Network Binarization via Contrastive Learning

Yuzhang Shang[1], Dan Xu[2], Ziliang Zong[3], Liqiang Nie[4], and Yan Yan[1]⋆

[1] Illinois Institute of Technology, USA
[2] Hong Kong University of Science and Technology, Hong Kong
[3] Texas State University, USA
[4] Harbin Institute of Technology, Shenzhen, China
yshang4@hawk.iit.edu, danxu@cse.ust.hk, ziliang@txstate.edu,
nieliqiang@gmail.com, and yyan34@iit.edu

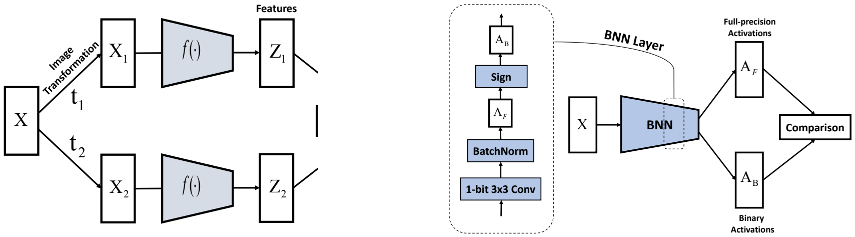
Abstract. Neural network binarization accelerates deep models by quantizing their weights and activations into 1-bit. However, there is still a huge performance gap between Binary Neural Networks (BNNs) and their full-precision (FP) counterparts. As the quantization error caused by weights binarization has been reduced in earlier works, the activations binarization becomes the major obstacle for further improvement of the accuracy. BNN characterises a unique and interesting structure, where the binary and latent FP activations exist in the same forward pass (*i.e.* $\text{Binarize}(\mathbf{a}_F) = \mathbf{a}_B$). To mitigate the information degradation caused by the binarization operation from FP to binary activations, we establish a contrastive learning framework while training BNNs through the lens of Mutual Information (MI) maximization. MI is introduced as the metric to measure the information shared between binary and the FP activations, which assists binarization with contrastive learning. Specifically, the representation ability of the BNNs is greatly strengthened via pulling the positive pairs with binary and FP activations from the same input samples, as well as pushing negative pairs from different samples (the number of negative pairs can be exponentially large). This benefits the downstream tasks, not only classification but also segmentation and depth estimation, *etc.* The experimental results show that our method can be implemented as a pile-up module on existing state-of-the-art binarization methods and can remarkably improve the performance over them on CIFAR-10/100 and ImageNet, in addition to the great generalization ability on NYUD-v2. The code is available at https://github.com/42Shawn/CMIM.

Keywords: Neural Network Compression, Network Binarization, Contrastive Learning, Mutual Information Maximization

1 Introduction

Although deep learning [27] has achieved remarkable success in various computer vision tasks such as image classification [25] and semantic image segmentation [5], its over-parametrization problem makes its computationally expensive

⋆ Corresponding author.



(a) Contrastive instance le

(b) *CMIM* for training BNN

Fig. 1. (a): In contrastive instance learning, the features produced by different transformations of the same sample are contrasted to each other. **(b)**: However BNN can yield the binary activations A_B and full-precision activations A_F (*i.e.* two transformations of an image both from the same BNN) in the same forward pass, thus the BNN can act as two image transformations in the literature of contrastive learning.

and storage excessive. To advance the development of deep learning in resource-constrained scenarios, several neural network compression paradigms have been proposed, such as network pruning [28,15], knowledge distillation [19,39] and network quantization [22]. Among the network quantization methods, the network binarization method stands out for quantizing weights and activations (*i.e.* intermediate feature maps) to ±1, compressing the full-precision counterpart 32×, and replacing time-consuming inner-product in full-precision networks with efficient xnor-bitcount operation in the BNNs [22].

However, severe accuracy drop-off always exists between full-precision models and their binary counterparts. To tackle this problem, previous works mainly focus on reducing the quantization error induced by weights binarization [38,29], and elaborately approximating binarization function to alleviate the the gradient mismatch issue in the backward propagation [31,37]. Indeed, they achieve the SoTA performance. Yet narrowing down the quantization error and enhancing the gradient transmission reach their bottlenecks [4,23], since the 1W32A (only quantizing the weights into 1-bit, remaining the activations 32-bit) models are capable of performing as well as the full-precision models [18,29], implying that the activations binarization becomes the main issue for further performance improvement.

To address the accuracy degradation caused by the activations binarization, a few studies are proposed to regulate the distributions of the binary activations, *e.g.* researchers in [10] design a distribution loss to explicitly regularize the activation flow; researchers in [23] propose to shift the thresholds of binary activation functions to unbalance the distribution of the binary activations. They heuristically design low-level patterns to analyze the distributions of binary activations, such as minimum of the activations and the balanced property of distributions. Nevertheless, they neglect the high-level indicators of the distribution and the unique characteristics of BNN, where the binary activations and latent full-precision activations co-exist in the same forward pass. Thus, we argue that the high-level properties of distributions, such as correlations and dependencies between binary and full-precision activations should be captured and utilized.

In this work, we explore introducing mutual information for BNNs, in which the mutual information acts as a metric to quantify the information amount shared by the binary and latent real-valued activations in BNNs. In contrast to the works mentioned above focusing on learning the distribution of binary activations, mutual information naturally captures statistical dependencies between variables, quantifying the degree of the dependence [11]. Based on this metric, we propose a novel method, termed as Network Binarization via **C**ontrastive Learning for **M**utual **I**nformation **M**aximization (***CMIM***). Specifically, we design a highly effective optimization strategy using contrastive estimation for mutual information maximization. As illustrated in Figure 1, we replace the data transformation module in contrastive learning with the exclusive structure in BNNs, where full-precision and binary activations are in the same forward pass. In this way, contrastive learning contributes to inter-class decorrelation of binary activations, and avoids collapse solutions. In other words, our method is built upon a contrastive learning framework to learn representative binary activations, in which we pull the binary activation closer to the full-precision activation and push the binary activation further away from other binary activations in the contrastive space. Moreover, by utilizing an additional MLP module to extract representations of activations, our method can explicitly capture higher-order dependencies in the contrastive space. To the best of our knowledge, it is the first work aiming at maximizing the mutual information of the activations in BNNs within a contrastive learning framework.

