 6　Comparison of speedup ratio of three compression methods in the training process

方法	加速比
Full-Precision	1
HWGQ-Net	3
HWGQ+BWN	30

3　结束语

本文提出一种神经网络压缩方法，采用近似ReLU的半波高斯量化器对输入部分进行量化，在反向传播阶段利用ReLU函数解决梯度不匹配问题。在此基础上，通过改进的2 bit均匀半波高斯量化器加速训练过程，并采用交替更新方法对已训练的二值模型进行缩放因子和二元权值微调，进一步提高神经网络模型测试精度。实验结果表明，在神经网络模型规模保持不变的情况下，该方法能明显提高模型测试精度并加快训练速度。下一步将研究不同稀疏度的半波高斯量化器对神经网络模型测试精度和加速效果的影响，并在满足模型压缩规模的条件下，将该半波高斯量化器与三值模型相结合进一步提高测试精度。

参考文献

[1] GUPTA S, AGRAWAL A, GOPALAKRISHNAN K, et al. Deep learning with limited numerical precision [C]// Proceedings of International Conference on Machine Learning. New York, USA: ACM Press, 2015: 1737-1746.

[2] COURBARIAUX M, BENGIO Y, DAVID J P. BinaryConnect: training deep neural networks with binary weights during propagation [C]//Proceedings of International Conference on Neural Information Processing Systems. Cambridge, USA: MIT Press, 2015: 3123-3131.

[3] COURBARIAUX M, HUBARA I, SOUDRY D, et al. Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or −1 [EB/OL]. [2020-02-10]. https://arxiv.org/abs/1602.02830.

[4] RASTEGARI M, ORDONEZ V, REDMON J, et al. XNOR-Net: ImageNet classification using binary convolutional neural networks [C]//Proceedings of European Conference on Computer Vision. Berlin, Germany: Springer, 2016: 525-542.

[5] CAI Zhaowei, HE Xiaodong, SUN Jian, et al. Deep learning with low precision by half-wave Gaussian quantization [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2017: 5918-5926.

[6] LI Fengfu, ZHANG Bo, LIU Bin. Ternary weight networks [EB/OL]. [2020-02-10]. https://arxiv.org/abs/1605.04711.

[7] ZHU Chenzhuo, HAN Song, MAO Huizi, et al. Trained ternary quantization [EB/OL]. [2020-02-10]. https://arxiv.org/pdf/1612.01064.pdf.

[8] ZHOU Shuchang, WU Yuxin, NI Zekun, et al. Dorefa-Net: training low bitwidth convolutional neural networks with low bitwidth gradients [EB/OL]. [2020-02-10]. https://arxiv.org/pdf/1606.06160.pdf.

[9] ZHOU Aojun, YAO Anbang, GUO Yiwen, et al. Incremental network quantization: towards lossless CNNs with low-precision weights [EB/OL]. [2020-02-10]. https://arxiv.org/pdf/1702.03044.pdf.

[10] HU Qinghao, WANG Peisong, CHENG Jian. From hashing to CNNs: training binary weight networks via hashing [C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2018: 3247-3254.

[11] LIN D D, TALATHI S S. Overcoming challenges in fixed point training of deep convolutional networks [EB/OL]. [2020-02-10]. https://arxiv.org/abs/1607.02241.

[12] GLOROT X, BORDES A, BENGIO Y. Deep sparse rectifier neural networks [C]//Proceedings of the 14th International Conference on Artificial Intelligences and Statistics. Washington D. C., USA: IEEE Press, 2011: 315-323.

[13] LLOYD S. Least squares quantization in PCM [J]. IEEE Transactions on Information Theory, 1982, 28(2): 129-137.

[14] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift [EB/OL]. [2020-02-10]. https://arxiv.org/pdf/1502.03167.pdf.

[15] PASCANU R, MIKOLOV T, BENGIO Y. On the difficulty of training recurrent neural networks [C]//Proceedings of International Conference on Machine Learning. Washington D. C., USA: IEEE Press, 2013: 1310-1318.

[16] LI Zefan, NI Bingbing. Performance guaranteed network acceleration via high-order residual quantization [EB/OL]. [2020-02-10]. https://arxiv.org/abs/1708.08687.

[17] WU Jiaxiang, LENG Cong, WANG Yuhang, et al. Quantized convolutional neural networks for mobile devices [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2016: 4820-4828.

[18] SHEN Fumin, SHEN Chunhua, LIU Wei, et al. Supervised discrete hashing [C]//Proceedings of IEEE Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2015: 37-45.

[19] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge [J]. International Journal of Computer Vision, 2015, 115(3): 211-252.

[20] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM, 2017, 60(6): 84-90.

[21] KINGMA D P, BA J. Adam: a method for stochastic optimization [EB/OL]. [2020-02-10]. https://arxiv.org/pdf/1412.6980.pdf.

编辑　陆燕菲