Pattern Recognition 143 (2023) 109780

Contents lists available at ScienceDirect

Pattern Recognition

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

Data-free quantization via mixed-precision compensation without fine-tuning

Jun Chen[a,1], Shipeng Bai[a,1], Tianxin Huang[a], Mengmeng Wang[a], Guanzhong Tian[b,*], Yong Liu[a,*]

[a] Institute of Cyber-Systems and Control, Zhejiang University, China
[b] Ningbo Innovation Center, Zhejiang University, China

ARTICLE INFO

Article history:
Received 24 October 2022
Revised 27 May 2023
Accepted 25 June 2023
Available online 26 June 2023

Keywords:
Neural network compression
Date-free quantization

ABSTRACT

Neural network quantization is a very promising solution in the field of model compression, but its resulting accuracy highly depends on a training/fine-tuning process and requires the original data. This not only brings heavy computation and time costs but also is not conducive to privacy and sensitive information protection. Therefore, a few recent works are starting to focus on data-free quantization. However, data-free quantization does not perform well while dealing with ultra-low precision quantization. Although researchers utilize generative methods of synthetic data to address this problem partially, data synthesis needs to take a lot of computation and time. In this paper, we propose a data-free mixed-precision compensation (DF-MPC) method to recover the performance of an ultra-low precision quantized model without any data and fine-tuning process. By assuming the quantized error caused by a low-precision quantized layer can be restored via the reconstruction of a high-precision quantized layer, we mathematically formulate the reconstruction loss between the pre-trained full-precision model and its layer-wise mixed-precision quantized model. Based on our formulation, we theoretically deduce the closed-form solution by minimizing the reconstruction loss of the feature maps. Since DF-MPC does not require any original/synthetic data, it is a more efficient method to approximate the full-precision model. Experimentally, our DF-MPC is able to achieve higher accuracy for an ultra-low precision quantized model compared to the recent methods without any data and fine-tuning process.

1. Introduction

In order to realize the deployment of deep neural networks on resource-constrained lightweight devices, a series of remarkable neural network compression techniques are gradually developing, including low-rank factorization [1], parameter and filters pruning [2–5], quantization [6–9] and knowledge distillation [10–13]. Among these neural network compression techniques, quantization is viewed as a more suitable scheme for hardware acceleration [14,15] than pruning and knowledge distillation. In this sense, this paper will focus on quantization.

Quantization can be divided into data-driven quantization and data-free quantization [16–19] according to whether it depends on the data. And data-driven quantization can be further subdivided into quantization-aware training [7,14,20,21] and post-training quantization [22–24] according to whether it depends on training/fine-tuning. However, the original training data is not always easily accessible, especially for privacy, security, and deployment in the field. Therefore, data-free quantization is a vital research direction to achieve a low-precision model without any original data and training.

The accuracy drop of data-free quantization is particularly dramatic when focusing on the ultra-low precision model. Thus, researchers are starting to utilize generative methods [17,25,26] to generate synthetic samples that resemble the distribution of the original dataset and achieve high accuracy. However, generative methods need to cost a lot of computation and time to synthesize data, which conflicts with the concept of data-free.

In this paper, we abandon the idea of data synthesis and restore the quantized error caused by the ultra-low precision quantization from the perspective of compensation. Inspired by a few works [27–29], we propose a data-free mixed-precision compensation (DF-MPC) method to achieve higher accuracy for an ultra-low precision quantization without any data and fine-tuning process, as

* Corresponding authors.
E-mail addresses: junc@zju.edu.cn (J. Chen), shipengbai@zju.edu.cn (S. Bai), 21725129@zju.edu.cn (T. Huang), mengmengwang@zju.edu.cn (M. Wang), gztian@zju.edu.cn (G. Tian), yongliu@iipc.zju.edu.cn (Y. Liu).
[1] Both authors contributed equally to this work.