Overall, the contributions of this paper are three-fold:

- Considering the distributions of activations, we propose a novel contrastive framework to optimize BNNs, via maximizing the mutual information between the binary activation and its latent real-valued counterpart;
- We develop an effective contrastive learning strategy to achieve the goal of mutual information maximization for BNNs, and benefited from it, the representation ability of BNNs is strengthened for not only the classification task but also downstream CV tasks;
- Experimental results show that our method can significantly improve the existing SoTA methods over the classification task on CIFAR-10/100 and ImageNet, *e.g.* 6.4% on CIFAR-100 and 3.0% on ImageNet. Besides, we also demonstrate the great generalization ability of the proposed *CMIM* on other challenging CV tasks such as depth estimation and semantic segmentation.

2 Related Work

In [22], the researchers introduce the sign function to binarize weights and activations to 1-bit, initiating the studies of BNNs. In this work, the straight-through estimator (STE) [2] is utilized to approximate the derivative of the sign function. Following the seminal art, copious studies contribute to improving the performance of BNNs. For example, Rastegari *et al.* [38] disclose that the quantization error between the full-precision weights and the corresponding binarized weights is one of the major obstacles degrading the representation capabilities of BNNs. Reducing the quantization error thus becomes a fundamental research direction to improve the performance of BNNs. Researchers propose XNOR-Net [38] to introduce a scaling factor calculated by L1 norm for both weights and activation functions to minimize the quantization error. Inspired by XNOR-Net, XNOR++ [3] further learns both spatial and channel-wise scaling factors to improves the performances. Bi-Real [31] proposes double residual connections with full-precision downsampling layers to mitigate the excessive gradient vanishing issue caused by binarization. ProxyBNN [18] designs a proxy matrix as a basis of the latent parameter space to guide the alignment of the weights with different bits by recovering the smoothness of BNNs. ReActNet [32] implements binarization with MobileNet [21] instead of ResNet, and achieves SoTA performance.

Nevertheless, we argue that those methods focusing on narrowing down the quantization error and enhancing the gradient transmission reach their bottleneck (*e.g.* 1W32A ResNet-18 trained by ProxyBNN achieves 67.7% Top-1 accuracy on ImageNet, while full-precision version is only 68.5%). Because they neglect the activations in BNNs, especially the relationship between the binary and latent full-precision activations. We treat them as discrete variables and investigate them under the metric of mutual information. By maximizing the mutual information via contrastive learning, the performance of BNNs is further improved. The experimental results show that *CMIM* can consistently improve the aforementioned methods by directly adding our *CMIM* module on them.

3 Training BNNs via Contrastive Learning for Mutual Information Maximization

3.1 Preliminaries

We define a K-layer Multi-Layer Perceptron (MLP). For simplification, we discard the bias term of this MLP. Then the network $f(\mathbf{x})$ can be denoted as:

$$f(\mathbf{W}^1, \cdots, \mathbf{W}^K; \mathbf{x}) = (\mathbf{W}^K \cdot \sigma \cdot \mathbf{W}^{K-1} \cdot \cdots \cdot \sigma \cdot \mathbf{W}^1)(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is the input sample and $\mathbf{W}^k : \mathbb{R}^{d_{k-1}} \longmapsto \mathbb{R}^{d_k} (k = 1, ..., K)$ stands for the weight matrix connecting the $(k-1)$-th and the k-th layer, with d_{k-1} and d_k representing the sizes of the input and output of the k-th network layer, respectively. The $\sigma(\cdot)$ function performs element-wise activation operation on the input feature maps.

Based on those predefined notions, the sectional MLP $f^k(\mathbf{x})$ with the front k layers of the $f(\mathbf{x})$ can be represented as:

$$f^k(\mathbf{W}^1, \cdots, \mathbf{W}^k; \mathbf{x}) = (\mathbf{W}^k \cdot \sigma \cdots \sigma \cdot \mathbf{W}^1)(\mathbf{x}). \tag{2}$$

And the MLP f can be seen as a special case in the function sequence $\{f^k\}(k \in \{1, \cdots, K\})$, *i.e.* $f = f^K$.

Binary Neural Networks. Here, we review the general binarization method in [8,22], which maintains latent full-precision weights $\{\mathbf{W}_F^k\}(k \in \{1, \cdots, K\})$ for gradient updates, and the k-th weight matrix $\mathbf{W}_F^k$ is binarized into ±1, obtaining the binary weight matrix $\mathbf{W}_B^k$ by a binarize function (normally $sgn(\cdot)$), *i.e.* $\mathbf{W}_B^k = sgn(\mathbf{W}_F^k)$. Then the intermediate activation map (full-precision) of the k-th layer is produced by $\mathbf{A}_F^k = \mathbf{W}_B^k \mathbf{A}_B^{k-1}$. Finally, the same sign function is used to binarize the full-precision activations into binary activations as $\mathbf{A}_B^k = sgn(\mathbf{A}_F^k)$ (see Fig.1 b), and the whole forward pass of a BNN is performed by iterating this process for L times.

