- where⊙denotes the inner product for vectors with bitwiseoperations XNOR and Bitcount.

- minimizing the quantization error

- For binary models, therepresentation ability of their parameters is limited to two val-ues, which makes the information carried by neurons easy tolose.

-

- 二元神经网络的求解空间

- 仅通过最小化量化误差来保证一个好的二值化网络是不够的和困难的。

- 天秤座（平衡）参数二值化

- jointly considers both quanti-zation error and information loss

- If we only pursue the goal of minimizing quantization error,the information entropy of the quantized parameters can beclose to zero in extreme case

- $$\min J(Q_x(\mathbf{x})) - \lambda \mathcal{H}(Q_x(\mathbf{x})).$$

- evenly distributed

- 均匀分布

- subtracting the mean of full-precisionweights

- subtracting the mean of full-precisionweights

- standardization and balance operations

- more stable during the training

- can maximize the information entropy

- bit-shift scalar

- our Libra Parameter Binarization for the for-ward propagation can be presented as below:

- To further minimize the quantization error and avoid extraexpensive floating-point calculation in previous binarizationmethods, Libra-PB introduces an integer bit-shift scalarsto expand the representation ability of binary weights.

- the in-formation entropy ofQx(x)is larger thansign(x)

- 0.5 and 0.2

- an implicit rectifier

- For example, [13] pro-posed the specific degeneration problem of binarization andsolved it using a specially designed additional regularizationloss

- 二值化前参数分布的影响

- and promise the optimalsolution by maximizing the information entropy

- be directly plugged into the standard trainingpipeline with a very limited extra computation cost.

- 反向传播中的误差衰减估计器

- 不连续

- huge information loss

- g(w)denotes theapproximation of thesignfunction

- straightly passes

- The Identity function straightly passes the gradient in-formation of output values to input values and completely neglects the effect of binarization

- 完全忽略了二值化的影响

- 梯度误差很大，在反向传播过程中会累积

- retain correct gradient information

- takes the clipping attribute of binariza-tion into account

- pass the gradient information inside the clipping interval

- 逐渐逼近符号函数（原函数）

- 开始的时候需要更大的梯度用于网络的更新1

-

(c) EDE
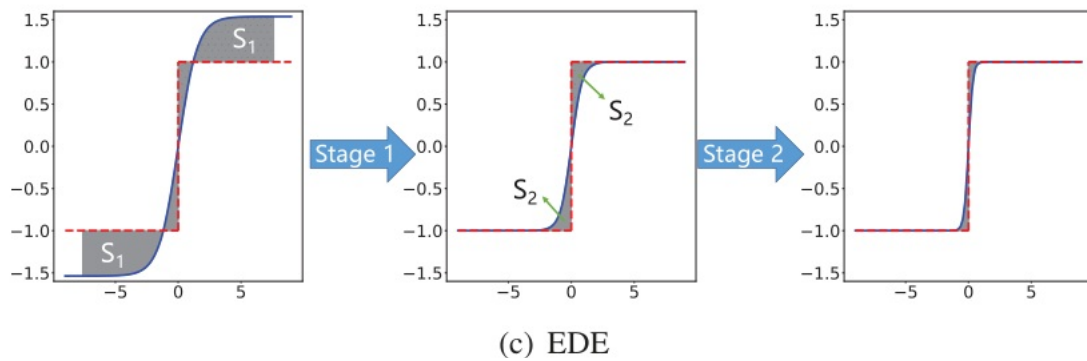
- y=x

- y=clip(-1,1,x)

- outside the clamping interval, it cannot be updatedanymore

- 一旦超出范围，网络难以更新

- 确保足够的更新可能性是至关重要的，特别是在培训过程的开始阶段。

- Itis crucial to ensure enough updating possibility, especiallyduring the beginning of the training process.

- his characteristic greatly harms the updating abil-ity of backward propagation, which can be proved by the factthat ReLU is a superior activation function compared withTanh.

- loses the gradient information ofquantization

- make a balance and acquire the optimal approximation ofbackward gradient, we design Error Decay Estimator:

- 把两者的优势结合起来

- $$t = T_{\min} 10^{\frac{i}{N} \times \log \frac{T_{\max}}{T_{\min}}}, \quad k = \max(\frac{1}{t}, 1)$$

- S1在Stage1坍缩 S2在Stage2收缩

- Get current t and k by Eq

- k早期取值1/t(因为开始的时候t小于1)，k的取值>1，因此阴影部分面积比较大

- $g'(x) = kt(1 - \tanh^2(tx))$

- Ours0C1× C2+ C1

- Bit-shift.

- obviously more concentrated around 0.

- 这一现象意味着大多数权重的符号在优化过程中很容易发生变化

- Thisphenomenon means the signs of most weights are easy tochange during the process of optimization,

- 应该是自动求微分

- the convolutional layers

- 除了第一个层和最后一个层外，所有的卷积层和全连接层都是二值化的

- Bi-Real [38]structure of ResNet

- 这应该是一个针对二值化改进的ResNet

- Hardtanh

- trained from scratch (ran-dom initialization)

- mostly follow the hyper-parameter settings of their originalpapers

- 尽可能使用了原始文章中的超参数

- SGD

- 从信息熵的角度对权值进行简单而有效的变换

- which simultaneously reduces the information lossof both weights and activations

- by a large margin

- 大量实验

- Vanilla binarization suffers a largereduction in the information entropy of the binary activations.

- The weight distributions are shown in theupper part.

- the blue line is the derivative of EDE

- 下面的图是导数图像

- derivative

- STE

- sign

- all weights can be effectively updated

- there is much data outside the range [-1,+1], thus there should not be much clipping which will beharmful to update ability

- 早期，很多都不在[-1,1]之间

- 同时，有很多值都在0附近，如果使用符号函数将会出现较大问题

- EDE keeps the derivative similartoIdentityfunction at this stage to ensure the derivativearound zero not too large

- 通过P3可以发现，权重的峰值逐渐向-1/+1（二值化目标）靠拢

- Thus we can slowly increase thevalue of derivative and approximate a standardsignfunc-tion to reduce the gradient mismatch.

- In all cases, our IR-Netobtains the best performance

- we use the original ResNet structure or the Bi-Real struc-ture.

- to 4.3%.

- IR-Net over ResNet-18 and ResNet-34

- outperforms

- 胜过

- 甚至超过了更高bit量化的模型

-

- 1、2上标的意思

- Raspberry Pi 3B

- daBNN

- We have to point out that tillnow there have been very few studies that reported theirinference speed in real-world devices, especially when using1-bit binarization.

- the model size of IR-Net can be greatly reduced and thebit-shift scales in IR-Net bring almost no extra inferencetime and storage consumption.

- 这里选择Ours1和2有什么小心思吗

- 摘要总结：

- simultaneouslyminimizing both quantization error and information loss ofparameters by balanced and standardized weights in forwardpropagation

- 在前向传播中，通过平衡的和标准化的权值来最小化参数的量化误差和信息损失

- 通过在反向传播中逐步逼近信号函数，从而减小梯度的信息损失，共同考虑更新能力和精确梯度的信息损失。

- minimizingthe information loss of gradients by gradually approximatingthesignfunction in backward propagation, jointly consid-ering the updating ability and accurate gradients.

