**IR²Net: information restriction and information recovery for accurate binary neural networks**

Ping Xue¹ · Yang Lu¹,²,³ · Jingfei Chang¹ · Xing Wei¹,²,⁴ · Zhen Wei¹,²,³

Received: 5 January 2022 / Accepted: 15 March 2023 / Published online: 27 March 2023
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

**Abstract**

Weight and activation binarization can efficiently compress deep neural networks and accelerate model inference, but they cause severe accuracy degradation. Existing optimization methods for binary neural networks (BNNs) focus on fitting full-precision networks to reduce quantization errors and suffer from a tradeoff between accuracy and efficiency. In contrast, considering information loss and the mismatch between model capacity and input information quantity caused by network binarization, we propose Information Restriction and Information Recovery Network (IR²Net) to stimulate the potential of BNNs and achieve improved network accuracy by restricting the input information and recovering feature information. The proposed approach includes (1) information restriction, which evaluates the feature information extracted from the input by a BNN, discards some of the information it cannot focus on, and reduces the amount of the input information to match the model capacity; and (2) information recovery: due to the information loss incurred during forward propagation, the extracted feature information of the network is not sufficient for supporting accurate classification. Shallow feature maps with richer information are selected, and these feature maps are fused with the final feature maps to recover the extracted feature information and further enhance the model capacity to match the amount of input information. In addition, the computational cost is reduced by streamlining the information recovery method to strike a better tradeoff between accuracy and efficiency. Experimental results demonstrate that our approach still achieves comparable accuracy even with a ~ 10x floating-point operations (FLOPs) reduction for ResNet-18. The models and code are available at [https://github.com/pingxue-hfut/IR2Net](https://github.com/pingxue-hfut/IR2Net).

**Keywords** BNNs · Information restriction · Information recovery · Relevant information

---

¹ School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
² Anhui Mine IOT and Security Monitoring Technology Key Laboratory, Hefei 230009, China
³ Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei University of Technology, Hefei 230009, China
⁴ Intelligent Manufacturing Institute of Hefei University of Technology, Hefei 230009, China
1 Introduction

Deep Convolutional Neural Networks (CNNs) have made much progress in a wide variety of computer vision applications [1–4]. However, as research has advanced, the depth of networks has expanded from a few layers to hundreds of layers [5–8]. The massive numbers of parameters and ultrahigh computational complexity levels of CNNs greatly constrain their deployment, especially under the conditions of applications with high real-time requirements or limited computational resources. To solve this problem, various compression techniques for CNNs have emerged. Tensor factorization [9, 10] is the factorization of one original large-size weight tensor into several low-rank tensors to reduce the computation of convolutional operations and accelerate network inference. Network pruning [11–13] reduces model redundancy by pruning convolutional kernels or channels. An efficient architecture design [14–17] replaces conventional convolutional layers with well-designed lightweight modules. Knowledge distillation [18–20] attempts to transfer knowledge from complex networks (teachers) to compact networks (students). Additionally, quantization [21–25] replaces 32-bit weights and activations with low-bit (e.g., 16-bit) ones to reduce both memory footprint and computational complexity. The extreme form of quantization is binarization. Compared with 32-bit floating-point networks, network binarization constrains both weights and activations to \{-1, +1\}, i.e., the parameters of binary neural networks (BNNs) need only 1-bit representation, which greatly reduces the storage requirement. Furthermore, while binarizing network weights and activations, the computationally intensive matrix multiplication and addition operations in full-precision networks are replaced with low-cost XNOR and bitcount, greatly reducing the network inference delay. Therefore, benefiting from a high compression ratio, good acceleration, and energy savings, network binarization has been considered one of the most promising techniques for network compression and is the focus of this work.

Network binarization has attracted much attention due to its advantages in compression and acceleration. Although much progress has been made, existing binarization methods still suffer from a tradeoff between accuracy and efficiency. For example, XNOR-Net [27] and Bi-Real Net [28] improve upon the accuracy of BNNs with negligible extra computational cost, and a large accuracy gap remains between them and their full-precision counterparts (e.g., greater than 10%); whereas Group-Net [29] and MeliusNet [30] achieve comparable accuracy to that of full-precision networks, they introduce a noticeable additional computational cost (e.g., \(\sim 4x\)), which significantly offsets the advantages of network binarization. Therefore, one of the motivations in this work is to strike a better tradeoff between accuracy and computational complexity for BNNs.

In addition, the performance degradation of BNNs is mainly caused by their limited representational capability. BNNs represent weights and activations with 1-bit, which means the theoretical representation precision is only \(1/2^{31}\) with respect to the full-precision counterparts. The limited representational capability, i.e., model capacity, of BNNs may lead to two drawbacks: the mismatch between the amount of input information and model capacity, and severe information loss during the forward propagation. Figure 1 visualizes the intermediate feature maps of a BNN and its full-precision counterpart utilizing the attention map method [26]. As shown at level 4 (i.e., the last feature maps of the model), the full-precision network (FP) with higher model capacity can focus on much larger information regions of interest (the highlighted regions of the feature maps), covering the foreground information (the fish) that is relevant to the classification and part of the redundant information (the regions beyond the fish), whereas the BNN with lower model capacity is scattered to focus on smaller portions of both relevant and redundant information, which is only able to extract very limited information; in addition, the information loss incurred during the forward propagation of the BNN is also evident in the flow of the feature maps from low to high levels. IR-Net [31] and BBG [32] reduce the information loss in the forward propagation by balancing and normalizing weights to achieve maximum information entropy, which improves...
the network accuracy to some extent. However, these methods do not consider the mismatch between the amount of input information and model capacity, and they exhibit significant accuracy degradation on large-scale datasets (e.g., ImageNet).

To solve the aforementioned problems, from the perspective of the representational capability of BNNs themselves, we propose IR^2Net, a binarization approach for enhancing BNNs by restricting input information and recovering feature information: (1) intuitively, different human individuals have different learning abilities, for those with strong learning abilities, more information can be provided for their learning and refining, whereas for those with weak learning abilities, redundant information must be discarded to achieve better learning, and this is also true for networks. IR^2Net introduces the information restriction method to restrict the input information and regularize the networks, thus forcing BNNs to focus on the more critical information with their limited representational capability, and matching the amount of information input to the model capacity; (2) for information loss during the forward propagation in BNNs, IR^2Net leverages the information recovery method to fuse the shallow feature information with the final feature information before the classifier (or other task-specific modules) to fix the information loss and further enhance the model capacity to match the amount of input information.

With the abovementioned designs, the proposed IR^2Net can effectively force BNNs to focus on important information, defend against information loss in the forward propagation, and then achieve advanced performance and a good tradeoff between accuracy and efficiency for various networks and datasets.

The main contributions can be summarized as follows.