Mutual Information and Contrastive Learning. For two discrete variables $\mathbf{X}$ and $\mathbf{Y}$, their mutual information (MI) can be defined as [26]:

$$I(\mathbf{X}, \mathbf{Y}) = \sum_{x,y} P_{\mathbf{XY}}(x, y) \log \frac{P_{\mathbf{XY}}(x, y)}{P_{\mathbf{X}}(x) P_{\mathbf{Y}}(y)}, \tag{3}$$

where $P_{\mathbf{XY}}(x, y)$ is the joint distribution, $P_{\mathbf{X}}(x) = \sum_y P_{\mathbf{XY}}(x, y)$ and $P_{\mathbf{Y}}(y) = \sum_x P_{\mathbf{XY}}(x, y)$ are the marginals of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

Mutual information quantifies the amount of information obtained about one random variable by observing the other random variable. It is a dimensionless quantity with (generally) units of bits, and can be considered as the reduction in uncertainty about one random variable given knowledge of another. High mutual information indicates a large reduction in uncertainty and *vice versa* [26]. In the content of binarization, considering the binary and full-precision activations as random variables, we would like them share as much information as possible, since the binary activations are proceeded from their corresponding full-precision activations. Theoretically, the mutual information between those two variables should be maximized.

Our motivation can also be testified from the perspective of RBNN [29]. In RBNN, Lin *et al.* devise a rotation mechanism leading to around 50% weight flips which maximizes the information gain, $H(\mathbf{a}_B^{k,i})$. As MI can be written in another form as $I(\mathbf{X}, \mathbf{Y}) = H(\mathbf{X}) - I(\mathbf{X} \mid \mathbf{Y})$, the MI between binary and FP activations can be formulated as:

$$I(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}) = H(\mathbf{a}_B^{k,i}) - I(\mathbf{a}_B^{k,i} \mid \mathbf{a}_F^{k,j}), \tag{4}$$

in which maximizing the first term on the right can partially lead to maximizing the whole MI. In this work, we aim to universally maximize the targeted MI.

Recently, contrastive learning is proven to be an effective approach to MI maximization, and many methods based on contrastive loss for self-supervised learning are proposed, such as Deep InfoMax [20], Contrastive Predictive Coding [34], MemoryBank [42], Augmented Multiscale DIM [1], MoCo [16] and SimSaim [7]. These methods are generally rooted in NCE [13] and InfoNCE [20] which can serve as optimizing the lower bound of mutual information [36]. Intuitively, the key idea of contrastive learning is to pull representations in positive pairs close and push representations in negative pairs apart in a contrastive space, and thus the major obstacle for resorting to the contrastive loss is to define the negative and positive pairs.

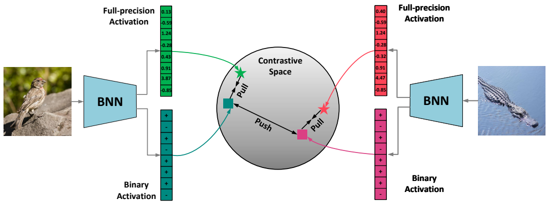


Fig. 2. Feeding two images into a BNN, and obtaining the three pairs of binary and full-precision activations. Our goal is to embed the activations into a contrastive space, then learn from the pair correlation with the contrastive learning task in Eq. 13.

3.2 Contrastive Learning for Mutual Information Maximization

In this section, we formalize the idea of constructing a contrastive loss based on Noise-Contrastive Estimation (NCE) to maximize the mutual information between the binary and the full-precision activations. Particularly, we derive a novel *CMIM* loss for training BNNs, where NCE is introduced to avoid the direct mutual information computation by estimating it with its lower bound in Eq. 9. Straightforwardly, the binary and full-precision activations from samples can be pull close, and activations from different samples can be pushed away, which corresponds to the core idea of contrastive learning.

For binary network f_B and its latent full-precision counterpart f_F in the same training iteration, the series of their activations $\{\mathbf{a}_B^k\}$ and $\{\mathbf{a}_F^k\}(k \in \{1, \cdots, K\})$, where $\mathbf{A}_B^k = (\mathbf{a}_B^{k,1}, \cdots, \mathbf{a}_B^{k,N})$ and $\mathbf{A}_F^k = (\mathbf{a}_F^{k,1}, \cdots, \mathbf{a}_F^{k,N})$ can be considered as a series of variables. The corresponding variables $(\mathbf{a}_B^k, \mathbf{a}_F^k)$ should share more information, *i.e.* the mutual information of the same layer's output activations $I(\mathbf{a}_B^k, \mathbf{a}_F^k)(k \in \{1, \cdots, K\})$ should be maximized to enforce them mutually dependent.

To this end, we introduce the contrastive learning framework into our targeted binarization task. The basic idea of contrastive learning is to compare different views of the data (usually under different data augmentations) to calculate similarity scores [34,20,1,16,7]. This framework is suitable for our case, since the binary and full-precision activations can be seen as two different views. For a training batch with N samples, the samples can be denoted as: $\{\mathbf{x}_i\}(i \in \{1, \cdots, N\})$. We feed a batch of samples to the BNN and obtain KN^2 pairs of activations $(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$, which augments the data for the auxiliary task. We define a pair containing two activations from the same sample as positive pair, *i.e.* if $i = j$, $(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})_+$ and *vice versa*. The core idea of contrastive learning is to discriminate whether a given pair of activation $(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$ is positive or negative, *i.e.*, inferring the distribution $P(D \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$, in which D is the variable decides whether $i = j$ or $i \neq j$. However, we can not directly compute the distribution $P(D \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$ [13], and we introduce its variational approximation

$$q(D \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}), \tag{5}$$

which can be calculated by our models. Intuitively, $q(D \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$ can be treated as a binary classifier, which can classify a given pair $(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$ into positive or negative.

