Neurocomputing 511 (2022) 67–77

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Dual-discriminator adversarial framework for data-free quantization

Zhikai Li [a,b], Liping Ma [a], Xianlei Long [a,b], Junrui Xiao [a,b], Qingyi Gu [a,*]

[a] Institute of Automation, Chinese Academy of Sciences, East Zhongguancun Road, Haidian District, Beijing, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Jingjia Road, Huairou District, Beijing, China

ARTICLE INFO

Article history:
Received 24 March 2022
Revised 20 July 2022
Accepted 4 September 2022
Available online 9 September 2022
Communicated by Zidong Wang

Keywords:
Model compression
Quantized neural networks
Data-free quantization

ABSTRACT

Thanks to the potential to address the privacy and security issues, data-free quantization that generates samples based on the prior information in the model has recently been widely investigated. However, existing methods failed to adequately utilize the prior information and thus cannot fully restore the real-data characteristics and provide effective supervision to the quantized model, resulting in poor performance. In this paper, we propose Dual-Discriminator Adversarial Quantization (DDAQ), a novel data-free quantization framework with an adversarial learning style that enables effective sample generation and learning of the quantized model. Specifically, we employ a generator to produce meaningful and diverse samples directed by two discriminators, aiming to facilitate the matching of the batch normalization (BN) distribution and maximizing the discrepancy between the full-precision model and the quantized model, respectively. Moreover, inspired by mixed-precision quantization, i.e., the importance of each layer is different, we introduce layer importance prior to both discriminators, allowing us to make better use of the information in the model. Subsequently, the quantized model is trained with the generated samples under the supervision of the full-precision model. We evaluate DDAQ on various network structures for different vision tasks, including image classification and object detection, and the experimental results show that DDAQ outperforms all baseline methods with good generality.

1. Introduction

Deep neural network (DNN) models have demonstrated remarkable effectiveness in a variety of computer vision tasks [1–6]; nevertheless, their high model complexity makes deployment and real-time inference on edge devices challenging [7–11]. Therefore, model quantization, which converts 32-bit floating-point parameters to low-precision values, is extensively used to produce efficient networks suitable for hardware deployment [12–19,70]. To mitigate the accuracy degradation caused by quantization, quantization-aware training (QAT) methods fine-tune the parameters with the help of original training data [20–22]. However, in many practical scenarios concerning user privacy and security, original data is not available [23–25], such as medical data and bio-metric data, rendering the QAT methods no longer applicable [26–28].

As a result, post-training quantization (PTQ) methods are proposed to eliminate the complex fine-tuning process [29–31]; however, they still typically require a small amount of training data for calibration, and suffer from non-trivial accuracy gaps, especially for low-precision (*e.g.*, 4-bit) quantization [32]. To address the above issues, recent works have proposed data-free quantization [27,32,33,26], which does not require any original data but produces powerful results, making it a new research hotspot. One approach is to reconstruct samples based on the batch normalization (BN) distribution [27,33,26], intending to match the real-data statistics contained in the full-precision model's BN layers, and then use the generated samples for parameter calibration. Another notable approach models the discrepancy between the full-precision model and the quantized model as an adversarial game [32], thereby achieving sample generation and knowledge transfer from the full-precision model to the quantized model. However, there is still a large performance gap between these two approaches and real-data-driven QAT methods, since neither approach fully utilizes the information in the model, i.e., the former ignores the information interaction between models and hence cannot ensure the generality of the quantized model, while the latter disregards the real-data statistics stored in the full-precision model, leading to a mismatch between the distribution of the generated samples and the real data.

* Corresponding author.
E-mail addresses: lizhikai2020@ia.ac.cn (Z. Li), liping.ma@ia.ac.cn (L. Ma), longxianlei2017@ia.ac.cn (X. Long), xiaojunrui2020@ia.ac.cn (J. Xiao), qingyi.gu@ia.ac.cn (Q. Gu).

https://doi.org/10.1016/j.neucom.2022.09.076
0925-2312/

In addition, none of the previous methods in the data-free quantization community take into account the layer importance prior, i.e., different layers in a DNN model have different contributions to the overall performance, which is a potentially key attribute of the model highlighted in the mixed-precision quantization methods [34–37]. For instance, DSG [27] performs layerwise sample enhancement (LSE) by assigning the same weight to each layer using an identity matrix, which is based on the faulty assumption that each layer is equally important; ZAQ [32] indiscriminately sums the discrepancies between the intermediate layers of the full-precision model and the quantized model, which also ignores the difference in the importance of the layers. Consequently, existing methods all fail to take advantage of the valuable attribute of layer importance prior, resulting in unsatisfactory performance.

In this paper, we propose a novel data-free quantization framework, named Dual-Discriminator Adversarial Quantization (DDAQ), to overcome the aforementioned issues. DDAQ efficiently exploit the information contained in the model, considering not only the BN statistics in the full-precision model, but also the information interactions between the models that are achieved by the min–max adversarial game of the discrepancy between the full-precision model and the quantized model. Specifically, we train the generator in an adversarial learning fashion using two discriminators, one to minimize the matching loss of the BN layer distribution and the other to maximize the discrepancy between the full-precision model and the quantized model, and subsequently, the generated samples are utilized to train the quantized model under the supervision of the full-precision model. Furthermore, we introduce Hessian-based layer importance prior to both discriminators, allowing us to fully utilize the valuable information in the model, and thus promoting more diverse sample generation and more efficient knowledge transfer. The overall workflow is illustrated in Fig. 1.

To sum up, our contributions are as follows:

- We propose DDAQ, an adversarial learning framework with two discriminators to support effective sample generation and knowledge transfer for data-free quantization. DDAQ considers both BN layer distribution and model interaction, allowing us to efficiently exploit the information contained in the model.
- We introduce layer importance prior to the framework by enhancing the discriminator losses with the average Hessian trace, which enables us to further leverage the information in the model.
- The diverse samples generated by DDAQ satisfy the real-data statistics stored in the BN layers and increase the discrepancy between models, thus facilitating more effective knowledge transfer from the full-precision model to the quantized model.
- We conduct extensive experiments on various model structures for image classification and object detection tasks, and consistently outperform state-of-the-art (SOTA) data-free quantization methods, demonstrating the effectiveness and generality of DDAQ.

2. Related works

Model quantization is a promising approach for reducing the memory consumption and computation cost of a DNN model, thus ensuring its real-time processing on edge devices [38–44]. To address the accuracy gap between the full-precision model and the quantized model, various QAT methods have been proposed. DoReFa [21] approximates the gradient propagation of the quantized model using straight-through estimator (STE) [45]. LSQ [46] and PACT [47] take the step size or activation clipping value as a learnable parameter to achieve low-bit quantization. LQ-Nets [20] determines the quantization levels using the quantization error minimization algorithm, while APoT [48] limits all quantization levels to the sum of powers-of-two terms. FQN [49] utilizes channel-wise quantization to compress the networks for object detection tasks. Other works attempt to use periodic regularization to assist quantization [50,51] or progressively quantify the network [52]. However, the above methods all require original training data for fine-tuning, and the unavailability of original data for many scenarios makes these methods inapplicable.

