高效还原式二值神经网络

曾 凯[1,2],万子鑫[1,2],王铭涛[1,2],沈 韬[1,2*]
(1. 昆明理工大学信息工程与自动化学院,云南昆明 650500;2. 云南省计算机技术应用重点实验室,云南昆明 650500)

摘 要: 将权重分布、激活分布和梯度尽可能地还原为原始全精度网络数据,能够极大提高二值网络的推理能力. 然而,现有方法将正向传播中的还原操作直接作用于二值数据,同时用以控制反向传播的梯度近似函数均为固定或手动方式确定,导致二值网络的还原效率有待改进. 针对这一问题,构建了高效还原式二值神经网络. 首先提出面向信息熵最大的分布恢复方法,通过对原始全精度权重均值平移和模长缩放,使量化后的二值权重直接具备分布最大还原特性,同时采用基于简单统计的平移和缩放因子,极大提高了权重和激活的还原效率;进一步提出基于自适应分布近似的梯度函数,根据当前全精度数据的实际分布,以P分位动态确定当前梯度的更新范围,进而自适应改变近似函数的形状,使训练过程中的梯度得到高效更新,从而提高了模型的收敛能力. 在保证执行效率提升的前提下,通过理论分析证实了本文方法能够使二值数据达到最大程度还原. 与当前现有的先进二值网络模型相比本文方法实验结果表现优异,其中针对ResNet-18和ResNet-20量化的分布还原操作计算时间开销分别下降了60%和67%;同时在CIFAR-10数据集上针对VGG-Small二值量化取得93.0%的准确率,在ImageNet数据集上针对ResNet-18二值量化取得61.9%的准确率,均为当前二值神经网络的最佳性能表现. 相关代码开源在https://github.com/sjmp525/IA/tree/ER-BNN.

关键词: 二值神经网络;信息还原;信息熵最大;自适应梯度
基金项目: 云南省杰出青年人才项目(No.202301AV070003);云南省重大科技专项(No.202302AG050009);云南省重大科技专项(No.202202AD080013)
中图分类号: TP391;TP301 **文献标识码:** A **文章编号:** 0372-2112(2025)02-0568-13
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20240640

Efficient Restoration for Binary Neural Networks

ZENG Kai[1,2], WAN Zi-xin[1,2], WANG Ming-tao[1,2], SHEN Tao[1,2*]
(1. *Faculty of Information Engineering and Automation*, *Kunming University of Science and Technology*, *Kunming*, *Yunnan* 650500, *China*;
2. *Yunnan Key Laboratory of Computer Technologies Application*, *Kunming University of Science and Technology*, *Kunming*, *Yunnan* 650500, *China*)

Abstract: Restoring the weight distribution, activation distribution, and gradient to the original full precision network data as much as possible can greatly improve the inference ability of the binary network. However, existing methods directly apply the restoration operation in forward propagation to binary data, and the gradient approximation functions for back-propagation are fixed or manually determined, resulting in the need for improvement in the restoration efficiency of binary networks. To address this problem, the efficient restoration method is investigated for binary neural networks. Firstly, a distribution recovery method for maximizing information entropy is proposed. By shifting the original full precision weight mean and scaling the modulus, the quantized binary weight directly has the characteristic of maximum distribution restoration. At the same time, a simple statistical translation and scaling factor is used to greatly improve the restoration efficiency of weight and activation. Furthermore, it is proposed a gradient function based on adaptive distribution approximation, which dynamically determines the update range of the current gradient in the P-percentile according to the actual distribution of the current full precision data. It adaptively changes the shape of the approximation function to efficiently update the gradient during the training process, thereby improving the convergence ability of the model. On the premise of ensuring the improvement of execution efficiency, theoretical analysis has confirmed that the method proposed in this paper can achieve maximum restoration of binary data. Compared with the existing advanced binary network models, the experimental results of our method show excellent performance, with a 60% and 67% reduction in computational time for the distribution restoration operation quantization of ResNet-18 and ResNet-20, respectively. An accuracy of 93.0% is achieved for VGG-Small

收稿日期:2024-07-08;修回日期:2024-10-22;责任编辑:孙瑶
*通讯作者:沈韬

binary quantization on the CIFAR-10 dataset, and 61.9% is achieved for ResNet-18 binary quantization on the ImageNet dataset, both of which are the best performance of the current binary neural network. The relevant code is available inhttps://github.com/sjmp525/IA/tree/ER-BNN.

Key words: binary neural network; information restoration; maximum information entropy; adaptive gradient

Foundation Item(s): Yunnan Outstanding Young Talent Project (No.202301AV070003); Yunnan Major Science and Technology Project (No.202302AG050009); Yunnan Major Science and Technology Project (No.202202AD080013)

1 引言

伴随边缘计算、高性能推理等实际需求被提出,深度神经网络的轻量化研究受到了广大学者和科研人员的关注. 其中,二值神经网络(Binarized Neural Networks,BNNs)将全精度模型量化为–1和+1,显著减少参数占用空间并加快运算速度,成为了网络轻量化的主要实现手段. 然而,二值神经网络通过使用不可导的符号函数量化全精度权重,产生的巨大信息损失导致网络性能严重下降. 因此,如何在保证推理效率的前提下,进一步优化模型的推理能力是二值神经网络的研究重点[1].

BNNs性能优化的瓶颈问题在于如何使模型的信息熵最大化[2]. 近年来,二值数据重塑是BNNs优化最为广泛使用的解决方案[3]. 该方法通过使–1和+1数量达到均衡以实现最大化信息熵,从而极大提高了模型的表达能力. 然而,重塑过程严重改变了原始权重和激活的分布特性,导致二值与全精度的特征图产生较大偏差. 因此,如何在信息熵最大化的前提下将特征分布差异降到最小仍需进一步研究.

为实现以上目标,Tu等人[4]首次提出了二值集神经网络AdaBin. 该方法通过对权重和激活进行平移和缩放,将–1和+1动态扩展为多类实值集合(如–1.34和+2.59),以此最大程度还原出全精度权重和激活的分布特性. 与重塑方法相比,在同样保证信息熵最大的前提下有效减小了量化误差. 然而,目前已有的二值集神经网络仍存在以下两方面不足.

(1) 正向推理中的平移与缩放操作低效

人工神经网络的权重通常在训练收敛后得到固定,而二值集网络每次推理均都需要对权重平移和缩放,两种思想相悖表明二值集的卷积过程仍需改进;另一方面,激活平移和缩放因子的实时化精细计算进一步增加了推理开销,而基于历史数据的简单统计方式更高效且对推理精度影响不大. 由此可见,在最大化信息熵和分布还原的前提下,权重和激活还原的执行效率需要进一步优化.

(2) 反向梯度动态还原的有效性不足

已有的二值集网络未考虑反向梯度的还原问题. 现有梯度还原研究中,用于还原符号函数真实梯度的近似函数均为固定或手动调整方式确定,导致权重更新范围的选择缺乏有效依据. 这一问题使近似梯度与真实符号函数梯度间产生了不确定偏差,陷入局部最优的模型难以高效收敛. 事实上,权重更新范围与权重和激活的实时分布状态密切相关. 如何自适应改变近似函数形状并动态确定权重更新范围,进而提高梯度还原的有效性,仍存在进一步改进的可能性.

针对现存问题,本文提出高效还原二值神经网络(Efficient Restoration for Binary Neural Networks,ER-BNN),主要贡献如图1所示.

(1)面向信息熵最大的分布恢复方法(Information-entropy Maximization Recovery,IMR). 在网络正向传播中,通过对原始全精度权重均值平移和模长缩放,使量化后的二值权重直接具备分布最大还原特性,并通过理论分析确立本方法仍满足信息熵最大准则,有效避免了推理过程中的冗余还原操作;进一步提出了基于简单统计的平移和缩放因子选择方法,减小了因子的实时化精细计算开销并将推理精度控制在可接受范围内,极大提高了二值激活的还原效率.

(2)基于自适应分布近似的梯度函数(Adaptive Data-distribution Approximation,ADA). 在网络反向传播中,根据当前全精度数据的实际分布,以P分位动态确定当前梯度的更新范围,进而自适应改变近似函数的形状. 通过理论分析表明,ADA近似函数在训练末期能够与符号函数保持高度相似,进而使近似梯度逼近真实符号函数梯度,提高了现有固定或手动式近似函数的梯度还原效率,使训练过程中的梯度得到有效更新,从而提高了模型的收敛能力.