With the Bayes' theorem, the posterior probability of two activations from the positive pair can be formalized as:

$$q(D=1 \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}) = \frac{q(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j} \mid D=1)\frac{1}{N}}{q(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j} \mid D=1)\frac{1}{N} + q(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j} \mid D=1)\frac{N-1}{N}}. \tag{6}$$

The probability of activations from negative pair is $q(D=0 \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}) = 1 - q(D=1 \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$. To simplify the NCE derivative, several works [13,42,41] build assumption about the dependence of the variables, we also use the assumption that the activations from positive pairs are dependent and the ones from negative pairs are independent, *i.e.* $q(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j} \mid D=1) = P(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$ and $q(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j} \mid D=0) = P(\mathbf{a}_B^{k,i})P(\mathbf{a}_F^{k,j})$. Hence, the above equation can be simplified as:

$$q(D=1 \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}) = \frac{P(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})}{P(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}) + P(\mathbf{a}_B^{k,i})P(\mathbf{a}_F^{k,j})(N-1)}. \tag{7}$$

Performing logarithm to Eq. 7 and arranging the terms, we can achieve

$$\log q(D=1 \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}) \leq \log \frac{P(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})}{P(\mathbf{a}_B^{k,i})P(\mathbf{a}_F^{k,j})} - \log(N-1). \tag{8}$$

Taking expectation on both sides with respect to $P(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$, and combining the definition of mutual information in Eq. 3, we can derive the form of mutual information as:

$$\overbrace{I(\mathbf{a}_B^{k}, \mathbf{a}_F^{k})}^{\text{targeted }\mathbf{MI}} \geq \overbrace{\mathbb{E}_{P(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})}\left[\log q(D=1 \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})\right]}^{\text{optimized lower bound}} + \log(N-1), \tag{9}$$

where $I(\mathbf{a}_B^{k}, \mathbf{a}_F^{k})$ is the mutual information between the binary and full-precision distributions of our targeted object. Instead of directly maximizing the mutual information, maximizing the lower bound in the Eq. 9 is a practical solution.

However, $q(D=1 \mid \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})$ is still hard to estimate. Thus, we introduce critic function h with parameter ϕ (*i.e.* $h(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}; \phi)$) as previous contrastive learning works [34,20,1,41,6]. Basically, the critic function h needs to map $\mathbf{a}_B^{k}, \mathbf{a}_F^{k}$ to $[0,1]$ (*i.e.* discriminate whether a given pair is positive or negative). In practice, we design our critic function for our BNN case based on the critic function in [41]:

$$h(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}) = \exp\left(\frac{< \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j} >}{\tau}\right)/C, \tag{10}$$

in which $C = \exp(\frac{< \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j} >}{\tau}) + N/M$, M is the number of all possible pairs, as well as τ is a temperature parameter that controls the concentration level of the distribution [19].

The activations of BNN have their properties can be used here, *i.e.*

$$sgn(\mathbf{a}_F^{k,i}) = \mathbf{a}_B^{k,i} \quad \text{and} \quad < \mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,i} >= \|\mathbf{a}_F^{k,i}\|_1 \tag{11}$$

Thus, the critic function in Eq. 10 can be further simplified as follows:

$$h(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}) = \exp(\frac{< sgn(\mathbf{a}_F^{k,i}), \mathbf{a}_F^{k,j} >}{\tau}) = \begin{cases} \exp(\frac{\|\mathbf{a}_F^{k,i}\|_1}{\tau}) & i = j, \\ \exp(\frac{< sgn(\mathbf{a}_F^{k,i}), \mathbf{a}_F^{k,j} >}{\tau}) & i \neq j \end{cases} \tag{12}$$

Critic in the view of activation flip. Eq. 12 reveals the working mechanism of *CMIM* from a perspective of activation flip. Specifically, by turning the + activation into −, binary activation in the critic can pull the activations in positive pair close and push the ones in the negative pair away via inner product. For example, suppose $\mathbf{a}_F^{k,1}$=$(0.3, -0.4, -0.6)$ and $\mathbf{a}_F^{k,2}$=$(0.6, -0.9, 0.7)$, and then $\mathbf{a}_B^{k,1}$=$(+1, -1, -1)$ is the anchor. Thus, for the positive pair, $< sgn(\mathbf{a}_B^{k,1}), \mathbf{a}_F^{k,1} >$ $=0.3\times(+1)+(-0.4)\times(-1)+(-0.6)\times(-1)$=$\|\mathbf{a}_F^{k,1}\|_1$ maximizing their similarity score; and for the negative pair, $< sgn(\mathbf{a}_B^{k,1}), \mathbf{a}_F^{k,2} >$ $=0.6\times(+1)+(-0.9)\times(-1)+$ $\overbrace{(0.6)\times(-1)}^{flipped}$ gradually minimizing the score, where the flipped term serve as a penalty for the negative pair. In this way, the binary anchor pull the positive full-precision activation close, and push the negative full-precision ones away by flipping numbers in the full-precision activations. Note that the process is iteratively operated during training, and thus all the binary activations can play the role as anchor, which eventually leads to better representation capacity in the contrastive space.

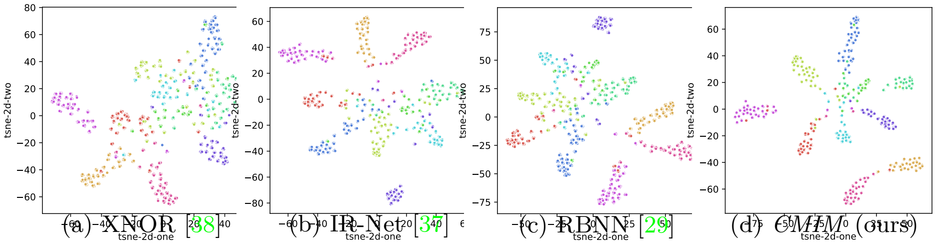
Loss Function. We define the contrastive loss function $\mathcal{L}_{NCE}^k$ between the k-th layer's activations $\mathbf{A}_B^k$ and $\mathbf{A}_F^k$ as: $\mathcal{L}_{NCE}^k =$

$$\mathbb{E}_{q(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}|D=1)}\left[\log h(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j})\right] + N\mathbb{E}_{q(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}|D=0)}\left[\log(1 - h(\mathbf{a}_B^{k,i}, \mathbf{a}_F^{k,j}))\right]. \tag{13}$$

We would comment on the above loss function from the perspective of contrastive learning. The first term of positive pairs is optimized for capturing more intra-class correlations and the second term of negative pairs is for inter-class decorrelation. Because the pair construction is instance-wise, the number of negative samples theoretically can be the size of the entire training set, *e.g.* 1.2 million for ImageNet. With those additional hand-craft designed contrastive pairs for the proxy optimization problem in Eq. 13, the representation capacity of BNNs can be further improved, as many contrastive learning methods demonstrated [7,34,20,1].