The PTQ methods are proposed to eliminate the computation cost of fine-tuning and thus improve the quantization efficiency. DFQ [29] and ACIQ [30] rely on correction strategies to equalize weights and remove biases. AdaRound [31] adaptively rounds weights based on the data and the task loss. However, the PTQ methods result in severe performance degradation, and they are

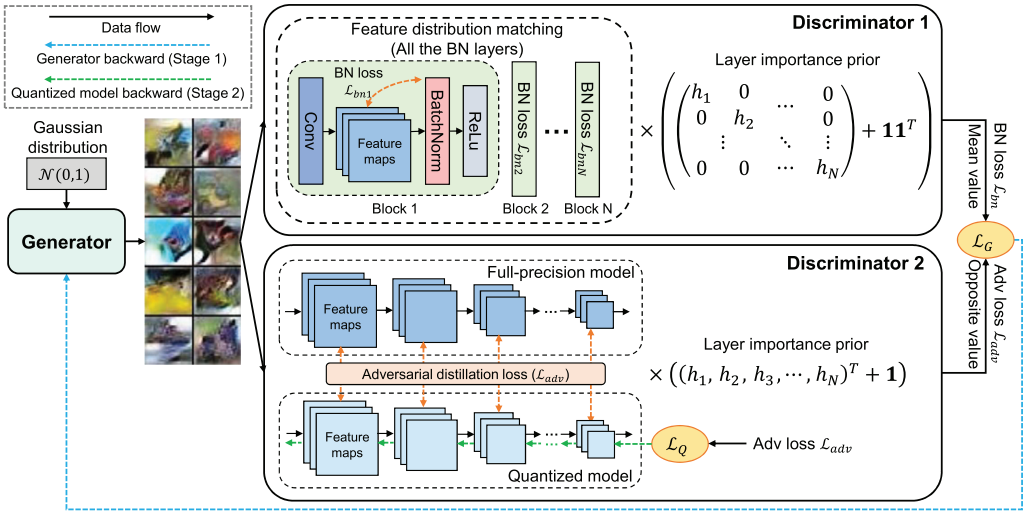


Fig. 1. The overall workflow of DDAQ. The training procedure is conducted in a two-stage adversarial learning manner. In the first stage, the full-precision model and the quantized model are fixed, while the generator is trainable and synthesizes fake samples from Gaussian noise. Two discriminators are employed to direct the training of the generator, which facilitate BN distribution matching and model discrepancy maximization, respectively. In the second stage, the full-precision model and the generator are fixed while the quantized model is trainable, and the generated samples are utilized to train the quantized model under the supervision of the full-precision model. In addition, we introduce layer importance prior (average Hessian trace h_i) to the framework to enhance the discriminator loss.

not genuinely data-free since a limited amount of data is required for calibration.

Data-free quantization, which can compress models without access to any real data, is a technique that is highly desired in many scenarios concerning privacy and security [17], and is therefore receiving increasing attention. The idea is to follow the prior information in the full-precision model to generate samples, and then use them to train the quantized model. ZeroQ [26] reconstructs samples to match the real-data statistics stored in the full-precision model's BN layers. DSG [27] slackens the BN matching constraint and assigns different attention to specific layers for different samples to ensure diverse sample generation. GDFQ [33] introduces class prior (category label) information combined with BN statistics to synthesize samples, limiting its inability to extend to high-level tasks such as object detection. ZAQ [32] models the discrepancy between the full-precision model and the quantized model as an adversarial game, and achieves sample generation and knowledge transfer in an adversarial learning manner. However, the performance of the existing methods is still far from satisfactory, as they all fail to fully use the information in the model, leading to low generality of the quantized model [26,27,33] and mismatch of the generated samples [32].

Layer importance prior is highlighted in the mixed-precision quantization methods [34–37], which is a potentially key attribute of the model, i.e., the importance of each layer for the final performance is different. A popular metric for layer importance evaluation is the average Hessian trace used in HAWQ-V2 [34], which is second-order information that can be calculated using the Hutchinson algorithm [53,54] for fast computation. In addition, layer importance prior also has wide range of applications in other scenarios, including filter pruning [55] and feature representation enhancement that improves the discriminative capability of DNN models [56–60]. In this work, we introduce the average Hessian trace to the proposed framework, on which two enhancement matrices are developed to improve the two discriminator losses, respectively, to further exploit the information in the model, promoting more effective sample generation and knowledge transfer.

3. Methodology

In this section, we first describe the preliminary of uniform quantization, generative adversarial nets (especially the dual-discriminator architecture), and layer importance prior represented by the average Hessian trace. The overall adversarial learning pipeline of DDAQ is then summarized and introduced. Afterward, we introduce the training of the generator, and in particular, describe in detail how to establish two discriminators to apply data-free quantization to the dual-discriminator framework. Finally, the training of the quantized model is presented to achieve knowledge transfer from the full-precision model.

3.1. Preliminary

Uniform quantization: Quantization converts the floating-point parameters θ^p (weights and activations) in the pretrained full-precision model to low-precision fixed-point values θ^q, thus reducing the model complexity. Uniform quantization is the simplest and most hardware-friendly method, which is defined as follows:

$$\theta^q = \left[\frac{\theta^p - Z}{\Delta}\right], \Delta = \frac{\max(\theta^p) - \min(\theta^p)}{2^k - 1} \tag{1}$$

where $[\cdot]$ is the rounding function, Z is the zero point, and k is the quantization bit-width.

Generative adversarial nets: The vanilla GAN [61] utilizes a generator G to map the noise vector $\boldsymbol{z}$ drawn from a prior $p_{\boldsymbol{z}}(\boldsymbol{z})$ (*e.g.*, Gaussian distribution) to the desired data $G(\boldsymbol{z})$, while a discriminator D is employed to distinguish the real data $\boldsymbol{y}$ from the generated data. Specifically, D and G play a two-player minimax game in an adversarial learning style with the following value function $V(G,D)$:

$$\min_G \max_D \; V(G,D) = \mathbb{E}_{\boldsymbol{y}\sim p_{data}(\boldsymbol{y})}[\log D(\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

At this point, the loss function $\mathscr{L}_{G_1}$ to be minimized for training the generator G is:

$$\mathscr{L}_{G_1} = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{3}$$

To further improve the capability of the generator, D2GAN [62] play a three-player minimax game with two discriminators D_1 and D_2 to guide sample generation, where D_1 encourages the real data while D_2 encourages the generated data. In this case, the generator is optimized depending on both D_1 and D_2 with the following loss function:

$$\mathscr{L}_{G_2} = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[-D_1(G(\boldsymbol{z}))] + \gamma\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(D_2(G(\boldsymbol{z})))] \tag{4}$$

where the two parts aim to fool the discriminators D_1 and D_2, respectively, and γ is a hyperparameter to balance the effect of the two discriminators.

Layer importance prior: HAWQ-V2 [34] theoretically proves that the average Hessian trace can represent the impact of each layer's perturbation on the overall performance of the model, and thus it can be used to measure the importance of each layer (please see Appendix for the proved Lemma). The Hessian matrix $H_i \in \mathbb{R}^{m_i \times m_i}$ of layer i in the pretrained model is the second derivative of the loss ℓ w.r.t. the parameters $\theta_i \in \mathbb{R}^{m_i}$ of layer i, which is calculated as follows:

$$H_i = \frac{\partial g_{\theta_i}}{\partial \theta_i} = \frac{\partial^2 \ell}{\partial \theta_i^2} \tag{5}$$

The Hessian trace can be estimated by a low computational overhead method [53], which is formulated as follows:

$$Tr(H_i) = Tr(H_i I) = Tr(H_i \mathbb{E}[vv^T]) = \mathbb{E}[Tr(H_i vv^T)] = \mathbb{E}[v^T H_i v] \tag{6}$$

where v is a random vector drawn from Rademacher distribution (or Gaussian distribution $\mathcal{N}(0,1)$).