- Althoughmany binarization methods have improved the accuracy ofthe model by minimizing the quantization error in forwardpropagation, there remains a noticeable performance gapbetween the binarized model and the full-precision one.

- 许多二值化方法通过最小化前向传播中的量化误差

- For example, Bi-Real Net [38]is targeted at the latter by adding on a full-precision short-cut to the quantized activations, which achieves significantperformance improvement.

- 例如，Bi-Real Net [38]通过在量化激活中添加一个全精度的捷径，从而实现了显著的性能提高。

- mainly categorizedinto either improving the updating ability or reducing themismatching area betweensignfunction and the approxi-mation one

- where in prac-tice strong updating ability is usually highly required whenthe training process starts and small gradient error becomesmore important at the end of the training.

- 在实践中，当训练过程开始时，通常需要很强的更新能力，而在训练结束时，小的梯度误差变得更加重要。

- $Q_w(\hat{w}_{std}) = B_w \ll \gg s = sign(\hat{w}_{std}) \ll \gg s$

- $Q_a(a) = B_a = sign(a);$

- $z = (B_w \odot B_a) \ll \gg s$

Forward and Backward Information Retention
for Accurate Binary Neural Networks

Haotong Qin[1], Ruihao Gong[1], Xianglong Liu[1,2*], Mingzhu Shen[1], Ziran Wei[4], Fengwei Yu[3], Jingkuan Song[5]

[1]State Key Lab of Software Development Environment, Beihang University
[2]Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beihang University
[3]SenseTime Research   [4]Beijing University of Posts and Telecommunications
[5]Center for Future Media, University of Electronic Science and Technology of China

{qinhaotong, gongruihao, xlliu}@nlsde.buaa.edu.cn, lavieenrosesmz@outlook.com,
yufengwei@sensetime.com, {weiziran125, jingkuan.song}@gmail.com

Abstract

*Weight and activation binarization is an effective approach to deep neural network compression and can accelerate the inference by leveraging bitwise operations. Although many binarization methods have improved the accuracy of the model by minimizing the quantization error in forward propagation, there remains a noticeable performance gap between the binarized model and the full-precision one. Our empirical study indicates that the quantization brings information loss in both forward and backward propagation, which is the bottleneck of training accurate binary neural networks. To address these issues, we propose an **I**nformation **R**etention **Net**work (IR-Net) to retain the information that consists in the forward activations and backward gradients. IR-Net mainly relies on two technical contributions: (1) Libra Parameter Binarization (Libra-PB): simultaneously minimizing both quantization error and information loss of parameters by balanced and standardized weights in forward propagation; (2) Error Decay Estimator (EDE): minimizing the information loss of gradients by gradually approximating the* sign *function in backward propagation, jointly considering the updating ability and accurate gradients. We are the first to investigate both forward and backward processes of binary networks from the unified information perspective, which provides new insight into the mechanism of network binarization. Comprehensive experiments with various network structures on CIFAR-10 and ImageNet datasets manifest that the proposed IR-Net can consistently outperform state-of-the-art quantization methods.*

1. Introduction

Deep neural networks (DNNs), especially convolutional neural networks (CNNs), have been well demonstrated in a wide variety of computer vision applications such as image classification [30, 46, 47, 49, 50, 56, 63, 53], object detection [17, 16, 42, 45, 34] and semantic segmentation [15, 65]. Traditional CNNs are usually with massive parameters and high computational complexity for the requirement of high accuracy. Consequently, deploying the most advanced deep CNN models requires expensive storage and computing resources, which largely limits the applications of DNNs on portable devices such as mobile phones and cameras. Binary neural networks are appealing to the community for their tiny storage usage and efficient inference [10, 52, 37, 64, 55, 19], which results from the binarization of both weights and activations and the efficient convolution implemented by bitwise operations. Although much progress has been made on binarizing DNNs, the existing quantization methods remains a significant drop of accuracy compared with the full-precision counterparts [14, 39, 2, 28, 57, 48, 9, 40, 43].

The performance degradation of binary neural networks is mainly caused by the limited representation ability and discreteness of binarization, which results in severe information loss in both forward and backward propagation. In the forward propagation, when the activations and weights are restricted to two values, the model's diversity sharply decreases, while the diversity is proved to be the key of pursuing high accuracy of neural networks [54]. Two approaches are widely used to increase the diversity of neural networks: increasing the number of neurons or increasing the diversity of feature maps. For example, Bi-Real Net [38] is targeted at the latter by adding on a full-precision shortcut to the quantized activations, which achieves significant performance improvement. However, with the additional floating-point add operation, Bi-Real Net inevitably faces worse efficiency than vanilla binary neural networks.

The diversity implies the ability to carry enough information during the forward propagation, meanwhile, accurate

gradients in the backward propagation provide correct information for optimization. However, during the training process of binary neural networks, the discrete binarization always leads to inaccurate gradients and the wrong optimization direction. To better deal with the discreteness, different approximations of binarization for backward propagation have been studied [8, 38, 11, 32, 7], mainly categorized into either improving the updating ability or reducing the mismatching area between `sign` function and the approximation one. Unfortunately, the difference between the early and later training stages is always be ignored, where in practice strong updating ability is usually highly required when the training process starts and small gradient error becomes more important at the end of the training. It is insufficient to acquire as much information reflected from the loss function as possible only focusing on one point.

To solve the above-mentioned problems, this paper is the first to study the model binarization from the view of information flow and proposes a novel **I**nformation **R**etention **Net**work (IR-Net) (see the overview in Fig 1). Our goal is to train highly accurate binarized models by retaining the information in the forward and backward propagation: (1) IR-Net introduces a balanced and standardized quantization method called *Libra Parameter Binarization* (Libra-PB) in the forward propagation. With Libra-PB, we can minimize the information loss in forward propagation by maximizing the information entropy of the quantized parameters and minimizing the quantization error, which ensures a high diversity. (2) In backward propagation, IR-Net adopts the *Error Decay Estimator* (EDE) to calculate gradients and minimizes the information loss by better approximating the `sign` function, which ensures sufficient updating at the beginning and accurate gradients at the end of the training.

Our IR-Net presents a new and practical perspective to understand how the binarized network works. Besides the strong capability of preserving the information forward/backward in the deep network, it also enjoys good versatility and can be optimized in a standard network training pipeline. We evaluate our IR-Net with image classification tasks on the CIFAR-10 and ImageNet datasets. The experimental results show that our method performs remarkably well across various network structures such as ResNet-20, VGG-Small, ResNet-18, and ResNet-34, surpassing previous quantization methods by a wide margin. Our code is released at https://github.com/htqin/IR-Net.