Combining the series of NCE loss from different layers $\{\mathcal{L}_{NCE}^k\}, (k = 1, \cdots, K)$, the overall loss $\mathcal{L}$ can be defined as:

$$\mathcal{L} = \lambda \sum_{k=1}^{K} \frac{\mathcal{L}_{NCE}^k}{\beta^{K-1-k}} + \mathcal{L}_{cls}, \tag{14}$$



(a) XNOR [38] (b) IR-Net [37] (c) RBNN [29] (d) *CMIM* (ours)

Fig. 3. t-SNE [33] visualization of the activations representing for random 10 classes in CIFAR-100. Every color represents a different class. We can clearly witness the improvement of our method for learning better binary representations.

where $\mathcal{L}_{cls}$ is the classification loss respect to the ground truth, λ is used to control the degree of NCE loss, β is a coefficient greater than 1, and we denote the *CMIM* loss as $\mathcal{L}_{CMIM} = \sum_{k=1}^{K} \frac{\mathcal{L}_{NCE}^{k}}{\beta^{K-1-k}}$. Hence, the β^{K-1-k} decreases with k increasing and consequently the $\frac{\mathcal{L}_{NCE}^{k}}{\beta^{K-1-k}}$ increases. In this way, the activations of latter layer can be substantially retained, which leads to better performance in practice. The complete training process of *CMIM* is presented in **Algorithm** 1 in the Supplemental Materials.

3.3 Discussion on CMIM

Besides the theoretical formulation from the perspective of mutual information maximization, we also provide an intuitive explanation about *CMIM*. As illustrated in Fig. 2, we strengthen the representation ability of binary activations (see Fig. 3) via designing a proxy task under the contrastive learning framework. By embedding the activations to the contrastive space and pull-and-push the paired embeddings, the BNNs can learn better representations from this difficult yet effective auxiliary contrastive learning task. Note that even though we only pick up two images to formulate Fig. 2, the actual number of negative samples can be huge in practice (*e.g.* 16,384 for training ResNet-18 on ImageNet), benefit from the MemoryBank [42] technique.

With this property, we speculate that the contrastive pairing works as the data augmentation, which contributes to our method. This additional pairing provides more information for training the BNNs, thus *CMIM* can be treated as an overfitting-mitigated module. We also conduct experiments in the Section 4.2 and 4.3 to validate our speculation.

Comparison with other contrastive learning methods. The key idea of contrastive learning is to pull representations close in positive pairs and push representations apart in negative pairs in a contrastive space. Several self-supervised learning methods are rooted in well-established idea of the mutual information maximization, such as Deep InfoMax [20], Contrastive Predictive Coding [34], MemoryBank [42], Augmented Multiscale DIM [1], MoCo [16] and SimSaim [7]. These are based on NCE [13] and InfoNCE [20] which can be seen as a lower bound on mutual information [36]. In the meantime, Tian *et. al.* [41] and Chen *et. al.* [6] generalize the contrastive idea into the content of knowledge distillation (KD) to pull-and-push the representations of teacher and student.

Our formulation of *CMIM*-BNN absorbs the core idea (*i.e.* construct the appropriate positive and negative pairs for contrastive loss) of the existing contrastive learning methods, especially the contrastive knowledge distillation methods, CRD [41] and WCoRD [6]. However, our approach has several differences from those methods. Firstly, our work can not be treated as a simply application with the teacher-and-student framework. In KD, the teacher is basically fixed to offer additional supervision signals and is not optimizable. But in our formulation, we leverage the exclusive structure of BNN, where FP and binary activations exist in the same forward pass, *i.e.* only one BNN is involved, without using another network as a teacher. Therefore, the accuracy improvement of the BNN trained by our method is purely benefited from the activation alignment in a contrastive way, rather than a more accurate teacher network. Secondly, due to the particular structure of BNNs (Eq. 11), our critic function is largely different from the normal critic in contrastive learning (see Eq. 11 and Eq. 12). Importantly, the critic functions of CRD and WCoRD must utilize a fully-connected layer over the representations to transform them into the same dimension and further normalize them by L_2 norm before the inner product, but ours does not. In the literature of binarization, our designed critic function act as an activation flip as we discussed below Eq. 12. Thirdly, instead of only using the activation of the final layer, we align the activations layer-by-layer with a hyperparameter to adjust the weight of each layer as shown in Eq. 14, which is a more suitable design for BNN. In conclusion, using contrastive objective as a tool to realize mutual information maximization for our network binarization is new.

4 Experiments

In this section, we first conduct experiments to compare with existing state-of-the-art methods in image classification. Following popular settings in most studies, we use CIFAR-10/100 [24] and ImageNet ILSVRC-2012 [9] to validate the effectiveness of our proposed binarization method. Besides comparing our method with the SoTA methods, we design experiments in semantic segmentation and depth estimation tasks on the NYUD-v2 [40] dataset to testify the generalization ability of our method. Meanwhile, we conduct a series of ablation studies to verify the effectiveness of our proposed technique, and we empirically explain the efficacy of *CMIM* from the perspective of mitigating overfitting. All experiments are implemented using PyTorch [35] with one NVIDIA RTX 6000 while training on CIFAR-10/100 and NYUD-v2, and four GPUs on ImageNet.
Experimental Setup. On CIFAR-10/100, the BNNs are trained by *CMIM* for 400 epochs with batch size of 256, initial learning rate of 0.1 and cosine learning rate scheduler. We adopt SGD optimizer with momentum of 0.9 and weight decay of 1e-4. On ImageNet, binary models are trained for 100 epochs with batch size of 256. SGD optimizer is applied with momentum of 0.9, weight

Table 1. Top-1 accuracy (%) on CIFAR-10 (C-10) and CIFAR-100 (C-100) test set. The higher the better. W/A denotes the bit number of weights/activations.