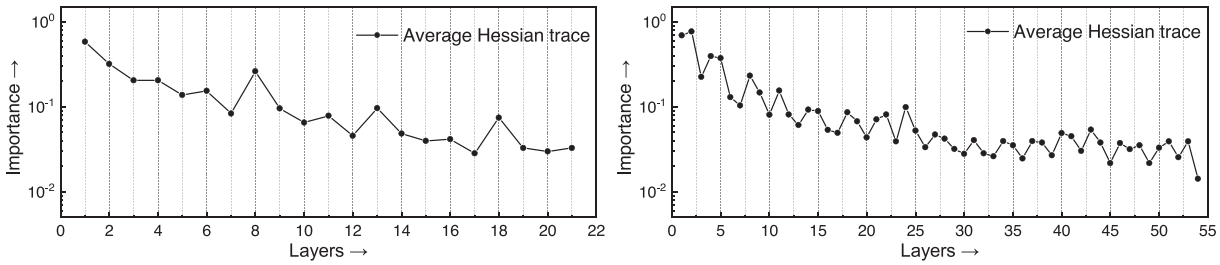
To eliminate the effect of the layer's memory footprint, we normalize the Hessian trace with the number of parameters m_i and obtain the average Hessian trace as follows:

$$h_i = \frac{Tr(H_i)}{m_i} \tag{7}$$

The average Hessian trace of each layer in ResNet model on ImageNet is illustrated in Fig. 2, showing that the importance of each layer for the final performance is significantly different.

3.2. The DDAQ pipeline

The DDAQ framework is performed in an adversarial learning manner, and the overall pipeline is summarized in Algorithm 1. In DDAQ, we only need the full-precision model to produce the quantized model; no original data is required. This is achieved by dividing the whole process into two stages which are performed alternately in an adversarial game, where the generator G is employed to produce diverse samples that satisfy the real-data distribution and maximize the discrepancy between models, while the quantized model Q learns the useful knowledge of the full-precision model with the help of the generated samples and thus minimizes the discrepancy between models.



(a) ResNet-18 (b) ResNet-50

Fig. 2. The average Hessian trace of different layers in pre-trained ResNet-18 and ResNet-50 on ImageNet. The x-axis represents each layer in the model, and the y-axis indicates the importance of each layer to the final model performance. As one can see, different layers have significantly different importance.

Algorithm 1: The DDAQ Pipeline

Algorithm 1: The DDAQ Pipeline

Input: A pretrained full-precision model P with parameters θ^p.

Output: A quantized model Q with parameters θ^q.

Initialize the generator G with parameters θ^g;

Initialize the quantized model Q by Eq. 1;

The following is the fine-tuning procedure.

for $t = 1, 2, \ldots, T$ **do**

  # Training of the Generator

  Randomly produce noise $\boldsymbol{z} \sim \mathcal{N}(0, 1)$;

  Generate samples $G(\boldsymbol{z})$ with the generator G;

  Input $G(\boldsymbol{z})$ into the two discriminators;

  Calculate $\mathcal{L}_{bn}$ by Eq. 11; # Discriminator 1

  Calculate $\mathcal{L}_{adv}$ by Eq. 15; # Discriminator 2

  Combine $\mathcal{L}_{bn}$ and -$\mathcal{L}_{adv}$ to obtain $\mathcal{L}_G$ by Eq. 16;

  Fix θ^q, update θ^g by back-propagation of $\mathcal{L}_G$;

  # Training of the Quantized Model

  Randomly produce noise $\boldsymbol{z} \sim \mathcal{N}(0, 1)$;

  Generate samples $G(\boldsymbol{z})$ with the generator G;

  Input $G(\boldsymbol{z})$ into discriminator 2;

  Calculate $\mathcal{L}_Q$ by Eq. 17;

  Fix θ^g, update θ^q by back-propagation of $\mathcal{L}_Q$;

end

In the first stage, we first randomly produce a batch of Gaussian noise $\boldsymbol{z} \sim \mathcal{N}(0, 1)$, and feed it to the generator G to obtain the samples $G(\boldsymbol{z})$. Then the generated samples $G(\boldsymbol{z})$ are input to two discriminators to obtain two loss functions for training the generator, where $\mathscr{L}_{bn}$ is utilized to facilitate the matching of BN statistics and $\mathscr{L}_{adv}$ is utilized to increase the discrepancy between the full-precision model and the quantized model. Finally, we combine the two losses to obtain $\mathscr{L}_G$ and perform back-propagation to update the parameters θ^g of the generator while fixing the parameters θ^q of the quantized model.

In the second stage, we start by producing Gaussian noise $\boldsymbol{z} \sim \mathcal{N}(0, 1)$ and generating samples $G(\boldsymbol{z})$ in the same way. The full-precision model and the quantized model then perform inference with the generated samples $G(\boldsymbol{z})$. We obtain the loss function $\mathscr{L}_Q$ which aims to reduce the discrepancy of the feature maps in the corresponding layers during the inference of the two models, and finally use it for back-propagation to update θ^q while fixing θ^g.

3.3. Training of the generator

Since D2GAN requires real data for training, data-free quantization cannot be directly applied to this architecture. Therefore, our interest is how to define two discriminators to direct the training of the generator without original data. To this end, we only utilize the prior knowledge available in the full-precision model to train the generator, with one discriminator reflecting the statistics of the real data and the other discriminator performing the information interaction between models. Specifically, the generator is a lightweight four-layer upsampling convolutional neural network, which follows GDFQ [33] and ZAQ [32]; discriminator 1 contains only the full-precision model, and discriminator 2 contains both the full-precision model and the quantized model, with the specific training strategy and loss functions below.

Discriminator 1: The statistics (i.e., the mean and standard deviation) encoded in the BN layers of the full-precision model can represent the distribution of the original training data. Discriminator 1 contains only the full-precision model itself, and it is employed to facilitate the matching of BN statistics. Specifically, we learn the input data to best match the BN statistics and thus make it approximate the real-data distribution. The matching loss $\mathscr{L}_{bn_i}$ of layer i is calculated as follows:

$$\mathscr{L}_{bn_i} = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}\left[||\tilde{\boldsymbol{\mu}}_i^g - \boldsymbol{\mu}_i||_2^2 + ||\tilde{\boldsymbol{\sigma}}_i^g - \boldsymbol{\sigma}_i||_2^2\right] \tag{8}$$

where $||\cdot||_2^2$ denotes the square of L_2 norm. Here, $\tilde{\boldsymbol{\mu}}_i^g/\tilde{\boldsymbol{\sigma}}_i^g$ are the running mean/standard deviation of the feature distribution at the *i-th* BN layer during model inference when the input is the generated samples $G(\boldsymbol{z})$, and $\boldsymbol{\mu}_i/\boldsymbol{\sigma}_i$ are mean/standard deviation information encoded in the *i-th* BN layer of the full-precision model.

Furthermore, we introduce layer importance prior to enhance the BN statistics matching. Specifically, instead of simply summing as in ZeroQ [26], we combine the matching loss $\mathscr{L}_{bn_i}$ of each layer into a row vector $\mathbf{L}_1 \in \mathbb{R}^{1\times N}$ (N is the total number of the layers), which is represented as follows:

$$\mathbf{L}_1 = \left[\mathscr{L}_{bn_1}, \mathscr{L}_{bn_2}, \ldots, \mathscr{L}_{bn_N}\right] \tag{9}$$

Then we define the enhancement matrix $\mathbf{X}_1 \in \mathbb{R}^{N\times N}$ based on the layer importance prior as follows:

$$\mathbf{X}_1 = \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ 0 & h_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_N \end{bmatrix} + \mathbf{1}\mathbf{1}^T \tag{10}$$

where h_i is the average Hessian trace of the *i-th* layer in the full-precision model, and $\mathbf{1}$ is an N-dimension column vector of all ones. The enhancement matrix $\mathbf{X}_1$ encourages a batch of generated samples to pay different attention to different layers (i.e., the i-th sample is more influenced by the i-th layer's BN statistics), and this attention varies with the layer importance prior, which can potentially ensure the diversity and validity of the samples.

Finally, the loss function $\mathscr{L}_{bn}$ of the generator directed by discriminator 1 is defined as:

$$\mathscr{L}_{bn} = \frac{1}{N}\cdot\mathbf{1}^T(\mathbf{L}_1\mathbf{X}_1) \tag{11}$$

Here, the result $\mathscr{L}_{bn}$ is a row vector, where the i-th element focuses more on the i-th BN layer and it will act on the update of the i-th sample in the batch.