(3)通过与当前先进二值网络模型进行对比以验证本文方法的有效性. 与当前现有的先进二值网络模型相比本文方法实验结果表现优异,其中针对Resnet-18和Resnet-20量化的分布还原操作计算时间分别下降了60%和67%;同时在CIFAR-10数据集上针对VGG-Small二值量化取得93.0%的准确率,在ImageNet数据集上针对ResNet-18二值量化取得61.9%的准确率,均为当前二值神经网络的最佳性能表现. 相关代码开源在:https://github.com/sjmp525/IA/tree/ER-BNN.

2 相关工作

在本节中,根据二值神经网络中信息流的前向传

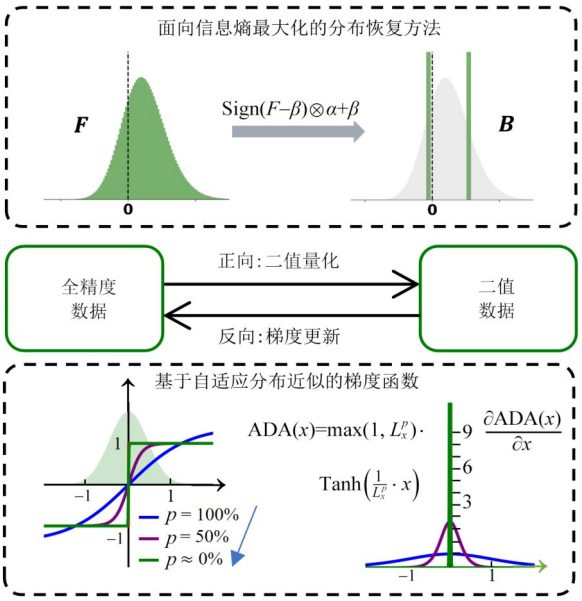


图1 二值数据高效还原方法

播和反向传播过程,回顾本领域量化优化和梯度优化的相关研究.

2.1 正向传播量化优化

二值神经网络对全精度激活和全精度权重进行二值化操作,获得更小的模型尺寸和更快的推理速度. 但是,参数的二值化会导致严重的信息丢失,与原始全精度数据之间存在巨大的量化误差,导致性能严重下降. 针对此问题,XNOR-Net(XNOR-bitcount binary convolutional neural Networks)[5]、Bi-Real(Binarizing Real-network)[6]、Real-to-Bin(binary neural networks with Real-to-Binary convolutions)[7]等方法通过优化二值网络结构,以提高模型推理能力,但未考虑二值数据的信息熵大小,其网络表征能力有待进一步提高. ReAct(binary neural network with generalized Activation functions)[8]、IR-Net(Information Retention binary neural Networks)[9]、ANE(binary neural networks with Adaptive Neural Encoder)[10]提出二值数据重塑方法,使-1和+1尽可能达到均衡,从而最大化了信息熵,极大提高了模型的信息容量和表达能力. 这些方法的重塑过程严重改变了原始权重和激活的分布特性,导致二值与全精度的特征图产生较大偏差. AdaBin[4]首次提出最优二进制集思想,在满足信息熵最大的前提下有效还原了原始二值数据分布,全面提高了二值网络的推理能力. 但AdaBin每次推理均都需对权重平移和缩放,同时激活平移和缩放因子的实时化精细计算进一步增加了推理开销.

通过以上分析,还原操作能够在满足信息熵最大的前提下使分布特性得到最大保留,是在正向传播过程中优化二值网络的有效方法. 但现有还原操作效率不足,仍需进一步改进.

2.2 反向传播量化优化

在二值神经网络中,利用阈值固定或可变的符号函数对网络的参数进行二值化操作. 然而,由于符号函数的不可导性,在反向传播期间权重将无法得到更新. 因此,使用近似函数代替符号函数进行反向梯度传播的方法一直被广泛研究. 直通估计器(Straight Through Estimator,STE)因其简洁性和高效性,首先被广泛使用. 它假设反向传播中$\mathrm{Sign}(x)=x$,但符号函数和线性函数之间的差异巨大,造成的梯度误差严重降低了二值神经网络的性能. 之后的梯度近似方法主要可以分为两大类:形状固定的近似函数和手动调整的近似函数. BNN(Binarized Neural Networks)[11]假设-1和1区间内的梯度为1,其余为0. 这种方式在一定程度上降低了梯度误差,但仍然存在巨大的梯度误差,关键问题在于一旦数据的绝对值大于1,将不会进行梯度更新. Bi-Real[6]中提出使用分段多项式函数$\mathrm{Ploy}(x)$,再次进一步减少了梯度误差,但是并没有彻底解决梯度误差和梯度消失的问题. IR-Net[9]和RBNN(Rotated Binary Neural Network)[12]分别设计了一个可以手动调整的近似函数,即Error Decay Estimator函数$\mathrm{EDE}(x)$和Training-aware函数$\mathrm{TWA}(x)$. 它们可以根据训练的时期,手动调整近似函数的形状,逐步逼近Sign函数,有效地缓解了梯度误差和梯度消失.

综上,在现有的梯度优化方法中,固定或手动给定的近似函数形状与网络数据没有有效关联全精度分布. 因此在反向传播时,无法平衡梯度的准确性和有效性,模型容易陷入局部最优.

3 方法

在本节中,首先回顾二值网络的基本理论. 进一步以原始全精度网络为基准,给出二值权重和激活的高效还原方法,理论分析了二值权重和激活在满足信息熵最大化的前提下,使其分布同时得到最大程度还原;进一步设计了梯度高效还原方法,给出自适应分布近似梯度函数,讨论了本文方法能够逼近原始符号函数. 最后给出了整体算法流程.

3.1 理论基础

二值神经网络是参数量化的一种极致情况,网络中的参数被限制在两种状态(+1,-1). 以卷积神经网络为例,卷积层的输入全精度激活F_A和全精度权重F_W将通过Sign函数进行二值化处理,生成对应的二值激活B_A和二值权重B_W. Sign函数的前向传播过程可以表述如下:

$$\text{Forward: } \mathrm{Sign}(x)=\begin{cases}+1, & \text{if } x\geqslant 0\\ -1, & \text{otherwise}\end{cases} \tag{1}$$

使用XNOR和Bitcount两种位运算操作,可以替换

传统全精度卷积操作中的乘法和加法操作,因此传统的全精度卷积可以被如下的二值卷积代替:

$$F_A \otimes F_W = \left(B_A \oplus B_W\right) \odot ak \tag{2}$$

其中,$\otimes$表示传统全精度卷积,$\oplus$表示二值卷积,$\odot$表示逐元素乘法. 大多数BNN引进了全精度缩放因子来降低Sign函数对前向数据带来的量化误差,以降低网络中正向信息流的信息损失. 使用α和k表示二值激活B_A和二值权重B_W的比例因子,它们分别由最小化量化误差$\left\| F_A - \alpha B_A \right\|_2^2$和$\left\| F_W - kB_W \right\|_2^2$得到.

在反向传播期间,通常采用直通估计器(Straight Through Estimator,STE)来解决Sign函数的不可导问题. 具体来说,使用近似函数,例如Clip函数,代替Sign函数将梯度反向传播给全精度权重以进行更新,公式化如下:

$$\text{Backward}: \frac{\partial \text{Sign}(x)}{\partial x} = \frac{\partial \text{Clip}(x)}{\partial x} = \begin{cases} 1, & \text{if } |x| \leqslant 1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

3.2　信息熵最大分布恢复的二值量化

本节讨论信息熵最大分布恢复的二值量化(Information-entropy Maximization Recovery,IMR)方法. 在二值神经网络内部,理想情况下的二值数据应该满足以下两方面的要求:(1)最小化信息损失;(2)最小化量化误差. 基于此,本节首先给出均值平移和模长缩放方法,分析了本方法能够使信息熵最大,满足了最小化信息损失要求. 基于均值和模长操作,给出权重和激活的高效还原方法,进一步讨论了该方法下二值分布和原始全精度分布的相似性,表明本方法能够满足最小化量化误差需求.