Topology	Method	Bit-width (W/A)	Acc.(%) (C-10)	Acc.(%) (C-100)
ResNet -20	Full-precision	32/32	92.1	70.7
	DoReFa [44]	1/1	79.3	-
	QSQ [12]	1/1	84.1	-
	SLB [43]	1/1	85.5	-
	LNS [14]	1/1	85.8	-
	IR-Net [37]	1/1	86.5	65.6
	RBNN [29]	1/1	87.0	66.0
	IR-Net + *CMIM*	1/1	**87.3**	**68.1**
	RBNN + *CMIM*	1/1	**87.6**	**68.2**
ResNet -18	Full-precision	32/32	93.0	72.5
	RAD [10]	1/1	90.5	-
	Proxy-BNN [18]	1/1	91.8	67.2
	IR-Net [37]	1/1	91.6	64.5
	RBNN [29]	1/1	92.2	65.3
	IR-Net + *CMIM*	1/1	**92.2**	**71.2**
	RBNN + *CMIM*	1/1	**92.8**	**71.4**
VGG -small	Full-precision	32/32	94.1	73.0
	XNOR [38]	1/1	90.5	-
	DoReFa [44]	1/1	90.2	-
	RAD [10]	1/1	90.5	-
	QSQ [12]	1/1	90.0	-
	SLB [43]	1/1	92.0	-
	Proxy-BNN [18]	1/1	91.8	67.2
	IR-Net [37]	1/1	90.4	67.0
	RBNN [29]	1/1	91.3	67.4
	IR-Net + *CMIM*	1/1	**92.0**	**70.0**
	RBNN + *CMIM*	1/1	**92.2**	**71.0**

Table 2. Top-1 and Top-5 accuracy on ImageNet. † represents the architecture which varies from the standard ResNet architecture but in the same FLOPs level.

Topology	Method	BW (W/A)	Top-1 (%)	Top-5 (%)
ResNet-18	Full-precision	32/32	69.6	89.2
	ABC-Net [30]	1/1	42.7	67.6
	XNOR-Net [38]	1/1	51.2	73.2
	BNN+ [22]	1/1	53.0	72.6
	DoReFa [44]	1/2	53.4	-
	XNOR++ [3]	1/1	57.1	79.9
	BiReal [31]	1/1	56.4	79.5
	IR-Net [37]	1/1	58.1	80.0
	RBNN [29]	1/1	59.9	81.0
	BiReal + CMIM	1/1	**60.1**	**81.3**
	IR-Net + CMIM	1/1	**61.2**	**83.0**
	RBNN + CMIM	1/1	**62.5**	**84.2**
	ReActNet [32]†	1/1	69.4	85.5
	ReActNet + CMIM†	1/1	**71.0**	**86.3**
ResNet-34	Full-precision	32/32	73.3	91.3
	ABC-Net [30]	1/1	52.4	76.5
	XNOR-Net [38]	1/1	53.1	76.2
	BiReal [31]	1/1	62.2	83.9
	XNOR++ [3]	1/1	57.1	79.9
	IR-Net [37]	1/1	62.9	84.1
	LNS [14]	1/1	59.4	81.7
	RBNN [29]	1/1	63.1	84.4
	IR-Net + CMIM	1/1	**64.9**	**85.8**
	RBNN + CMIM	1/1	**65.0**	**85.7**

decay of 1e-4, initial learning rate of 0.1 with cosine learning rate scheduler (for fair comparison, we also use ADAM optimizer in some ResNet-variant settings).

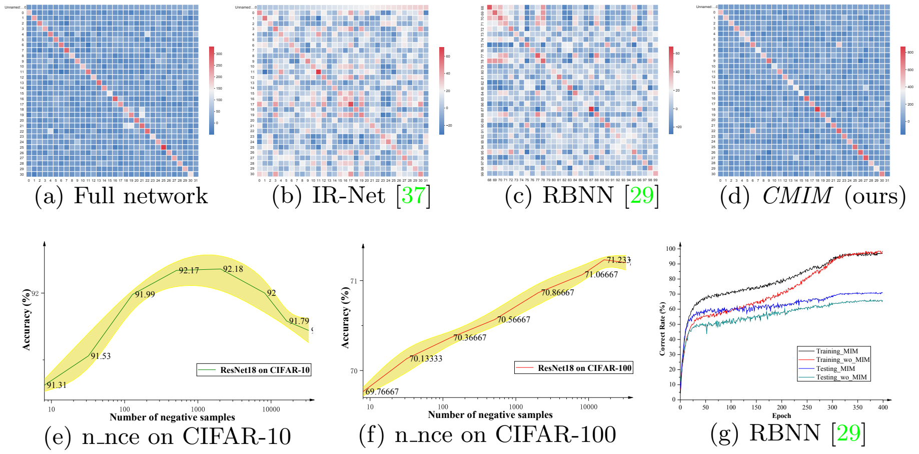
4.1 Experimental Results

CIFAR-10/100 are widely-used image classification datasets, where each consists of 50K training images and 10K testing images of size 32×32 divided into 10/100 classes. 10K training images are randomly sampled for cross-validation and the rest images are utilized for training. Data augmentation strategy includes random crop and random flipping as in [17] during training.