Discriminator 2: This discriminator considers the information interaction between models and maximizes the discrepancy between the full-precision model and the quantized model, which can both improve the diversity of generated samples and promote the generality of the quantized model after knowledge transfer. We calculate not only the output discrepancy between models, but also the discrepancy of the feature maps in the intermediate layers. The discrepancy is modeled with the adversarial distillation loss, which is calculated as follows:

$$\mathscr{L}_{adv_i} = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}\left[\frac{1}{N}||\mathscr{P}_i(G(\boldsymbol{z})) - \mathscr{Q}_i(G(\boldsymbol{z}))||_1\right] \tag{12}$$

Where $\mathscr{P}_i(G(\boldsymbol{z}))$ and $\mathscr{Q}_i(G(\boldsymbol{z}))$ are the feature maps of layer i in the full-precision model and the quantized model when the input is the generated sample $G(\boldsymbol{z})$, respectively. Note that we simply use the naive L_1 norm in order to prove the validity of the framework itself, rather than more advanced losses such as KL divergence.

Then, as in discriminator 1, we combine the discrepancies of each layer into a row vector $\mathbf{L}_2 \in \mathbb{R}^{1\times N}$ rather than simply adding them up like ZAQ [32], which is denoted as:

$$\mathbf{L}_2 = \left[\mathscr{L}_{adv_1}, \mathscr{L}_{adv_2}, \ldots, \mathscr{L}_{adv_N}\right] \tag{13}$$

In discriminator 2, the enhancement matrix $\mathbf{X}_2 \in \mathbb{R}^{N\times 1}$ is formulated as:

$$\mathbf{X}_2 = [h_1, h_2, \ldots, h_N]^T + \mathbf{1} \tag{14}$$

The enhancement matrix $\mathbf{X}_2$ motivates us to focus more on the discrepancies between the more important layers rather than treating all layers equally when calculating adversarial distillation losses, thus allowing us to obtain more effective information about the discrepancies.

Finally, the loss function of the discrepancy between the full-precision model and the quantized model is defined as:

$$\mathscr{L}_{adv} = \mathbf{L}_2\mathbf{X}_2 \tag{15}$$

Training loss of the generator: Our aim is to generate samples that can simultaneously fool both discriminators. Therefore, as in Eq. 4, we combine the losses corresponding to the two aforementioned discriminators, minimizing the distribution matching loss $\mathscr{L}_{bn}$ and maximizing the discrepancy loss $\mathscr{L}_{adv}$, to obtain the final objective function for training the generator as follows:

$$\mathscr{L}_G = \alpha\mathscr{L}_{bn} - \beta\mathscr{L}_{adv} \tag{16}$$

where α and β are hyperparameters to balance the two losses $\mathscr{L}_{bn}$ and $\mathscr{L}_{adv}$.

3.4. Training of the quantized model

The training process requires the participation of both the full-precision model and the quantized model. Specifically, in this process, the quantized model is trained with the generated samples under the supervision of the full-precision model, i.e., we achieve the knowledge transfer from the full-precision model to the quantized model with the help of the generated samples. To this end, we utilize the adversarial distillation loss in discriminator 2 to minimize the discrepancy between the two models, thus motivating the quantized model to learn useful knowledge (i.e., driving the quantized model to mimic the full-precision model). Note that since discriminator 1 contains only the full-precision model, it has no effect on the training of the quantized model and we can ignore it. More formally, the loss function for training the quantized model is defined as follows:

$$\mathscr{L}_Q = \mathscr{L}_{adv} = \mathbf{L}_2\mathbf{X}_2 \tag{17}$$

Note that $\mathscr{L}_Q$ directly forms an adversarial relationship with $(-\mathscr{L}_{adv})$, which makes it possible for us to solve optimization problems with the help of the adversarial learning paradigm.

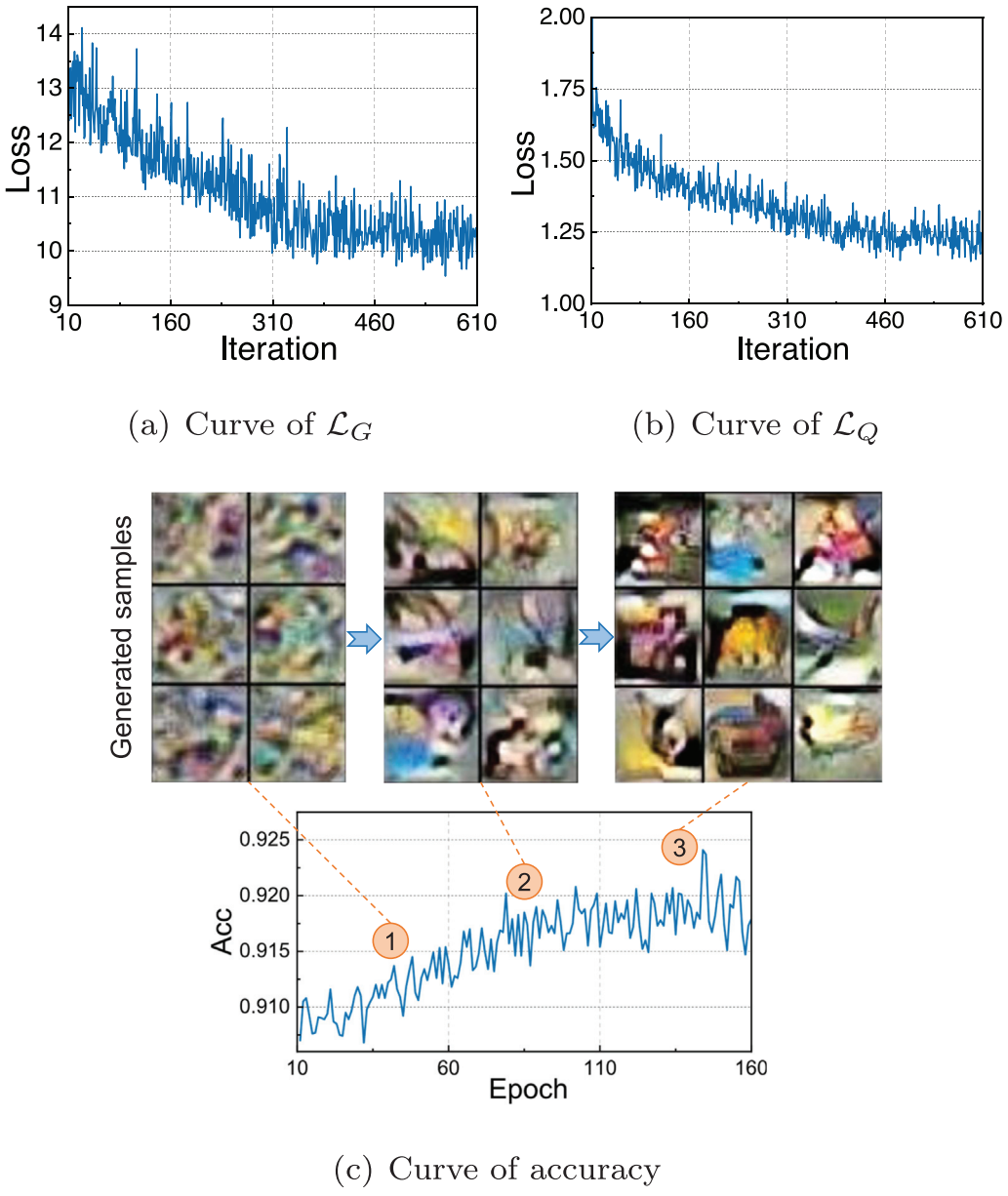


Fig. 3. Training losses and test accuracy of ResNet-20 on CIFAR-10 during fine-tuning. $\mathscr{L}_G$ and $\mathscr{L}_Q$ decrease in the adversarial, making the quantized model accuracy steadily improve. We visualize the results of the intermediate epochs, showing a gradual increase in the validity and semantics of the generated samples (in 32×32 resolution).