2. Related Work

Network binarization aims to accelerate the inference of neural networks and save memory occupancy without much accuracy degradation. One approach to speed up low-precision networks is to utilize bitwise operations. By directly binarizing the 32-bit parameters in DNNs including weights and activations, we can achieve significant acceler-
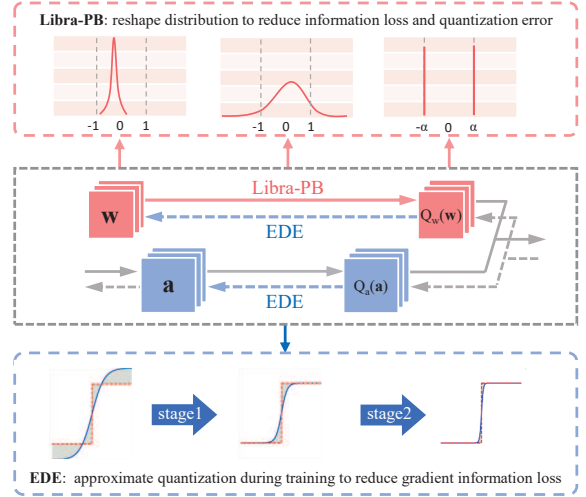

Figure 1: Overview of our IR-Net training for a convolutional layer, consisting of Libra Parameter Binarization (Libra-PB) in forward propagation and Error Decay Estimator (EDE) in backward propagation. Libra-PB changes the weight distribution in forward propagation to retain the information of weights and activations. And the shape change of EDE during the entire training process reduces the gradient information loss in backward propagation.

ations and memory reductions. XNOR-Net [44] utilizes a deterministic binarization scheme and minimizes the quantization error of the output matrix by employing some scalars in each layer. TWN [33] and TTQ [62] enhance the representation ability of neural networks with more available quantization points. ABC-Net [36] recommends using more binary bases for weights and activations to improve accuracy, while compression and acceleration ratios are reduced accordingly. [8] proposed HWGQ considering the quantization error from the aspect of activation function. [59] further proposed LQ-Net with more training parameters, which achieved comparable results on the ImageNet benchmark but increased the memory overhead.

Compared with other model compression methods, *e.g.*, pruning [21, 20, 23] and matrix decomposition [58, 51], network binarization can greatly reduce the memory consumption of the model, and make the model fully compatible with bitwise operations to get good acceleration. Although much progress has been made on network binarization, the existing quantization methods still cause a significant drop of accuracy compared with the full-precision models, since great information loss still exists in the training of binary neural networks. Therefore, to retain the information and ensure a correct information flow during the forward and backward propagation of binarized training, IR-Net is designed.

3. Preliminaries

The main operation in deep neural networks is expressed as:

$$z = \mathbf{w}^\top \mathbf{a}, \qquad (1)$$

where $\mathbf{w} \in \mathbb{R}^n$ indicates the weight vector, $\mathbf{a} \in \mathbb{R}^n$ indicates the input activation vector computed by the previous network layer.

The goal of network binarization is to represent the floating-point weights and/or activations with 1-bit. In general, the quantization can be formulated as:

$$Q_x(\mathbf{x}) = \alpha \mathbf{B_x}, \qquad (2)$$

where $\mathbf{x}$ denotes floating-point parameters including floating-point weights $\mathbf{w}$ and activations $\mathbf{a}$, and $\mathbf{B_x} \in \{-1, +1\}^n$ denotes binary values including binary weights $\mathbf{B_w}$ and activations $\mathbf{B_a}$. α denotes scalars for binary values including α_w for weights and α_a for activations. And we usually use `sign` function to get $\mathbf{B_x}$:

$$\mathbf{B_x} = \text{sign}(\mathbf{x}) = \begin{cases} +1, & if\ \mathbf{x} \geq 0 \\ -1, & otherwise. \end{cases} \qquad (3)$$

With the quantized weights and activations, the vector multiplications in the forward propagation can be reformulated as

$$z = Q_w(\mathbf{w})^\top Q_a(\mathbf{a}) = \alpha_w \alpha_a (\mathbf{B_w} \odot \mathbf{B_a}), \qquad (4)$$

where $\odot$ denotes the inner product for vectors with bitwise operations XNOR and Bitcount.

In the backward propagation, the derivative of the `sign` function is zero almost everywhere, which makes it incompatible with backward propagation, since exact gradients for the original values before the discretization (pre-activations or weights) would be zeroed. So "Straight-Through Estimator (STE) [5]" is generally used to train binary models, which propagates the gradient through `Identity` or `Hardtanh` function.

4. Information Retention Network

In the paper, we point out that the bottleneck of training highly accurate binary neural networks mainly lies in the severe information loss of the training process. Information loss caused by the forward `sign` function and the backward approximation for gradient greatly harms the accuracy of binary neural networks. In this paper, we propose a novel model, **I**nformation **R**etention **Net**work (IR-Net), which retains the information in the training process and acquires highly accurate binarized models.

4.1. Libra Parameter Binarization in Forward Propagation

In the forward propagation, the quantization operation brings information loss. Many quantized convolutional neural networks, including binarized models [44, 35, 59], find the optimal quantizer by minimizing the quantization error :

$$\min J(Q_x(\mathbf{x})) = \|\mathbf{x} - Q_x(\mathbf{x})\|^2, \qquad (5)$$

where $\mathbf{x}$ indicates the full-precision parameters, $Q_x(\mathbf{x})$ denotes the quantized parameters and $J(Q_x(\mathbf{x}))$ denotes the quantization error between full-precision and binary parameters. The objective function (Eq. (5)) assumes that quantized models should completely follow the pattern of full-precision models. However, this is not always true especially when extremely low bit-width is applied. For binary models, the representation ability of their parameters is limited to two values, which makes the information carried by neurons easy to lose. The solution space of binary neural networks also quite differs from that of full-precision neural networks. Therefore, without retaining the information through networks, it is insufficient and difficult to promise a good binarized network only by minimizing the quantization error.

To retain the information and minimize the information loss in forward propagation, we propose Libra Parameter Binarization (Libra-PB) that jointly considers both quantization error and information loss. For a random variable $b \in \{-1, +1\}$ obeying Bernoulli distribution, whose probability mass function is

$$f(b) = \begin{cases} p, & if\ b = +1 \\ 1-p, & if\ b = -1, \end{cases} \qquad (6)$$

where p is the probability of taking the value $+1$, $p \in (-1, 1)$, and each element in $\mathbf{B_x}$ can be viewed as a sample of b. The entropy of $Q_x(\mathbf{x})$ in Eq. (2) can be calculated by:

$$\mathcal{H}(Q_x(\mathbf{x})) = \mathcal{H}(\mathbf{B_x}) = -p\ln(p) - (1-p)\ln(1-p). \quad (7)$$


