Multi-bit Adaptive Distillation for Binary Neural Networks

Ying Nie
ying.nie@huawei.com
Kai Han
kai.han@huawei.com
Yunhe Wang
yunhe.wang@huawei.com

Noah's Ark Lab,
Huawei Technologies

Abstract

Binary neural networks (BNNs) represent weights and activations using 1-bit values, which has extremely lower memory costs and computational complexities, but usually suffer from severe accuracy degradation. Knowledge distillation is an effective way to improve the performance of BNN by inheriting the knowledge from higher-bit network. However, faced with the accuracy gap and bit gap between 1-bit network and different higher-bit networks, it is uncertain which higher-bit network is more suitable to be the teacher of a certain BNN. Therefore, we propose a novel multi-bit adaptive distillation(MAD) method for maximally integrating the advantages of various bit-width teacher networks(*e.g.* 2-bit, 4-bit, 8-bit and 32-bit). In practice, intermediate features and output logits of teachers will be simultaneously utilized for improving the performance of BNN. Moreover, an adaptive knowledge adjusting scheme is explored to dynamically adjust the contribution of different teachers in the distillation process. Comprehensive experiments conducted on CIFAR-10/100 and ImageNet datasets with various network architectures demonstrate the superiorities of MAD over many state-of-the-arts binarization methods. For instance, without introducing any extra inference calculations, our binarized ResNet-18 achieves 1.5% improvement for BirealNet binarization method on ImageNet.

1 Introduction

Deep convolution neural networks (CNNs) have achieved notable progress in many computer vision tasks. However, the remarkable performance of CNNs usually relies on millions of parameters and billions of floating-point operations (FLOPs). For example, ResNet-18 [10] has about 11.7M parameters and requires 1.8B FLOPs for processing a single 224×224 image. To cater for deep learning on resource-limited platforms like mobile phones, it is necessary to explore portable deep neural networks.

A series of methods have been proposed to develop portable CNNs, including network pruning [9, 22, 31], low-bit quantization [6, 15, 27], knowledge distillation [12, 28, 37] and lightweight network architecture design [5, 13, 20]. Wherein, network binarization is a kind of quantization method to shrink the size of the desired network to the extreme. Compared

with vanilla 32-bit networks, binary neural networks (BNNs) adopt 1-bit values to represent weights and activations, which not only achieves a ∼32x model compression but also extremely reduces the computational complexity by a factor of ∼58x [27].

Directly reducing the bit-width of weights and activations from 32 to 1 usually results in a severe accuracy degradation due to the non-differentiable quantization function and its poor capacity. Generally, the accuracy of a given neural architecture will decrease as the number of bits decreases, and the difficulty of optimization will increase accordingly. Knowledge distillation can provide an effective method to improve the performance of 1-bit student network by inheriting the knowledge from 32-bit teacher [23, 35, 43, 44]. The full precision 32-bit teacher network is the best one in terms of its accuracy, but whether it is the most appropriate teacher to teach the 1-bit student network? Obviously, the output by a 1-bit network could be significant different from that of a 32-bit network, due to their large capacity gap. Imagine that forcing a pupil to understand the advanced course prepared by a university professor is usually hard. But if there are multiple teachers of different levels, the student can choose the teachers and their corresponding courses that are most beneficial for his/her own study. Neural networks of any bit-width larger than 1 can be taken as the teacher network for the binary network during the distillation. However, existing works rarely explore multiple teacher networks of different bit-widths for network binarization, let alone optimizing the choice of teachers.

In this paper, we propose a novel multi-bit adaptive distillation framework by utilizing the various and diverse knowledge in multiple pre-trained teachers of different bit-widths (*e.g.* 2-bit, 4-bit, 8-bit and 32-bit). Both knowledge inherited in logits from classification layer and features from intermediate layers are utilized together to teach the student network. To better aggregate knowledge from multi-bit teachers, a group of coefficients are introduced to combine both features and logits in teacher networks, and further develop an adaptive knowledge adjusting scheme to adjust the contribution of different teachers dynamically. During distillation, the parameters of student network and the learnable coefficients will be updated jointly on the training dataset. The training process of binary student network is guided by the most suitable teacher combination to obtain better performance. To the best of our knowledge, this is the first time to explore multiple teacher networks with different bit-widths for distilling binary neural network adaptively. Extensive experiments conducted on various datasets and networks demonstrate the effectiveness of the proposed algorithm over the state-of-the-art methods for training BNNs.

