TBNN: totally-binary neural network for image classification*

ZHANG Qingsong[1,2], SUN Linjun[2], YANG Guowei[1], LU Baoli[2], NING Xin[2], and LI Weijun[2]**

1. School of Electronic Information, Qingdao University, Qingdao 266071, China

2. Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China

(Received 27 June 2022; Revised 12 October 2022)

Most binary networks apply full precision convolution at the first layer. Changing the first layer to the binary convolution will result in a significant loss of accuracy. In this paper, we propose a new approach to solve this problem by widening the data channel to reduce the information loss of the first convolutional input through the sign function. In addition, widening the channel increases the computation of the first convolution layer, and the problem is solved by using group convolution. The experimental results show that the accuracy of applying this paper's method to state-of-the-art (SOTA) binarization method is significantly improved, proving that this paper's method is effective and feasible.

Document code: A **Article ID:** 1673-1905(2023)02-0117-6

DOI https://doi.org/10.1007/s11801-023-2113-2

Model compression[1,2] is a popular area of artificial intelligence (AI) research in recent years. Effective model compression methods can significantly reduce the memory consumption of the network. In general, model compression can be broadly classified into four categories, model pruning, low-rank decomposition, model quantization, and knowledge distillation. Among the model compression methods, model quantization[3] is the most effective. In model quantization, binary quantization maximizes the reduction in model size. Binary quantization will quantify all the model activations and weights as +1 and −1. Compared to deep neural networks with full precision, binary networks have a small memory size and fast inference. It enjoys a 32× memory compression ratio and up to 58× practical computational reduction on the central processing unit (CPU). This makes it possible for binary networks to operate efficiently on embedded devices. However, one of the inevitable problems with binary networks is the severe drop in accuracy.

The main reason for the degradation in performance of binarized neural networks is the limited amount of information that can be expressed in binarization, which causes binarized neural networks to lose a large amount of information during forwarding and backward propagation. To solve this problem, many current methods start with weights and activation functions to reduce information loss. For example, the circulant binary convolutional network (CBCN)[4] proposes a circular convolution operation that increases the capacity of binarized convolutional features and a Gaussian function as an approximation to the gradient of the sign function. CBCN reduces information loss by preserving more information from a weighting perspective. ReActNet[5] transforms the sign function and the parametric rectified linear unit (PReLU) function adds learnable parameter variables and allows the model to automatically learn the best offset and scaling values for each layer. Most binary networks use full precision weights and activation for the first layer of the network. If the binarization is done at the first layer, it will cause a serious loss of information and force a more serious loss of accuracy. When the first layer of convolutional inputs and weights are quantized to +1 and −1, the expressiveness of the model decreases significantly, leading to a decrease in accuracy. Experimental evidence shows that the accuracy of binary networks such as DoReFa-Net[6] and regularizing activation distribution (RAD)[7] decreases significantly after quantizing the first layer of convolution. The main reason for the large loss of information during the binarization of the first convolutional layer is the binarization of the activation. To solve the information loss caused by the first layer of convolutional binarization activation, this paper proposes to process the data before they are fed into the binary network. By turning the input data into binary and changing 0 to −1, each binary bit occupies one channel, turning the original 3 channels of data into 24 channels of data. While preserving as much information as possible about the data, it also reduces the loss of information caused by the activation value in the sign function through the first layer. In this paper, the method

* This work has been supported by the National Natural Science Foundation of China (No.62172229).

** E-mail: ygw_ustb@163.com

is evaluated using an image classification task on the CIFAR10 dataset and the SVHN dataset. The experimental results show that the method is effective.