1. We propose IR^2Net, the first improving BNNs from the perspective of the mismatch between model capacity and input information quantity caused by network binarization.
2. An information restriction method is designed to restrict input information by the generated attention masks so that the amount of the input information matches the model capacity, and then the representational capability of the network is fully utilized without introducing additional cost.
3. An information recovery method is proposed to resist the information loss in the forward propagation and further mitigate the mismatch problem by fusing shallow and deep information; a compact information recovery method is also proposed to reduce the computational cost of the information recovery method and empower the network to tradeoff accuracy and efficiency.
4. Extensive experimental evaluations demonstrate that the proposed IR^2Net achieves new state-of-the-art performance on both CIFAR-10 and ImageNet and also has good versatility.

The rest of this paper is organized as follows. In Sect. 2, we discuss the related work on network binarization and efficient architecture design. In Sect. 3, we describe the preliminaries of BNNs. Then, we present the motivation, design methodology, and training process in Sect. 4. Experimental results and analysis of our approach are reported in Sect. 5, and the conclusion and future work are given in Sect. 6.

2 Related work

2.1 Network binarization

The pioneering study of network binarization dates back to BNN [33], which obtains competitive accuracy on small datasets (including MNIST, SVHN [34], and CIFAR-10 [35]), yet encounters severe performance degradation while on large-scale datasets (e.g., ImageNet [36]). Therefore, substantial research efforts have focused on minimizing the accuracy gap between BNNs and full-precision networks. The enhancement of BNNs usually requires the introduction of additional computational effort. Some works focus on using a fractional amount of real-valued operations in exchange for significant accuracy gains. For instance, XNOR-Net [27] improves the performance of BNNs on ImageNet to some extent by introducing real-valued scaling factors. XNOR-Net++ [37] makes further improvements by fusing the separated weight and activation scaling factors into one, which is learned discriminatively via backpropagation. Bi-Real Net [28] connects the real-valued activation of adjacent layers to enhance the network representational capability. Real-to-Bin [38] obtains the activation scaling factors via SE [39]. RBNN [40] further reduces the quantization error from the perspective of intrinsic angular bias. Whereas some other works relax the constraints on the additional computational complexity for higher accuracy. ABC-Net [41] uses a linear combination of multiple binary bases to approximate real-valued weights and activations. Least Squares [42] reduces the residual between real-valued activations and binary activations by utilizing a high-order approximation scheme. CBCN [43] enhances the diversity of intermediate feature maps by rotating the weight matrix. MeliusNet [30] implements a Dense Block and Improvement Block to improve the feature capability and quality, respectively. Group-Net [29] and BENN [44] use multiple BNNs for combination or ensemble to obtain significant
improvement. Although great progress has been made, existing methods mainly focus on fitting full-precision networks to reduce quantization errors and suffer from a tradeoff between accuracy and efficiency. Whereas in this paper, we improve the network performance by starting from the problem of mismatch between model capacity and the amount of input information in BNNs, while drawing on the idea of efficient architecture design to empower the networks to tradeoff accuracy and efficiency [31, 32] are the closest works to ours, the main difference being that they reduce information loss by balancing and normalizing weights to maximize information entropy, whereas our proposed information restriction and information recovery not only reduce information loss but further alleviate the mismatch between model capacity and the amount of input information caused by network binarization, and strike a better tradeoff between accuracy and efficiency.

2.2 Efficient architecture design

The main point of this line is to design compact architecture for model compression and acceleration. AlexNet [1] introduces group convolution to overcome the GPU memory constraints by partitioning input feature channels into mutually exclusive groups for convolution independently. However, group operation blocks the information interaction among groups, so ShuffleNet [15] introduces channel shuffle operation on top of group convolution to restore the connections among groups. IGC [45] uses two successive interleaved group convolutions to achieve complementarity. Xception [46] implements a depth-separable convolution that factorizes a standard convolution into depthwise convolution and pointwise convolution. MobileNet [14] uses depthwise-separable convolution to lighten the network. Based on the similarity between feature maps, GhostNet [16] introduces the ghost module to replace the conventional convolution to build compact neural networks. Both efficient architecture design and network binarization are aimed at compressing the network and accelerating the inference, and the two directions of work are orthogonal. Although we focus on network binarization in this paper, we are inspired by efficient architecture design to further lighten the proposed information recovery method and implement a compact information recovery method that empowers BNNs to tradeoff accuracy and efficiency while reducing the computational cost.

3 Preliminaries

In full-precision convolutional neural networks, the basic operation can be formalized as:

$$z = \omega_r \otimes A_r$$

where $\omega_r$ indicates the real-valued weights, $A_r$ is the real-valued input activations, and $\otimes$ the real-valued convolution. During the inference, the real-valued convolution operation contains a large number of floating-point operations and is computationally intensive. Network binarization aims to represent weights and activations with only 1-bit. By constraining weights and activations to $\{-1, +1\}$, the convolution operation can be implemented using efficient XNOR and bitcount, which is given as follows:

$$\omega_b = \text{sign}(\omega_r), \quad A_b = \text{sign}(A_r)$$

$$z = \omega_b \oplus A_b$$

where $\omega_b$ and $A_b$ denote the binary weights and input activations, respectively, and $\oplus$ the binary convolution. $\text{sign}(\cdot)$ is the binarization function, which is used to convert real-valued weights and activations into binary ones, and the function takes the form as:

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

Usually, binarization causes performance degradation and most methods [27, 28, 37, 38, 40, 47] introduce real-valued scaling factors to reduce quantization error and the binary convolution operation is replaced as:

$$z = x\beta(\omega_b \oplus A_b)$$

where $\alpha$ and $\beta$ are the scaling factors for the weights and activations, respectively (which may not be used simultaneously). Unlike these methods, in this paper, considering the limited representational capability of BNNs, we optimize BNNs via information restriction and information recovery, so that the scaling factors can be safely removed (although they could also be retained for compatibility with existing optimization methods).

4 Method

In this section, we present the proposed Information Restriction and Information Recovery Network (IR2Net) for network binarization. An overview of the proposed IR2Net is illustrated in Fig. 2. IR2Net is composed of two methodologies, information restriction and information recovery, for matching the model capacity and resisting the information loss. Specifically, the information restriction method evaluates the model capacity based on the output feature maps of the penultimate layer, analyzes the extracted information that the network can acquire from current input, and discards some information in each sample that it cannot pay attention to, thereby reducing the
amount of input information to match the limited model capacity of BNNs; while the information recovery method takes the penultimate layer outputs as the primary information, re-extracts the shallow feature maps as the supplementary information, and counteracts the information loss during the propagation by fusing the primary information with the re-extracted supplementary information, thus further enhancing the model capacity to match the large amount of input information.