2 Related Works

Binarization. The pioneering work BNN [15] turned both weights and activations into -1 and 1, it also verified the feasibility and benefit of binary neural networks. Rastegari *et al*. [27] proposed a better method for estimating floating-point values by adding a scale factor to the binary values instead of simply taking a sign function. DoReFaNet [42] explored how the different bit-widths of weights, activations and gradients affect the performance. BirealNet [21] enhanced the representational capability of BNNs by adding identity shortcut between all the intermediate convolutional layers. Yang *et al*. [34] and Gong *et al*. [7] investigated a differentiable soft quantization scheme to gradually approximate the standard quantization function. Lin *et al*. [18] analyzed the influence of angular bias on the quantization error and then creatively introduced a rotated BNN. In addition, exploring more effective neural architecture search method for BNNs has been discussed in [3, 4, 17].

Distillation. Knowledge distillation was first proposed in [12], which is an effective approach to improve the student model's performance by inheriting knowledge from a teacher model. FitNets [28] combined the soft output and intermediate features through introducing point-wise convolutional transforming layers. Yim *et al*. [36] defined the knowledge as inner product between features from two layers in teacher network, which can optimize student network faster. Apart from transferring between a static pre-trained teacher and a student, Zhang *et al*. [41] presented a deep mutual learning strategy by collaboratively training an ensemble of students. Converting both the information in teacher and student to the same space where the distance is easier to measure are studied in [8, 32]. Besides, distilled by multiple teachers in other tasks are also investigated in [29, 33, 37, 39]. To bridge the gap between the small student and the larger teacher, Mirzadeh *et al*. [25] and Son *et al*. [30] employed intermediate-sized networks (teacher assistant) and achieved promising result. In addition to the classification tasks, knowledge distillation was also introduced in the Generative Adversarial Networks (GANs) [16].

Distillation for BNNs. Aside from exploring more accurate quantization functions, there are also many works on improving the performance of BNNs by distillation. Zhuang *et al*. [42] proposed to inherit the strong representational ability of single full-precision teacher network to the low-bit student network with the same architecture. Zhou *et al*. [43] and Ye *et al*. [35] further applied the distillation paradigm by single 32-bit teacher to the binary network and achieved higher performance. Real-to-Binary-Net [24] and ReActNet [23] achieved state-of-the-art performance by redesigning binary-friendly network architectures and adopting effective training techniques such as two-step training and distillation by single full precision network. To the best knowledge of ours, existing works rarely explore multiple teacher networks of different bit-widths for teaching BNN dynamically.

3 Preliminaries

Quantized Neural Networks. In all our following experiments, we use the DoReFaNet method [42] to quantize the 32-bit network into multiple low-bit(*i.e*. 2-bit, 4-bit and 8-bit) networks to get all our teachers. A general quantization function is first introduced:

$$quantize_k(x) = \frac{1}{2^k-1} round((2^k-1)x), \tag{1}$$

where k denotes the number of bit-width. The activations are clipped to [0,1] and then quantized to low-bit values:

$$a^q = quantize_k(\text{clip}(a^f, 0, 1)), \tag{2}$$

where $\text{clip}(a^f, 0, 1) = \max(0, \min(1, a^f))$. Similarly, the weights are transformed to [0,1] and then quantized to low-bit values:

$$w^q = 2quantize_k(\frac{tanh(w^f)}{2max(|tanh(w^f)|)} + 0.5) - 1, \tag{3}$$

Different from low-bit neural networks, BNNs use 1-bit values to replace floating-point numbers of weights and activations. In particular, every element $w^f \in \mathbb{R}$ in the weights is binarized into a binary value:

$$w^b = \alpha \cdot \text{sign}(w^f), \tag{4}$$

where α is the scale factor, and $\text{sign}(\cdot)$ is the sign function that outputs -1 for negative numbers and $+1$ otherwise. For the scale factor, the current approaches usually use the absolute mean of current layer's weights or make it learnable [40, 42]. As for activations, each element a^f is binarized similarly:

$$a^b = \text{round}(\text{clip}(a^f, 0, 1)), \tag{5}$$

Knowledge Distillation. Knowledge distillation is an effective approach to improve the student model's performance. Given the groundtruth one-hot label vector $\mathbf{y} \in \{0,1\}^C$ and the output logits of both student and teacher network, *i.e.* $\mathbf{y}^s \in \mathbb{R}^C$ and $\mathbf{y}^t \in \mathbb{R}^C$, where C is the number of classes. The knowledge distillation is achieved by minimizing the following loss function:

$$\mathcal{L}_{KL}\left(\sigma(\frac{\mathbf{y}^t}{T}),\sigma(\frac{\mathbf{y}^s}{T})\right) = \left\langle \sigma(\frac{\mathbf{y}^t}{T}), \log\frac{\sigma(\mathbf{y}^t/T)}{\sigma(\mathbf{y}^s/T)} \right\rangle, \tag{6}$$

where $\mathcal{L}_{KL}$ refers to the Kullback–Leibler divergence, σ represents softmax function. T is the temperature coefficient and when T increases, the probability distribution produced by the softmax function becomes softer. A series of knowledge distillation methods are proposed to make full use of the teacher networks, such as Fitnets [28] and attention transfer [38]. Inspired by the progress in knowledge distillation, we propose to take advantage of teachers to help the training of BNNs, where the teacher can be higher-bit quantized networks with much better performance.