However, while widening the number of convolution channels in the first layer, the problem of increased computational effort (floating-point operations per second, FLOPs) arises. There are a number of lightweight networks that offer ways to reduce computational effort, and MobileNet[8] proposes deeply separable convolutions that are $1/K^2$ the computational effort of standard convolutions (K is the convolutional kernel size). ShuffleNet[9] uses group convolution and channel shuffle to greatly reduce the computational effort of the model while maintaining accuracy. MobilVit[10] combines the strengths of convolutional neural networks (CNNs) and vision transformers (ViTs) to build a lightweight and low latency network for mobile vision. In this paper, we widen the original red-green-blue (RGB) channels by 8 times each to get 24 channels. To reduce the computation, we need to perform group convolution, and each group should be a set of data containing RGB channels. The ideas in ShuffleNet are therefore a natural fit for this paper. So we adopt the idea of ShuffleNet and use group convolution to solve the problem of increasing computational effort (FLOPs). This solves the problem of increased computation caused by the first layer of channel expansion. In addition, one of the problems with group convolution is that communication between different groups is required, otherwise, it will reduce the feature extraction capability of the network. Therefore, channel shuffle is used to solve this problem, where the input data is divided into 8 groups and each group is taken one channel per 8 channels using channel shuffle. This method allows the first layer to reduce the increase in computational effort due to channel expansion without compromising accuracy.

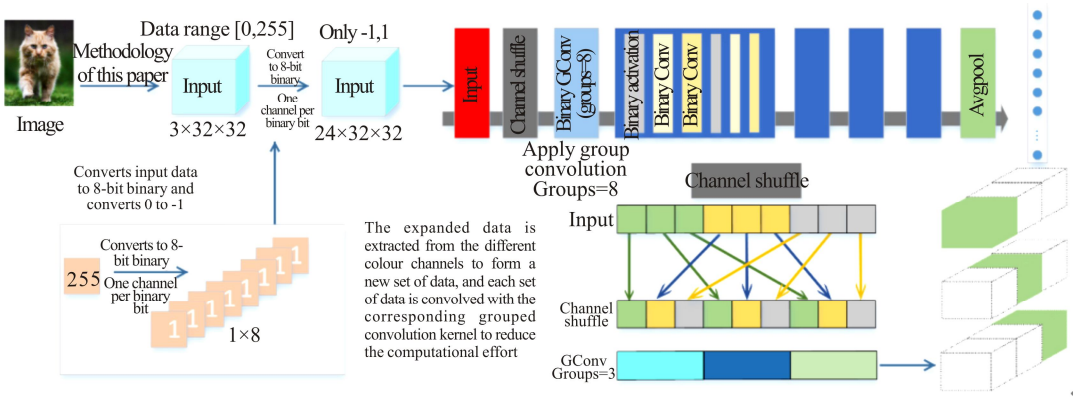

Fig.1 Overall framework of the methodological thought in this paper

The aim of neural network binarization is to speed up the inference of a neural network without reducing accuracy and to reduce the memory footprint. Binary quantization of 32-bit activations and weights in deep neural networks can save a lot of memory and significantly speed up inference. However, many current binary networks are not fully binarized, as binarizing the first layer of convolution results in a huge loss of information, and most binary networks do not process the inputs much, essentially focusing their efforts entirely on the processing of weights and back-propagation approximations. For example, information retention (IR)-Net[11] balances and normalizes weights and minimizes quantization error and loss of parameter information in forwarding propagation, using an error decay estimator (EDE) to approximate the sign function and reduce the loss of gradient information. Bi-Real-Net[12] proposes to use a custom segmentation function for back propagation instead of the sign function since the derivative of the sign function cannot be used for training. There are also networks that make a binary approximation to the input and output. For instance, accurate binary convolutional (ABC)-Net[13] uses linear combinations of bases of multiple binary weights to approximate full precision weights and linear combinations of multiple binary activations to approximate true activations for inputs, reducing information loss. However, few networks currently process the input and the first layer is convolved with full precision convolution. Designing robust network baselines is also a current direction for binary networks. FTBNN[14] re-investigates and tunes proper non-linear modules to fix that contradiction. But a robust baseline network does not solve the problem of accuracy degradation caused by the first layer of convolutional binarization. BinaryDuo[15] uses the gradient of the smoothed loss function to better estimate the gradient mismatch in a quantized neural network. RAD uses distribution loss to explicitly regularize the activation flow, and develop a framework to systematically formulate the loss. Although RAD normalizes the input from the loss function, the first layer convolution is still full accuracy. If the first layer of the existing binary network is convolved using binarized convolution, it is

difficult for the current binarization method to have high accuracy. This paper focuses on the first layer of convolution binarization leading to a large amount of information loss by proposing a new method that starts from the sign function property and converts the data to +1 and −1 at the input to avoid information loss in the first layer.