4. Experimental results

In this section, we evaluate various model structures [63–65] on CIFAR-10 [66] and ImageNet [67] datasets for image classification task and COCO [68] dataset for object detection task, and compare the experimental results with the SOTA data-free quantization methods to demonstrate the advantages of the proposed DDAQ.

4.1. Implementation details

All implementations are performed based on PyTorch. First, we obtain the pretrained full-precision model from pytorchcv[1]. Then we use PyHessian[2] [54] to calculate the average Hessian trace as the layer importance evaluation metric. And we quantize the weights and activations of the model into the corresponding bit-widths, including the first and last layers. The specific training process includes two parts, warm-up and fine-tuning, which are described below.

Warm-up: In the first few epochs, the generated samples are close to Gaussian noise and are useless for training the quantized model. Therefore, we only train the generator in the warm-up. Specifically, the warm-up epochs are set to 10, 30, and 30 on CIFAR-10, ImageNet, and COCO datasets, respectively.

Fine-tuning: As shown in Algorithm 1, the fine-tuning is performed in a two-stage adversarial game, and no fine-tuning means that the quantized model is updated only once. We use the Adam [69] optimizer to learn both the generator and quantized model, and the gradient back-propagation and updating of the quantized model is done using naive STE [45]. For the discrepancy calculation in discriminator 2, we make simplifications to reduce the computation, *e.g.*, we only consider the feature map of the last layer of each residual block in ResNet. For the balance coefficients α and β of the generator loss, we set them both to 0.5 after a simple grid

[1] https://pypi.org/project/pytorchcv
[2] https://github.com/amirgholami/PyHessian

Table 1
Quantization results on CIFAR-10 and ImageNet datasets. We abbreviate fine-tuning as "FT"("–" means without fine-tuning and "✓" means with fine-tuning) and bit-precision as "Prec." (W*x*A*y* means weight with *x*-bit and activation with *y*-bit). Our proposed DDAQ outperforms the SOTA data-free quantization methods in both no fine-tuning and fine-tuning cases, including ZeroQ [26], DSG [27], GDFQ [33], and ZAQ [32]. "*" denotes the results reproduced from the source code, and all other results are obtained from the original paper.

Dataset	Model	Method	FT	Prec.	Size(MB)	BitOps(G)	Top-1(%)
CIFAR-10	ResNet-20	Baseline		FP32	1.03	41.7	94.03
		ZeroQ	–	W4A4	0.130	0.652	85.39
		DSG	–	W4A4	0.130	0.652	87.75
		DDAQ (ours)	–	W4A4	0.130	0.652	**90.70**
		GDFQ	✓	W4A4	0.130	0.652	90.25
		ZAQ*	✓	W4A4	0.130	0.652	91.03
		DDAQ (ours)	✓	W4A4	0.130	0.652	**92.41**
ImageNet	ResNet-18	Baseline		FP32	44.6	1858	71.47
		ZeroQ	–	W4A4	5.58	29.0	26.04
		DSG	–	W4A4	5.58	29.0	34.53
		DDAQ (ours)	–	W4A4	5.58	29.0	**58.44**
		GDFQ	✓	W4A4	5.58	29.0	60.60
		ZAQ*	✓	W4A4	5.58	29.0	61.34
		DDAQ (ours)	✓	W4A4	5.58	29.0	**62.91**
	ResNet-50	Baseline		FP32	97.8	3951	77.72
		ZeroQ*	–	W4A6	12.2	92.6	67.82
		DDAQ (ours)	–	W4A6	12.2	92.6	**73.30**
		GDFQ*	✓	W4A6	12.2	92.6	73.52
		DDAQ (ours)	✓	W4A6	12.2	92.6	**74.56**
		ZeroQ	–	W6A6	18.3	139	75.56
		DSG	–	W6A6	18.3	139	76.07
		DDAQ (ours)	–	W6A6	18.3	139	**76.58**
		GDFQ	✓	W6A6	18.3	139	76.59
		DDAQ (ours)	✓	W6A6	18.3	139	**76.98**
	MobileNetV2	Baseline		FP32	14.0	307	73.03
		ZeroQ	–	W4A4	1.75	4.80	3.31
		DDAQ (ours)	–	W4A4	1.75	4.80	**49.59**
		GDFQ	✓	W4A4	1.75	4.80	51.30
		DDAQ (ours)	✓	W4A4	1.75	4.80	**52.99**
		ZeroQ	–	W6A6	2.63	10.8	69.62
		DDAQ (ours)	–	W6A6	2.63	10.8	**70.30**
		GDFQ	✓	W6A6	2.63	10.8	70.98
		DDAQ (ours)	✓	W6A6	2.63	10.8	**71.62**

search. Fig. 3 shows the training losses of ResNet-20 on CIFAR-10, where $\mathscr{L}_G$ and $\mathscr{L}_Q$ drop in an adversarial fashion, resulting in increased accuracy.

4.2. Performance test for image classification

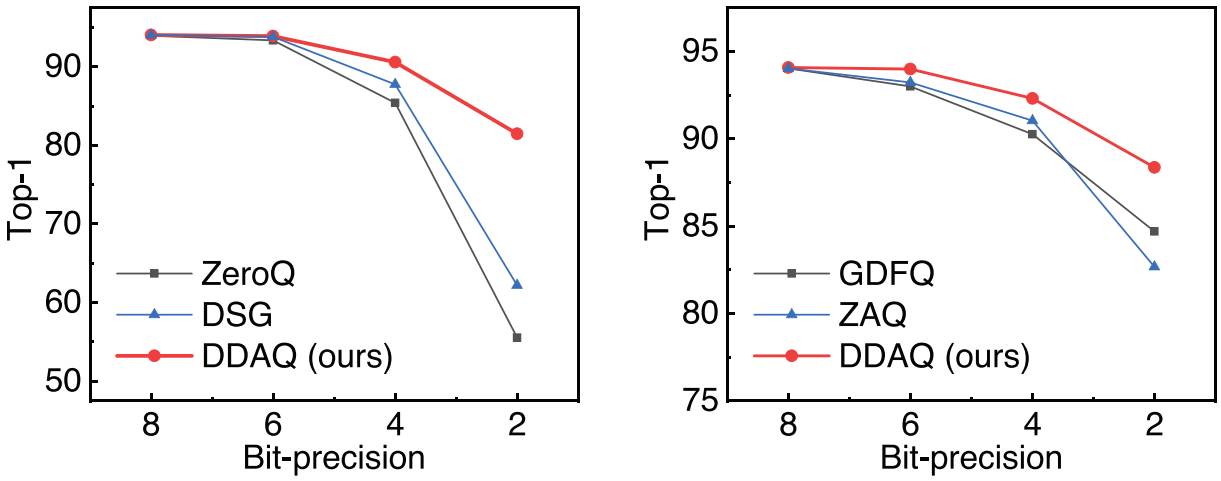
We start by discussing the effectiveness of DDAQ on image classification tasks, where DDAQ is applied to quantify various pre-trained models on CIFAR-10 and ImageNet datasets. We compare DDAQ with the SOTA data-free quantization methods including ZeroQ [26], DSG [27], GDFQ [33], and ZAQ [32] in different configurations (i.e., fine-tuning and bit-precision), and the quantization results are reported in Table 1. For the quantization with W4A4 for ResNet-20 on CIFAR-10, DDAQ improves by 2.95% and 1.38% without and with fine-tuning, respectively. In particular, DDAQ without fine-tuning even obtains 0.45% higher accuracy than GDFQ with fine-tuning.