4 Learning from Multi-bit to 1-bit

Most of BNNs utilize STE to enable end-to-end training, when both the weights and activations in network are quantized into 1-bit values, the difficulty of optimizing BNNs is hard. Besides, full precision network or quantized networks with ≥ 2 bit-width perform much better than the corresponding BNN as illustrated in the blue bars in Fig. 1. It would be helpful to use the higher-bit network with higher accuracy to guide the training of BNN.

Accuracy Gap and Bit Gap. There are two factors may influence the effect of knowledge distillation: 1) the performance of teacher network, 2) the bit gap between teacher network and student network. The full-precision network has the highest accuracy compared with other lower-bit versions. Higher performance may provide more potential to the student. Additionally, the bit gap between 2-bit teacher and 1-bit student is smaller, *i.e.* the similarity between their outputs is higher so that a smaller bit gap is easier for the student to learn from teacher. We then conduct a toy experiment to validate the influence of these two factors. For fairness, we repeat the experiment for 4 times with different random seeds, and report the median value. The red bars in Fig. 1 displays the performance of binarized ResNet-20 distilled by teachers with different bit-widths on CIFAR-10 dataset. Here, the DoReFaNet binarization method is used to binarize the weights and activations. From Fig. 1, we can see that although 32-bit teacher has higher accuracy, 1-bit student distilled by 32-bit teacher achieves lower accuracy than 2-bit teacher. In addition, although 4-bit teacher has smaller bit gap, 4-bit distillation achieves lower accuracy than 8-bit distillation. Thus, to better guide the knowledge transfer, we need to achieve a balance between higher accuracy gap and smaller bit gap of the teacher networks.

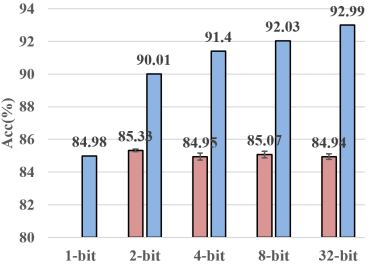


Figure 1: The accuracy charts of ResNet-20 on CIFAR-10, where blue bars denote the accuracy of ResNet-20 with various bit-widths, red bars denote the accuracy of binarized ResNet-20 distilled by different single higher bit-width teacher.

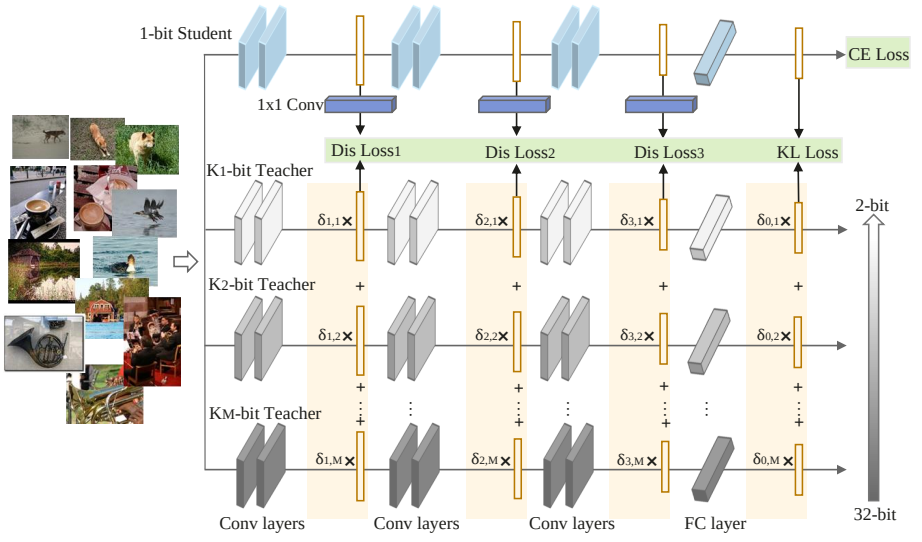


Figure 2: The diagram of the proposed method. Each teacher network is assigned a set of learnable coefficients. The overall loss consists of the cross-entropy loss of logits, KL-divergence of logits, and distance loss of intermediate layers' features, which are used together to update the binary student network, the point-wise convolutional layers and the learnable coefficients simultaneously.