Binary networks take up less memory and are faster than other methods such as pruning and matrix decomposition. Great progress has been made in several aspects of binary networks, but the first convolutional layer of binarized networks often uses full precision, which can lead to a large accuracy drop if the first layer is binarized. The method in this paper is proposed to solve this problem.

Many previous literatures have focused on the treatment of weights and the activation is binarized directly through the sign function, so that the convolution of the first layer is mostly full precision. Assume that replacing the first layer of convolution with a binarized convolution results in a substantial loss of accuracy. Direct binarization of picture information to +1 and −1 at the first layer can lose a lot of information and lead to a loss of accuracy. In this paper, we want to replace the first layer of convolution with a binarized convolution, where the data is binarized in the first layer and with as little loss of accuracy as possible.

Image data contains a wealth of information prior to binarization. Binarization of the input data would result in a significant loss of information. Many binary networks use full precision data in the first layer of convolution to maximize the retention of the original information. It was demonstrated experimentally that by replacing the first full-accuracy convolutional layer of a binary network with a convolution in which both weights and activation are binarized, the accuracy of the network decreases significantly. In this paper, we want to change the first layer of convolution to binarized convolution, so that all the convolution layers become binarized convolution. It is important to retain as much information as possible about the input data in order to have a small decrease in accuracy after changing the first convolution layer.

Images are data within [0, 255] which are essentially represented in binary, so each number in the original image is converted to an 8-bit unsigned binary representation of that number. However, binary only has 0 and 1, 0 and 1 only get 1 after passing the sign function, which lacks the negative part. In this paper, all the zeros in the binary are changed to −1 (as shown in Fig.2), so that the data remains unchanged after passing the sign function, which is equivalent to no loss of information after passing the sign function, maximizing the loss of information in the original image after the data enters the first layer of binary convolution. After each number in the original picture has been changed to binary, each number is represented by 8 bits of binary. The original 3-channel image needs to be widened to 24 channels, spreading the information from the original image over 24 channels to be fed into the network.

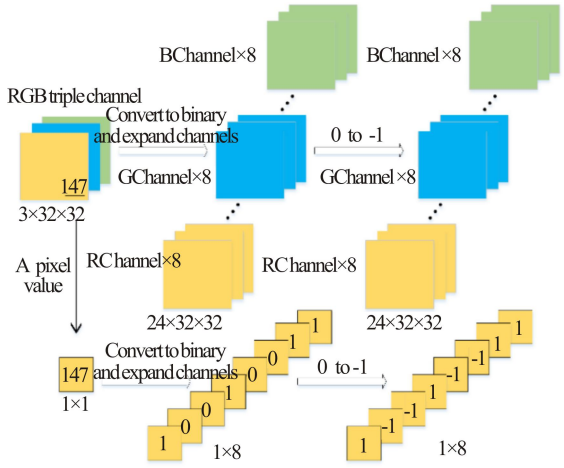

Fig.2 Input to binary and channel expansion

The concept of group convolution was originally introduced in AlexNet for distributing models across two graphics processing units (GPUs). Its effectiveness is now well demonstrated in ResNeXt[16]. The deep separable convolution proposed in Xception[17] encapsulates the idea of separable convolution in the inception series. MobileNet[8] utilizes deeply separable convolution and achieves state-of-the-art (SOTA) results in a lightweight model. Channel shuffle enables group convolution to obtain different information from different groups so that the inputs and outputs will be fully correlated.

As shown in Fig.4(b), the input graph is X (W_1, H_1, C_1).

Step 1: Calculate the size of each feature map as (W_1, H_1, C_1/g) for a total of g groups.