https://doi.org/10.1016/j.patcog.2023.109780
0031-3203/

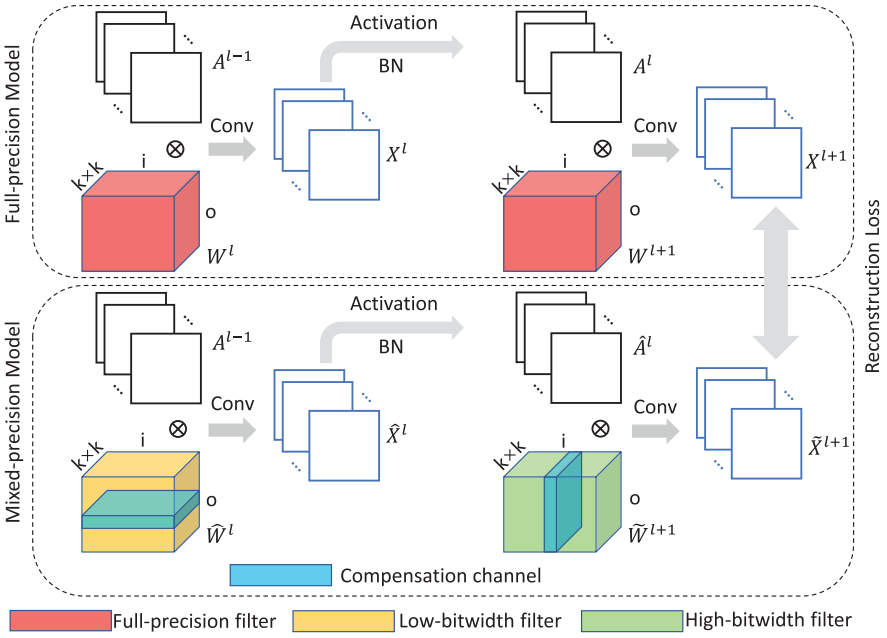


Fig. 1. The overview of our DF-MPC method, where the filter in the lth layer is quantized to low-bitwidth and the filter in the $(l+1)$th layer is quantized to high-bitwidth. The output of $(l+1)$th convolutional layer can be restored by multiplying the compensation coefficient with respect to the input channel of the high-bitwidth filter, which is equivalent to multiplying the compensation coefficient with respect to the output channel of the low-bitwidth filter. Note that the reconstruction loss is the output difference of $(l+1)$th layer from the pre-trained full-precision model and its layer-wise mixed-precision quantized model.

depicted in Fig. 1. In summary, we make three main contributions, as shown below:

- In two adjacent layers of a neural network, we assume that the quantized error caused by a low-precision quantized layer can be restored via the reconstruction of a high-precision quantized layer. Specifically, we quantize the weights in one layer into low precision values (e.g., 2-bit) and then recover the performance by reconstructing relatively higher precision (e.g., 6-bit) weights in the next layer. The layer-wise mixed-precision compensation assumption is described in Section 4.1.
- Based on the mixed-precision compensation assumption, we mathematically formulate the reconstruction loss between the pre-trained full-precision model and its mixed-precision quantized model. Without any fine-tuning process and original/synthetic data, we can achieve layer-wise mixed-precision quantization (e.g., 2/6-bit) only relying on our compensation method. The reconstruction loss is formulated in Section 4.2.
- Based on the reconstruction loss, we theoretically deduce the closed-form solution by minimizing the reconstruction loss of the feature maps to restore the quantized error caused by the low precision weight. The global minimum is solved in Section 4.3. Furthermore, we verify the effectiveness of our compensation method through experiments on multiple datasets (CIFAR10, CIFAR100, and ImageNet) with multiple network structures (ResNet, DenseNet121, VGG16, and MobileNetV2).

2. Related work

Quantization is a kind of model compression method, which accelerates the forward inference phase by converting a full-precision model to a low-precision model (with respect to weights or activations). Whether the low-precision model needs any data or fine-tuning, quantization can continue to be subdivided into the following three classes.

2.1. Quantization-aware training (QAT)

Since the low-precision representations of weights and activations will cause an accuracy drop, quantization-aware training (QAT) aims to reduce the accuracy drop by retraining or fine-tuning the low-precision with training/validation data [14,20]. Especially for the ultra-low precision (e.g., binary [7,21] and ternary [30,31]), QAT can also obtain a satisfactory quantized model.

However, the training process for QAT is computationally expensive and time-consuming. Specifically, the training time and memory of QAT far exceed full precision model training due to simulating quantization operators [32]. On the other hand, in some private or secure situations, the original training/validation data is not easy to access.

2.2. Post-training quantization (PTQ)

Post-training quantization (PTQ) aims to obtain an accurate low-precision model without any fine-tuning process. Therefore, PTQ requires relatively less computation and time consumption than QAT. Specifically, Banner et al. [22] proposed the 4-bit post-training quantization method that introduces a per-channel allocation and bias-correction, and approximates the optimal clipping value analytically from the distribution of the tensor. Zhao et al. [23] proposed outlier channel splitting that requires no additional training and works on commodity hardware. Nagel et al. [24] found a good solution to the per-layer weight-rounding mechanism via a continuous relaxation, but this method still requires a small amount of unlabelled data.