On the large-scale ImageNet dataset, DDAQ also shows significant advantages for various models in different configurations. For instance, in the case of W4A4 without fine-tuning, DDAQ quantifies ResNet-18 with 58.44% accuracy, which is 1.69× higher than DSG. When quantizing ResNet-50 with W6A6, DDAQ achieves 76.98% accuracy, which is only 0.75% lower than the full-precision model at a 5.3× compression rate of model size. In addition, DDAQ remains robust to low-bit quantization of the light-weight model MobileNetV2. The accuracy of DDAQ with W4A4 is 46.28% and 1.69% higher than ZeroQ and GDFQ without and with fine-tuning, respectively, and DDAQ with W6A6 can obtain a quantized model of size 2.63 MB with 71.62% accuracy.

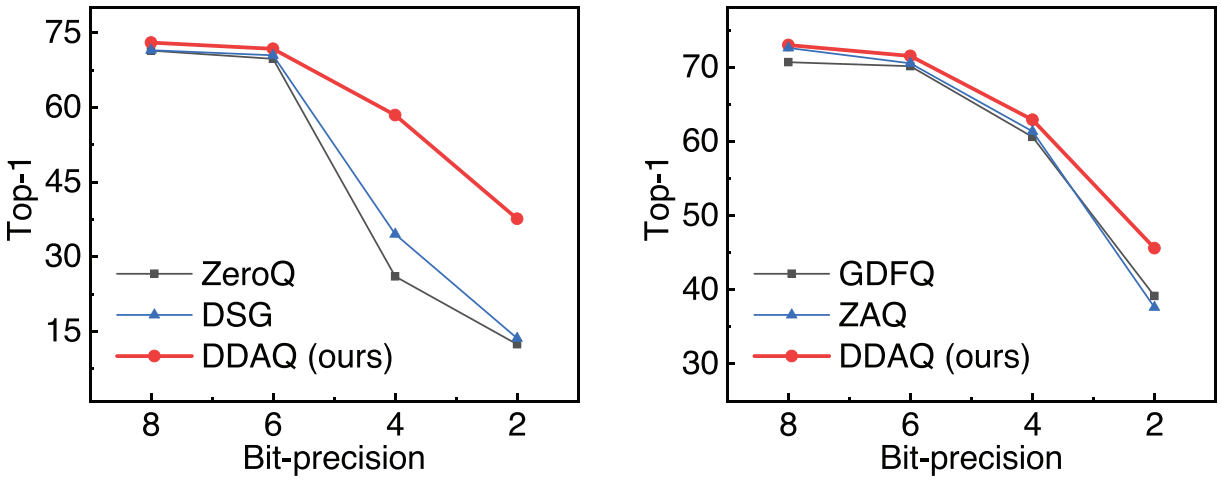
To demonstrate the robustness of DDAQ for quantization bit-precisions, we compare the results of 2-, 4-, 6-, and 8- bit quantization for ResNet-20 on CIFAR-10 and ResNet-18 on ImageNet, as illustrated in Fig. 4. We can clearly see that DDAQ always achieves the highest performance at different bit-precisions without and with fine-tuning. Particularly, the SOTA methods have non-trivial accuracy degradation for low-precision (*e.g.*., 2-bit) quantization, while DDAQ still maintains a satisfactory performance.

4.3. Performance test for object detection

In addition to image classification, object detection is also of significant value in real-world applications. Here, DDAQ is extended to the object detection task to sufficiently demonstrate its generality. Specifically, we evaluate the effectiveness of DDAQ on the SOTA single-stage model RetinaNet [65] with ResNet-50 as the backbone, and the quantization results on COCO dataset are shown in Table 2. For the practical implementation, similar to the classification task, we only calculate the discriminator losses depending on the backbone ResNet-50 and perform knowledge transfer for the backbone network. From the results, in the absence of fine-tuning, DDAQ achieves superior performance over ZeroQ at various quantization bit-precisions, *e.g.*, about 1% mAP improvement at both W4A4 and W4A6 settings. Benefiting from fine-tuning, DDAQ can further improve accuracy, and in particular, DDAQ with W4A6 can achieve comparable results to FQN [49] with W4A4 that requires real data for fine-tuning.



(a) ResNet-20 on CIFAR-10



(b) ResNet-18 on ImageNet

Fig. 4. Comparison of quantization results with different bit-precisions on CIFAR-10 and ImageNet datasets. (left): without fine-tuning; (right): with fine-tuning. Our proposed DDAQ consistently outperforms the SOTA methods at any bit-precision, thus proving its promising robustness and generality.

Table 2
Quantization results of RetinaNet on COCO dataset. Our proposed DDAQ outperforms the SOTA data-free quantization method ZeroQ [26] at all bit-precisions, and the quantization with W4A6 can achieve comparable results to FQN [49] with W4A4, which relies on real data for fine-tuning.

Model	Method	FT	Prec.	Size(MB)	BitOps(T)	mAP
RetinaNet	Baseline		FP32	145	128	73.03
	FQN	✓	W4A4	18.1	1.99	32.5
	ZeroQ	–	W4A4	18.1	1.99	20.3
	DDAQ (ours)	–	W4A4	18.1	1.99	**21.6**
	DDAQ (ours)	✓	W4A4	18.1	1.99	**23.2**
	ZeroQ	–	W4A6	18.1	2.99	30.1
	DDAQ (ours)	–	W4A6	18.1	2.99	**31.0**
	DDAQ (ours)	✓	W4A6	18.1	2.99	**32.3**
	ZeroQ	–	W6A6	27.2	4.49	36.5
	DDAQ (ours)	–	W6A6	27.2	4.49	**36.7**
	DDAQ (ours)	✓	W6A6	27.2	4.49	**37.0**

Table 3
Ablation study on effect of different loss functions of generator G for ResNet-18 with W4A4 on ImageNet.

$\mathscr{L}_{bn}$ (Discriminator 1)	$\mathscr{L}_{adv}$ (Discriminator 2)	FT	Top-1	FT	Top-1
–	–	–	22.11	✓	25.39
✓	–	–	42.82	✓	58.98
–	✓	–	35.76	✓	61.62
✓	✓	–	**58.44**	✓	**62.91**

Table 4
Ablation study on layer importance prior for ResNet-18 with W4A4 on ImageNet.

$\mathbf{X}_1$ (Discriminator 1)	$\mathbf{X}_2$ (Discriminator 2)	FT	Top-1	FT	Top-1
–	–	–	55.65	✓	61.82
✓	–	–	56.72	✓	62.22
–	✓	–	57.05	✓	62.37
✓	✓	–	**58.44**	✓	**62.91**

Table 5
Ablation study on effect of different weighting styles in loss functions of generator G for ResNet-18 with W4A4 on ImageNet.

Discriminator 1	Discriminator 2	FT	Top-1	FT	Top-1
$\mathscr{L}'_{bn}$	$\mathscr{L}'_{adv}$	–	57.30	✓	62.29
$\mathscr{L}'_{bn}$	$\mathscr{L}_{adv}$	–	57.51	✓	62.56
$\mathscr{L}_{bn}$	$\mathscr{L}'_{adv}$	–	58.12	✓	62.72
$\mathscr{L}_{bn}$	$\mathscr{L}_{adv}$	–	**58.44**	✓	**62.91**

4.4. Ablation study

We perform three ablation studies to verify the effect of components of the proposed DDAQ using ResNet-18 with W4A4 on ImageNet dataset. First, the effect of two discriminators on the training of the generator is investigated, as shown in Table 3. It can be seen that in both no fine-tuning and fine-tuning cases, the losses $\mathscr{L}_{bn}$ and $\mathscr{L}_{adv}$ directed by two discriminators both contribute significantly to the final performance. In particular, without fine-tuning, the absence of $\mathscr{L}_{bn}$ and $\mathscr{L}_{adv}$ produces 26.7% and 38.8% accuracy degradation, respectively. From the results, the effects of the two discriminators are superimposed, indicating that the two discriminators are focused on different aspects and are independent.