Step 2: The size of a single convolution kernel per group is (C_1/g, k, k). A convolutional kernel is divided into g groups of C_2 kernels, the number of kernels in each group is C_2/g, and the size of each group is (C_2/g, C_1/g, k, k). The number of convolutional kernel parameters in a set at this point is

$$params = k^2 \times \frac{C_1}{g} \times \frac{C_2}{g}. \tag{1}$$

Step 3: The output feature map size is (W_2, H_2, C_2), and the total number of parameters of the process is

$$params = \left(k^2 \times \frac{C_1}{g} \times \frac{C_2}{g} \right) \times g. \tag{2}$$

The total FLOPs are

$$FLOPs = \left(k^2 \times \frac{C_1}{g} \times \frac{C_2}{g} \times W_2 \times H_2 \right) \times g. \tag{3}$$

The input channels for the first layer of convolution are widened to 24 channels, which results in an increase in the amount of computation (FLOPs). To solve this problem, this paper adopts the idea of group convolution

in ShuffleNet. Dividing the first layer of convolution into 8 groups makes the computation the same as for the original 3 channels. After the data is expanded, the original channels become 8 channels each. Before feeding into the convolution, the data needs to be channel shuffled (as shown in Fig.4). The expanded channel data are grouped, so that each group of 3 channels comes from each of the 8 channels after the original 3-channel expansion of the data. Tab.1 gives the amount of computation (FLOPs) for the first layer of the network before and after using group convolution and reverting to 3-channel convolution (FLOPs) after applying group convolution.

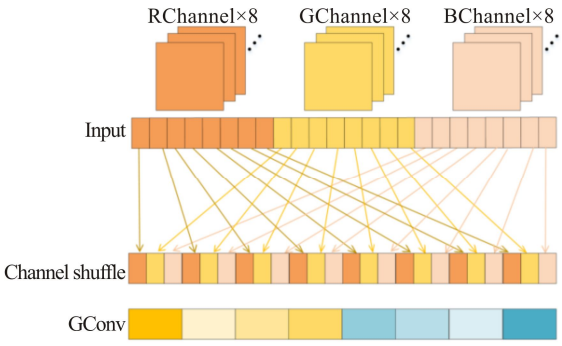

Fig.3 Schematic of channel shuffle (GConv stands for group convolution)

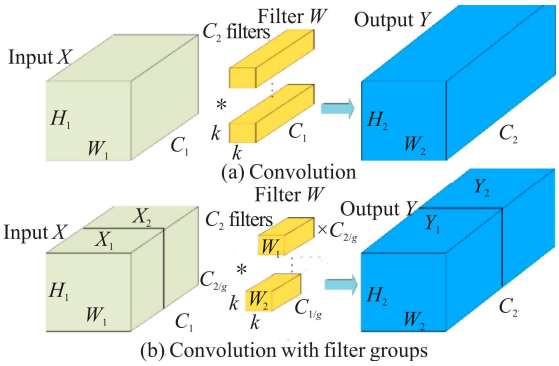

Fig.4 Schematic diagram of group convolution (*g* represents the number of groups, *H* and *W* represent the width and height of the input, respectively, C_1 represents the number of channels at input, and C_2 represents the number of channels at the output)

The experiments in this paper are based on the Pytorch deep learning framework. The network is based on ResNet18 and experiments are conducted on the CIFAR10 and SVHN datasets. The methods in this paper are applied to some SOTA binary networks to demonstrate the effectiveness.

Columns 4 to 6 of Tab.2 and Tab.3 show the accuracy of the binary network with the first layer of convolution retained as full accuracy, the first layer of convolution replaced by convolution with only binarized weights, and the binary network with the first layer of convolution fully binarized. The data in the two tables show that binarization of the convolution weights in the first layer results in a small decrease in accuracy, but binarization of the activation in the first layer results in a significant decrease in accuracy. Therefore, the first layer of convolutional binarization activation is responsible for the significant drop in accuracy. The last columns of Tab.2 and Tab.3 show the results after applying the method proposed in this paper to various binarization networks. The improvement in accuracy from the last columns of Tab.2 and Tab.3 demonstrates the effectiveness of the proposed method, with varying degrees of improvement for different networks. The experimental results show that the accuracy of the input data can be improved after changing the first convolution layer of the binary network to a binarized convolution layer by applying the method of this paper, proving that the method can reduce the information loss of the original image after the first layer of binarization. The method in this paper can be widely applied to many binary networks.