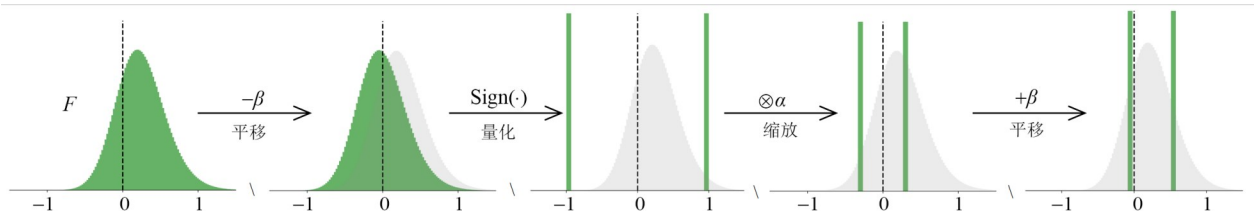


图2　信息熵最大化分布恢复的二值量化过程

3.2.1　均值平移与模长缩放

全精度数据F通过sign函数生成二值数据B. 最大化互信息可以最大化从全精度数据到二值化数据的信息流,旨在尽可能多的保留F中的信息:

$$\arg\max_{F,B} I(F;\ B) = -H(B|F) + H(B) \tag{4}$$

其中,$I(F;\ B)$是F和B的互信息,$H(B|F)$是给定F时B的条件熵,$H(B)$是信息熵. 式(4)中的条件熵$H(B|F)$可以表示为

$$\begin{aligned} H(B|F) &= \sum_f P(f) H\left(b|F=f\right) \\ &= -\sum_f P(f) \sum_b P\left(b|f\right) \log\left(P\left(b|f\right)\right) \end{aligned} \tag{5}$$

因为使用具有0阈值的确定性符号函数Sign为量化器,所以对于给定全精度数据F,生成的二值数据B只有一种确定的可能性,即式(5)中存在$P(b|f)=1$,最终$H(B|F)=0$. 因此,可以得出结论,最大化全精度数据F和二值数据B的互信息$I(F;\ B)$等价于最大化二值数据的信息熵$H(B)$,式(5)可以进而改写为

$$\begin{aligned} \arg\max_{F,B} I(F;\ B) &= \arg\max H(B) \\ &= -\sum_b P(b) \log P(b) \\ &= -P(1)\log P(1) - \left(1-P(1)\right)\log\left(1-P(1)\right) \end{aligned} \tag{6}$$

其中,$P(1)$是取值+1的概率,可以通过$H(B)$对$P(1)$的求导来计算出式(6)的最大值:

$$\begin{aligned} \frac{\mathrm{d}H(B)}{\mathrm{d}P(1)} &= -\left(\log P(1) + \frac{1}{\ln 2}\right) + \left(\log\left(1-P(1)\right) + \frac{1}{\ln 2}\right) \\ &= \log\left(\frac{1-P(1)}{P(1)}\right) \end{aligned} \tag{7}$$

根据式(7)可以得到当导数为0时,$P(1)=0.5$,二值化数据的信息熵将达到最大值0.693. 这表明,在二值神经网络中,满足信息熵最大的二值数据应呈现均匀分布,即概率质量函数$P(B)$满足以下公式条件:

$$P(B) = \begin{cases} 0.5, & \text{if } B=+1 \\ 0.5, & \text{if } B=-1 \end{cases} \tag{8}$$

通过减去全精度数据的均值,可以确保生成的二值数据的信息熵始终维持在最大值,即$\overline{B} = \text{Sign}\left(\overline{F}\right) = \text{Sign}\left(F - \text{Mean}(F)\right)$. 然而,这种操作会使新生成的二值数据$\overline{B}$与原始全精度数据$F$之间的量化误差增加. 量化误差QE用于衡量全精度数据F和二值数据B之间的分布相似程度,其定义如下式(9)所示,其中$P(F)$表示F的概率密度函数.

$$\text{QE} = \int_{-\infty}^{+\infty} P(F)(F-B)^2 \mathrm{d}F \tag{9}$$

在信息熵最大的前提下,为了使生成的二值数据$\overline{B}$与原始全精度数据F的分布最大程度匹配,对二值数据

$\overline{B}$采取进一步的缩放操作和平移操作. 具体而言,首先,对于全精度数据$F\in R^{C\times W\times H}$,其模长定义为$\sqrt{\sum_i^n (F_i)^2}$. 对于任意的二值向量$B\in\{+1,-1\}^{C\times W\times H}$,其模长始终等于$\sqrt{C\times W\times H}$,如图3所示. 为了弥补两者之间的模长差距,引入缩放因子$\alpha=\sqrt{\sum_i^n\left(\overline{F}_i\right)^2\Big/\left(C\times W\times H\right)}$对$\overline{B}$进行进一步的缩放操作,以减小量化误差.

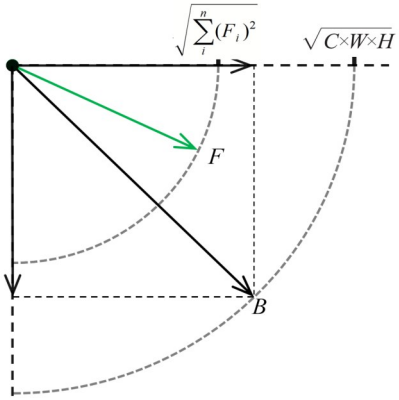


图3 模长示意图

生成的二值数据$\overline{B}$的中心位置始终等于0,而原始全精度数据F的中心为均值Mean(F). 为了对齐二值数据$\overline{B}$和原始全精度数据F的分布中心,引入平移因子$\beta=\text{Mean}(F)$对$\overline{B}$进行平移操作. 通过这两种操作,最终得到理想化的二值数据$\tilde{B}$:

$$\tilde{B}=\overline{B}\times\alpha+\beta=\text{Sign}(F-\beta)\times\alpha+\beta \tag{10}$$

新的二值数据$\tilde{B}\in(\alpha+\beta,-\alpha+\beta)^{C\times W\times H}$仍然满足信息熵最大的要求,其概率质量函数$P(\tilde{B})$如下式(11)所示. 更重要的是,$\tilde{B}$的分布与原始全精度数据$F$得到了最大程度的匹配,有助于在BNN的前向传播过程中,更有效地还原出原始全精度数据的原始信息,如图2所示.

$$P(\tilde{B})=\begin{cases}0.5, & \text{if } \tilde{B}=\alpha+\beta\\ 0.5, & \text{if } \tilde{B}=-\alpha+\beta\end{cases} \tag{11}$$

3.2.2 基于简单统计的激活还原

通过式(10)保持二值激活与全精度激活的最大相似性,可以让BNN中的原始全精度特征信息流高效地向下传递,进而提升网络性能:

$$\tilde{B}_a=\overline{B}_a\times\alpha_a+\beta_a=\text{Sign}(F_a-\beta_a)\times\alpha_a+\beta_a \tag{12}$$

然而,与训练后就保持不变的全精度权重不同,全精度激活F_a的分布并不稳定,它会随着输入图片的不同而不断变化. 通过式(10)来直接计算得出对应的缩放因子α_a和平移因子β_a,公式为$\alpha_a=\sqrt{\left(F_a-\text{Mean}(F_a)\right)_i^2\Big/\left(C\times W\times H\right)}$和$\beta_a=\text{Mean}(F_a)$,会在模型的推理阶段带来额外的计算成本,从而显著降低BNN的推理效率.

为了避免引入额外的非必要推理成本,本文提出了基于历史训练数据的简单统计方法. 在训练过程中,统计之前1 000个mini-batch中缩放因子α和平移因子β的具体数值. 在使用训练完成的模型进行推理时,将基于这些统计数据分别计算均值,代替作为推理时使用的缩放因子α和平移因子β. 对于任意的缩放因子或平移因子δ,在第t次训练迭代后,存在以下的计算关系:

$$\delta_{\text{inference}}^t=\frac{\delta_{\text{train}}^t+\delta_{\text{train}}^{t-1}+\cdots+\delta_{\text{train}}^{t-\min(t,1\,000)+1}}{\min(t,1\,000)} \tag{13}$$

使用基于1 000个mini-batch中缩放因子α和平移因子β的算数平均值作为估计,与直接计算因子值的方式相比,在提高执行效率的前提下其精度下降范围在可接受范围内,本文在消融实验中对此进行了充分讨论.