4.1 Multi-bit Distillation

Multi-bit Logits Distillation. Based on the above observations, we propose to train the BNN by learning from multiple pre-trained teachers with different bit-widths, as shown in Fig. 2. In particular, given M teachers whose bit-widths are $k_1, k_2, \cdots, k_M$, respectively, we utilize all of them to teach the student. Denoting the output logits of the m-th teacher as $\mathbf{y}_m^t \in \mathbb{R}^C$, we integrate these outputs to form the multi-bit logits of all the teachers:

$$\mathbf{y}^t = \sum_{m=1}^{M} \delta_{0,m} \mathbf{y}_m^t, \tag{7}$$

where $\delta_{0,m}$ is the coefficient indicating the importance of m-th teacher's output and it satisfies $\sum_{m=1}^{M} \delta_{0,m} = 1$, which can be realized simply by softmax function. Denoting the logits of 1-bit student network as $\mathbf{y}^s \in \mathbb{R}^C$, we utilize the multi-bit logits in Eq. 7 to guide the training of student network and the loss function is the same as Eq. 6.

Multi-bit Feature Distillation. To provide richer knowledge from teachers, we distill not only the logits from final classification layer, but also the features from intermediate layers. For example, we choose the feature output by every stage to compute the distance loss between intermediate layers for ResNet [10]. Consider the i-th layer ($i = 1, 2, \cdots$) whose output feature is $F_i^s \in \mathbb{R}^{h \times w \times c}$ where h, w, c represent height, width and channels, respectively. We integrate multiple teachers with different bit-widths by adding their activations to get the final full-precision activations,

$$F_i^t = \sum_{m=1}^{M} \delta_{i,m} F_{i,m}, \tag{8}$$

where $\delta_{i,m}$ is the coefficient for the i-th layer in the m-th teacher and it also satisfies $\sum_{m=1}^{M} \delta_{i,m} = 1$, $F_{i,m}$ is the corresponding quantized activations. However, if we directly calculate the

distance loss of intermediate layers between teachers and student, it will not improve the performance of student. The features in intermediate layers is not one-to-one correspondence by channel between teacher networks, or between the teacher and student network. Inspired by Fitnets [28], we construct point-wise convolutional transforming layers on top of the selected intermediate layers of the student network to transform the features. The transformed features are $r_i(F_i^s)$ using the transforming layer r_i. In addition, if we also transform the mixed feature of multiple teachers by a transforming layer, the weight in convolutional transforming layers of student and teachers will easily all fall into zero values as the training progresses, so we keep the mixed features of teachers unchanged.

The distance between the features of multi-bit teachers and the transformed features of student is calculated to monitor the progress of training BNN. Here we use smoothed-L_1 distance owing to its smoother gradient and robustness:

$$\mathcal{L}_{Dis} = \sum_{i=1}^{N} Dis\left(F_i^t, r_i(F_i^s)\right) = \sum_{i=1}^{N}\sum_{x\in\mathcal{A}} smooth_{L_1}(x), \tag{9}$$

where N refers to the number of selected intermediate layers and

$$\mathcal{A} = F_i^t - r_i(F_i^s) \tag{10}$$

$$smooth_{L_1}(x) = \begin{cases} 0.5(x)^2, & \text{if } |x| < 1, \\ |x| - 0.5, & \text{otherwise}, \end{cases} \tag{11}$$

By combining the cross-entropy loss of logits, KL divergence of logits in Eq. 6 and distance loss of intermediate layers' features in Eq. 9, we achieve the objective function:

$$\begin{aligned} \mathcal{L}_{All} = \mathcal{L}_{CE} + \alpha\mathcal{L}_{KL} + \beta\mathcal{L}_{Dis} = \langle \mathbf{y}, \log\sigma(\mathbf{y}^s)\rangle + \\ \alpha\left\langle \sigma(\frac{\mathbf{y}^t}{T}), \log\frac{\sigma(\mathbf{y}^t/T)}{\sigma(\mathbf{y}^s/T)}\right\rangle + \beta\sum_{i=1}^{N}\sum_{x\in\mathcal{A}} smooth_{L_1}(x), \end{aligned} \tag{12}$$

where α and β are the trade-off hyper-parameters. Given positive $\delta_{0,m}$ and $\delta_{i,m}$ values, optimizing the loss $\mathcal{L}_{All}$ can guide the student BNN to learn from the multiple teacher networks.

4.2 Adaptive Knowledge Adjusting

The importance of each individual teacher is not absolutely static. Instead, the optimal teacher importance may change at different steps during the training process. Hence beyond the manual-setting of δ values, we propose an adaptive knowledge adjusting scheme to make the coefficients in Eq. 7 and Eq. 8 learnable and changeable. All trainable parameters in our proposed framework include three parts: the student network's parameters θ_s, the parameters θ_r in transforming layers equipped with binary student network and the learnable coefficients δ indicating the importance of different teachers. It should be noted that the transforming layers do not participate in the inference of student model, so the complexity of BNN is the same as the original one. All the trainable parameters are optimized jointly under the supervision of the overall loss in Eq. 12. The gradients of weights in ordinary neural layers can be computed using standard back-propagation algorithm. As for the introduced coefficients δ, the gradients can be calculated as

Table 1: Accuracy comparisons on CIFAR-10 and CIFAR-100.