Tab.1 Amount of computation for the first layer of the network (ResNet-18.Conv0 represents the first layer of binary convolution with an input channel of 3, ResNet-18.Conv01 represents the first layer of convolution with an input channel of 24, and ResNet-18.Conv02 represents the first layer of convolution after applying group convolution)

Layer	MFLOPs	Params
ResNet-18.BinaryConv0	1.77	1 728
ResNet-18.BinaryConv01	14.16	13 824
ResNet-18.BinaryConv02	1.77	1 728

In this part, we investigate the behaviors and effects of the proposed data on binary spread channel processing and group convolution with channel shuffle techniques on BNN performance.

Data to binary dilation reduces the loss of information from the first layer of convolutional binarization and is the most important method to improve accuracy, while group convolution and channel shuffle reduce the additional computation due to dilation and do not reduce the accuracy of the network when used only in the first layer.

As shown in Tab.4, data on binary spread channel processing is the main method to improve the accuracy, and using group convolution with channel shuffle can reduce the increase in the computation of the first convolutional spread channel without decreasing the accuracy. From the data in Tab.2 and Tab.4, it is concluded that by changing the first full precision convolutional layer of the network to a weighted binarized convolutional layer, the accuracy of the network decreases less when the input is still full precision, so the reason for the significant decrease in accuracy

is the loss of a large amount of information directly in the first binarized input layer.

Tab.2 Performance of SOTA method on CIFAR10 dataset (*Acc* (%) (w) represents the accuracy of the original binary network with only the first layer of convolution changed to binarized weights, *Acc* (%) (w, a) represents the accuracy of the original binary network with both the weights and activation of the first layer of convolution binarized, and *Acc* (%) (ours) represents the accuracy after applying the proposed method)

Topology	Method	Bit-width (w/a)	*Acc* (%)	*Acc* (%) (w)	*Acc* (%) (w, a)	*Acc* (%) (ours)
ResNet-18	FP	32/32	93.0	-	-	-
	DoReFa	1/1	85.5	84.6	71.8	**72.9**
	BNN[18]	1/1	89.5	88.9	73.5	**79.6**
	Bi-Real-Net	1/1	90.6	90.1	75.2	**80.0**
	RAD	1/1	90.7	89.8	75.7	**82.8**
	DSQ[19]	1/1	90.9	88.6	76.2	**82.4**
	ReAct-Net	1/1	91.6	88.9	76.9	**82.6**
	IR-Net	1/1	91.7	88.5	78.3	**83.1**

Tab.3 Performance of SOTA method on SVHN dataset

Topology	Method	Bit-width (w/a)	*Acc* (%)	*Acc* (%) (w)	*Acc* (%) (w, a)	*Acc* (%) (ours)
ResNet-18	FP	32/32	96.1	-	-	-
	DoReFa	1/1	94.3	94.2	75.0	**93.4**
	BNN	1/1	95.1	95.0	77.9	**95.0**
	Bi-Real-Net	1/1	95.6	95.5	77.9	**95.2**
	RAD	1/1	95.1	95.0	78.3	**93.8**
	DSQ	1/1	95.5	95.0	78.1	**94.7**
	ReAct-Net	1/1	95.6	95.4	78.3	**94.6**
	IR-Net	1/1	95.2	95.1	78.2	**94.7**

Tab.4 Ablation study

Method	Bit-width (w/a)	*Acc* (%)
FP	32/32	93.0
IR-Net	1/1	78.2
GConv with channel shuffle	1/1	78.3
Data on binary spread channel processing	1/1	83.1
TBNN	1/1	**83.1**

In this paper, we propose a method for processing the input data to address the problem that the current binary networks change the first layer convolution of the network resulting in a significant decrease in network accuracy. The data is transformed using the characteristics of the sign function to change the data into an 8-bit binary representation, and a further measure is to change the 0 in binary to −1. The loss of information from the input through the sign function is significantly reduced. On the other hand, group convolution is applied to the convolution layer to recover the original 3-channel computation, in order to cope with the increased computation of the convolution layer caused by the addition of input channels. At the same time, channel shuffle allows the interoperability of information across groups. The effectiveness was verified by applying the method proposed in this paper on several SOTA networks based on ResNet-18 networks, CIFAR10 and SVHN datasets for experiments.

Statements and Declarations

The authors declare that there are no conflicts of interest related to this article.