3.2.3 面向全精度重塑的权重还原

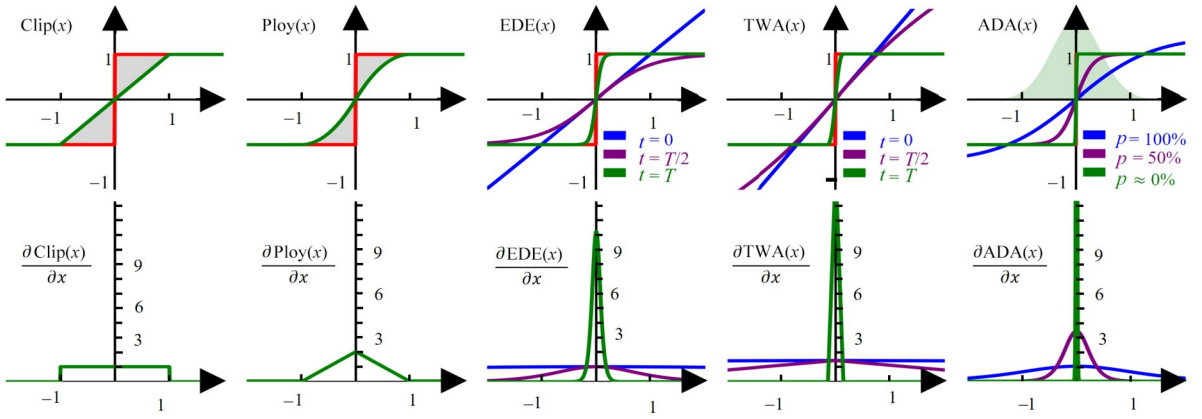
直接使用式(10)对生成的二值权重分布进行缩放平移以生成实数域二值权重,将会导致额外的推理计算成本. 事实上,二值权重在训练完成后便固定下来,不会像二值激活那样实时变化. 同时,为了降低二值卷积的计算量,本文提出对原始全精度权重的分布进行重塑,使其二值量化后的权重直接满足分布还原最大化要求,避免了推理过程中的平移和缩放操作. 该过程可公式化如下:

$$\tilde{F}_W=\frac{F_W-\beta_W}{\alpha_W}=\frac{F_W-\text{Mean}(F_W)}{\sqrt{\sum_i^n\left(F_W-\text{Mean}(F_W)\right)_i^2\Big/\left(C\times W\times H\right)}} \tag{14}$$

此时,重塑后的全精度权重$\tilde{F}_w$的均值始终为0,同时其模长始终为$\sqrt{C\times W\times H}$. 经过符号函数Sign直接生成对应的二值权重$\tilde{B}_W\in(-1,+1)$后,即式$\tilde{B}_W=\text{Sign}(\tilde{F}_W)$. $\tilde{B}_W$的信息熵仍然可以始终保持最大,同时最大化了与重塑全精度权重$\tilde{F}_W$之间的相似性. 此外,全精度权重$\tilde{F}_W$与二值权重$\tilde{B}_W$的模长相同,能够使BNN的量化误差达到最小.

3.3 自适应分布近似的梯度函数

本节讨论自适应分布近似的梯度函数(Adaptive Datadistribution Approximation, ADA). 在二值网络的反向传播过程中,为消除符号函数Sign的不可导性,通常使用可微的近似函数替代符号函数Sign,使网络参数可以进行反向梯度更新,如图4所示. 一般的近似函数Approx(x)有如下形式:



(a) 分段线性函数与梯度计算　(b) 分段多项式函数与梯度计算　(c) Error Decay Estimator 函数与梯度计算　(d) Training-aware 函数与梯度计算　(e) 自适应分布近似函数与梯度计算

图4　常见的近似函数形状对比

$$\frac{\delta \mathrm{Sign}(x)}{\delta x}=\frac{\delta \mathrm{Approx}(x)}{\delta x} \tag{15}$$

先前的工作中,提出的梯度近似方法可以分为以下两大类:(1)形状固定的近似函数,例如分段线性函数 Clip(x)、分段多项式函数 Ploy(x);(2)手动调整的近似函数,例如 Error Decay Estimator 函数 EDE(x)、Training-aware 函数 TWA(x). 具体如下:

$$\mathrm{Clip}(x)=\begin{cases}+1, & x\geqslant 1\\ x, & -1<x<1\\ -1, & x\leqslant 1\end{cases} \tag{16}$$

$$\mathrm{Ploy}(x)=\begin{cases}+1, & x\geqslant 1\\ 2x-x^2, & 0<x<1\\ 2x+x^2, & -1<x\leqslant 0\\ -1, & x\leqslant -1\end{cases} \tag{17}$$

$$\mathrm{EDE}(x)=a\cdot\tanh(b\cdot x) \tag{18}$$

$$\mathrm{TWA}(x)=\begin{cases}c\cdot\left(-\mathrm{Sign}(x)\cdot\dfrac{d^2x^2}{2}+\sqrt{2}\cdot d\cdot x\right), & |x|<\dfrac{\sqrt{2}}{d}\\ c\cdot\mathrm{Sign}(x), & \text{else}\end{cases} \tag{19}$$

使用 T 表示训练时期的总次数,t 表示当前时期. 在 EDE(x) 中,存在 $a=\max(1,\ 1/b)$,$b=10^{2t/T-1}$. 在 TWA(x)中,存在 $c=\max(1,\ 1/d)$,$d=10^{3t/T-2}$.

对于以上五种近似函数,Clip(x)是最粗略的估计,与符号函数 Sign(x)存在巨大的梯度误差. 此外,一旦数据的绝对值大于1,将不会进行更新,存在梯度饱和问题. Ploy(x)进一步减少了梯度误差,但是并没有彻底解决梯度误差和梯度饱和问题. 相比之下,EDE(x)和 TWA(x)根据训练的时期,手动调整近似函数的形状,在训练过程中逐步逼近符号函数 Sign(x),有效地缓解了这两个问题. 然而,由于手动给定的近似函数形状与网络数据的具体分布无关,因此难以有效使近似梯度得到有效还原.

本文提出自适应梯度近似函数,该方法能够根据数据的分布情况自适应的改变近似函数形状,使网络参数得到更有效的梯度更新. ADA 函数可以表示为如下计算公式:

$$\mathrm{ADA}(x)=\max\left(1,L_x^p\right)\cdot\tanh\left(\frac{1}{L_x^p}\cdot x\right) \tag{20}$$

其中,L_F^p 表示在二值化操作之前的全精度数据 F 在分位值 p 时的取值. 由于 Sign(x) 函数具有关于 Y 轴的对称性,所以在计算 L_F^p 时,将分别计算 Y 轴的左侧值 $L_{|lF|}^p$ 和右侧值 L_{rF}^p,取二者中的较大值作为最终的 L_F^p,以确保尽可能多的全精度数据可以得到有效更新.

$$L_F^p=\max\left(L_{|lF|}^p,L_{rF}^p\right) \tag{21}$$

为了凸显 ADA 方法的优越性,分位值 $p\in(0,1)$的改变采用最简单的线性策略,即 $p=1-t/T$. 随着 p 的改变,ADA(x)会根据当前输入数据的分布情况自适应的生成适当的 L_f^p,从而引导近似函数改变形状,如图4所示. 对于产生的梯度误差,可以通过衡量 Sign(x) 函数与近似函数 ADA(x)之间的阴影面积 S 来进行评估,具体评估公式如下:

$$\begin{aligned}S&=\int_{-\infty}^{+\infty}\left|\mathrm{Sign}(F)-\max\left(1,L_F^p\right)\cdot\tanh\left(\frac{1}{L_F^p}\cdot x\right)\right|\mathrm{d}x\\&=2\int_0^{+\infty}\left(1-\max\left(1,L_F^p\right)\cdot\tanh\left(\frac{1}{L_F^p}\cdot x\right)\right)\mathrm{d}x\\&=2\cdot\max\left(1,L_F^p\right)\cdot L_F^p\cdot\log\left(\tanh\left(\frac{1}{L_F^p}\cdot x\right)+1\right)\Bigg|_{x=0}^{x=+\infty}\\&=2\cdot\max\left(1,L_F^p\right)\cdot L_F^p\cdot\log(2)\end{aligned} \tag{22}$$

根据式(22)可以推出,随着训练的进行,p 将逐渐

接近0,此时存在 $\lim_{p\to 0} L_F^p = 0$,由此可以推导出 $S=0$. 即ADA(x)可以极大地逼近Sign(x),使梯度误差接近0. 同时,它不会因为数据在$[-1,+1]$之外而无法更新,大大缓解了梯度饱和问题. 在网络的整个训练中,对ADA(x)进行分析,其优势在于:(1)在训练初期,所有数据都可以得到反向梯度,保证了充足的更新能力;(2)在训练中期,越来越多的数据被钳制为$[-1,+1]$,进一步减少梯度误差,同时保持$[-1,+1]$之外的更新能力;(3)在训练末期,近似函数的形状将和符号函数Sign高度相似,可在0的附近提供精确梯度信息.

3.4 总体算法流程

训练时算法主要包含:(1)正向传播中对权重、激活进行平移缩放还原操作,计算并存储推理时所需的平移缩放因子;(2)根据分布的P分位值自适应调整近似函数,使其动态逼近符号函数Sign(x),反向传播中还原出准确梯度. 推理时算法直接读取满足最大化还原分布的二值权重,并根据训练时存储的平移缩放因子生成满足最大化还原分布的二值激活. 具体过程如算法1与算法2所示.