Before detailing our IR²Net, we draw a distinction between the input information, the information in 'Information Restriction', and the information in 'Information Recovery'. The input information is the total information contained in the input, including the relevant information and the redundant information. The relevant information is defined as [48], i.e., the information of the input relevant to the task (e.g., classification), and the rest is the redundant information. The goal of a model is to extract the relevant information from input for a specific task. Therefore, The information in 'Information Restriction' is the redundant information of the input, and The information in 'Information Recovery' the relevant information. IR²Net is inspired by the intuitive idea that the model capacity needs to match the amount of input information. As shown in Fig. 3, $I_D$ denotes the information contained in an image $D$, $I_F$ is the relevant information about the object that needs to be classified, and $L_N$ the model capacity. On the one hand, if the model capacity of a network $L_N \geq I_F$, the network is theoretically capable of accurately classifying $D$; whereas if $L_N < I_F$, the amount of the relevant information exceeds the model capacity of the network, it can only classify correctly with a certain probability. On the other hand, the goal of a network is to extract the relevant information from input. However, given an image $D$ and a network with limited model capacity $L_N < I_D$, the information extracted by the network may contain both relevant and redundant information in practice, as mentioned in Fig. 1. Intuitively, the more relevant information extracted (the regions where the blue/red circle overlaps with the pink one in Fig. 3), the more beneficial it is for a specific task, and vice versa. In addition, the larger the gap between the model capacity and the amount of input information, the larger the non-overlapping regions are likely to be. The details of IR²Net to address these issues are elaborated on below.

4.1 Information restriction

Generally, the model capacity of a BNN is limited with respect to that of the full-precision counterpart. As shown in Fig. 1, the information regions extracted by the BNN are much smaller than that of the full-precision network. Motivated by this, we propose the Information Restriction method (IRes), which attempts to reduce the amount of input information to match the limited model capacity by restricting the redundant information. When the redundant information is removed, the information extracted by a network is restricted to the relevant information regions, so that more relevant information can be extracted for networks with certain model capacity, thus improving network performance, i.e., transferring the state of 'Example 1' to that of 'Example 2' in Fig. 3.

The spatial location of the relevant information varies from image to image, so static information restriction or manual annotation is not appropriate. Usually, in CNNs, a
network uses the stack of convolutional blocks as a feature extractor to extract the feature information of the input image, and the last linear layer as a classifier to classify the input image with the extracted information to accomplish the classification. Therefore, we analyze the information extracted by the network based on the outputs of the extractor. Specifically, as shown in Fig. 4, we use the attention map $F_A$ generated on the basis of the output feature maps from the penultimate layer $A_I$ as the relevant information extracted by the network:

$$F_A = \Psi_{\text{attention}}(A_I)$$

$$\Psi_{\text{attention}}(\cdot) = \sum_{i=1}^{C} | \cdot |^2$$

The generated attention map first performs bilinear upsampling to make its spatial dimension the same as the input image:

$$F'_A = \text{Upsample}(F_A)$$

The value of each element in the attention map represents the relevance level of that pixel in the input image for the task. By setting a threshold $\tau$, we put the value of the elements with lower relevance levels to 0 and the higher...
ones to 1 to generate an attention mask $F_m$ that masks the input image $D$ to achieve information restriction, as follows:

$$F_m = \Psi_{\text{Threshold}}(F_A')$$
$$D_m = F_m \odot D$$

(7)

where $\odot$ denotes Hadamard product, $D_m$ is the masked image. $\Psi_{\text{Threshold}}$ is used to generate the mask matrices, expressed as:

$$\Psi_{\text{Threshold}}(x) = \begin{cases} 
1, & x \geq \tau \\
0, & \text{otherwise}
\end{cases}$$

(8)

It is worth noting that since the input data is variable, the range of values of the generated attention maps also varies. Therefore, the product of the mean value of the attention map and the hyperparameter $\lambda \in [0, 1]$ is used as a threshold to avoid it being out of a reasonable range:

$$\tau = \lambda \times \text{Mean}(F_A')$$

(9)

In addition, since the generation of the attention mask requires prior knowledge, and obtaining the knowledge introduces extra computational cost, thus the information restriction method is only performed in the training phase. The original image is fed into the network first to obtain $\text{Loss}_{\text{original}}$ and an attention mask, and then the attention mask and the original image are used to generate the masked image which is fed into the network again to evaluate $\text{Loss}_{\text{masked}}$. $\text{Loss}_{\text{masked}}$ is used as a regularization term to merge with $\text{Loss}_{\text{original}}$ to obtain $\text{Loss}_{\text{total}}$ for backpropagation, thus forcing the network to focus on the relevant information within its limited model capacity (i.e., improving the overlap of the regions between the extracted information and the relevant information as in Fig. 3) without any negative impact on the model inference delay. The final loss function is defined as:

$$\text{Loss}_{\text{total}} = \mu \text{Loss}_{\text{original}} + (1 - \mu) \text{Loss}_{\text{masked}}$$

(10)

where $\mu \in [0, 1]$ is a tradeoff coefficient to balance the two losses, which is set to $\mu = 0.5$ in all experiments of this paper. The specific workflow of the information restriction method is summarized in Algorithm 1.

---

**Algorithm 1** The workflow of IRes.

**Input:** image dataset $S$, initial network $N$.

**Output:** the trained binary network $N_B$.

**Training:**

Split the dataset $S$ into mini-batch $\{b_1, b_2, ..., b_n\}$.

Calculate the original loss $\text{Loss}_{\text{original}}$:

$$\begin{align*}
\text{output}_i, A_i & = N(b_i) \\
\text{Loss}_{\text{original}} & = \text{criterion}(\text{output}_i, \text{target})
\end{align*}$$

Calculate $\text{Loss}_{\text{masked}}$:

According to equations (5-7), each image in $b_i$ is masked to generate $b_i^m$:

$$\begin{align*}
\text{output}_i & = N(b_i^m) \\
\text{Loss}_{\text{masked}} & = \text{criterion}(\text{output}_i, \text{target})
\end{align*}$$

Calculate $\text{Loss}_{\text{total}}$ according to equation (10).

Perform backpropagation based on $\text{Loss}_{\text{total}}$ and update the network:

$$N_B = \text{Update}(N)$$

Repeat the above process until the training is finished and return $N_B$.

**Inference**

Calculate the output based on the input image $D$:

$$\text{output} = N_B(D)$$
4.2 Information recovery

The information restriction method reduces the total amount of the input information by restricting the redundant information of the input to maximize the matching with the model capacity. Meanwhile, the network is trained on the masked input with less redundant information, which improves the overlap between the extracted information and the relevant information, thus improving the network performance. However, when the model capacity of the network is exceedingly limited, the relevant information extracted by the network may not be sufficient to perform classification (or other tasks) effectively even if the regions overlap highly (e.g., the right part of the ‘Example 2’ in Fig. 3); in addition, the network suffers from severe information loss in the forward propagation when binarized, which also impairs the accuracy. Therefore, improving the model capacity of the network and fixing its information loss is essential. Motivated by this, the Information Recovery method (IRec) is proposed to compensate for the relevant information extracted by the network but lost during the forward propagation by aggregating multi-level feature information, thus improving the final extracted information of the network to resist the information loss, and augmenting the model capacity to match the amount of input information (e.g., the left part of the ‘Example 2’ in Fig. 3). The details of the information recovery method are presented in Fig. 5.

The output feature maps of the penultimate layer $A_l$ are used as the primary information $F_{last} = A_l \in R^{C_l \times H_l \times W_l}$, and the shallow feature maps as the supplementary information to compensate for the relevant information lost during the forward propagation. Since selecting overmuch shallow information will introduce a large amount of computational cost, only the output feature maps of some of the layers (as shown in Fig. 2) are picked as follows: (1) the output feature maps of the first convolutional layer. Existing binarization methods usually keep the first layer as real-valued, so the output feature maps of the first layer can retain more information; (2) the output feature maps of the convolutional layer before downsampling layers. Widely used network architectures usually contain only a small number of downsampling layers, which are selected to ensure the introduction of less computational complexity while avoiding the information loss caused by downsampling.