If we only pursue the goal of minimizing quantization error, the information entropy of the quantized parameters can be close to zero in extreme case. Therefore, Libra-PB combines the quantization error and information entropy of quantized values as objective function, which is defined as

$$\min J(Q_x(\mathbf{x})) - \lambda\mathcal{H}(Q_x(\mathbf{x})). \qquad (8)$$

Under the Bernoulli distribution assumption, when $p = 0.5$, the information entropy of the quantized values takes the maximum value. This means the quantized values should be evenly distributed. Therefore, we balance weights with zero-mean attribute by subtracting the mean of full-precision weights. Moreover, to make the training more stable without negative effect deriving from the weight magnitude and thus the gradient, we further normalize the balanced weight. The standardized balanced weights $\hat{\mathbf{w}}_{\text{std}}$ are obtained through standardization and balance operations as follows:

$$\hat{\mathbf{w}}_{\text{std}} = \frac{\hat{\mathbf{w}}}{\sigma(\hat{\mathbf{w}})}, \quad \hat{\mathbf{w}} = \mathbf{w} - \overline{\mathbf{w}}. \qquad (9)$$

where the $\sigma(\cdot)$ denotes the standard deviation. $\hat{\mathbf{w}}_{\text{std}}$ has two characteristics: (1) $zero-mean$, which maximizes the obtained binary weights' information entropy. (2) $unit-norm$,
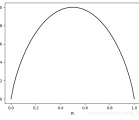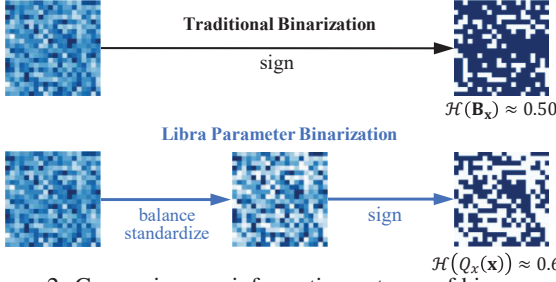
Figure 2: Comparison on information entropy of binary weights quantized with Libra-PB $Q_x(\mathbf{x})$ and `sign` function, respectively. Owing to the balance characteristic brought by Libra-PB, the information entropy of $Q_x(\mathbf{x})$ is larger than `sign(x)`, where $Q_x(\mathbf{x})$ and `sign(x)` have a probability of 0.5 and 0.2 to take value 1 under Bernoulli distribution, respectively.

which makes the full-precision weights involved in binarization more dispersed. Therefore, compared with the direct use of the balanced progress, the use of standardized balanced progress makes the weights steadily updated, and makes the binary weights $Q_w(\hat{\mathbf{w}}_{std})$ more stable during the training.

Since the value of $Q_w(\hat{\mathbf{w}}_{std})$ depends on the sign of $\hat{\mathbf{w}}_{std}$ and the distribution of $\mathbf{w}$ is almost symmetric [24, 4], the balanced operation can maximize the information entropy of quantized $Q_w(\hat{\mathbf{w}}_{std})$ on the whole. And when Libra-PB is used for weights, the information flow of activations in the network can also be maintained. Supposing quantized activations $Q_a(\mathbf{a})$ have mean $\mathbb{E}[Q_a(\mathbf{a})] = \mu\mathbf{1}$, the mean of z can be calculated by:

$$\mathbb{E}[z] = Q_w(\hat{\mathbf{w}}_{std})^\top \mathbb{E}[Q_a(\mathbf{a})] = Q_w(\hat{\mathbf{w}}_{std})^\top \mu\mathbf{1}. \quad (10)$$

Because of using Libra-PB for weights in each layer, we have $Q_w(\hat{\mathbf{w}}_{std})^\top \mathbf{1} = 0$, the mean of output is zero. Therefore, the information entropy of activations in each layer can be maximized, which means that the information in activations can be retained.

To further minimize the quantization error and avoid extra expensive floating-point calculation in previous binarization methods, Libra-PB introduces an integer bit-shift scalar s to expand the representation ability of binary weights. The optimal bit-shift scalar can be solved by:

$$\mathbf{B}_\mathbf{w}^*, s^* = \underset{\mathbf{B}_\mathbf{w}, s}{\arg\min} \|\hat{\mathbf{w}}_{std} - \mathbf{B}_\mathbf{w} \lll s\|^2 \quad s.t. \ s \in \mathbb{N} \quad (11)$$

where $\lll$ stands for left or right bit-shift. $\mathbf{B}_\mathbf{w}^*$ is calculated by $\mathbf{B}_\mathbf{w}^* = \text{sign}(\hat{\mathbf{w}}_{std})$, thus s^* can be solved as:

$$s^* = \text{round}(\log_2(\|\hat{\mathbf{w}}_{std}\|_1/n)). \quad (12)$$

where n and $\|\hat{\mathbf{w}}_{std}\|_1$ denote the dimension and L1-norm of the vector, respectively.

Therefore, our Libra Parameter Binarization for the forward propagation can be presented as below:

$$Q_w(\hat{\mathbf{w}}_{std}) = \mathbf{B}_\mathbf{w}\lll s = \text{sign}(\hat{\mathbf{w}}_{std})\lll s, \\ Q_a(\mathbf{a}) = \mathbf{B}_\mathbf{a} = \text{sign}(\mathbf{a}). \quad (13)$$

The main operations in IR-Net can be expressed as:

$$z = (\mathbf{B}_\mathbf{w} \odot \mathbf{B}_\mathbf{a}) \lll s. \quad (14)$$

As shown in Fig 2, the parameters quantized by Libra-PB have the maximum information entropy under the Bernoulli distribution. We call our binarization method "Libra Parameter Binarization" because the parameters are balanced before the binarization to retain information.

Note that Libra-PB serves an implicit rectifier that reshapes the data distribution before binarization. In the literature, a few studies also realized this positive effect on the performance of BNNs and adopted empirical settings to redistribute parameters [44, 13]. For example, [13] proposed the specific degeneration problem of binarization and solved it using a specially designed additional regularization loss. Different from these works, we first straightforwardly present the information view to rethink the impact of parameter distribution before binarization, and promise the optimal solution by maximizing the information entropy. Moreover, in this framework, Libra-PB can accomplish the distribution adjustment by simply balancing and standardizing the weights before the binarization. This means that our method can be easily and widely applied to various neural network architectures and be directly plugged into the standard training pipeline with a very limited extra computation cost.

4.2. Error Decay Estimator in Backward Propagation

Limited by the discontinuity of binarization, approximation of gradients is inevitable for the backward propagation. Thus, huge information loss is caused because the influence of quantization cannot be exactly modeled with the approximation. The approximation can be formulated as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial Q_w(\hat{\mathbf{w}}_{std})} \frac{\partial Q_w(\hat{\mathbf{w}}_{std})}{\partial \mathbf{w}} \approx \frac{\partial \mathcal{L}}{\partial Q_w(\hat{\mathbf{w}}_{std})} g'(\mathbf{w}), \quad (15)$$

where $L(\mathbf{w})$ represents the loss function, $g(\mathbf{w})$ denotes the approximation of the `sign` function and $g'(\mathbf{w})$ is the derivative of $g(\mathbf{w})$. There are two common practices for approximation used in previous works:

$$\text{Identity}: y = x \quad \text{or} \quad \text{Clip}: y = \text{Hardtanh}(x). \quad (16)$$

The `Identity` function straightly passes the gradient information of output values to input values and completely neglects the effect of binarization. As shown in the shaded area of Fig 3(a), the gradient error is huge and will accumulate during the backward propagation. It is highly required to retain correct gradient information to avoid unstable training rather than ignore the noise caused by `Identity` for utilizing the Stochastic Gradient Descent Algorithm.