此外,在图5中给出了激活平移因子和缩放因子的计算实例. 具体而言,根据算法1首先在训练时将实时计算得到每个mini-batch中激活数据的平移因子 $\beta\in(0.47,0.83,-0.45,\cdots)$和缩放因子$\alpha\in(1.55,1.36,0.39,\cdots)$并缓存(上限1 000个),然后计算出其对应的均值 $\beta_{\text{inference}}^{t=3}=0.28$、$\beta_{\text{inference}}^{t=3}=0.11$并存储. 最后,将根据算法2在推理时将重新读取$\beta_{\text{inference}}^{t}$和$\alpha_{\text{inference}}^{t}$,根据式(10)直接参与到输入全精度激活的二值化还原过程,以避免生成因子时所需的实时精细化计算过程,从而提升推理效率.

算法1 训练时ER-BNN的正向传播和反向传播流程

输入:全精度激活 F_a,全精度权重 F_W

输出:满足最大化还原分布要求的二值激活 $\tilde{B}_a$、二值权重 $\tilde{B}_w$

激活平移因子与缩放因子$\beta_{\text{inference}}^{t}$、$\alpha_{\text{inference}}^{t}$

自适应近似梯度 $\frac{\delta\text{Sign}(x)}{\delta x}$

正向传播过程:

(1)通过IMR算法计算二值激活 $\tilde{B}_a$

$$\beta_a=\text{Mean}(F_a)$$

$$\alpha_a=\sqrt{\frac{\sum_i^n\left(F_a-\text{Mean}(F_a)\right)_i^2}{C\cdot W\cdot H}}$$

$$\tilde{B}_a=\text{Sign}(F_a-\beta_a)\cdot\alpha_a+\beta_a$$

$$\beta_{\text{inference}}^{t}=\frac{\beta_a^t+\beta_a^{t-1}+\cdots+\beta_a^{t-\min(t,1\,000)+1}}{\min(t,1\,000)}$$

$$\alpha_{\text{inference}}^{t}=\frac{\alpha_a^t+\alpha_a^{t-1}+\cdots+\alpha_a^{t-\min(t,1\,000)+1}}{\min(t,1\,000)}$$

(2)通过IMR算法计算二值权重 $\tilde{B}_W$

$$\beta_W=\text{Mean}(F_W)$$

$$\alpha_W=\sqrt{\frac{\sum_i^n\left(F_W-\text{Mean}(F_W)\right)_i^2}{C\cdot W\cdot H}}$$

$$\tilde{F}_W=\frac{F_W-\beta_w}{\alpha_W}$$

$$\tilde{B}_W=\text{Sign}(\tilde{F}_W)$$

反向传播过程:

(1)通过ADA算法更新网络梯度

$$P=1-\frac{t}{T}$$

$$L_F^p=\max\left(L_{|\text{lx}|}^p,L_{\text{rx}}^p\right)\cdot\frac{\delta\text{Sign}(x)}{\delta x}=\frac{\delta\text{ADA}(x)}{\delta x}$$

$$=\max\left(1,L_x^p\right)\cdot\frac{1}{L_x^p}\left(1-\tanh^2\left(\frac{1}{L_x^p}\cdot x\right)\right)$$

算法2 推理时ER-BNN的正向传播流程

输入:全精度激活 F_a,二值权重 $\tilde{B}_w$

激活平移因子与缩放因子$\beta_{\text{inference}}^{t}$、$\alpha_{\text{inference}}^{t}$

输出:满足最大化还原分布要求的二值激活 $\tilde{B}_a$、二值权重 $\tilde{B}_w$

正向传播过程:

(1)计算二值激活 $\tilde{B}_a$

$$\tilde{B}_a=\text{Sign}(F_{\text{a}}-\beta_{\text{inference}}^{t})\cdot\alpha_{\text{inference}}^{t}+\beta_{\text{inference}}^{t}$$

4 实验

本节首先介绍数据集与实验设置,进而通过消融实验、对比实验和复杂度分析验证本文方法的有效性和优越性.

4.1 数据集与实验设置

4.1.1 数据集

实验使用了三个广泛用以二值网络效果验证的数据集:MNIST、CIFAR-10和ImageNet.

(1)MNIST. 包含6万张训练图片和1万张测试图片,均为灰度图像数据集. 每张图片的大小为28×28,代表数字0到9. 为了保持此基础数据集的挑战性,在训练和测试过程中未应用卷积、数据增强、预处理或其他操作.

(2)CIFAR-10. 由6万张32×32大小的RGB图像组成,覆盖10个类别,包括飞机、汽车、鸟、猫、鹿、狗、青蛙、马、轮船、卡车. 其中,5万张图像用于训练,1万张图像用于测试. 在训练过程中,采用常规的数据增强技术,包括在图像两侧各填充四个像素、随机裁剪和随机水平翻转. 而在测试时,则对原始RGB图像的单一视图进行直接评估.

(3)ImageNet. 包含大约120万张训练图像和5万张验证图像,分布在1 000个类别中. 在训练阶段,采用随机裁剪、调整大小为224×224、随机水平翻转的常规数据增强方法. 在测试阶段,采用大小224×224的中心裁剪方法进行评估.

4.1.2 实验设置

算法使用PyTorch V2编程开发,并在配备英伟达3080Ti GPU的服务器上训练所有模型. 本文提出的信息还原方法可灵活部署于各类二值神经网络,为全面验证本文方法的有效性,实验的基线网络和训练细节的设置如下.

(1)网络结构. 在MNIST中,为了确保公平比较,采用与BNN[11]中相同的多层感知器(MLP)网络结构,它由3个具有2 048个二进制单元的隐藏层组成. 在CIFAR-10和ImageNet上,使用广泛流行的网络结构进行了评估,包括ResNet-18、ResNet-20和VGG-Small. 为了确保公平,除网络的第一层和最后一层外,对剩余的所有全精度卷积层都进行了二值化处理. 对于ResNet-20和ResNet-18,采用了Bi-Real中提出的双跳连接,以进一步增强网络性能. 此外,IR-Net方法在ResNet-18网络结构上没有采用双跳连接,本章在相关实验中保持了相同的设置,以保证对比实验的公平性.

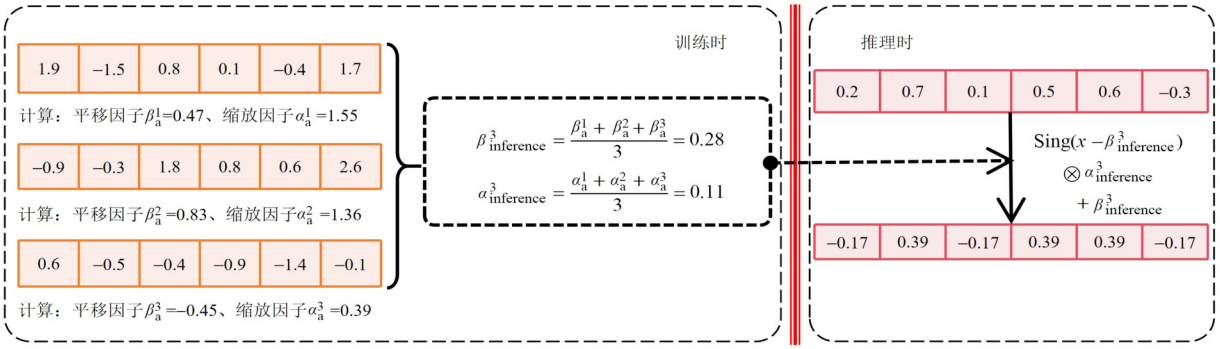


图5　激活平移缩放因子的计算实例

(2)训练细节. ER-BNN是从零开始训练的,没有使用任何的预训练模型,与典型的单阶段训练方法一致. 在MNIST中,使用Dropout对二值模型进行正则化,并采用Adam优化器,批量大小设置为256,与BNN[11]中使用的配置一致. 在CIFAR-10和ImageNet数据集上的实验中,选择RPRelu作为激活函数. 使用随机梯度下降(Stochastic Gradient Descent,SGD),且动量设置为0.9. 此外,使用余弦退火调节器调整学习率,且初始学习率统一设置为0.1. 对于CIFAR-10、MNIST和ImageNet数据集,训练批量大小统一设置为256.

4.2 消融实验

本节首先在基线模型中分别对两个模块进行消融,以整体验证本文分布还原(Information-entropy Maximization Recovery,IMR)和自适应近似(Adaptive Data-distribution Approximation,ADA)方法的对二值网络精度的提升能力;后续通过可视化以及与其他模型单独结合的方式,进一步验证本文方法的有效性.

4.2.1 网络整体消融实验

本节展示了各种配置在CIFAR-10数据集上的消融实验结果,具体包括全精度网络、二值基线、施加IMR方法后的二值基线、施加ADA方法后的二值基线、由IMR和ADA组成的完整ER-BNN算法. 注意在二值基线中,使用分段线性函数Clip(x)作为符号函数的近似函数,使用RPRelu(x)作为非线性激活函数.