Second, we evaluate the effect of Hessian-based layer importance prior in both no fine-tuning and fine-tuning cases, and the results are shown in Table 4. As we can see, the average Hessian trace is effective in enhancing the losses of discriminators, especially the enhancement matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ both contribute more than 1% accuracy improvement without fine-tuning. In addition, the enhancements to BN statistics and discrepancies between layers are also superimposable and non-interfering with each other.

Finally, the effect of different weighting styles in loss functions of the generator is evaluated, as shown in Table 5. Specifically, we replace the loss functions with $\mathscr{L}'_{bn}$ and $\mathscr{L}'_{adv}$, where $\mathscr{L}'_{bn}$ is weighted using $\mathbf{X}_2$ and $\mathscr{L}'_{adv}$ is weighted using $\mathbf{X}_1$ as follows:

$$\mathscr{L}_{bn}\prime = \mathbf{L}_1\mathbf{X}_2, \quad \mathscr{L}_{adv}\prime = \frac{1}{N}\cdot \mathbf{1}^T(\mathbf{L}_2\mathbf{X}_1) \tag{18}$$

As we can see, $\mathscr{L}_{bn}$ weighted with $\mathbf{X}_1$ can ensure the sample diversity in the batch while promoting BN distribution matching, which is important for the case without fine-tuning. For instance, replacing to weighting with $\mathbf{X}_2$ leads to about 1% performance degradation. In addition, replacing to $\mathscr{L}'_{adv}$ also produces accuracy loss, thus it is more effective to consider the information interactions between models of the overall sample batch.

5. Conclusions

We have proposed DDAQ, a data-free quantization method that potentially enables high-accuracy model compression without any original training data. DDAQ is performed in an adversarial learning paradigm that alternately trains the generator and quantized model, with two main innovations: first, the training of the generator depends on dual discriminators that facilitate sample distribution matching and model interaction, respectively; second, we introduce Hessian-based layer importance prior to the framework and thus allowing for more diverse sample generation and more effective knowledge transfer. Extensive experiments have been conducted on various model structures for image classification and object detection tasks, and DDAQ consistently outperforms the SOTA methods, especially with 23.91% and 1.57% improvements for low-precision W4A4 quantization of ResNet-18 on ImageNet without and with fine-tuning, respectively, fully demonstrating its effectiveness and generality.

In the future, we plan to explore the BN-free scheme for data-free quantization, which can be applied to vision transformers.

CRediT authorship contribution statement

Zhikai Li: Conceptualization, Methodology, Software. **Liping Ma:** Validation, Writing - original draft. **Xianlei Long:** Formal analysis, Visualization, Investigation. **Junrui Xiao:** Data curation, Writing - original draft. **Qingyi Gu:** Conceptualization, Supervision, Project administration.

Declaration of Competing Interest

The 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.

Acknowledgements

This work was partially supported by the Scientific Instrument Developing Project of the Chinese Academy of Sciences (No. YJKYYQ20200045).

Appendix A. Appendix

The Lemma proved by HAWQ-V2 [34] is shown below, indicating that the average Hessian trace can represent the importance of each layer. Please refer to the original HAWQ-V2 paper [34] for the detailed proof of the Lemma.

First, we assume that the model is twice differentiable and has converged to a local minima W^*. Given this assumption, the Lemma is established as follows.

Lemma: When we quantize two layers (denoted by B_1 and B_2) with same amount of perturbation, namely $||\Delta W_1^*||_2^2 = ||\Delta W_2^*||_2^2$, we will have:

$$\mathscr{L}(W_1^* + \Delta W_1^*, W_2^*, \cdots, W_L^*) \leqslant \mathscr{L}(W_1^*, W_2^* + \Delta W_2^*, W_3^*, \cdots, W_L^*) \tag{19}$$

if

$$\frac{1}{m_1} Tr(H_i) \leqslant \frac{1}{m_2} Tr(H_i). \tag{20}$$

where $\mathscr{L}(\cdot)$ is the loss of the model, $Tr(\cdot)$ is the function of calculating the trace, and $H_i \in \mathbb{R}^{m_i \times m_i}$ is the Hessian matrix of layer i. The layer's average Hessian trace can represent the impact of this layer on the overall performance of the model after being perturbed, thus it can be used as a measure of importance.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[2] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inform. Process. Syst. 25 (2012) 1097–1105.
[3] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1653–1660.
[4] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1137–1149.
[5] F. Shao, L. Chen, J. Shao, W. Ji, S. Xiao, L. Ye, Y. Zhuang, J. Xiao, Deep learning for weakly-supervised object detection and localization: A survey, Neurocomputing 496 (2022) 192–207.
[6] J.-X. Mi, J. Feng, K. Huang, Designing efficient convolutional neural network structure: A survey, Neurocomputing 489 (2022) 139–156.
[7] X. Li, H. Jiang, R. Zhang, F. Tian, S. Huang, D. Xu, Robustness-aware 2-bit quantization with real-time performance for neural network, Neurocomputing 455 (2021) 12–22.
[8] S. Han, H. Mao, W.J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint arXiv:1510.00149.
[9] T. Choudhary, V. Mishra, A. Goswami, J. Sarangapani, A comprehensive survey on model compression and acceleration, Artif. Intell. Rev. 53 (7) (2020) 5113–5155.
[10] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531.
[11] Z. Li, L. Ma, X. Long, Y. Chen, H. Deng, F. Yan, Q. Gu, Hardware-oriented algorithm for high-speed laser centerline extraction based on hessian matrix, IEEE Trans. Instrum. Meas. 70 (2021) 1–14.
[12] C. Zhu, S. Han, H. Mao, W.J. Dally, Trained ternary quantization, arXiv preprint arXiv:1612.01064.
[13] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, Z. Zhang, Improving neural network quantization without retraining using outlier channel splitting, in: International conference on machine learning, PMLR, 2019, pp. 7543–7552.
[14] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, X.-S. Hua, Quantization networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7308–7316.
[15] T. Liang, J. Glossner, L. Wang, S. Shi, X. Zhang, Pruning and quantization for deep neural network acceleration: A survey, Neurocomputing 461 (2021) 370–403.
[16] J. Achterhold, J.M. Koehler, A. Schmeink, T. Genewein, Variational network quantization, in: International Conference on Learning Representations, 2018.
[17] A. Gholami, S. Kim, Z. Dong, Z. Yao, M.W. Mahoney, K. Keutzer, A survey of quantization methods for efficient neural network inference, arXiv preprint arXiv:2103.13630.
[18] P. Peng, M. You, W. Xu, J. Li, Fully integer-based quantization for mobile convolutional neural network inference, Neurocomputing 432 (2021) 194–205.
[19] L. Enderich, F. Timm, W. Burgard, Symog: Learning symmetric mixture of gaussian modes for improved fixed-point quantization, Neurocomputing 416 (2020) 310–315.
[20] D. Zhang, J. Yang, D. Ye, G. Hua, Lq-nets: Learned quantization for highly accurate and compact deep neural networks, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 365–382.
[21] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, Y. Zou, Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients, arXiv preprint arXiv:1606.06160.
[22] R. Krishnamoorthi, Quantizing deep convolutional networks for efficient inference: A whitepaper, arXiv preprint arXiv:1806.08342.
[23] H. Yin, P. Molchanov, J.M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N.K. Jha, J. Kautz, Dreaming to distill: Data-free knowledge transfer via deepinversion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8715–8724.
[24] H. Chen, Y. Wang, C. Xu, Z. Yang, C. Liu, B. Shi, C. Xu, C. Xu, Q. Tian, Data-free learning of student networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3514–3522.
[25] H. Zhao, X. Sun, J. Dong, H. Yu, H. Zhou, Dual discriminator adversarial distillation for data-free model compression, arXiv preprint arXiv:2104.05382.
[26] Y. Cai, Z. Yao, Z. Dong, A. Gholami, M.W. Mahoney, K. Keutzer, Zeroq: A novel zero shot quantization framework, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13169–13178.
[27] X. Zhang, H. Qin, Y. Ding, R. Gong, Q. Yan, R. Tao, Y. Li, F. Yu, X. Liu, Diversifying sample generation for accurate data-free quantization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15658–15667.
[28] Z. Li, L. Ma, M. Chen, J. Xiao, Q. Gu, Patch similarity aware data-free quantization for vision transformers, arXiv preprint arXiv:2203.02250.
[29] M. Nagel, M. v. Baalen, T. Blankevoort, M. Welling, Data-free quantization through weight equalization and bias correction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1325–1334.
[30] R. Banner, Y. Nahshan, E. Hoffer, D. Soudry, Aciq: Analytical clipping for integer quantization of neural networks, arXiv preprint arXiv:1810.05723.
[31] M. Nagel, R.A. Amjad, M. Van Baalen, C. Louizos, T. Blankevoort, Up or down? adaptive rounding for post-training quantization, in: International Conference on Machine Learning, PMLR, 2020, pp. 7197–7206.
[32] Y. Liu, W. Zhang, J. Wang, Zero-shot adversarial quantization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1512–1521.
[33] S. Xu, H. Li, B. Zhuang, J. Liu, J. Cao, C. Liang, M. Tan, Generative low-bitwidth data free quantization, European Conference on Computer Vision, Springer (2020) 1–17.
[34] Z. Dong, Z. Yao, Y. Cai, D. Arfeen, A. Gholami, M.W. Mahoney, K. Keutzer, Hawq-v2: Hessian aware trace-weighted quantization of neural networks, arXiv preprint arXiv:1911.03852.
[35] K. Wang, Z. Liu, Y. Lin, J. Lin, S. Han, Haq: Hardware-aware automated quantization with mixed precision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8612–8620.
[36] H. Yang, L. Duan, Y. Chen, H. Li, Bsq: Exploring bit-level sparsity for mixed-precision neural network quantization, arXiv preprint arXiv:2102.10462.
[37] H. Yu, Q. Han, J. Li, J. Shi, G. Cheng, B. Fan, Search what you want: Barrier panelty nas for mixed precision quantization, European Conference on Computer Vision, Springer (2020) 1–16.
[38] N. Morgan, et al., Experimental determination of precision requirements for back-propagation training of artificial neural networks, in: Proc. Second Int'l. Conf. Microelectronics for Neural Networks, Citeseer, 1991, pp. 9–16.
[39] T.-W. Chin, I. Pierce, J. Chuang, V. Chandra, D. Marculescu, One weight bitwidth to rule them all, European Conference on Computer Vision, Springer (2020) 85–103.
[40] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio, Binarized neural networks, arXiv preprint arXiv:1602.02505.
[41] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnor-net: Imagenet classification using binary convolutional neural networks, in: European conference on computer vision, Springer, 2016, pp. 525–542.
[42] E. Park, S. Yoo, P. Vajda, Value-aware quantization for training and inference of neural networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 580–595.
[43] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko, Quantization and training of neural networks for efficient integer-arithmetic-only inference, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2704–2713.
[44] Z. Li, Q. Gu, I-vit: Integer-only quantization for efficient vision transformer inference, arXiv preprint arXiv:2207.01405.
[45] Y. Bengio, N. Léonard, A. Courville, Estimating or propagating gradients through stochastic neurons for conditional computation, arXiv preprint arXiv:1308.3432.
[46] S.K. Esser, J.L. McKinstry, D. Bablani, R. Appuswamy, D.S. Modha, Learned step size quantization, arXiv preprint arXiv:1902.08153.
[47] J. Choi, Z. Wang, S. Venkataramani, P.I.-J. Chuang, V. Srinivasan, K. Gopalakrishnan, Pact: Parameterized clipping activation for quantized neural networks, arXiv preprint arXiv:1805.06085.
[48] Y. Li, X. Dong, W. Wang, Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks, arXiv preprint arXiv:1909.13144.