Since QAT is fully trained on the entire training data, PTQ's performance tends to be inferior to QAT's regardless of the bit width quantization, which is also the bottleneck of PTQ. And compared to QAT, PTQ is also still not completely free from the original data dependence.

2.3. Data-free quantization (DFQ)

Compared to QAT and PTQ, data-free quantization (DFQ) requires neither training/validation data nor fine-tuning/training process. In particular, Nagel et al. [16] could greatly recover the accuracy of low-precision models by applying weight equalization and bias correction. However, it suffers a huge accuracy drop while the bit width is less than 6-bit. Cai et al. [17] utilized synthetic data to achieve mixed-precision quantization, but it is also difficult to deal with the accuracy drop below 4-bit.

Recently, researches on DFQ seem to turn to data sampling and generation. Zhang et al. [18] proposed a sample generation method that enhances the diversity of data by slacking the alignment of feature statistics in the BN layer and designing a layerwise enhancement. Choi et al. [19] proposed a method that uses superposed latent embeddings to generate synthetic boundary supporting samples, and confirmed that samples near the boundary can improve the performance of a low-precision model. Although DFQ based on data synthesis does not use the original training/validation data, it costs a lot of computation and time to synthesize the data.

3. Problem formulation of data-free quantization

In this section, we present the problem of data-free quantization with the corresponding full-precision pre-trained model.

3.1. Background and notations

Given a neural network model with L layers, we denote $\mathcal{W}^l \in \mathbb{R}^{o\times i\times k\times k}$ and $\mathcal{A}^{l-1} \in \mathbb{R}^{i\times w\times h}$ as the weight in the lth layer and activation in the $(l-1)$th layer, where o represents the size of output channels, i represents the size of input channels, $k\times k$ is the size of kernel filters and $w\times h$ is the size of activation maps. Then we obtain the feature maps $\mathcal{X}^l \in \mathbb{R}^{o\times w\times h}$

$$\mathcal{X}^l = \mathcal{W}^l \otimes \mathcal{A}^{l-1}, \tag{1}$$

where $\otimes$ is the standard convolution operation. By introducing the activation function f and a batch normalization BN, we can finally output the activation map based on the feature map

$$\mathcal{A}^l = f\big(\mathrm{BN}\left(\mathcal{X}^l\right)\big). \tag{2}$$

Subsequently, we consider the ternary weight tensor in the lth layer that consists of three quantized values $\{-1, 0, +1\}$ and a scaling factor α^l

$$\hat{\mathcal{W}}^l = \begin{cases} +1, & \text{if } \mathcal{W}^l > \Delta^l \\ 0, & \text{if } \left|\mathcal{W}^l\right| \le \Delta^l \\ -1, & \text{if } \mathcal{W}^l < -\Delta^l \end{cases}. \tag{3}$$

Based on Ternary Weight Networks [30], we can obtain the optimized layer-wise values of the threshold Δ^l and the scaling factor α^l

$$\begin{aligned} \Delta^l &= 0.7\mathbb{E}\big(\left|\mathcal{W}^l\right|\big) \\ \alpha^l &= \mathop{\mathbb{E}}_{j\in\left\{j\left|\mathcal{W}^l(j)\right|>\Delta^l\right\}}\big(\left|\mathcal{W}^l(j)\right|\big). \end{aligned} \tag{4}$$

Since the layer-wise scaling factor α^l can be absorbed into a batch normalization, we can omit α^l and use Eq. (3) to represent the ternary weight tensor directly.

The new feature map $\hat{\mathcal{X}}^l$ will deviate from the original feature map $\mathcal{X}^l$ when we consider the quantization of the weight tensor, resulting in a rapid accuracy drop of the neural network without fine-tuning, i.e.,

$$\hat{\mathcal{X}}^l = \hat{\mathcal{W}}^l \otimes \mathcal{A}^{l-1} \neq \mathcal{X}^l. \tag{5}$$

3.2. Problem statement

Therefore, we consider reconstructing the weight tensor in the next layer to compensate the feature map in the next layer such that we can recover the accuracy of the low-precision model. Note that we choose a relatively high-precision quantization for the weight tensor of the next layer $\tilde{\mathcal{W}}^{l+1}$ because it is required to compensate the quantized error caused by $\hat{\mathcal{W}}^l$ as much as possible. And we can apply the uniform quantization with k-bit based on DoReFa-Net [20]

$${}^k\mathcal{Q}(\cdot) = \frac{2}{2^k-1}\,\mathrm{round}\left[(2^k-1)\left(\frac{\cdot}{2\max|\cdot|}+\frac{1}{2}\right)\right]-1. \tag{6}$$

Similarly, we omit the layer-wise scaling factor $\max|\cdot|$ as it can be absorbed into a batch normalization. And we need to make the reconstruction loss between the new feature map and the original feature map as small as possible.