最佳准确度以粗体在表1所示. 将IMR方法单独应用于基准模型时,三种网络结构的准确度分别提高了1.8%、1.7%和2.1%;将ADA方法单独应用于基准模型时,三种网络结构的准确度分别提高了1.2%、1.5%和1.4%. 这一结果验证了本文IMR和ADA的有效性. 将两者相结合,本文分方法在三种网络结构上分别提高了2.5%、2.5%和3.1%的准确度. 这一结果表明,IMR和ADA可以进一步叠加以提升二值网络能力.

表1　本文方法在CIFAR-10数据集上的消融实验

消融过程	位宽	准确率/%		
		VGG-Small	ResNet-20	ResNet-18
全精度	32/32	94.1	91.7	94.8
基线	1/1	90.5	86.0	90.7
基线+IMR	1/1	92.3	87.7	92.6
基线+ADA	1/1	91.7	87.5	92.1
基线+IMR+ADA	1/1	**93.0**	**88.5**	**93.8**

4.2.2 信息熵最大分布恢复有效性验证

为了更深入地评估IMR方法的有效性,本实验对二值神经网络的中间层特征进行了可视化. 如图6所示,在使用传统的符号函数Sign对全精度数据特征进行二值化时,固定的阈值生成的二值特征存在严重的失真现象,导致信息大量损失. 相反的,使用本文提出的IMR方法后,最大程度地保留了从全精度数据到二值数据的信息流,有助于为二值卷积操作提供具有丰富表征能力的中间二值特征.

另一方面,在激活的恢复方法中没有采用实时精细

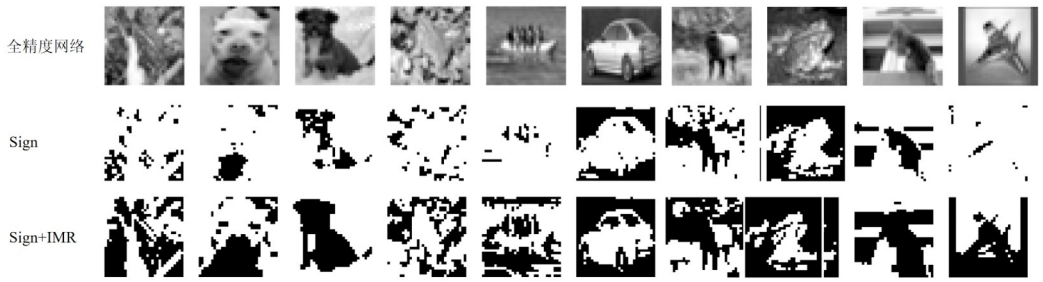


图6 IMR中间特征消融实验可视化结果

化因子计算方式，而是使用了基于历史数据的简单统计方法，其目的是避免反复计算缩放因子α_a和平移因子β_a，以降低额外的非必要推理成本. 表2对比了本文提出的简单统计方法与实时精细计算方法的准确率和每批次大小为256的平均时间开销. 结果表明，在VGG-Small和ResNet-20模型上，准确率仅下降了0.2%和0.4%，而计算时间开销却大幅降低了56%和31%. 由此可以得出，基于训练数据的简单统计方法对性能的影响较小，以轻微的性能下降换取时间代价的大幅降低是值得的，处于可接受的范围内.

表2 因子计算时间消融实验

网络模型	因子计算方法消融	位宽	准确率/%	因子计算时间开销/ms
VGG-Small	ER-BNN+实时精细计算	1/1	93.2	18
	ER-BNN+简单统计	1/1	**93.0**	**8**
ResNet-20	ER-BNN+实时精细计算	1/1	88.9	77
	ER-BNN+简单统计	1/1	**88.5**	**53**

4.2.3 自适应分布近似(ADA)有效性验证

为了更深入地评估ADA方法的有效性，进一步将ADA方法单独添加到AdaBin中进行测试. AdaBin对二值神经网络的正向传播过程进行了全面优化，但没有对反向传播过程进行专门的处理. 在向AdaBin添加ADA方法后，如表3所示，其性能获得了进一步的提升，3种网络结构的准确度分别提高了0.5%、0.3%和0.4%.

表3 ADA在AdaBin模型上的消融实验结果

网络模型	二值方法基线与消融	位宽	准确率/%
VGG-Small	AdaBin[4]	1/1	92.3
	AdaBin[4] + ADA	1/1	**92.8**
ResNet-20	AdaBin[4]	1/1	88.2
	AdaBin[4] + ADA	1/1	**88.5**
ResNet-18	AdaBin[4]	1/1	93.1
	AdaBin[4] + ADA	1/1	**93.5**

4.2.4 消融实验结果分析

消融实验分别验证了本文IMR和ADA的有效性. 其主要原因有以下两点.

(1)IMR实现了二值数据信息熵最大化，保持了信息流数据的多样性，从而最小化了信息损失并增强了网络的表征能力；IMR最大程度还原出原始全精度数据的分布，从而使量化损失达到最小，保障了数据的完整性和准确性. 通过两方面工作，有效提升了二值网络的推理能力.

(2)ADA根据数据的实时分布情况，自适应地调整近似函数的形状，能够更加有效地刻画梯度的更新范围. 在训练过程中逐渐逼近符号函数Sign，最终能够达到梯度的最大程度还原梯度，并使梯度误差接近0，有效缓解了梯度饱和问题，平衡了梯度的准确性和有效性，进而提高了推理精度.

4.3 对比试验

本节首先在不同数据集上整体对比本文方法与当前主要的二值网络方法，进一步从分布还原和梯度还原两方面与其他同类方法进行对比，以验证本文方法的优越性.

4.3.1 不同数据集上的准确性对比

(1)MNIST数据集上的对比实验

将本文方法与多种二值神经网络优化方法进行了比较，包括BNN[11]、XNOR[5]、LAB2[13]和Si-BNN[14]，结果列于表4. ER-BNN方法取得了最高的准确度，仅比全精度MLP网络低0.01%，能够同性能的代替全精度神经网络. 同时，这表明所提出的ER-BNN方法不仅适用于卷积神经网络CNN，也适用于全连接网络MLP，可以显著提升性能.

(2)CIFAR-10数据集上的对比实验

表5展示了在CIFAR-10上不同BNN方法的性能比较结果，包括VGG-Small上的XNOR、BNN、Si-BNN、IR-Net[9]、ANE[10]、SD-BNN[15]、DIR-Net[16]和RBNN[12]；ResNet-20上的DSQ[17]、ANE、SLB[18]、Bi-Real[6]、LNS[19]、RBNN、IR-Net和SD-BNN；以及ResNet-18上的XNOR、ANE、

BNSC[20]、IR-Net、RBNN、ReAct[8]、SD-BNN、BNAS[21]和DIR-Net.

表4　ER-BNN方法在MNIST数据集上的对比实验结果

网络模型	二值方法	位宽	准确率/%
MLP	Full-precision	32/32	98.81
	BNN[11]	1/1	98.53
	XNOR[5]	1/1	98.47
	LAB2[13]	1/1	98.62
	Si-BNN[14]	1/1	98.74
	ER-BNN(本文方法)	1/1	**98.80**

表5　ER-BNN方法在CIFAR-10数据集上的对比实验结果

网络模型	二值方法	位宽	准确率/%
VGG-Small	全精度	32/32	91.7
	XNOR[5]	1/1	89.8
	BNN[11]	1/1	89.9
	Si-BNN[14]	1/1	90.2
	IR-Net[9]	1/1	90.4
	ANE[10]	1/1	90.8
	SD-BNN[15]	1/1	90.8
	DIR-Net[16]	1/1	91.1
	RBNN[12]	1/1	91.3
	ER-BNN(本文方法)	1/1	**93.0**
ResNet-20	全精度	32/32	91.7
	DSQ[17]	1/1	84.1
	ANE[10]	1/1	85.3
	SLB[18]	1/1	85.5
	Bi-Real[6]	1/1	85.7
	LNS[19]	1/1	85.8
	RBNN[12]	1/1	86.5
	IR-Net[9]	1/1	86.5
	SD-BNN[15]	1/1	86.9
	ER-BNN(本文方法)	1/1	**88.5**
ResNet-18	全精度	32/32	93.0
	XNOR[5]	1/1	90.2
	ANE[10]	1/1	90.9
	BNSC[20]	1/1	91.1
	IR-Net[9]	1/1	91.5
	RBNN[12]	1/1	92.2
	ReAct[8]	1/1	92.3
	SD-BNN[15]	1/1	92.5
	BNAS[21]	1/1	92.7
	DIR-Net[16]	1/1	92.8
	ER-BNN(本文方法)	1/1	**93.8**

在所有情况下,ER-BNN都取得了最高的性能. 由图8中可以看出准确度稳步提高,损失也稳定收敛. 具体而言,相较于在前向传播过程中使用Libra参数二值化(Libra Parameter Binarization, Libra-PB),并在反向传播过程中使用误差衰减估计器(Error Decay Estimator, EDE)的IR-Net,ER-BNN在三种网络结构上的准确度分别高出2.6%、2.0%和2.3%. 同样,与在前向过程中减小角度误差,并在反向过程中采用训练近似感知(Training-aWare Approximation, TWA)的RBNN相比,ER-BNN在三种结构上的准确度分别高出1.7%、2.0%和2.9%. 与同时结合教师中间特征和教师软标签进行离线知识蒸馏的ANE相比,ER-BNN在三种结构上的准确度分别高出2.2%、3.2%和2.9%. 此外,与采用两步训练策略的二值方法相比(第一步训练时对激活进行二值化,第二步训练时对权重和激活同时进行二值化),所提出的ER-BNN仍保持领先地位,例如在ResNet-18上准确度比ReAct高1.5%,在ResNet-20上准确度比Bi-Real高2.8%.