Model	Method	Bit-Width (W/A)	CIFAR-10 (%)	CIFAR-100 (%)
VGG-Small	FP	32/32	93.78	73.42
	8-bit	8/8	93.81	74.31
	4-bit	4/4	93.65	74.02
	2-bit	2/2	93.33	73.27
	BNN [15]	1/1	89.90	-
	XNOR [27]	1/1	89.80	67.18
	EXP-Net [43]	1/1	90.19	64.54
	DSQ [7]	1/1	91.72	-
	IR-Net [26]	1/1	90.4	-
	RBNN [18]	1/1	91.3	-
	DoReFaNet [42]	1/1	91.20	70.17
	MAD	1/1	**92.28** $\pm$ 0.11	**71.23** $\pm$ 0.20
ResNet-20	FP	32/32	92.99	69.43
	8-bit	8/8	92.03	66.73
	4-bit	4/4	91.40	67.35
	2-bit	2/2	90.01	62.53
	DSQ [7]	1/1	84.11	-
	XNOR [27]	1/1	85.33	54.06
	IR-Net [26]	1/1	85.4	-
	DoReFaNet [42]	1/1	84.78	54.37
	MAD	1/1	**85.91** $\pm$ 0.12	**55.03** $\pm$ 0.18

$$\frac{\partial \mathcal{L}_{All}}{\partial \delta_{0,m}} = \frac{\partial \mathcal{L}_{All}}{\partial \mathbf{y}^t} \circ \frac{\partial \mathbf{y}^t}{\partial \delta_{0,m}} = \alpha \left\langle \frac{\partial \mathcal{L}_{KL}}{\partial \mathbf{y}^t}, \mathbf{y}^t_m \right\rangle, \tag{13}$$

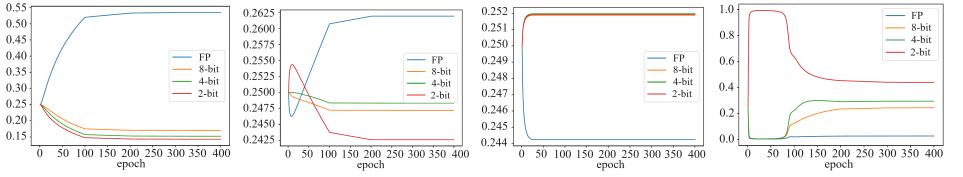
$$\frac{\partial \mathcal{L}_{All}}{\partial \delta_{i,m}} = \frac{\partial \mathcal{L}_{All}}{\partial F^t_i} \circ \frac{\partial F^t_i}{\partial \delta_{i,m}} = \beta \left\langle \frac{\partial \mathcal{L}_{Dis}}{\partial \mathrm{Vec}(F^t_i)}, \mathrm{Vec}(F_{i,m}) \right\rangle, \tag{14}$$

where $\circ$ means the multiplication in the chain rule, $\mathrm{Vec}(\cdot)$ is the vectorization operation for transforming the features from four dimensional matrix format(*i.e.* $[n,c,h,w]$, where n, c, h, w represent the batch size, channel, height and width of features, respectively) to two dimensional vector(*i.e.* $[n, c \times h \times w]$). The value of $\frac{\partial \mathcal{L}_{KL}}{\partial \mathbf{y}^t}$ and $\frac{\partial \mathcal{L}_{Dis}}{\partial \mathrm{Vec}(F^t_i)}$ can be obtained via the standard chain rule back-propagation. All the trainable parameters are updated simultaneously using SGD or Adam optimizer.

5 Experiments

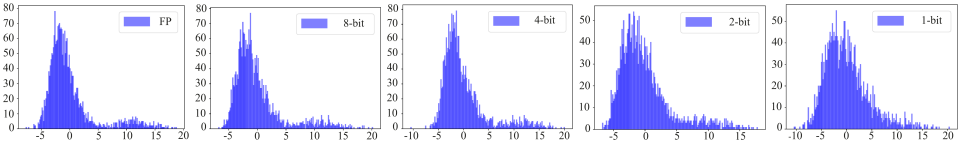
5.1 Experimental Setup

We use four teachers with different bit-widths including 32-bit, 8-bit, 4-bit and 2-bit to teach BNN, and the quantization method of these low-bit teacher networks is described in previous preliminaries section. For ResNet-20 and ResNet-18 [10], we choose the features output by the end of every stage to compute the distance loss of intermediate features. For VGG-Small [40], we choose the features output by every MaxPooling operation to compute the distance loss. In addition, we also conduct experiment on lightweight model based on ReActNet-A [23] since its backbone structure is MobileNetV1 [14]. For CIFAR-10 dataset, the hyperparameters α and β in Eq. 12 are empirically set to 1 and 0.2, respectively. For CIFAR-100 and ImageNet datasets, α and β in Eq. 12 are set to 1 and 0.1, respectively.