For ResNet-20, we compare with DoReFa [44], QSQ [12], SLB [43], LNS [14], IR-Net [37] and RBNN [29]. For ResNet-18, RAD [10], Proxy-BNN [18], IR-Net and RBNN are chosen to be the benchmarks. For VGG-small, our method is compared with IR-Net and RBNN, *etc.*

As presented in Table 1, *CMIM* constantly outperforms other SOTA methods. On CIFAR-100, our method achieves 2.5%, 6.1% and 4.0% performance improvement with ResNet-20, ResNet-18 and VGG-small architectures, respectively. To show the pile-up property, we add *CMIM* on different baseline methods, and we can obviously observe the accuracy gain with *CMIM*.

ImageNet is a dataset with 1.2 million training images and 50k validation images equally divided into 1K classes. ImageNet has greater diversity, and its


(a) Full network (b) IR-Net [37] (c) RBNN [29] (d) *CMIM* (ours)

(e) n_nce on CIFAR-10 (f) n_nce on CIFAR-100 (g) RBNN [29]

Fig. 4. In-depth analysis on different aspects of the proposed approach including correlation maps (a-d) the effect of number of negative samples in contrastive mutual information maximization (e,f), and training and testing curves (g).

image size is 469×387 (average). We report the single-crop evaluation result using 224×224 center crop from images.

For ResNet-18, we compare our method with XNOR-Net [38], ABC-Net [30], DoReFa [44], BiReal [31], XNOR++ [3], IR-Net [37], RBNN [29]. For ResNet-34, we compare our method with BiReal, IR-Net and RBNN, *etc.* All experimental results are either taken from their published papers or reproduced by ourselves using their code. As demonstrated in Table 2, our proposed method exceeds all the methods in both top-1 and top-5 accuracy. Particularly, *CMIM* achieves around 1.3% Top-1 accuracy gain with ResNet-18 architecture, as well as 1.9% Top-1 accuracy improvement with ResNet-34 architecture, compared with the SoTA RBNN method.

4.2 Number of Negative Samples in *CMIM*

The number of negative samples n_nce is an important hyper-parameter in our method, which ensures the estimation accuracy level of the optimized distribution in Eq. 9. We perform experiments with ResNet18 on CIFAR-100 for parameter analysis of n_nce, with range from 2^0 to 2^{15}. As the results in Fig. 4(f) and 4(e) presented, the accuracy arises with increasing n_nce, which also validates our speculation in the Section 3.2 that the contrastive pairing module, serving as a data augmentation module in training, contributes to the performance improvement of *CMIM*.

4.3 Mitigate Overfitting

A good training objective should consistently improve the model performance in testing set [42]. We investigate the relation between the training and the testing performance *w.r.t.* training iterations. Fig. 4(g) shows that (1) the binary ResNet-18 can reach 100% on training set of CIFAR-100, which means its representative ability is enough for this dataset; (2) the testing performance of the BNN trained with *CMIM* loss is much better on the final stage, while the training performance is relatively lower. This is a clear sign of mitigating overfitting. In addition, as the results shown in the Table 3, we can observe the phenomenon that the accuracy gain on CIFAR-100 is more noticeable than the gain on ImageNet. This phenomenon can also be explained from the perspective of mitigating overfitting. Since the contrastive pairing (data augmentation for the proxy contrastive learning task) plays a significant role in improving the performance of BNNs, and the data for training is sufficient on ImageNet than on CIFAR. The overfitting issue is not that severe on ImageNet. Hence, our binarization method could be more suitable for relatively data-deficient tasks.

4.4 Ablation study

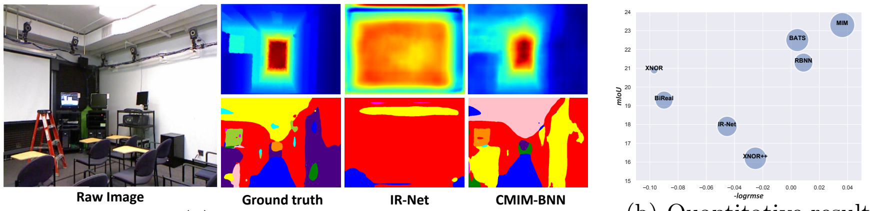
We conduct a series of ablative studies of our proposed method in CIFAR-10/100 and ImageNet datasets with the ResNet18 architecture. By adjusting the coefficient λ in the loss function $\mathcal{L}_{CMIM}$ (Eq. 14), where $\lambda = 0$ equals to no *CMIM* loss are added as our baseline. In the ablative studies, we introduce IR-Net [37] as our baseline on all the datasets. The results are shown in Table 3. With λ increasing, the improving performance validates the efficacy of *CMIM* loss.

Table 3. Ablation study of *CMIM*. The results are presented in the form of accuracy rate (%). $\lambda = 0$ denotes no *CMIM* loss added, serving as our baseline.

λ Dataset	0 (baseline)	0.2	0.4	0.8	1.6	3.2	6.4	12.8
CIFAR-10	87.59	90.92	91.63	92.06	**92.18**	91.89	91.32	91.01
CIFAR-100	64.53	68.21	69.31	70.67	70.86	71.09	**71.19**	71.17
ImageNet-1K	58.03	59.29	59.99	**61.22**	61.17	61.02	60.64	59.7

4.5 Generalization Ability

To study the dependence of the binary activations from the same layer, we visualize the correlation matrix of those activations by using the shade of the color to represent the cosine similarity of two activations. Red stands for two activations are similar and blue *vice versa*. As shown in Figs. 4(a)-4(d), *CMIM* captures more intra-class correlations (diagonal boxes are redder) and alleviates more inter-class correlations (non-diagonal boxes are bluer). Those intensified representative activations are constructive for fine-tuning down-stream tasks.