Additionally, the selected shallow feature information $F_i \in R^{G_i \times H_i \times W_i}, i \in [1, 2, ...]$ is the output feature maps of different layers with varying spatial dimensions. The information recovery method uses adaptive pooling to make the spatial dimensions of $F_i$ and $F_{last}$ consistent, i.e.,

$$F_i = \text{Adaptive AvgPool}(F_i), \ i \in [1, 2, ...]$$

$$F_i \in R^{G_i \times H_i \times W_i}$$

(11)

After concatenating the shallow information corrected for spatial dimension $F_i$ with $F_{last}$, the information is fused using $1 \times 1$ convolution for information recovery. The concatenation operation is defined as:

$$F_{cat} = \text{Concatenate}(F_i', F_2', ..., F_{last})$$

$$F_{cat} \in R^{\sum C_i \times H_i \times W_i}$$

(12)

and the fusion operation is as:

$$F_{fused} = \text{NonLinear} (\text{BN}(\text{Conv1} \times 1(F_{cat})))$$

$$F_{fused} \in R^{C_{fused} \times H_{fused} \times W_{fused}}$$

(13)

where $F_{fused}$ is the final fused information, $\text{BN}(\cdot)$ the batch normalization, and $\text{NonLinear}(\cdot)$ the nonlinear activation function (e.g., Hardtanh, PReLU, etc.). Notably, the dimensionality of $F_{fused}$ is the same as that of $A_l$, so there is no need to adjust the subsequent modules of the network.

4.3 Compact information recovery

The information recovery method compensates for the relevant information lost in the forward propagation process by re-extracting the shallow feature information, thus enriching the final information extracted by the network and enhancing the model capacity for further matching with the amount of input information. However, its use of $1 \times 1$ convolution induces a fair amount of computational complexity. To alleviate this problem, inspired by [1, 15, 39], we modify the IRes and propose the redesigned Compact Information Recovery method (CIRec), which reduces the computational cost by group convolution and dimensionality reduction, and then the number of groups and the ratio of dimensionality reduction can be adjusted on demand to tradeoff accuracy and efficiency. The compact information recovery method can be regarded as a generalized version of the information recovery method, and the details are illustrated in Fig. 6.

The $1 \times 1$ convolution in the information recovery method can achieve effective fusion of the feature information but with a considerable computational cost. Group convolution [1] may significantly reduce the computational complexity, but the group operation hinders the information interaction between groups, which defeats the original purpose of information fusion. Channel shuffle [15] enables effective recovery of information interactions between groups though, empirical study shows that the convolution operation can achieve better fusion. Therefore, the compact information recovery method replaces the channel shuffle with $1 \times 1$ convolution and uses two
convolutions to form a bottleneck similar to SE [39]. The first $1 \times 1$ convolution is used for channel information interaction and dimensionality reduction:

$$F_{\text{channel}} = \text{NonLinear}(BN(Conv1 \times 1(F_{\text{cat}})))$$

$$F_{\text{channel}} \in R^{H_s \times W_s}$$

(14)

where $r$ is the reduction ratio, and the second $3 \times 3$ group convolution for spatial information interaction and dimensionality reconstruction:

$$F_{\text{spatial}} = \text{NonLinear}(BN(\text{Group Conv3} \times 3(F_{\text{channel}}, g)))$$

$$F_{\text{spatial}} \in R^{(C_c - \frac{r}{2})\times H_s \times W_s}$$

(15)

where $g$ denotes the number of groups. $r$ and $g$ are employed to jointly adjust the computational complexity, with $r$ for coarse tuning and $g$ for fine-tuning. Notably, to further save the computational cost, the compact information recovery method does not take the output of the second convolution as the final output, but obtains the fused information by concatenating the outputs of the two convolutions [16]:

$$F_{\text{fused}} = \text{Concatenate}(F_{\text{channel}}, F_{\text{spatial}})$$

$$F_{\text{fused}} \in R^{C_c \times H_s \times W_s}$$

(16)

The details of the computational complexity will be analyzed in Sect. 5.

5 Experiments

To evaluate the proposed methods, we carry out comprehensive experiments on the benchmarks CIFAR-10 [35] and ImageNet [36], using VGG-Small [49], ResNet-20, and ResNet-18 [8] as network backbones, respectively. Experimental results demonstrate the superiority of IR$^2$Net. In the following, the basic setup of the experiments is stated first, including an introduction to the datasets and a description of the implementation details; and then, a series of ablation experiments are conducted on CIFAR-10; finally, a comparison of our solution with some state-of-the-arts is presented in terms of performance and complexity.

5.1 Experimental setting

1. Datasets
   CIFAR-10: The CIFAR-10 dataset consists of 60,000 $32 \times 32$ images divided into 10 categories, 50,000 of which are the training set and the remaining 10,000 are the test set.
   ImageNet: Compared to CIFAR-10, ImageNet is more challenging because of its larger size and more diverse categories. There are several versions of this dataset, of which the widely used version ILSVRC12 is adopted in this paper. ILSVRC12 is divided into 1000 categories and contains about 1.2 million training images and 50,000 test images.

2. Implementation Details
   The proposed methods perform in an end-to-end manner so that all existing training schemes for BNNs are applicable theoretically. Among the experiments, IR$^2$Net is implemented with Pytorch with the following setup.
   Network structure: VGG-Small, ResNet-20, and ResNet-18 are employed as backbones on CIFAR-10, and ResNet-18 on ImageNet, respectively. Consistent with other binarization methods, all convolutional and fully-connected layers are binarized except for the first and last one of the network; for the activation function,hardtanh is chosen when on the CIFAR-10 dataset [31], and PReLU is used while on ImageNet [38, 47].
   Training strategy: Since the sign function is not differentiable, Straight-Through Estimator (STE) [50] or its variants [25, 28] are required, and the gradient approximation of Bi-Real Net [28] is employed in this paper. For the training method, our IR$^2$Net is trained from scratch on CIFAR-10 without leveraging any pre-trained model; whereas on ImageNet, following [38, 47], the two-step training method of [51] is adopted. We mostly follow their original papers for the rest settings, if not otherwise specified.
   Complexity measurement: We measure the computational complexity of the methods with the number of operations, which is calculated in line with Real-to-Bin [38]. In addition, following ReActNet [47], we count the binary operations (BOPs) and floating-point
5.2 Ablation study

To investigate the effectiveness of the components in the proposed IR²Net, we perform ablation studies on CIFAR-10. In all these experiments, ResNet-20 with Bi-Real Net [28] structure is used as the backbone and trained from scratch.