The `Clip` function takes the clipping attribute of binarization into account to reduce the gradient error. But it can only
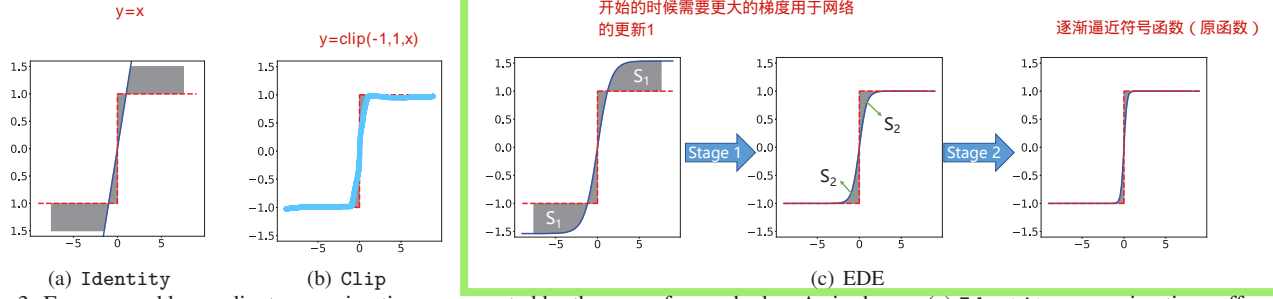
Figure 3: Error caused by gradient approximation, represented by the area of gray shades. As is shown, (a) `Identity` approximation suffers huge error. (b) `Clip` approximation does not update the values outside the clipping interval. (c) EDE maintains the updating ability at early stage and progressively reduces the error. S_1 shrinks during Stage 1 by decreasing the clipping value and S_2 shrinks during Stage 2 by increasing the derivate.

pass the gradient information inside the clipping interval. It can be seen in Fig 3(b) that for parameters outside [-1, +1], the gradient is clamped to 0. This means once the value jumps outside the clamping interval, it cannot be updated anymore. This characteristic greatly harms the updating ability of backward propagation, which can be proved by the fact that ReLU is a superior activation function compared with Tanh. Thus the `Clip` approximation increases the difficulty for optimization and decreases the accuracy in practice. It is crucial to ensure enough updating possibility, especially during the beginning of the training process.

`Identity` function loses the gradient information of quantization while `Clip` function loses the gradient information outside the clipping interval. There is a contradiction between these two kinds of gradient information loss. To make a balance and acquire the optimal approximation of backward gradient, we design Error Decay Estimator:

$$g(x) = k \tanh tx \qquad (17)$$

where $g(x)$ is the backward approximation substitute for the forward `sign` function, k and t are control variables varying during the training process:

$$t = T_{\min} 10^{\frac{i}{N} \times \log \frac{T_{\max}}{T_{\min}}}, \quad k = \max(\frac{1}{t}, 1) \qquad (18)$$

where i is the current epoch and N is the number of epochs, $T_{\min} = 10^{-1}$ and $T_{\max} = 10^{1}$.

To retain the information deriving from loss function in the backward propagation, EDE introduces a progressive two-stage approach to approximate gradients.

Stage 1: Retain the updating ability of the backward propagation algorithm. We keep the gradient estimation function's derivative value close to one, and then progressively reduce the clipping value from a large number to one. With this rule, our estimation function evolves from `Identity` to `Clip` approximation, which ensures the update ability at the early stage of training.

Stage 2: Retain the accurate gradients for parameters around zero. We keep the clipping value as one and gradually push the derivative curse to the shape of the staircase function. With this rule, our estimation function evolves from `Clip` approximation to `sign` function, which ensures the consistency of forward and backward propagation.

The shape change of EDE for each stage is shown in Fig 3(c). Our EDE updates all parameters in the first stage, and further makes the parameters more accurate in the second stage. Based on the two-stage estimation, EDE reduces the gap between the forward binarization function and the backward approximation function and meanwhile all parameters can be reasonably updated.

Algorithm 1 Forward and backward propagation for BNN training by the proposed IR-Net.

1: **Require**: the input data $\mathbf{a} \in \mathbb{R}^n$, pre-activation $z \in \mathbb{R}$, full-precision weights $\mathbf{w} \in \mathbb{R}^n$.
2: **Forward propagation**
3:    Compute binary weight by Libra-PB [Eq. (13)]:
      $\hat{\mathbf{w}}_{\text{std}} = \frac{\mathbf{w} - \bar{\mathbf{w}}}{\sigma(\mathbf{w} - \bar{\mathbf{w}})}, \quad s^* = \texttt{round}(\log_2 \frac{\|\hat{\mathbf{w}}_{\text{std}}\|_1}{n})$
      $Q_w(\hat{\mathbf{w}}_{\text{std}}) = \mathbf{B_w} \lll s = \texttt{sign}(\hat{\mathbf{w}}_{\text{std}}) \lll s$
4:    Compute balanced binary input data [Eq. (13)]:
      $Q_a(\mathbf{a}) = \mathbf{B_a} = \texttt{sign}(\mathbf{a});$
5:    Calculate the output: $z = (\mathbf{B_w} \odot \mathbf{B_a}) \lll s$
6: **Back propagation**
7:    Update the $g'(\cdot)$ via EDE:
      Get current t and k by Eq. (18)
      Update the $g'(\cdot)$: $g'(x) = kt(1 - \tanh^2(tx))$
8:    Calculate the gradients w.r.t. $\mathbf{a}$:
      $\frac{\partial \mathcal{L}}{\partial \mathbf{a}} = \frac{\partial \mathcal{L}}{\partial Q_a(\mathbf{a})} g'(\mathbf{a})$
9:    Calculate the gradients w.r.t. $\mathbf{w}$:
      $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial Q_w(\hat{\mathbf{w}}_{\text{std}})} g'(\hat{\mathbf{w}}_{\text{std}}) 2^s$
10: **Parameters Update**
11:    Update $\mathbf{w}$: $\mathbf{w} = \mathbf{w} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$, where η is learning rate.

4.3. Analysis and Discussions

The training process of our IR-Net is summarized in Algorithm 1. In this section, we will analyze IR-Net from different aspects.

4.3.1 Complexity Analysis

Since Libra-PB is applied on weights, there is extra operation for binarizing activations in IR-Net. And in Libra-PB, with the novel bit-shift scales, the computation costs are reduced compared with the existing solutions with floating-point scalars (*e.g.*, XNOR-Net, and LQ-Net), as shown in

Table 1: The additional floating-point operations consumed by different binarization methods.

Method	Float Operations	Bitwise Operations
XNOR-Net	C_1	$C_1 \times C_2$
LQ-Net	C_1	$C_1 \times C_2$
Ours	0	$C_1 \times C_2 + C_1$

* $C_1 = w_{\text{out}} \times h_{\text{out}} \times c_{\text{out}}$ and $C_2 = w_{\text{k}} \times h_{\text{k}} \times c_{\text{in}}$, where $c_{\text{out}}, c_{\text{in}}, w_{\text{k}}$, $h_{\text{k}}, w_{\text{out}}, h_{\text{out}}$ denote the number of output channels, input channels, kernel width, kernel height, output width, and output height, respectively. The Bitwise operation mainly consists of XNOR, Bitcount and Bit-shift.

Table 1. Later, we further test the real speed of deployment on hardware and the results can be seen in the Deployment Efficiency Section.