(3)ImageNet数据集上的对比实验

对于大规模的ImageNet数据集,在ResNet-18网络结构上进行实验. 如表6所示,将ER-BNN与其它最先进的二值神经网络方法进行了比较,包括XNOR、Bi-Real、IA-BNN[22]、IR-Net、BNAS、DGRL[23]、Si-BNN、RBNN、DIR-Net. EM-BNN表现出显著的性能优势. 具体而言,与使用两步训练策略的Bi-Real相比,ER-BNN的准确度提高了5.5%. 此外,与使用预训练教师的中间特征进行离线知识蒸馏的DGRL相比,准确度提高了2.4%. 更重要的是,与同样对二值神经网络的正向和反向过程都优化的IR-Net和RBNN相比,ER-BNN的准确度分别高出3.8%和2.0%.

表6　ER-BNN方法在ImageNet数据集上的对比实验结果

网络模型	二值方法	位宽	准确率/%
ResNet-18	全精度	32/32	69.6
	XNOR[5]	1/1	51.2
	Bi-Real[6]	1/1	56.4
	IA-BNN[22]	1/1	57.2
	IR-Net[9]	1/1	58.1
	BNAS[21]	1/1	58.8
	DGRL[23]	1/1	59.5
	Si-BNN[14]	1/1	59.7
	RBNN[12]	1/1	59.9
	DIR-Net[16]	1/1	60.4
	ER-BNN(本文方法)	1/1	**61.9**

4.3.2　不同分布还原方法对比

本节通过对比不同方法的还原效率以验证本文方法IMR的先进性. 根据调研,AdaBin是除本文以外的唯一分布还原研究. 通过理论分析,本文方法与AdaBin均可实现分布的最大化还原,因此本节主要对比两者还原操作的计算效率. 如图7所示,针对Resnet18和Resnet20的量化操作,本文方法在一个Epoch中分布

还原计算时间分别下降了60%和67%.

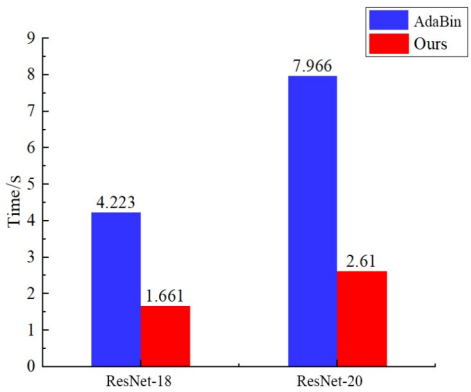


图7 不同分布还原方法推理时间

4.3.3 不同梯度近似函数对比

本节以IR-Net为基线,分别选择Clip(x)(固定)、EDE(x)(手动)、和本文提出的ADA(x)(自适应)等3种近似函数进行对比,在CIFAR-10数据集上的训练结果如图8所示. 相比于固定和手动近似函数,本文方法能够更加高效地近似真实梯度,进而获得更快的收敛效果. 此外,以IR-Net作为基线模型,将典型常用的手动调整函数EDE(x)与本文提出的自适应近似函数ADA(x)进行进一步的对比,在CIFAR-10数据集上的分类精度结果如表7所示. 使用ADA(x)后的IR-Net取得更佳的性能表现,将分类精度分别提升了1.1%、0.9%和0.8%.

表7 ADA方法与EDE方法的对比实验结果

Model	近似梯度函数	位宽	准确率/%
VGG-Small	IR-Net[9]+EDE	1/1	90.4
	IR-Net[9]+ADA	1/1	**91.5**
ResNet-20	IR-Net[9]+EDE	1/1	86.5
	IR-Net[9]+ADA	1/1	**87.4**
ResNet-18	IR-Net[9]+EDE	1/1	91.5
	IR-Net[9]+ADA	1/1	**92.3**

4.3.4 模型计算量对比

表8展示了ER-BNN在内存占用和计算成本两个方面的优势. 内存存储大小的计算方法为全精度参数数量乘32再加上二进制参数的数量. 操作总数(Operations Per second, OPs)使用二进制操作(Binary Operations Per second, BOPs)和浮点操作(FLoating Point Operations Per second, FLOPs)进行计算,即OPs=BOPs/64+FLOPs,因为包括XNOR和Pop-Count在内的二值卷积运算可由主流CPU并行执行64次.

表8 ER-BNN的内存、OPs、Acc对比实验结果

二值网络	位宽	存储大小/Mbit	OPs/M	准确率/%
Full-precision	32/32	374.1	1 810	69.6
XNOR[5]	1/1	33.7	167	51.2
Bi-Real[6]	1/1	33.6	163	56.4
IR-Net[9]	1/1	33.3	163	58.1
DGRL[23]	1/1	33.6	163	59.5
ER-BNN	1/1	**33.3**	**163**	**61.9**

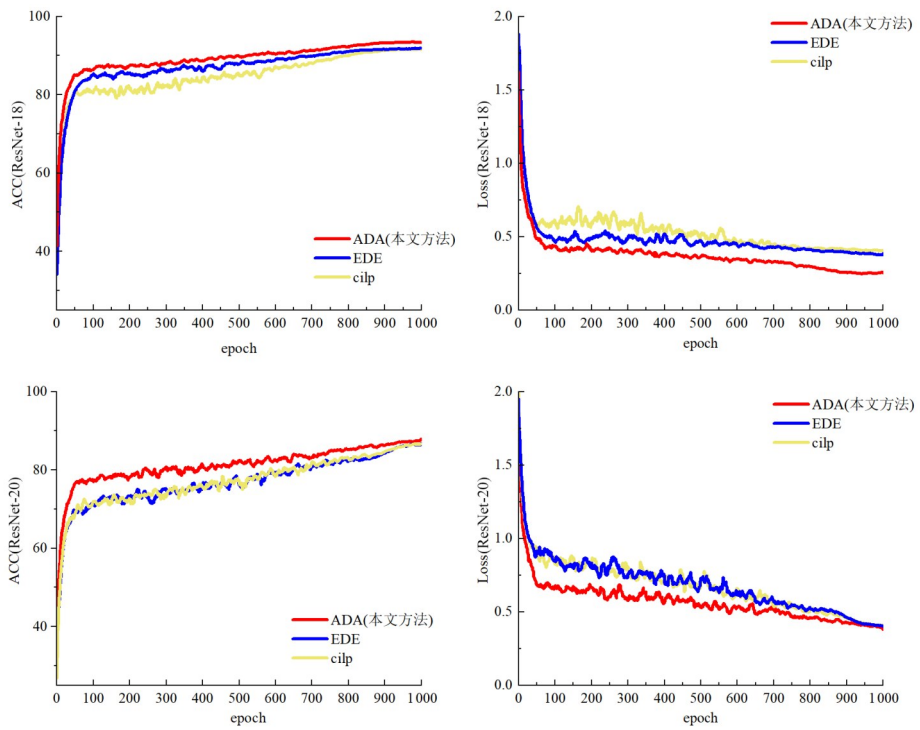


图8 不同近似函数的收敛曲线

可以看到,ER-BNN的内存占用仅是全精度网络的8.9%,计算成本仅是全精度网络的9.0%. 在参数的二值化过程中,ER-BNN只引入了一些额外的加法操作,但可以忽略不计. 与广泛使用的IR-Net相比,ER-BNN的内存和计算量相差不大,但在准确度上取得了3.8%的巨大提升.