By introducing the coefficient vector $\mathbf{c} = [c_1, c_2, \ldots, c_i]^T \ge 0$ whose each component corresponds each input channel of the weight tensor in the $(l+1)$th layer, we give the jth channel of reconstructed weight tensor as follows:

$$\tilde{\mathcal{W}}_j^{l+1} = c_j \cdot {}^k\mathcal{Q}\big(\mathcal{W}_j^{l+1}\big). \tag{7}$$

We hope to find an optimal $\mathbf{c}$ such that the reconstructed feature map $\tilde{\mathcal{X}}_t^{l+1}$ is close to the original feature map $\mathcal{X}_t^{l+1}$, i.e.,

$$\begin{aligned} \tilde{\mathcal{X}}_t^{l+1} &= \tilde{\mathcal{W}}_{t,j}^{l+1} \otimes \hat{\mathcal{A}}_j^l + \sum_{m=1,m\neq j}^{i} \mathcal{W}_{t,m}^{l+1} \otimes \mathcal{A}_m^l \\ &\approx \mathcal{X}_t^{l+1} = \sum_{m=1}^{i} \mathcal{W}_{t,m}^{l+1} \otimes \mathcal{A}_m^l. \end{aligned} \tag{8}$$

where the shapes of the weight and activation tensors are $o\times i\times k\times k$ and $i\times w\times h$, respectively. As a result, $\tilde{\mathcal{X}}_t^{l+1}$ and $\mathcal{X}_t^{l+1}$ indicate the tth output channel of the feature map. Note that when we use the same notation j or m to indicate the channel of the weight and activation, it means that their dimensions are the same and correspond to each other in the computation.

Consequently, our problem aims to find a coefficient vector $\mathbf{c}$ to minimize the reconstruction loss based on a full-precision pre-trained model $\mathcal{W}$ without any training process and data, i.e.,

$$\min_{\mathbf{c}} \sum_{t=1}^{o} \|\tilde{\mathcal{X}}_t^{l+1} - \mathcal{X}_t^{l+1}\|_2^2, \tag{9}$$

Note that we apply the mixed-precision quantization, i.e., one layer low-bitwidth (ternary) and one layer high-bitwidth that is used for compensation. The mixed-precision structures of some main deep neural networks are shown in Fig. 2.

Although we consider restoring the quantized error for the ternary values, our method is not limited to the ternary case, but is also applicable to higher precision case (even the same as the precision of the quantized filter). For example, we have different mixed-precisions, such as 2/6-bit, 3/6-bit, 6/6-bit etc. Note that in this paper, we use the ternary filter just to distinguish it from the quantized filter.

4. Proposed method of mixed-precision compensation

In this section, we theoretically give the layer-wise mixed-precision compensation assumption for the reconstruction loss of Eq. (9). According to this assumption, we present our data-free