4.3.2 Stabilize Training

In Libra Parameter Binarization, weight standardization is introduced for reducing the gap between full-precision weights and the binarized ones, avoiding the noise caused by binarization. Fig 4 shows the data distribution of weights without standardization, obviously more concentrated around 0. This phenomenon means the signs of most weights are easy to change during the process of optimization, which directly causes unstable training of binary neural networks. By redistributing the data, weight standardization implicitly sets up a bridge between the forward Libra-PB and backward EDE, contributing to a more stable training of binary neural networks.

5. Experiments

In this section, we conduct experiments on two benchmark datasets: CIFAR-10 [29] and ImageNet (ILSVRC12) [12] to verify the effectiveness of the proposed IR-Net and compare it with other state-of-the-art (SOTA) methods.

IR-Net: We implement our IR-Net using PyTorch because of its high flexibility and powerful automatic differentiation mechanism. When constructing a binarized model, we simply replace the convolutional layers in the origin models with the binary convolutional layer binarized by our method.

Network Structures: We employ the widely-used network structures including VGG-Small [59], ResNet-18, ResNet-20 for CIFAR-10, and ResNet-18, ResNet-34 [22] for ImageNet. To prove the versatility of our IR-Net, we evaluate it on both the normal structure and the Bi-Real [38] structure of ResNet. All convolutional and fully-connected layers except the first and last one are binarized, and we select Hardtanh as our activation function instead of ReLU when we binarize the activation.

Initialization: Our IR-Net is trained from scratch (random initialization) without leveraging any pre-trained model. To evaluate our IR-Net on various network architectures, we
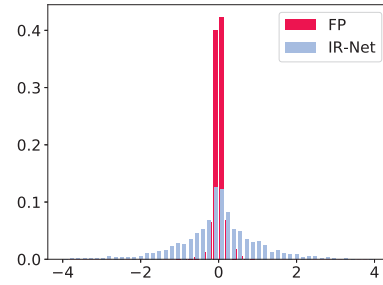

Figure 4: Full-precision weights (in red) in neural networks have a small data range and always gather around 0, and thus their signs are highly possible to flip in backward propagation. IR-Net balances and standardizes the weights (in blue) before the binarization for stabilizing training.

mostly follow the hyper-parameter settings of their original papers [44, 59, 38]. Among the experiments, we apply SGD as our optimization algorithm.

5.1. Ablation Study

In this part, we investigate the behaviors and effects of the proposed Libra-PB and EDE techniques on BNN performance.

5.1.1 Effect of Libra-PB

Our Libra-PB can maximize the information entropy of binary weights and binary activations in IR-Net by adjusting the distribution of weights in the network. Since an explicit balance operation is used before the binarization, the binary weights of each layer in the network have maximum information entropy. Binary activations affected by binary weights in IR-Nets also have maximum information entropy. To demonstrate the information retention of Libra-PB in IR-Net, in Fig 5, we show the information loss of each layer's binary activations in the networks quantized by vanilla binarization and Libra-PB. Vanilla binarization suffers a large reduction in the information entropy of the binary activations. In the network quantized by Libra-PB, the activation of each layer is close to the maximum information entropy under the Bernoulli distribution. And in the forward propagation, the information loss of binary activations caused by vanilla binarization accumulates layer by layer. Fortunately, the results in Fig 5 show that Libra-PB can retain the information in the binary activation of each layer.

5.1.2 Effect of EDE

To demonstrate the necessity and effect of our well-designed EDE, we show the data distribution of weights in different stages of training, as shown in Fig 6. The figures in the first row show the distribution and the figures in the second row show the corresponding derivative curve. Among the derivative curves, the blue one represents EDE and the
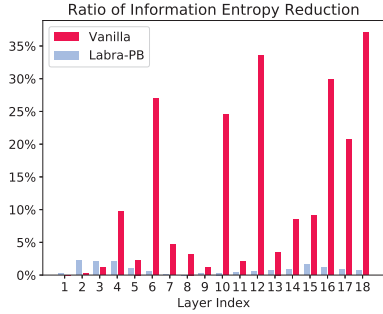
Figure 5: Libra-PB's effect on information entropy of activations of each layer in ResNet-20. The ratio of information entropy reduction (compared with the maximum information entropy under the Bernoulli distribution) quantized by Libra-PB is significantly larger than vanilla binarization.
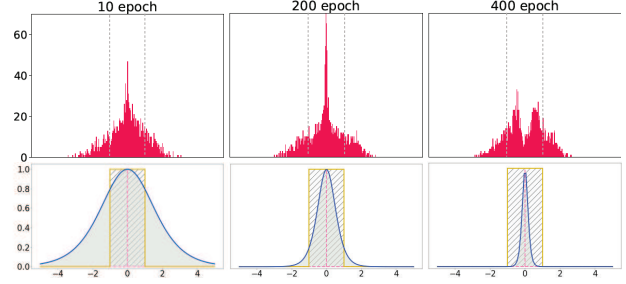

Figure 6: EDE takes full account of the impact of weight distributions (after standardization) in different epochs (10, 200 and 400) during training. The weight distributions are shown in the upper part. In the lower part, the blue line is the derivative of EDE, the yellow and the pink lines are the derivative of STE and `sign` function, respectively. And the shade indicates the error between the derivative of EDE/STE and that of the `sign` function. In the first stage of EDE, all weights can be effectively updated. In the second stage, the weights around 0 are more accurately updated due to the decrease of gradient error.

yellow one represents the common STE (with clipping). It can be seen that during the first stage of EDE (epoch 10 to epoch 200 in Fig 6), there is much data outside the range [-1, +1], thus there should not be much clipping which will be harmful to update ability. Besides, the peakedness of weight distribution is high and much data gather around zero at the beginning of training. EDE keeps the derivative similar to `Identity` function at this stage to ensure the derivative around zero not too large, and thus avoids severely unstable training. Fortunately, with the binarization introduced into training, the weights will gradually approach -1/+1 in the later stages of training. Thus we can slowly increase the value of derivative and approximate a standard `sign` function to reduce the gradient mismatch. The visualized results prove that our EDE approximation for backward propagation agrees with the real data distribution, which is the key to improving accuracy.

5.1.3 Ablation Performance

We further investigate the performance using different parts of IR-Net with the ResNet-20 model on CIFAR-10, which helps understand how our IR-Net works in practice. Table 2 shows the performance in different settings. From the table, we can see that using Libra-PB or EDE alone can improve the accuracy, and the weight standardization in Libra-PB also plays an important role. Moreover, the improvements brought by these parts together can be superimposed, that is why our method can train highly accurate binarized models.

5.2. Comparison with SOTA methods

We further comprehensively evaluate IR-Net by comparing it with the existing SOTA methods.

5.2.1 CIFAR-10

Table 3 lists the performance using different methods on CIFAR-10, including RAD [13] over ResNet-18 (based

Table 2: Ablation study for IR-Net.