5.2.1 Effect of information restriction and information recovery

Table 1 shows the performance of each component (W/A represents the number of bits used in weight/activation quantization). As seen in the table, both the information restriction and information recovery methods work well independently and significantly improve the accuracy. Specifically, a 1% absolute accuracy gain is obtained with the information restriction method with respect to the baseline, whereas even a 2.3% increase is achieved with the information recovery method. The possible reason for the difference in the effectiveness of the two methods is that the information restriction method is mainly used to reduce the amount of the input information to match the model capacity so that the regions between the extracted information and the relevant information are aligned, whereas the information recovery method straightly strengthens the model capacity of the network for further matching with the amount of the input information and resists the information loss in the forward propagation. However, although the information recovery method significantly improves the accuracy, it introduces a high computational cost, which can be mitigated by using the compact information recovery method instead, which balances the accuracy and efficiency by adjusting the hyperparameters $r$ and $g$. Table 1 uses the setting $r = 4$ and $g = C_I$, with $C_I$ denoting the number of input channels for the group convolution. $r$ and $g$ are strategically chosen as described in Sect. 5.3. Finally, IR²Net achieves a 2% accuracy increase with respect to the baseline using the combination of the information restriction method and the compact information recovery method, indicating that the effects of the two components can be superimposed.

5.2.2 Impact of hyperparameter $\lambda$

IR²Net introduces three hyperparameters, of which $r$ and $g$ are mainly used to tradeoff accuracy and efficiency on demand. In contrast, the hyperparameter $\lambda$ introduced in equation (9) is used to control the ratio of the information restriction, i.e., if $\lambda = 0$, it means that no information restriction is used; while the larger $\lambda$ is, the higher the restriction ratio is. Therefore, we study the impact of $\lambda$ with various values on the network accuracy, and the experimental results are plotted in Fig. 7. As seen in the figure, on the one hand, when $\lambda$ is small, the accuracy is improved and with less fluctuation, compared with not using information restriction; while $\lambda$ is too large (e.g., $\lambda = 1$), the accuracy decreases significantly. This indicates that the method is robust to $\lambda$ to a certain extent, but when $\lambda$ is exceedingly large, it impairs the training of the network due to too much restriction instead. On the other hand, when $\lambda \in [0.15, 0.75]$, a larger $\lambda$ can obtain better accuracy by using only the information restriction method, whereas the opposite is true when using both the information restriction and information recovery methods. This suggests that when the information recovery method is not used, the network is with less representational capability and requires a higher information restriction ratio to reduce the amount of input information to match the model capacity; whereas the information recovery method is used, the model capacity of the network has been enhanced to further match a larger amount of input information, thus

| Method       | Bit-width (W/A) | Accuracy (%) |
|--------------|-----------------|--------------|
| FP           | 32/32           | 90.8         |
| Baseline     | 1/1             | 85.2         |
| IRes         | 1/1             | 86.2         |
| IRec         | 1/1             | 87.5         |
| CIRec        | 1/1             | 86.9         |
| IR²Net (IRes + CIRec) | 1/1 | 87.2 |
requiring only a lower information restriction rate, which verifies the mismatch between the model capacity and the amount of input information as mentioned previously. In particular, based on the analysis of \( \lambda \) with different values, we safely set \( \lambda = 0.15 \) in all experiments in this paper, if not stated otherwise.

### 5.3 Comparison with state-of-the-art methods

We further compare the proposed IR\(^2\)Net with existing state-of-the-art methods on CIFAR-10 and ImageNet, respectively, to comprehensively evaluate the performance of IR\(^2\)Net.

**CIFAR-10:** On the CIFAR-10 dataset, we compare the performance of existing binarization methods with that of IR\(^2\)Net using VGG-Small, ResNet-20, and ResNet-18 as backbones, respectively. Noticeably, given that most existing methods use real-valued scaling factors, the FLOPs introduced by are:

\[
Q_{\text{scale}} = \sum_{i=2}^{l_N-1} C_i \times H_i \times W_i
\]

where \( l_N \) denotes the number of network layers, \( i \in [2, l_N - 1] \), the first and last real-valued layers are excluded; \( C_i, H_i, \) and \( W_i \) indicate the output channels, height, and width of the \( i \)-th layer, respectively. And the additional FLOPs of IR\(^2\)Net are the sum of the computational cost of the two convolutions in Fig. 6 (the FLOPs introduced by the information restriction method are zero during the inference):

\[
Q_{\text{CIRec}} = \frac{C_n}{r} \times C_{in} \times H_n \times W_n \times K_1 \times K_1 + \left( \frac{C_n}{g} - \frac{C_n}{r} \right) \times \frac{C_n}{g} \times H_n \times W_n \times K_2 \times K_2
\]

where \( C_{in}, C_n, H_n, \) and \( W_n \) denote the input channels, output channels, height, and width of the compact information recovery method, respectively, and \( K_1, K_2 \) the convolution kernel size. To keep IR\(^2\)Net less computational cost, ensure \( Q_{\text{CIRec}} \leq Q_{\text{scale}} \) by adjusting \( r \) and \( g \), the settings are given in Table 2. And the experimental results are listed in Table 3, it shows that our method obtains the best accuracy on all three network backbones with large margins with respect to existing methods. Particularly, over VGG-Small, our IR\(^2\)Net even narrows the accuracy gap between the binary model and its full-precision counterpart to 0.2%.

**ImageNet:** We further investigate the performance of IR\(^2\)Net on ImageNet. Similar to most methods, we conduct experiments with the ResNet-18 backbone for a fair comparison. Table 4 presents the results (the number times 1/1 indicates the multiplicative factor), where -A/B/C/D indicate different combinations of \( r \) and \( g \) for trading off accuracy and efficiency, the details of which are provided in Table 5. As seen in Table 4, even IR\(^2\)Net-C outperforms the other existing methods already, while IR\(^2\)Net-A obtains comparable accuracy to that of the full-precision counterpart, closing the gap to 1.1%.

**Visualization:** In addition, to verify the effect of IR\(^2\)Net on matching the model capacity with the information quantity for BNNs and resisting the information loss, we visualize the intermediate feature maps of IR\(^2\)Net. As shown in Fig. 8, the information (highlighted part in each figure) that IR\(^2\)Net equipped with the CIRec can extract is significantly improved with respect to the BNN; while with using the IRes, the attention of IR\(^2\)Net is more focused on the relevant information in comparison with the full-precision network; also, due to the different hyperparameter settings, which result in a gap in resisting the information loss, there are subtle differences in regions of the extracted information between IR\(^2\)Net-A and IR\(^2\)Net-C.