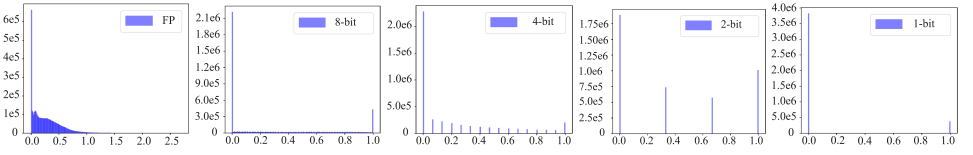


(a) first MP (b) second MP (c) third MP (d) classification layer

Figure 3: The evolution of four teachers' coefficients at different layers, where MP represents MaxPooling operation.



(a) logits of classification layer



(b) activations of first MP layer

Figure 4: The histogram distribution of logits and activations, where MP represents MaxPooling operation.

5.2 Experiments on CIFAR-10/100

We employ the widely-used network architectures including VGG-Small and ResNet-20 for CIFAR-10/100 dataset, and follow the general setting in BNNs to set the first and last layers as full-precision convolutional layers. For VGG-Small [40], we train 400 epochs and employ learning rate starting at 0.01 and decay the learning rate by a factor of 10 at 100, 200, and 300 epochs. For ResNet-20, we train 400 epochs and employ learning rate starting at 0.1 and decay the learning rate by a factor of 10 at 200, 300, and 375 epochs. SGD with momentum of 0.9 is used as our optimization algorithm with batch size of 128. Weight decay is set to 5e-4 when training binary VGG-Small, and 1e-4 when training binary ResNet-20. In order to get more accurate statistics, we repeat the experiment for 4 times with different random seeds, and report the median and standard deviation of classification accuracy in Table 1. In all cases, without changing the network architecture, our method obtains the best performance on both CIFAR-10/100. For example, there are about 1.08% and 1.13% improvement for VGG-Small and ResNet-20 binarized by DoReFaNet method on CIFAR-10, respectively.

Analysis on the importance of multi-teachers. To analyze the adaptive adjusting scheme for multiple teachers, we plot the evolution process of the coefficients at different layers for each teacher in VGG-Small on CIFAR-100. From Fig. 3, we can see that the full-precision teacher is always dominated at the first MP layer, and the 2-bit teacher is always dominated at the classification layer. In other words, the early layers of VGG-Small tend to choose high-precision teachers, and the later layers tend to choose teachers with similar outputs. For the second MP layer and third MP layer, the coefficients of four teachers have slightly change during training process. That is, as the training progresses, the binary student

will choose higher-bit teachers dynamically and targetedly at different layers.

Fig. 4 intuitively displays the histogram distribution of logits and activations in network, for simplicity, we only display the first MP layer's activations. From the histogram distribution of logits in classification layer, the distribution of 1-bit student is obviously closest to the distribution of 2-bit teacher, which can be attributed to the largest coefficients of 2-bit teacher in training process. Besides, in the distribution of activations of the first MP layer among four teachers, zeros are all in the majority, which can explains that zeros are also in the majority at this layer of 1-bit student network.

5.3 Experiments on ImageNet

Binarize non-compact network. For the large-scale ImageNet classification task, we firstly use the widely-used non-compact ResNet-18 as the backbone network. We use three binarization methods including DoReFaNet, BirealNet [21] and ReActNet-ResNet [23] to binarize weights and activations. DoReFaNet does not change the architecture of vanilla ResNet-18, while BirealNet equips every convolution layer with a shortcut layer. Based on the modified ResNet-18 structure of BirealNet, ReActNet-ResNet further proposed a channel-wise reshaping and shifting operation on activation first, then a two-step distillation scheme by a single full precision teacher. For DoReFaNet binarization method, we utilize both SGD and Adam as they influence the accuracy largely [1]. For BirealNet and ReActNet-ResNet binarization method, we only apply Adam for its effectiveness. In addition, when training the ReActNet-ResNet, we also follow their activation transformation operation and two-step training scheme but by multiple teachers. For models using SGD, we train 120 epochs with the initial learning rate of 0.2 and decay by a factor of 10 at 70, 90, and 110 epochs. For Adam, we employ learning rate starting at 0.002 and decay by a factor of 10 at 70, 90, and 110 epochs. Weight decay is set to 0 when training BNN.

Table 2: Accuracy of ResNet-18 on ImageNet.