Method	Bit-width (W/A)	Acc.(%)
FP	32/32	90.8
Binary	1/1	83.8
Libra-PB (without weight standardization)	1/1	84.3
Libra-PB (without bit-shift scales)	1/1	84.6
Libra-PB	1/1	84.9
EDE	1/1	85.2
IR-Net (Libra-PB & EDE)	1/1	**86.5**

on [31]), DoReFa-Net [61], LQ-Net [59], DSQ [18] over ResNet-20 (based on [3]) and BNN [27], LAB [25], RAD, XNOR-Net [44] over VGG-Small. In all cases, our IR-Net obtains the best performance. More importantly, our method gets a significant improvement over the SOTA methods when using 1-bit weights and 1-bit activations (1W/1A), whether we use the original ResNet structure or the Bi-Real structure. For example, in the 1W/1A bit-width setting, compared with SOTA on ResNet-20, the absolute accuracy increase is as high as 2.4%, and the gap to its full-precision (FP) counterpart is reduced to 4.3%.

5.2.2 ImageNet

For the large-scale ImageNet dataset, we study the performance of IR-Net over ResNet-18 and ResNet-34. Table 4 shows a number of SOTA quantization methods over ResNet-18 and ResNet-34, including BWN [44], HWGQ [35], TWN [33], LQ-Net [59], DoReFa-Net [61], ABC-Net [36], Bi-Real [38], XNOR++ [6], BWHN [26], SQ-BWN and SQ-TWN [1]. We can observe that when only quantizing weights over ResNet-18, IR-Net using 1-bit outperforms most other methods by large margins, and even surpasses TWN using 2-bit weights. And in the 1W/1A setting, the

bit

Table 3: Accuracy comparison with SOTA methods on CIFAR-10.

Topology	Method	Bit-width (W/A)	Acc.(%)
ResNet-18	FP	32/32	93.0
	RAD	1/1	90.5
	Ours[1]	1/1	**91.5**
ResNet-20	FP	32/32	91.7
	DoReFa	1/1	79.3
	DSQ	1/1	84.1
	Ours[1]	1/1	**85.4**
	Ours[2]	1/1	**86.5**
	FP	32/32	91.7
	DoReFa	1/32	90.0
	LQ-Net	1/32	90.1
	DSQ	1/32	90.2
	Ours[1]	1/32	**90.8**
VGG-Small	FP	32/32	91.7
	LAB	1/1	87.7
	XNOR	1/1	89.8
	BNN	1/1	89.9
	RAD	1/1	90.0
	Ours	1/1	**90.4**

Table 4: Accuracy comparison with SOTA methods on ImageNet.

Topology	Method	Bit-width (W/A)	Top-1(%)	Top-5(%)
ResNet-18	FP	32/32	69.6	89.2
	ABC-Net	1/1	42.7	67.6
	XNOR	1/1	51.2	73.2
	BNN+	1/1	53.0	72.6
	DoReFa	1/2	53.4	–
	Bi-Real	1/1	56.4	79.5
	XNOR++	1/1	57.1	79.9
	Ours[2]	1/1	**58.1**	**80.0**
	FP	32/32	69.6	89.2
	SQ-BWN	1/32	58.4	81.6
	BWN	1/32	60.8	83.0
	HWGQ	1/32	61.3	83.2
	TWN	2/32	61.8	84.2
	SQ-TWN	2/32	63.8	85.7
	BWHN	1/32	64.3	85.9
	Ours[1]	1/32	**66.5**	**86.8**
ResNet-34	FP	32/32	73.3	91.3
	ABC-Net	1/1	52.4	76.5
	Bi-Real	1/1	62.2	83.9
	Ours[2]	1/1	**62.9**	**84.1**
	FP	32/32	73.3	91.3
	Ours[1]	1/32	**70.4**	**89.5**

Top-1 accuracy of IR-Net is also significantly better than that of the SOTA methods (*e.g.*, 58.1% vs. 56.4% for ResNet-18). The experimental results prove that our IR-Net is more competitive than the existed methods.

5.3. Deployment Efficiency

To further validate the efficiency of IR-Net when deployed into the real-world mobile devices, we further implement our IR-Net on Raspberry Pi 3B, which has a 1.2 GHz 64-bit quad-core ARM Cortex-A53, and test its real speed in practice. We utilize the SIMD instruction SSHL on ARM NEON to make inference framework daBNN [60] compatible with our IR-Net. We have to point out that till now there have been very few studies that reported their inference speed in real-world devices, especially when using 1-bit binarization. In Table 5, we compare our IR-Net with the existing high-performance inference implementation including NCNN [41] and DSQ [18]. From the table, we can easily find that the inference speed of IR-Net is much faster, the model size of IR-Net can be greatly reduced and the bit-shift scales in IR-Net bring almost no extra inference time and storage consumption.

Table 5: Comparison of time cost of ResNet-18 with different bits (single thread).

Method	Bit-width (W/A)	Size (Mb)	Time (ms)
FP	32/32	46.77	1418.94
NCNN	8/8	–	935.51
DSQ	2/2	–	551.22
Ours (without bit-shift scales)	1/1	**4.20**	**252.16**
Ours	1/1	**4.21**	**261.98**

6. Conclusion

In this paper, we propose IR-Net to retain the information propagated in binary neural networks, mainly consisting of two novel techniques: Libra-PB for keeping diversity in forward propagation and EDE for reducing the gradient error in backward propagation. Libra-PB conducts a simple yet effective transformation on weights from the view of information entropy, which simultaneously reduces the information loss of both weights and activations, without additional operation on activations. Thus, the diversity of binary neural networks can be kept as much as possible and meanwhile the efficiency will not be harmed. Besides, the well-designed gradient estimator EDE retains the gradient information during backward propagation. Owing to the sufficient updating ability and accurate gradients, the performance with EDE surpasses that with STE by a large margin. Extensive experiments prove that the IR-Net consistently outperforms the existed state-of-the-art binary neural networks.

---

[1]Results of ResNet with normal structure [22].
[2]Results of ResNet with Bi-Real structure [38].

References

[1] Learning accurate low-bit deep neural networks with stochastic quantization. *BMVC*, 2017. 7

[2] T. Ajanthan, P. K. Dokania, R. Hartley, and P. H. S. Torr. Proximal mean-field for neural network quantization. In *IEEE ICCV*, 2019. 1

[3] akamaster. pytorch_resnet_cifar10. https://github.com/akamaster/pytorch_resnet_cifar10. 7

[4] R. Banner, Y. Nahshan, E. Hoffer, and D. Soudry. Post training 4-bit quantization of convolution networks for rapid-deployment. *CoRR, abs/1810.05723*, 2018. 4

[5] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3

[6] A. Bulat and G. Tzimiropoulos. Xnor-net++: Improved binary neural networks. *CoRR*, abs/1909.13863, 2019. 7

[7] A. Bulat, G. Tzimiropoulos, J. Kossaifi, and M. Pantic. Improved training of binary networks for human pose estimation and image recognition. *CoRR*, abs/1904.05868, 2019. 2

[8] Z. Cai, X. He, J. Sun, and N. Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *IEEE CVPR*, 2017. 2

[9] S. Cao, L. Ma, W. Xiao, C. Zhang, Y. Liu, L. Zhang, L. Nie, and Z. Yang. Seernet: Predicting convolutional neural network feature-map sparsity through low-bit quantization. In *IEEE CVPR*, 2019. 1

[10] M. Courbariaux, Y. Bengio, and J.-P. David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NeurIPS*. 2015. 1