[49] R. Li, Y. Wang, F. Liang, H. Qin, J. Yan, R. Fan, Fully quantized network for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2810–2819.
[50] A.T. Elthakeb, P. Pilligundla, F. Mireshghallah, T. Elgindi, C.-A. Deledalle, H. Esmaeilzadeh, Gradient-based deep quantization of neural networks through sinusoidal adaptive regularization, arXiv preprint arXiv:2003.00146.
[51] M. Naumov, U. Diril, J. Park, B. Ray, J. Jablonski, A. Tulloch, On periodic functions as regularizers for quantization of neural networks, arXiv preprint arXiv:1811.09862.
[52] A. Zhou, A. Yao, Y. Guo, L. Xu, Y. Chen, Incremental network quantization: Towards lossless cnns with low-precision weights, arXiv preprint arXiv:1702.03044.
[53] H. Avron, S. Toledo, Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix, J. ACM (JACM) 58 (2) (2011) 1–34.
[54] Z. Yao, A. Gholami, K. Keutzer, M.W. Mahoney, Pyhessian: Neural networks through the lens of the hessian, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 581–590.
[55] C. Sarvani, M. Ghorai, S.R. Dubey, S.S. Basha, Hrel: Filter pruning based on high relevance between activation maps and class labels, Neural Networks 147 (2022) 186–197.
[56] Q. Bi, K. Qin, H. Zhang, G.-S. Xia, Local semantic enhanced convnet for aerial scene recognition, IEEE Trans. Image Process. 30 (2021) 6498–6511.
[57] J. Gu, Y. Shen, B. Zhou, Image processing using multi-code gan prior, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 3012–3021.
[58] C. Fosco, V. Casser, A.K. Bedi, P. O'Donovan, A. Hertzmann, Z. Bylinskii, Predicting visual importance across graphic design types, in: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, 2020, pp. 249–260.
[59] Q. Bi, H. Zhang, K. Qin, Multi-scale stacking attention pooling for remote sensing scene classification, Neurocomputing 436 (2021) 147–161.
[60] Z. Gao, L. Wang, G. Wu, Lip: Local importance-based pooling, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3355–3364.
[61] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems (2014) 2672–2680.
[62] T.D. Nguyen, T. Le, H. Vu, D. Phung, Dual discriminator generative adversarial nets, arXiv preprint arXiv:1709.03831.
[63] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[64] M. Sandler, A.G. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation, arXiv preprint arXiv:1801.04381.
[65] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
[66] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009) 1–60.
[67] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vision 115 (3) (2015) 211–252.
[68] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, C.L. Zitnick, Microsoft coco captions: Data collection and evaluation server, arXiv preprint arXiv:1504.00325.
[69] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
[70] Z. Li, Q. Gu, I-ViT: integer-only quantization for efficient vision transformer inference, arXiv preprint arXiv:2207.01405.

Zhikai Li received the B.Sc. degree from Dalian University of Technology, Dalian, China, in 2020. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His current research interests include computer vision and efficient deep learning.

Liping Ma received the B.Sc. degree from Hunan Agricultural University, Hunan, China, in 2007 and the Ph.D. degree in mechanical engineering from Beijing Institute of Technology, Beijing, in 2015. He is currently an Associate Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include robot design, electromechanical system design, visual measurement and application.

Xianlei Long received the B.Sc. degree in Electrical Engineering from China University of Mining and Technology, Jiangsu, China, in 2017. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His research interests include high-speed image processing and computer vision.

Junrui Xiao received the B.Sc. degree from Xidian University, Shaanxi, China, in 2020. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His current research interests include computer vision and model compression.

Qingyi Gu (M'13) received the B.E. degree in Electronic and Information Engineering from Xi'an Jiaotong University, China, in 2005. He received the M.E. degree, and Ph.D. degree in Engineering, Hiroshima University, Japan, in 2010, and 2013 respectively. He is currently a professor with the Institute of Automation, Chinese Academy of Sciences, China. His primary research interest is high-speed image processing, and applications in industry and biomedicine.