### 5.4 Complexity analysis

Table 6 shows the computational cost of different binarization methods, where the ‘OPs gap’ column and ‘Accuracy gap’ column indicate the gap of OPs and Top-1 accuracy between existing methods and ours, respectively. The computational cost of IR\(^2\)Net-D is slightly higher than that of BNN and XNOR-Net, but there is a huge gap in accuracy. Whereas for the other methods, IR\(^2\)Net can achieve significant accuracy gains with less computational cost. In particular, IR\(^2\)Net-A obtains comparable accuracy to that of the full-precision one with \( \sim 10x \) computational cost reduction.
In addition, we compare the time and space complexity between the proposed method and existing state-of-the-art methods, as shown in Table 7, where the “FP” row is the time and space complexity of the original full-precision network. Consistent with existing studies, we use memory usage as a measure of space complexity, the number of operations as a measure of time complexity, and accuracy drop as the gap of Top-1 accuracy between the method and the original full-precision network. Since evaluating the performance of a binarization method requires considering both its compression effect (time/space complexity) and classification accuracy, the comparison of different binarization methods requires comparing the time/space complexity when the accuracy is comparable or comparing the accuracy when the time/space complexity is comparable. As can be seen from the table, the proposed IR²Net has higher accuracy than the others when the time/space complexity is comparable, i.e., IR²Net-D vs. Bi-Real Net, IR²Net-C vs. Real-to-Bin, ReActNet, and IR²Net-B vs. MeliusNet29/2, MeliusNet29; while IR²Net-A still reduces the accuracy gap between the BNN and the corresponding full-precision network to 1.1% when the time/space complexity is reduced by more than 8×.

Table 2 Settings of hyperparameters $r$ and $g$ on CIFAR-10 ($C_I$ is the number of input channels for the group convolution)

| Backbone     | Method        | $r$  | $g$  |
|--------------|---------------|------|------|
| VGG-Small    | FP            | 32   | $C_I$|
|              | XNOR-Net [27] | 1/1  | 89.8 |
|              | BNN [33]      | 1/1  | 89.9 |
|              | BNN-DL [52]   | 1/1  | 90.0 |
|              | IR-Net [31]   | 1/1  | 90.4 |
|              | BinaryDuo [53]| 1/1  | 90.4 |
|              | IR²Net        | 1/1  | 91.5 |
| ResNet-20    | FP            | 32/32| 90.8 |
|              | DSQ [25]      | 1/1  | 84.1 |
|              | IR-Net [31]   | 1/1  | 85.4 |
|              | IR²Net        | 1/1  | 86.3 |
|              | IR²Net+ [31]  | 1/1  | 86.5 |
|              | IR²Net+* [31] | 1/1  | 87.2 |
| ResNet-18    | FP            | 32/32| 93.0 |
|              | BNN-DL [52]   | 1/1  | 90.5 |
|              | IR-Net [31]   | 1/1  | 91.5 |
|              | RBNN [40]     | 1/1  | 92.2 |
|              | IR²Net        | 1/1  | 92.5 |

Table 3 Accuracy comparison between different methods on CIFAR-10 (* indicates the use of Bi-Real Net structure)

| Backbone     | Method        | Bit-width (W/A) | Accuracy (%) |
|--------------|---------------|-----------------|--------------|
| VGG-Small    | FP            | 32/32           | 91.7         |
|              | XNOR-Net [27] | 1/1             | 89.8         |
|              | BNN [33]      | 1/1             | 89.9         |
|              | BNN-DL [52]   | 1/1             | 90.0         |
|              | IR-Net [31]   | 1/1             | 90.4         |
|              | BinaryDuo [53]| 1/1             | 90.4         |
|              | IR²Net        | 1/1             | 91.5         |
| ResNet-20    | FP            | 32/32           | 90.8         |
|              | DSQ [25]      | 1/1             | 84.1         |
|              | IR-Net [31]   | 1/1             | 85.4         |
|              | IR²Net        | 1/1             | 86.3         |
|              | IR²Net+ [31]  | 1/1             | 86.5         |
|              | IR²Net+* [31] | 1/1             | 87.2         |
| ResNet-18    | FP            | 32/32           | 93.0         |
|              | BNN-DL [52]   | 1/1             | 90.5         |
|              | IR-Net [31]   | 1/1             | 91.5         |
|              | RBNN [40]     | 1/1             | 92.2         |
|              | IR²Net        | 1/1             | 92.5         |

Table 4 Accuracy comparison between different methods on ImageNet (if not specified, ResNet-18 is used as the backbone), where IR²Net-D/C/B/A denote the IR²Net obtained using different settings of hyperparameters $r$, $g$, respectively, and the settings are shown in Table 5

| Method         | Bit-width (W/A) | Top-1 (%) | Top-5 (%) |
|----------------|-----------------|-----------|-----------|
| FP             | 32/32           | 69.3      | 89.2      |
| BNN [33]       | 1/1             | 42.2      | –         |
| XNOR-Net [27]  | 1/1             | 51.2      | 73.2      |
| Bi-Real Net [28]| 1/1             | 56.4      | 79.5      |
| XNOR-Net++ [37]| 1/1             | 57.1      | 79.9      |
| IR-Net [31]    | 1/1             | 58.1      | 80.0      |
| BBG [32]       | 1/1             | 59.4      | –         |
| CI-BCNN [54]   | 1/1             | 59.9      | 84.2      |
| BinaryDuo [53]| 1/1             | 60.9      | 82.6      |
| Real-to-Bin [38]| 1/1             | 65.4      | 86.2      |
| ReActNet [47]  | 1/1             | 65.5      | –         |
| MeliusNet29/2 [30]| 1/1      | 65.7      | –         |
| MeliusNet29 [30]| 1/1             | 65.8      | –         |
| BENN [44]      | (1/1) × 6       | 61.1      | –         |
| CBCN [43]      | (1/1) × 4       | 61.4      | 82.8      |
| ABC-Net [41]   | (1/1) × 5       | 65.0      | 85.9      |
| Group-Net [29] | (1/1) × 4       | 66.3      | 86.6      |
| IR²Net-D       | 1/1             | 63.8      | 85.5      |
| IR²Net-C       | 1/1             | 66.6      | 87.0      |
| IR²Net-B       | 1/1             | 67.0      | 87.1      |
| IR²Net-A       | 1/1             | 68.2      | 88.0      |

Table 5 Settings of hyperparameters $r$ and $g$ on ImageNet ($C_I$ denotes the number of input channels for the group convolution; when $r = 1$, the compact information recovery method backs off to the original one and $g$ is not applicable)

| Method         | $r$  | $g$  |
|----------------|------|------|
| IR²Net-A       | 1    | –    |
| IR²Net-B       | 2    | $C_I$|
| IR²Net-C       | 4    | 8    |
| IR²Net-D       | 20   | $C_I$|
6 Conclusion

In this paper, we propose IR2Net, which contains two components of the information restriction and information recovery, from the perspective of the mismatch between the model capacity and the information quantity caused by network binarization. The information restriction method motivates reducing the amount of input information to match the model capacity of the network, improving the overlap between the extracted information of the network and the relevant information of the input, and then fully

![Comparison of the resulting feature maps](image)