Method	Bit-Width (W/A)	Top-1 (%)	Top-5 (%)
FP	32/32	71.0	89.8
8-bit	8/8	70.4	89.6
4-bit	4/4	69.8	89.1
2-bit	2/2	64.1	85.4
BNN [15]	1/1	42.2	67.1
ABC-Net [19]	1/1	42.7	67.6
QN [34]	1/1	53.6	75.3
Bop [11]	1/1	56.6	79.4
XNOR++ [3]	1/1	57.1	79.9
DGRL [35]	1/1	60.45	-
Real-to-Binary-Net [24]	1/1	65.4	86.2
XNOR [27]	1/1	51.2	73.2
MAD	1/1	**52.0**	**73.8**
DoReFaNet [42] (SGD)	1/1	52.5	76.5
MAD (SGD)	1/1	**53.4**	**77.5**
DoReFaNet [42] (Adam)	1/1	56.2	78.9
MAD (Adam)	1/1	**57.4**	**80.2**
BirealNet [21]	1/1	56.4	79.5
MAD	1/1	**57.9**	**80.3**
ReActNet-ResNet [23]	1/1	65.5	-
MAD	1/1	**66.5**	**86.5**

Table 2 shows the comparison of imagenet classification results between our work and the excellent work of other predecessors. When using DoReFaNet binarization method, without changing the network architecture, we achieve 0.9% and 1.2% Top-1 improvement for SGD and Adam respectively. When using BirealNet binarization method, we achieve 1.5% Top-1 improvement for Adam. In addition, we achieve 1.0% Top-1 improvement for ReActNet-ResNet using Adam, and owing to the adaptive distillation scheme by multiple teachers, the performance of the binarized shortcut-added ResNet-18 exceeds the vanilla 2-bit ResNet-18 network. When compared with other kd-based binarization methods including

DGRL [35], Real-to-Binary-Net [24] and ReActNet-ResNet [23]on the same ResNet-18 architecture, our ReActNet-ResNet-MAD achieves the highest accuracy.

Binarize lightweight network. Lightweight network has more practical value than non-compact network, so we conduct experiment on ReActNet-A [23], which is a strong binarized network modified from MobileNetV1. Considering that the accuracy of baseline ReActNet-A is close to that of some higher-bit networks *e.g.* 4-bit ResNet-18, we use the more powerful ResNet-50 as teachers here. Besides, to save the running memory in GPU, we only utilized pre-trained 8-bit, 4-bit and 2-bit as teachers since the performance of 8-bit is close to that of the full-precision. In addition, due to the different architecture of teacher networks and student network, we only calculate the cross-entropy loss of logits and the KL-divergence of logits to guide the learning of student network.

Table 3: Accuracy of ReActNet-A on ImageNet.

Method	Top-1(%)	Top-5(%)
ResNet-50 8-bit	76.5	93.0
ResNet-50 4-bit	76.1	92.5
ResNet-50 2-bit	72.3	89.6
ReActNet-A 1-bit [23]	69.4	-
MAD 1-bit	**70.2**	**88.7**

The classification accuracy is shown in Table 3, with the adaptive distillation by multiple higher-bit ResNet-50 teachers, we increase the Top-1 accuracy of ReActNet-A to 70.2%, surpassing the distillation by single static 32-bit teacher network.

We also conduct a series of ablation studies in the Supplementary File.

Memory and computational cost analysis. We take the BirealNet-ResNet18 as an example to demonstrate the superiority of the binary neural network in memory saving and computational cost reduction. We compare the memory and computational cost of the full-precision and binary neural network in Table 4. The memory calculation method is 32 times the full precision parameters plus 1 times the binary parameters [21]. In addition, we follow the principle of counting the total operations (OPs) in ReActNet [23], that is, first count the binary operations (BOPs) and the floating point operations (FLOPs) separately, and then the total operations is calculated by OPs = BOPs/64 + FLOPs.

Table 4: Analysis of Memory and Computational Cost for FP and Binary ResNet-18.

Bit-Width (W/A)	Memory (Mbit)	BOPs ($x10^8$)	FLOPs ($x10^8$)	OPs ($x10^8$)	Top-1(%)
32/32	374.1	0.0	18.1	18.1	71.0
1/1	33.6	15.6	1.39	1.63	57.9

From Table 4, compared with the full-precision neural network, the binary neural network achieves 91% memory savings and 91% reduction in computational cost. In addition, without introducing any extra inference calculations, our binarized ResNet-18 achieves 1.5% improvement for the BirealNet binarization method.

6 Conclusion

In this paper, we propose a novel multi-bit adaptive distillation method for improving the performance of BNN. We combine the knowledge from different layers of the higher-bit teachers to provide BNN with richer information. To better aggregate the knowledge from multiple teachers with different bit-widths, we introduce a simple yet effective adaptive knowledge adjusting scheme to adjust the contribution of teachers in the training process dynamically. Extensive experiments conducted on various datasets and networks indicate the effectiveness of the proposed method.