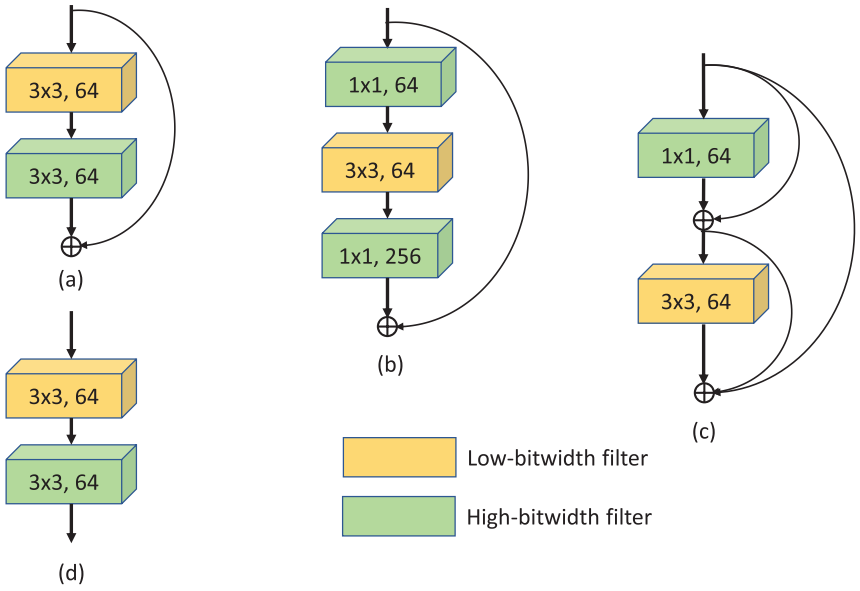


Fig. 2. The layer-wise mixed-precision structures of some main deep neural networks. (a): a building block for ResNet18/ResNet34. (b): a bottleneck block for ResNet50/ResNet101. (c): a dense block for DenseNet. (d): a building block for deep neural networks.

mixed-precision compensation method to recover the accuracy of the low-precision neural network.

4.1. Compensation assumption

In order to minimize Eq. (9) without any data and fine-tuning process, we assume that the quantized error of each filter with low-bitwidth can be partly compensated by reconstructing filters with high-bitwidth in the next layer. Then we further assume that the reconstructed filter consists of a linear combination of the high-bitwidth filters and the coefficient value, which is defined as Eq. (7).

Assumption 1. In order to minimize Eq. (9) with a data-free version, we propose a one-to-one channel-wise compensation assumption that the quantized error caused by the low-bitwidth quantization of each channel of the filter can be compensated by the high-bitwidth quantization of the corresponding channel of the filter in the next layer.

Without loss of generality, let the filter of the lth layer be quantized to low-bitwidth (ternary) such that the tth channel of the reconstruction loss in the $(l+1)$th layer can be represented as

$$\begin{aligned}
&\tilde{\mathcal{X}}_t^{l+1} - \mathcal{X}_t^{l+1} = \tilde{\mathcal{W}}_{t,j}^{l+1} \otimes \hat{\mathcal{A}}_j^l - \mathcal{W}_{t,j}^{l+1} \otimes \mathcal{A}_j^l \\
&= c_j \cdot {}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) \otimes \hat{\mathcal{A}}_j^l - \mathcal{W}_{t,j}^{l+1} \otimes \mathcal{A}_j^l \\
&= c_j \cdot \left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) - \mathcal{W}_{t,j}^{l+1}\right) \otimes \hat{\mathcal{A}}_j^l + c_j \cdot \mathcal{W}_{t,j}^{l+1} \otimes \hat{\mathcal{A}}_j^l - \mathcal{W}_{t,j}^{l+1} \otimes \mathcal{A}_j^l \\
&= c_j \cdot \left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) - \mathcal{W}_{t,j}^{l+1}\right) \otimes \hat{\mathcal{A}}_j^l + \mathcal{W}_{t,j}^{l+1} \otimes (c_j \cdot \hat{\mathcal{A}}_j^l - \mathcal{A}_j^l).
\end{aligned} \tag{10}$$

Note that the lth output channel size of $\mathcal{A}$ and $\mathcal{W}$ is equal to the $(l+1)$th input channel size of $\mathcal{W}$. For brevity, we first omit the activation function f and a batch normalization BN. Then the equations $\hat{\mathcal{A}}_j^l = \hat{\mathcal{X}}_j^l$ and $\mathcal{A}_j^l = \mathcal{X}_j^l$ hold. By introducing the two formulas

$$\begin{aligned}
\hat{\mathcal{A}}_j^l &= \hat{\mathcal{X}}_j^l = \textstyle\sum_{m=1}^i \hat{\mathcal{W}}_{j,m}^l \otimes \mathcal{A}_m^{l-1} \\
\mathcal{A}_j^l &= \mathcal{X}_j^l = \textstyle\sum_{m=1}^i \mathcal{W}_{j,m}^l \otimes \mathcal{A}_m^{l-1}.
\end{aligned} \tag{11}$$