(a) Visualized results

(b) Quantitative results

Fig. 5. Results of depth estimation and segmentation on NYUD-v2

To further evaluate the generalization capacity of the learned binary features, we transfer the learned binary backbone to the image segmentation and depth estimation on NYUD-v2 dataset. We follow the standard pipeline for fine-tuning. A prevalent practice is to pre-train the backbone network on ImageNet and fine-tune it for the downstream tasks. Thus, we conduct experiments with DeepLab heads with the binary ResNet18 backbone. While fine-tuning, the learning rate is initialized to 0.001 and scaled down by 10 times after every 10K iterations and we fix the binary backbone weights, only updating the task-specific heads layers. The results are presented in Fig. 5(b), X-axis is the depth estimation accuracy (-logrmse, higher is better), Y-axis is segmentation performance (mIoU, higher is better) and the size of dot denotes the performance of classification (bigger is better). The visualization results are presented in Fig. 5(a). We can observe that the models with backbone pre-trained by *CMIM* outperform other methods on both segmentation and depth estimation tasks.

5 Conclusion

In this paper, we investigate the activations of BNNs by introducing mutual information to measure the distributional similarity between the binary and full-precision activations. We establish a proxy task via contrastive learning to maximize the targeted mutual information between those binary and real-valued activations. We name our method *CMIM*-BNN. Because of the push-and-pull scheme in the contrastive learning, the BNNs optimized by our method have better representation ability, benefiting downstream tasks, such as classification and segmentation, *etc.* We conduct experiments on CIFAR, ImageNet (for classification) and NYUD-v2 (fine-tuning for depth estimation and segmentation). The results show that CMIM outperforms several state-of-the-art binarization methods on those tasks.

Acknowledgements. This research was partially supported by NSF CNS-1908658 (ZZ,YY), NeTS-2109982 (YY), Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No. 26202321 (DX), HKUST Startup Fund No. R9253 (DX) and the gift donation from Cisco (YY). This article solely reflects the opinions and conclusions of its authors and not the funding agents.

References

1. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. In: NeurIPS (2019) 5, 6, 7, 8, 9
2. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv:1308.3432 (2013) 3
3. Bulat, A., Tzimiropoulos, G.: Xnor-net++: Improved binary neural networks. In: BMVC (2019) 4, 11, 12
4. Cai, Z., He, X., Sun, J., Vasconcelos, N.: Deep learning with low precision by half-wave gaussian quantization. In: CVPR (2017) 2
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI (2017) 1
6. Chen, L., Wang, D., Gan, Z., Liu, J., Henao, R., Carin, L.: Wasserstein contrastive representation distillation. In: CVPR (2021) 7, 10
7. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR (2021) 5, 6, 8, 9
8. Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: Training deep neural networks with binary weights during propagations. In: NeurIPS (2016) 5
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 10
10. Ding, R., Chin, T.W., Liu, Z., Marculescu, D.: Regularizing activation distribution for training binarized deep networks. In: CVPR (2019) 2, 11
11. Gao, S., Ver Steeg, G., Galstyan, A.: Efficient estimation of mutual information for strongly dependent variables. In: AISTATS (2015) 3
12. Gong, R., Liu, X., Jiang, S., Li, T., Hu, P., Lin, J., Yu, F., Yan, J.: Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In: ICCV (2019) 11
13. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: AISTATS (2010) 5, 6, 7, 9
14. Han, K., Wang, Y., Xu, Y., Xu, C., Wu, E., Xu, C.: Training binary neural networks through learning with noisy supervision. In: ICML (2020) 11
15. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: ICLR (2016) 2
16. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020) 5, 6, 9
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 11
18. He, X., Mo, Z., Cheng, K., Xu, W., Hu, Q., Wang, P., Liu, Q., Cheng, J.: Proxybnn: Learning binarized neural networks via proxy matrices. In: CVPR (2020) 2, 4, 11
19. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NeurIPS (2014) 2, 7
20. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018) 5, 6, 7, 8, 9
21. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017) 4

22. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks. In: NeurIPS (2016) 2, 3, 5, 11
23. Kim, H., Park, J., Lee, C., Kim, J.J.: Improving accuracy of binary neural networks using unbalanced activation distribution. In: CVPR (2021) 2
24. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 10
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS (2012) 1
26. Kullback, S.: Information theory and statistics. Courier Corporation (1997) 5
27. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature (2015) 1
28. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. In: NeurIPS (1989) 2
29. Lin, M., Ji, R., Xu, Z., Zhang, B., Wang, Y., Wu, Y., Huang, F., Lin, C.W.: Rotated binary neural network. In: NeurIPS (2020) 2, 5, 9, 11, 12
30. Lin, X., Zhao, C., Pan, W.: Towards accurate binary convolutional neural network. In: NeurIPS (2017) 11, 12
31. Liu, Z., Luo, W., Wu, B., Yang, X., Liu, W., Cheng, K.T.: Bi-real net: Binarizing deep network towards real-network performance. IJCV (2020) 2, 4, 11, 12
32. Liu, Z., Shen, Z., Savvides, M., Cheng, K.T.: Reactnet: Towards precise binary neural network with generalized activation functions. In: ECCV (2020) 4, 11
33. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR (2008) 9
34. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 5, 6, 7, 8, 9
35. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019) 10
36. Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., Tucker, G.: On variational bounds of mutual information. In: ICML (2019) 5, 10
37. Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., Song, J.: Forward and backward information retention for accurate binary neural networks. In: CVPR (2020) 2, 9, 11, 12, 13
38. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: ECCV (2016) 2, 3, 4, 9, 11, 12
39. Shang, Y., Duan, B., Zong, Z., Nie, L., Yan, Y.: Lipschitz continuity guided knowledge distillation. In: ICCV (2021) 2
40. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012) 10
41. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: ICLR (2021) 7, 10
42. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018) 5, 7, 9, 13
43. Yang, Z., Wang, Y., Han, K., Xu, C., Xu, C., Tao, D., Xu, C.: Searching for low-bit weights in quantized neural networks. In: NeurIPS (2020) 11
44. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016) 11, 12