References

[1] Milad Alizadeh, Javier Fernández-Marqués, Nicholas D. Lane, and Yarin Gal. A systematic study of binary neural networks' optimisation. In *ICLR*, 2019.

[2] Adrian Bulat and Georgios Tzimiropoulos. Xnor-net++: Improved binary neural networks. *arXiv preprint arXiv:1909.13863*, 2019.

[3] Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. Bats: Binary architecture search. *arXiv preprint arXiv:2003.01711*, 2020.

[4] Hanlin Chen, Baochang Zhang, Xiawu Zheng, Jianzhuang Liu, Rongrong Ji, David Doermann, Guodong Guo, et al. Binarized neural architecture search for efficient object recognition. *International Journal of Computer Vision*, pages 1–16, 2020.

[5] Hanting Chen, Yunhe Wang, Chunjing Xu, Boxin Shi, Chao Xu, Qi Tian, and Chang Xu. Addernet: Do we really need multiplications in deep learning? *arXiv preprint arXiv:1912.13200*, 2019.

[6] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NeurIPS*, pages 3123–3131, 2015.

[7] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4852–4861, 2019.

[8] Jia Guo, Minghao Chen, Yao Hu, Chen Zhu, Xiaofei He, and Deng Cai. Spherical knowledge distillation. *arXiv preprint arXiv:2010.07485*, 2020.

[9] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[11] Koen Helwegen, James Widdicombe, Lukas Geiger, Zechun Liu, Kwang-Ting Cheng, and Roeland Nusselder. Latent weights do not exist: Rethinking binarized neural network optimization. In *NeurIPS*, 2019.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019.

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[15] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *NeurIPS*, pages 4107–4115, 2016.

[16] Qing Jin, Jian Ren, Oliver J Woodford, Jiazhuo Wang, Geng Yuan, Yanzhi Wang, and Sergey Tulyakov. Teachers do more than teach: Compressing image-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13600–13611, 2021.

[17] Dahyun Kim, Kunal Pratap Singh, and Jonghyun Choi. Learning architectures for binary networks. In *European Conference on Computer Vision*, pages 575–591. Springer, 2020.

[18] Mingbao Lin, Rongrong Ji, Zihan Xu, Baochang Zhang, Yan Wang, Yongjian Wu, Feiyue Huang, and Chia-Wen Lin. Rotated binary neural network. *Advances in Neural Information Processing Systems*, 33, 2020.

[19] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *Advances in Neural Information Processing Systems*, pages 345–353, 2017.

[20] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[21] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*, 2018.

[22] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Tim Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *ICCV*, 2019.

[23] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. *arXiv preprint arXiv:2003.03488*, 2020.

[24] Brais Martinez, Jing Yang, Adrian Bulat, and Georgios Tzimiropoulos. Training binary neural networks with real-to-binary convolutions. *arXiv preprint arXiv:2003.11535*, 2020.

[25] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020.

[26] Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2250–2259, 2020.

[27] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, pages 525–542. Springer, 2016.

[28] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[29] Muhamad Risqi U Saputra, Pedro PB de Gusmao, Yasin Almalioglu, Andrew Markham, and Niki Trigoni. Distilling knowledge from a deep pose regressor network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 263–272, 2019.

[30] Wonchul Son, Jaemin Na, and Wonjun Hwang. Densely guided knowledge distillation using multiple teacher assistants. *arXiv preprint arXiv:2009.08825*, 2020.

[31] Yunhe Wang, Chang Xu, Jiayan Qiu, Chao Xu, and Dacheng Tao. Towards evolutionary compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2476–2485, 2018.

[32] Yixing Xu, Chang Xu, Xinghao Chen, Wei Zhang, Chunjing Xu, and Yunhe Wang. Kernel based progressive distillation for adder neural networks. *arXiv preprint arXiv:2009.13044*, 2020.

[33] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan Yuille. Knowledge distillation in generations: More tolerant teachers educate better students. *arXiv preprint arXiv:1805.05551*, 2018.

[34] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *CVPR*, pages 7308–7316, 2019.

[35] Jianming Ye, Shiliang Zhang, and Jingdong Wang. Distillation guided residual learning for binary convolutional neural networks. *arXiv preprint arXiv:2007.05223*, 2020.

[36] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[37] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *SIGKDD*, 2017.

[38] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[39] Chenrui Zhang and Yuxin Peng. Better and faster: knowledge transfer from multiple self-supervised learning tasks via graph distillation for video classification. *arXiv preprint arXiv:1804.10069*, 2018.

[40] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *ECCV*, 2018.

[41] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

[42] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[43] Zhengguang Zhou, Wengang Zhou, Xutao Lv, Xuan Huang, Xiaoyu Wang, and Houqiang Li. Progressive learning of low-precision networks. *arXiv preprint arXiv:1905.11781*, 2019.

[44] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Towards effective low-bitwidth convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7920–7928, 2018.