[11] S. Darabi, M. Belbahri, M. Courbariaux, and V. P. Nia. BNN+: improved binary network training. *CoRR*, abs/1812.11800, 2018. 2

[12] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li. Imagenet: a large-scale hierarchical image database. In *IEEE CVPR*, 2009. 6

[13] R. Ding, T.-W. Chin, Z. Liu, and D. Marculescu. Regularizing activation distribution for training binarized deep networks. In *IEEE CVPR*, June 2019. 4, 7

[14] Z. Dong, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *IEEE ICCV*, 2019. 1

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge. *IJCV*. 1

[16] R. Girshick. Fast r-cnn. In *IEEE ICCV*, 2015. 1

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE CVPR*, 2014. 1

[18] R. Gong, X. Liu, S. Jiang, T. Li, P. Hu, J. Lin, F. Yu, and J. Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *IEEE ICCV*, 2019. 7, 8

[19] J. Gu, J. Zhao, X. Jiang, B. Zhang, J. Liu, G. Guo, and R. Ji. Bayesian optimized 1-bit cnns. In *IEEE ICCV*, 2019. 1

[20] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *ICLR*, 2016. 2

[21] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks. *NeurIPS*, 2015. 2

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, 2016. 6, 8

[23] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *IEEE ICCV*, Oct 2017. 2

[24] Z. He and D. Fan. Simultaneously optimizing weight and quantizer of ternary neural network using truncated gaussian approximation. In *IEEE CVPR*, 2019. 4

[25] L. Hou, Q. Yao, and J. T. Kwok. Loss-aware binarization of deep networks. *ICLR*, 2017. 7

[26] Q. Hu, P. Wang, and J. ChengT. From hashing to cnns: Training binary weight networks via hashing. In *AAAI*, 2018. 7

[27] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks. In *NeurIPS*. 2016. 7

[28] S. Jung, C. Son, S. Lee, J. Son, J.-J. Han, Y. Kwak, S. J. Hwang, and C. Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *IEEE CVPR*, 2019. 1

[29] A. Krizhevsky, V. Nair, and G. Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014. 6

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*. 2012. 1

[31] kuangliu, ypwhs, fducau, and bearpaw. pytorch-cifar. https://github.com/kuangliu/pytorch-cifar. 7

[32] F. Lahoud, R. Achanta, P. Márquez-Neila, and S. Süsstrunk. Self-binarizing networks. *CoRR*, abs/1902.00730, 2019. 2

[33] F. Li, B. Zhang, and B. Liu. Ternary weight networks. *CoRR*, abs/1605.04711, 2016. 2, 7

[34] R. Li, Y. Wang, F. Liang, H. Qin, J. Yan, and R. Fan. Fully quantized network for object detection. In *IEEE CVPR*, 2019. 1

[35] Z. Li, B. Ni, W. Zhang, X. Yang, and W. Gao. Performance guaranteed network acceleration via high-order residual quantization. In *IEEE ICCV*, 2017. 3, 7

[36] X. Lin, C. Zhao, and W. Pan. Towards accurate binary convolutional neural network. In *NeurIPS*. 2017. 2, 7

[37] C. Liu, W. Ding, X. Xia, B. Zhang, J. Gu, J. Liu, R. Ji, and D. S. Doermann. Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnns with circulant back propagation. In *IEEE CVPR*, 2019. 1

[38] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*, 2018. 1, 2, 6, 7, 8

[39] S. Morozov and A. Babenko. Unsupervised neural quantization for compressed-domain similarity search. In *IEEE ICCV*, 2019. 1

[40] M. Nagel, M. v. Baalen, T. Blankevoort, and M. Welling. Data-free quantization through weight equalization and bias correction. In *IEEE ICCV*, 2019. 1

[41] nihui, BUG1989, Howave, gemfield, Corea, and eric612. ncnn. https://github.com/Tencent/ncnn. 8

[42] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin. Libra r-cnn: Towards balanced learning for object detection. In *IEEE CVPR*, 2019. 1

[43] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe. Binary neural networks: A survey. *Pattern Recognition*, 2020. 1

[44] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, 2016. 2, 3, 4, 6, 7

[45] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*. 2015. 1

[46] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE CVPR*, 2015. 1

[48] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han. Haq: Hardware-aware automated quantization with mixed precision. In *IEEE CVPR*, 2019. 1

[49] Y. Wang, W. Gan, W. Wu, and J. Yan. Dynamic curriculum learning for imbalanced data classification. In *IEEE ICCV*, 2019. 1

[50] Y. Wang, W. Gan, J. Yang, W. Wu, and J. Yan. Dynamic curriculum learning for imbalanced data classification. In *ICCV*, October 2019. 1

[51] Y. Wang, C. Xu, C. Xu, and D. Tao. Packing convolutional neural networks in the frequency domain. *IEEE TPAMI*, 2019. 2

[52] Z. Wang, J. Lu, C. Tao, J. Zhou, and Q. Tian. Learning channel-wise interactions for binary convolutional neural networks. In *IEEE CVPR*, 2019. 1

[53] Y. Wu, Y. Wu, R. Gong, Y. Lv, K. Chen, D. Liang, X. Hu, X. Liu, and J. Yan. Rotation consistent margin loss for efficient low-bit face recognition. In *CVPR*, June 2020. 1

[54] B. Xie, Y. Liang, and L. Song. Diverse neural network learns true target functions. In *Artificial Intelligence and Statistics*, 2017. 1

[55] Y. Xu, X. Dong, Y. Li, and H. Su. A main/subsidiary network framework for simplifying binary neural networks. In *IEEE CVPR*, June. 1

[56] J. Yang, J. Fan, Y. Wang, Y. Wang, W. Gan, L. Liu, and W. Wu. Hierarchical feature embedding for attribute recognition. In *CVPR*, June 2020. 1

[57] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, and X.-s. Hua. Quantization networks. In *IEEE CVPR*, 2019. 1

[58] X. Yu, T. Liu, X. Wang, and D. Tao. On compressing deep models by low rank and sparse decomposition. In *IEEE CVPR*, 2017. 2

[59] D. Zhang, J. Yang, D. Ye, and G. Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *ECCV*, 2018. 2, 3, 6, 7

[60] J. Zhang, Y. Pan, T. Yao, H. Zhao, and T. Mei. dabnn: A super fast inference framework for binary neural networks on ARM devices. In *ACM MM*, 2019. 8

[61] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *CoRR*, abs/1606.06160, 2016. 7

[62] C. Zhu, S. Han, H. Mao, and W. J. Dally. Trained ternary quantization. In *ICLR*, 2017. 2

[63] F. Zhu, R. Gong, F. Yu, X. Liu, Y. Wang, Z. Li, X. Yang, and J. Yan. Towards unified int8 training for convolutional neural network. In *CVPR*, June 2020. 1

[64] S. Zhu, X. Dong, and H. Su. Binary ensemble neural network: More bits per network or more networks per bit? In *IEEE CVPR*, 2019. 1

[65] B. Zhuang, C. Shen, M. Tan, L. Liu, and I. Reid. Structured binary neural networks for accurate image classification and semantic segmentation. In *IEEE CVPR*, 2019. 1