**Table 6** Computational complexity analysis of different methods on ImageNet (if not specified, ResNet-18 is used as the backbone)

| Method          | BOPs \((\times 10^6)\) | FLOPs \((\times 10^8)\) | OPs \((\times 10^8)\) | OPs gap \((\times 10^8)\) | Accuracy gap (%) |
|-----------------|-------------------------|-------------------------|-----------------------|--------------------------|------------------|
| IR2Net-D        | 1.68                    | 1.48                    | 1.74                  | 0                        | 0                |
| BNN [33]        | 1.70                    | 1.31                    | 1.58                  | -0.16                    | -21.6            |
| XNOR-Net [27]   | 1.70                    | 1.33                    | 1.60                  | -0.14                    | -12.6            |
| Bi-Real Net [28]| 1.68                    | 1.49                    | 1.75                  | +0.01                    | -7.4             |
| IR2Net-C        | 1.68                    | 1.55                    | 1.81                  | 0                        | 0                |
| Real-to-Bin [38]| 1.68                    | 1.56                    | 1.82                  | +0.01                    | -1.2             |
| ReActNet [47]   | 1.68                    | 1.55                    | 1.81                  | 0                        | -1.1             |
| IR2Net-B        | 1.68                    | 1.59                    | 1.85                  | 0                        | 0                |
| MeliusNet29/2 [30]| –                     | –                      | 1.96                  | +0.11                    | -1.3             |
| MeliusNet29 [30]| –                      | –                      | 2.14                  | +0.29                    | -1.2             |
| IR2Net-A        | 1.68                    | 1.70                    | 1.96                  | 0                        | 0                |
| FP              | 0                      | 18.3                    | 18.3                  | +16.34                   | +1.1             |

**Table 7** Comparison of compression rates for time and space complexity

| Method          | Memory usage | Memory saving | OPs \((\times 10^8)\) | Speedup | Accuracy drop (%) |
|-----------------|--------------|---------------|------------------------|---------|-------------------|
| Bi-Real Net [28]| 33.6 Mbit    | 11.14×        | 1.75×\(\times 10^8\)  | 10.46×  | 12.9              |
| IR2Net-D        | 30.6 Mbit    | 12.23×        | 1.74×\(\times 10^8\)  | 10.52×  | 5.5               |
| Real-to-Bin [38]| 38.5 Mbit    | 9.72×         | 1.82×\(\times 10^8\)  | 10.06×  | 3.9               |
| ReActNet [47]   | 33.4 Mbit    | 11.20×        | 1.81×\(\times 10^8\)  | 10.11×  | 3.8               |
| IR2Net-C        | 35.3 Mbit    | 10.60×        | 1.81×\(\times 10^8\)  | 10.11×  | 2.7               |
| MeliusNet29/2 [30]| 40.0 Mbit  | 9.36×         | 1.96×\(\times 10^8\)  | 9.34×   | 3.6               |
| MeliusNet29 [30]| 40.8 Mbit    | 9.17×         | 2.14×\(\times 10^8\)  | 8.55×   | 3.5               |
| IR2Net-B        | 37.9 Mbit    | 9.87×         | 1.85×\(\times 10^8\)  | 9.89×   | 2.3               |
| FP              | 374.1 Mbit   | 1.0×          | 1.83×\(\times 10^8\)  | 1.0×    | 0                 |
| IR2Net-A        | 45.6 Mbit    | 8.21×         | 1.96×\(\times 10^8\)  | 9.34×   | 1.1               |
utilizing the representation capability; the information recovery method fuses multi-level feature information to enhance the model capacity of the network for further matching with the amount of the input information and resists the information loss incurred during the forward propagation. In addition, a compact information recovery method is further devised to reduce the computational cost of the information recovery method and tradeoff accuracy and efficiency. Experiments with various network structures on CIFAR-10 and ImageNet demonstrate the superiority of our IR²Net. Furthermore, future work should focus on combining network binarization with pruning to further trading off accuracy and efficiency. Additionally, the application of these binarization methods to other fields (e.g., semantic segmentation and natural language processing) requires further investigation.

Acknowledgments This work was supported in part by the Anhui Provincial Key Research and Development Program under Grant 202004a05020040, in part by the National Key Research and Development Program under Grant 2018YFC0604404, in part by Intelligent Network and New Energy Vehicle Special Project of Intelligent Manufacturing Institute of HUTF under Grant IMFWL2019003, and in part by Fundamental Research Funds for the Central Universities under Grant PA2021GDP0061.

Data Availability All data generated or analyzed during this study are included in this article.