References

[1] HE R, SUN S, YANG J, et al. Knowledge distillation as efficient pre-training：faster convergence, higher data-efficiency, and better transferability[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 19-24, 2022, New Orleans, Louisiana, USA. New York：IEEE, 2022：9161-9171.

[2] ZHANG L, CHEN X, TU X, et al. Wavelet knowledge distillation：towards efficient image-to-image translation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 19-24, 2022, New Orleans, Louisiana, USA. New York：IEEE, 2022：12464-12474.

[3] ZHONG Y, LIN M, NAN G, et al. IntraQ：learning synthetic images with intra-class heterogeneity for zero-shot network quantization[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 19-24, 2022, New Orleans, Louisiana, USA. New York：IEEE, 2022：12339-12348.

[4] LIU C, DING W, XIA X, et al. Circulant binary convolutional networks：enhancing the performance of 1-bit DCNNs with circulant back propagation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 13-19, 2019, Long Beach, CA, USA. New York：IEEE, 2019：2691-2699.

[5] LIU Z, SHEN Z, SAVVIDES M, et al. ReActNet：towards precise binary neural network with generalized activation functions[C]//European Conference on Computer Vision, August 23-28, 2020, Virtual. Cham：Springer, 2020：143-159.

[6] ZHOU S, WU Y, NI Z, et al. Dorefa-net：training low bit width convolutional neural networks with low bit width gradients[EB/OL]. (2016-06-20) [2022-06-22]. https：//arxiv.org/pdf/1606.06160.pdf.

[7] DING R, CHIN T W, LIU Z, et al. Regularizing activation distribution for training binarized deep networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 13-19, 2019, Long Beach, CA, USA. New York：IEEE, 2019：11408-11417.

[8] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets：efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17) [2022-06-22]. https：//arxiv.org/pdf/1704.04861.pdf.

[9] ZHANG X, ZHOU X, LIN M, et al. Shufflenet：an extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 18-22, 2018, Salt Lake City, UT, USA. New York：IEEE, 2018：6848-6856.

[10] MEHTA S, RASTEGARI M. Mobilevit：light-weight, general-purpose, and mobile-friendly vision transformer[EB/OL]. (2021-10-17) [2022-06-22]. https：//arxiv.org/pdf/2110.02178v2.pdf.

[11] QIN H, GONG R, LIU X, et al. Forward and backward information retention for accurate binary neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 13-19, 2020, Seattle, WA, USA. New York：IEEE, 2020：2250-2259.

[12] LIU Z, WU B, LUO W, et al. Bi-real net：enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm[C]//Proceedings of the European Conference on Computer Vision, September 8-14, 2018, Munich, Germany. Berlin, Heidelberg：Springer-Verlag, 2018：722-737.

[13] LIN X, ZHAO C, PAN W. Towards accurate binary convolutional neural network[J]. Advances in neural information processing systems, 2017, 30：345-353.

[14] SU Z, FANG L, GUO D, et al. FTBNN：rethinking non-linearity for 1-bit CNNs and going beyond[EB/OL]. (2010-09-29) [2022-06-22]. https：//www.xueshufan.com/reader/3096361616.

[15] KIM H, KIM K, KIM J, et al. Binaryduo：reducing gradient mismatch in binary activation network by coupling binary activations[EB/OL]. (2002-06-05) [2022-06-22]. https：//arxiv.org/pdf/2002.06517v1.pdf.

[16] XIE S, GIRSHICK R, DOLLÁR P, et al. Aggregated residual transformations for deep neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York：IEEE, 2017：1492-1500.

[17] CHOLLET F. Xception：deep learning with depth wise separable convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York：IEEE, 2017：1251-1258.

[18] HUBARA I, COURBARIAUX M, SOUDRY D, et al. Binarized neural networks[J]. Advances in neural information processing systems, 2016, 29：4107-4115.

[19] GONG R, LIU X, JIANG S, et al. Differentiable soft quantization：bridging full-precision and low-bit neural networks[C]//Proceedings of the IEEE International Conference on Computer Vision, October 27-November 3, 2019, Seoul, South Korea. New York：IEEE, 2019：4852-4861.