4.3.5　对比实验分析

对比实验结果表明,本文方法在还原计算效率上取得较大提升. 除此之外,推理精度也优于现有的先进方法. 特别是在CIFAR-10数据集上针对VGG-Small二值量化取得93.0%的准确率;在ImageNet数据集上对ResNet-18的二值量化取得61.9%的准确率,均为当前二值网络的最佳性能表现. 其如要原因如下.

(1)相比于原始全精度网络,IMR和ADA分别在分布和梯度两方面实现了最大程度还原,其效果通过叠加进而提高了二值网络能力. 具体内容已在消融实验中进行了分析.

(2)相比于其他还原方法,IMR在保证分布最大还原的前提下,将分布还原操作实施于原始全精度数据,进而避免了推理过程中的还原过程,提高了推理效率;另一方面,在梯度还原中采用了简单统计方法,避免了实时精细化计算带来的冗余开销,消融实验中已验证本方法能够极大提高计算效率并且对精度影响不大.

(3)现有近似梯度函数均为固定或手动式调整,使反向传播中权重难以有效更新. 本文ADA根据实时数据分布自适应调整近似函数形状,能够更加客观地刻画近似梯度函数,从而确定权重更新范围并有效更新梯度. 实验结果验证了梯度更新范围与分布具有相关性.

综上,在现有的梯度优化方法中,固定或手动给定的近似函数形状与网络数据没有有效关联全精度分布. 因此在反向传播时,无法平衡梯度的准确性和有效性,模型容易陷入局部最优.

5　结论与展望

本文介绍了所提出的信息最大化二值神经网络算法,其可以使网络中的正向量化信息、反向梯度信息最大程度的还原为原始全精度信息. 具体而言包含以下两种方法. 首先,提出信息熵最大化恢复方法方法(IMR),可以使二值化后的数据尽可能多的保留原始全精度信息,并且始终保持最大的信息熵,使参数的量化损失保持最小. 其次,提出自适应数据分布近似方法(ADA),能够根据全精度数据的分布情况自适应改变近似函数形状,逐渐逼近符号函数的原始真实梯度,使二值神经网络的参数得到更有效的梯度更新. 在三个数据集和四种网络结构上进行的广泛实验充分证明了所提出的ER-BNN算法的有效性和优越性.

对于二值神经网络的应用领域,目前主要集中在分类任务上,但其潜力应远不止于此. 模型二值化压缩的初衷是为低功耗移动设备、嵌入式设备提供使用深度神经网络的可能性,因此其实际部署应用至关重要. 然而,在语音、视频、点云以及其他时序信号领域,二值量化技术的应用仍然存在着诸多空白. 为充分发挥二值神经网络的优势,未来的研究应更加关注多个应用领域的特性,从而拓宽其可部署应用的范围,使更多的领域能够受益于这一前沿技术. 此外,为了提升二值神经网络在实际部署中的效能,应该专注于设计部署友好的二值网络结构. 虽然二值量化相较于其他高位量化模式在移植上更为便捷,但现有研究中往往难以兼顾算法的准确性与硬件的友好度. 更为罕见的是,少有团队致力于相关的专有硬件设计. 因此,在后续的工作中,应该特别关注硬件友好的二值神经网络设计,确保网络结构在实际部署中具备良好的硬件兼容度,从而充分发挥其性能优势.

参考文献

[1] 王子为, 鲁继文, 周杰. 基于自适应梯度优化的二值神经网络[J]. 电子学报, 2023, 51(2): 257-266.
WANG Z W, LU J W, ZHOU J. Learning adaptive gradients for binary neural networks[J]. Acta Electronica Sinica, 2023, 51(2): 257-266. (in Chinese)

[2] YUAN C, AGAIAN S S. A comprehensive review of binary neural network[J]. Artificial Intelligence Review, 2023, 56(11): 12949-13013.

[3] 袁海英, 成君鹏, 曾智勇, 等. Mobile _ BLNet:基于Big-Little Net的轻量级卷积神经网络优化设计[J]. 电子学报, 2023, 51(1): 180-191.
YUAN H Y, CHENG J P, ZENG Z Y, et al. Mobile_BLNet: Optimization design of lightweight convolutional neural network based on Big-Little Net[J]. Acta Electronica Sinica, 2023, 51(1): 180-191. (in Chinese)

[4] TU Z, CHEN X, REN P, et al. AdaBin: Improving binary neural networks with adaptive binary sets[C]//European Conference on Computer Vision (ECCV). Cham: Springer, 2022: 379-395.

[5] RASTEGARI M, ORDONEZ V, REDMON J, et al. XNOR-Net: ImageNet classification using binary convolutional neural networks[C]//European Conference on Computer Vision (ECCV). Cham: Springer, 2016: 525-542.

[6] LIU Z, LUO W, WU B, et al. Bi-Real Net: Binarizing deep network towards real-network performance[J]. International Journal of Computer Vision, 2020, 128(1): 202-219.

[7] MARTINEZ B, YANG J, BULAT A, et al. Training binary neural networks with real-to-binary convolutions[C]//Inter-

national Conference on Learning Representations (ICLR). Piscataway: IEEE, 2020: 1-13.
[8] LIU Z, SHEN Z, SAVVIDES M, et al. ReActNet: Towards precise binary neural network with generalized activation functions[C]//European Conference on Computer Vision (ECCV). Cham: Springer, 2020: 143-159.
[9] QIN H, GONG R, LIU X, et al. Forward and backward information retention for accurate binary neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 2250-2259.
[10] ZHANG S, GE F, DING R, et al. Learning to binarize convolutional neural networks with adaptive neural encoder[C]//International Joint Conference on Neural Networks (IJCNN). Piscataway: IEEE, 2021: 1-8.
[11] HUBARA I, COURBARIAUX M, SOUDRY D, et al. Binarized neural networks[J]. Advances in Neural Information Processing Systems, 2016, 29: 1-14.
[12] LIN M, JI R, XU Z, et al. Rotated binary neural network[C]//Neural Information Processing Systems (NeurIPS). Piscataway: IEEE, 2020: 7474-7485.
[13] LU H, YAO Q, KWOK J T. Loss-aware binarization of deep networks[C]//International Conference on Learning Representations (ICLR). Piscataway: IEEE, 2017: 24-26.
[14] WANG P S, HE X Y, LI G, et al. Sparsity-inducing binarized neural networks[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12192-12199.
[15] XUE P, LU Y, CHANG J, et al. Self-distribution binary neural networks[J]. Applied Intelligence, 2022, 52(12): 13870-13882.
[16] QIN H, ZHANG X, GONG R, et al. Distribution-sensitive information retention for accurate binary neural network[J]. International Journal of Computer Vision, 2023, 131(1): 26-47.
[17] GONG R, LIU X, JIANG S, et al. Differentiable soft quantization: Bridging full-precision and low-bit neural networks[C]//IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2019: 4852-4861.
[18] YANG Z, WANG Y, HAN K, et al. Searching for low-bit weights in quantized neural networks[J]. Advances in Neural Information Processing Systems, 2020, 33: 4091-4102.
[19] HAN K, WANG Y, XU Y, et al. Training binary neural networks through learning with noisy supervision[C]//International Conference on Machine Learning (ICML). Piscataway: IEEE, 2020: 4017-4026.
[20] WU L J, LIN X, CHEN Z C, et al. An efficient binary convolutional neural network with numerous skip connections for fog computing[J]. IEEE Internet of Things Journal, 2021, 8(14): 11357-11367.
[21] KIM D, SINGH K P, CHOI J. Learning architectures for binary networks[C]//European Conference on Computer Vision (ECCV). Cham: Springer, 2020: 575-591.
[22] KIM H, PARK J, LEE C, et al. Improving accuracy of binary neural networks using unbalanced activation distribution[C]//Computer Vision and Pattern Recognition (CVPR). Cham: Springer, 2021: 7862-7871.
[23] YE J, WANG J, ZHANG S. Distillation-guided residual learning for binary convolutional neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(12): 7765-7777.

作者简介

曾 凯 男，1985 年生，副教授，昆明理工大学硕士生导师．主要研究方向为模型压缩、边缘计算．
E-mail: zengkai@kust.edu.cn

王铭涛 男，现就读于昆明理工大学硕士研究生．主要研究方向为二值神经网络、计算机视觉．
E-mail: wangmingtao@stu.kust.edu.cn

万子鑫 男，2024 年取得昆明理工大学硕士学位．主要研究方向为神经网络量化与二值神经网络．
E-mail: wanzixin@stu.kust.edu.cn

沈 韬 男，教授，昆明理工大学博士生导师，云南省杰出基金项目获得者．主要研究方向为边缘计算、多源智能感知．
E-mail: shentao@kust.edu.cn