Declarations

Conflict of interest All authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
2. Wang Z, Lu J, Wu Z, Zhou J (2021) Learning efficient binarized object detectors with information compression. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2021.3050464
3. Tong Z, Xu P, Denœux T (2021) Evidential fully convolutional network for semantic segmentation. Appl Intel 51(9):6376–6399. https://doi.org/10.1007/s10489-021-03272-0
4. Ding Y, Ma Z, Wen S, Xie J, Chang D, Si Z, Wu M, Ling H (2021) AP-CNN: weakly supervised attention pyramid convolutional neural network for fine-grained visual classification. IEEE Trans Image Process 30:2826–2836. https://doi.org/10.1109/TIP.2021.3055617
5. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
6. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings
7. Szegedy C, Liu W, Jia Y, Sermanet P, Reed SE, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015, pp 1–9. https://doi.org/10.1109/CVPR.2015.7298594
8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
9. Denton EL, Zaremba W, Bruna J, LeCun Y, Fergus R (2014) Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in neural information processing systems 27: annual conference on neural information processing systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp 1269–1277
10. Wang P, Hu Q, Fang Z, Zhao C, Cheng J (2018) Deepsearch: a fast image search framework for mobile devices. ACM Trans Multim Comput Commn Appl 14(1):6–1622. https://doi.org/10.1145/3152127
11. Singh P, Verma VK, Rai P, Namboodiri VP (2020) Acceleration of deep convolutional neural networks using adaptive filter pruning. IEEE J Sel Top Signal Process 14(4):838–847. https://doi.org/10.1109/JSTSP.2020.2992390
12. Zhang Z, Kouzani AZ (2020) Implementation of dnns on iot devices. Neural Comput Appl 32(5):1327–1356. https://doi.org/10.1007/s00521-019-04550-w
13. Ding G, Zhang S, Jia Z, Zhong J, Han J (2021) Where to prune: using LSTM to guide data-dependent soft pruning. IEEE Trans Image Process 30:293–304. https://doi.org/10.1109/TIP.2020.3035028
14. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
15. Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: 2018 IEEE conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp 6848–6856. https://doi.org/10.1109/CVPR.2018.00716
16. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) Ghostnet: more features from cheap operations. In: 2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, pp 1577–1586. https://doi.org/10.1109/CVPR42600.2020.00165
17. Gao H, Wang Z, Cai L, Ji S (2021) Channelnets: compact and efficient convolutional neural networks via channel-wise convolutions. IEEE Trans Pattern Anal Mach Intell 43(8):2570–2581. https://doi.org/10.1109/TPAMI.2020.2975796
18. Li X, Li S, Omar B, Wu F, Li X (2021) Reskd: residual-guided knowledge distillation. IEEE Trans Image Process 30:4735–4746. https://doi.org/10.1109/TIP.2021.3066051
19. Gou J, Yu B, Maybank SJ, Tao D (2021) Knowledge distillation: a survey. Int J Comput Vis 129(6):1789–1819. https://doi.org/10.1007/s11263-021-01453-z
20. Tan C, Liu J, Zhang X (2021) Improving knowledge distillation via an expressive teacher. Knowl Based Syst 218:106837. https://doi.org/10.1016/j.knosys.2021.106837
21. Tung F, Mori G (2020) Deep neural network compression by inparallel pruning-quantization. IEEE Trans Pattern Anal Mach Intell 42(3):568–579. https://doi.org/10.1109/TPAMI.2018.2886192
22. Huang C, Liu P, Fang L (2021) MXQN: mixed quantization for reducing bit-width of weights and activations in deep convolutional neural networks. Appl Intell 51(7):4561–4574. https://doi.org/10.1007/s10489-020-02109-0
23. Sakai Y, Tamiya Y (2021) S-dfp: shifted dynamic fixed point for quantized deep neural network training. Neural Comput Appl 1–8
24. Hu S, Qiao GC, Chen TP, Yu Q, Liu Y, Rong LM (2021) Quantized stdp-based online-learning spiking neural network. Neural Comput Appl 33(19):12317–12332. https://doi.org/10.1007/s00521-021-05832-y
25. Gong R, Liu X, Jiang S, Li T, Hu P, Lin J, Yu F, Yan J (2019) Differentiable soft quantization: bridging full-precision and low-bit neural networks. In: 2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp 4851–4860. https://doi.org/10.1109/ICCV.2019.00495
26. Zagoruyko S, Komodakis N (2017) Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference track proceedings
27. Rastegari M, Ordonez V, Redmon J, Farhadi A (2016) Xnor-net: Imagenet classification using binary convolutional neural networks. In: Computer vision - ECCV 2016 - 14th European conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV, pp 525–542. https://doi.org/10.1007/978-3-319-46493-0_32
28. Liu Z, Luo W, Wu B, Yang X, Liu W, Cheng K (2020) Bi-real net: binarizing deep network towards real-network performance. Int J Comput Vis 128(1):202–219. https://doi.org/10.1007/s12663-019-01227-8
29. Zhuang B, Shen C, Tan M, Liu L, Reid ID (2019) Structured binary neural networks for accurate image classification and semantic segmentation. In: IEEE conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019, pp 413–422. https://doi.org/10.1109/CVPR.2019.00505
30. Bethge J, Bartz C, Yang H, Chen Y, Meinel C (2021) Meliusnet: an improved network architecture for binary neural networks. In: IEEE winter conference on applications of computer vision, WACV 2021, Waikoloa, HI, USA, January 3–8, 2021, pp 1438–1447. https://doi.org/10.1109/WACV48630.2021.00148
31. Qin H, Gong R, Liu X, Shen M, Wei Z, Yu F, Song J (2020) Forward and backward information retention for accurate binary neural networks. In: 2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, pp 2247–2256. https://doi.org/10.1109/CVPR42600.2020.00232
32. Shen M, Liu X, Gong R, Han K (2020) Balanced binary neural networks with gated residual. In: 2020 IEEE International conference on acoustics, speech and signal processing, ICASSP 2020, Barcelona, Spain, May 4–8, 2020, pp 4197–4201. https://doi.org/10.1109/ICASSP40776.2020.9054599
33. Hubara I, Courbariaux M, Soudry D, El-Yaniv R, Bengio Y (2016) Binarized neural networks. In: Annual conference on neural information processing systems 2016, December 5–10, 2016, Barcelona, Spain, pp 4107–4115
34. Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning
35. Torralba A, Fergus R, Freeman WT (2008) 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Trans Pattern Anal Mach Intell 30(11):1958–1970. https://doi.org/10.1109/TPAMI.2008.128
36. Deng J, Dong W, Socher R, Li L, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE computer society conference on computer vision and pattern recognition (CVPR 2009), 20–25 June 2009, Miami, Florida, USA, pp 248–255. https://doi.org/10.1109/CVPR.2009.5206848
37. Bulat A, Tzimiropoulos G (2019) Xnor-net++: improved binary neural networks. In: 30th British machine vision conference 2019, BMVC 2019, p 62 Cardiff, UK, September 9–12, 2019
38. Martínez B, Yang J, Bulat A, Tzimiropoulos G (2020) Training binary neural networks with real-to-binary convolutions. In: 8th international conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020
39. Hu J, Shen L, Albanie S, Sun G, Wu E (2020) Squeeze-and-excitation networks. IEEE Trans Pattern Anal Mach Intell 42(8):2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372
40. Lin M, Ji R, Xu Z, Zhang B, Wang Y, Wu Y, Huang F, Lin C (2020) Rotated binary neural network. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual
41. Lin X, Zhao C, Pan W (2017) Towards accurate binary convolutional neural network. In: Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA, pp 345–353
42. Pouransari H, Tu Z, Tuzel O (2020) Least squares binary quantization of neural networks. In: 2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR workshops 2020, Seattle, WA, USA, June 14–19, 2020, pp 2986–2996. https://doi.org/10.1109/CVPRW50498.2020.00357
43. Liu C, Ding W, Hu Y, Xia X, Zhang B, Liu J, Doermann D (2020) Circulant binary convolutional networks for object recognition. IEEE J Sel Top Signal Process 14(4):884–893. https://doi.org/10.1109/JSTSP.2020.2969516
44. Zhu S, Dong X, Su H (2019) Binary ensemble neural network: more bits per network or more networks per bit? In: IEEE conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019, pp 4923–4932. https://doi.org/10.1109/CVPR.2019.00506
45. Zhang T, Qi G, Xiao B, Wang J (2017) Interleaved group convolutions. In: IEEE international conference on computer vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pp 4383–4392. https://doi.org/10.1109/ICCV.2017.469
46. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: 2017 IEEE conference on computer vision and pattern recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017, pp 1800–1807. https://doi.org/10.1109/CVPR.2017.198
47. Liu Z, Shen Z, Savvides M, Cheng K (2020) Reactnet: Towards precise binary neural network with generalized activation functions. In: Computer vision - ECCV 2020 - 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV, pp 143–159. https://doi.org/10.1007/978-3-030-58568-6_9
48. Tishby N, Pereira FCN, Bialek W (2000) The information bottleneck method. arXiv:physics/0004057
49. Zhang D, Yang J, Ye D, Hua G (2018) Lq-nets: learned quantization for highly accurate and compact deep neural networks. In: Visual computing - ECCV 2018 - 15th European conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII, pp 373–390. https://doi.org/10.1007/978-3-030-01237-3_23
50. Bengio Y, Léonard N, Courville AC (2013) Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv:1308.3432
51. Bulat A, Tzimiropoulos G, Kossaifi J, Pantic M (2019) Improved training of binary networks for human pose estimation and image recognition. arXiv:1904.05868
52. Ding R, Chin T, Liu Z, Marculescu D (2019) Regularizing activation distribution for training binarized deep networks. In: IEEE conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019, pp 11408–11417. https://doi.org/10.1109/CVPR.2019.01167
53. Kim H, Kim K, Kim J, Kim J (2020) Binaryduo: reducing gradient mismatch in binary activation network by coupling binary activations. In: 8th international conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020

54. Wang Z, Lu J, Zhou J (2021) Learning channel-wise interactions for binary convolutional neural networks. IEEE Trans Pattern Anal Mach Intell 43(10):3432–3445. https://doi.org/10.1109/TPAMI.2020.2988262

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.