Theorem 1. *If there is no batch normalization and activation function between a feature map and its activation map based on Eq. (10) and (11), the reconstruction loss in the tth channel can be formulated as follows:*

$$\begin{aligned}
\tilde{\mathcal{X}}_t^{l+1} - \mathcal{X}_t^{l+1} = c_j \cdot \left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) - \mathcal{W}_{t,j}^{l+1}\right) \otimes \left(\sum_{m=1}^i \hat{\mathcal{W}}_{j,m}^l \otimes \mathcal{A}_m^{l-1}\right) \\
+ \mathcal{W}_{t,j}^{l+1} \otimes \left[\sum_{m=1}^i \left(c_j \cdot \hat{\mathcal{W}}_{j,m}^l - \mathcal{W}_{j,m}^l\right) \otimes \mathcal{A}_m^{l-1}\right].
\end{aligned} \tag{12}$$

Proof. See Appendix 4.4. □

For the term $\left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) - \mathcal{W}_{t,j}^{l+1}\right)$ of the above equation, its value is determined. Since ${}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right)$ has a relatively high-bitwidth, the value of this item is actually very small. When considering minimizing $\|\tilde{\mathcal{X}}_t^{l+1} - \mathcal{X}_t^{l+1}\|$, for the first row, we have a small constraint on c_j, i.e., a regularization term $\|\mathbf{c}\|$.

On the other hand, for the second row of the above equation, we can minimize the term $\|\sum_{m=1}^i \left(c_j \cdot \hat{\mathcal{W}}_{j,m}^l - \mathcal{W}_{j,m}^l\right) \otimes \mathcal{A}_m^{l-1}\|$ because the term $\mathcal{W}_{t,j}^{l+1}$ comes from the full-precision pre-trained model that is invariable.

In summary, we prioritize minimizing the equation

$$\|\sum_{m=1}^i \left(c_j \cdot \hat{\mathcal{W}}_{j,m}^l - \mathcal{W}_{j,m}^l\right) \otimes \mathcal{A}_m^{l-1}\|_2^2, \tag{13}$$

since $\|c_j\|$ is less restrictive than the above equation.

4.2. Data-free compensation

We now introduce a batch normalization BN with two statistics (scale γ and shift β) and two trainable quantities (mean μ and variance σ^2) [33]. By omitting the activation function f, we have the following two equations:

$$\begin{aligned}
\hat{\mathcal{A}}_j^l &= \mathrm{BN}\left(\hat{\mathcal{X}}_j^l\right) = \hat{\gamma}_j \frac{\hat{\mathcal{X}}_j^l - \hat{\mu}_j}{\hat{\sigma}_j} + \hat{\beta}_j \\
\mathcal{A}_j^l &= \mathrm{BN}\left(\mathcal{X}_j^l\right) = \gamma_j \frac{\mathcal{X}_j^l - \mu_j}{\sigma_j} + \beta_j.
\end{aligned} \tag{14}$$

Lemma 1. *If there is only batch normalization between a feature map and its activation map based on Eqs. (10) and (14), the reconstruction*

loss in the tth channel can be formulated as follows:

$$\begin{aligned}&\tilde{\mathcal{X}}_t^{l+1}-\mathcal{X}_t^{l+1}\\&=c_j\cdot\left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right)-\mathcal{W}_{t,j}^{l+1}\right)\otimes\hat{\mathcal{A}}_j^l+\mathcal{W}_{t,j}^{l+1}\otimes(c_j\cdot\hat{\mathcal{A}}_j^l-\mathcal{A}_j^l),\end{aligned}\tag{15}$$

where

$$\begin{aligned}c_j\cdot\hat{\mathcal{A}}_j^l-\mathcal{A}_j^l=&\sum_{m=1}^{i}\left(\frac{c_j\hat{\gamma}_j\cdot\hat{\mathcal{W}}_{j,m}^l}{\hat{\sigma}_j}-\frac{\gamma_j\cdot\mathcal{W}_{j,m}^l}{\sigma_j}\right)\otimes\mathcal{A}_m^{l-1}\\&+\left(\frac{\gamma_j}{\sigma_j}\mu_j-\frac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\hat{\mu}_j\right)+(c_j\hat{\beta}_j-\beta_j).\end{aligned}\tag{16}$$

Proof. See Appendix 4.4. □

In order to minimize the reconstruction loss $\sum_{t=1}^{o}\|\tilde{\mathcal{X}}_t^{l+1}-\mathcal{X}_t^{l+1}\|_2^2$, we analyse that most of this loss actually come from the term $\|c_j\cdot\hat{\mathcal{A}}_j^l-\mathcal{A}_j^l\|_2^2$ based on Eq. (10). For the expansion of this term, it is actually the above equation. And the summation term occupies a large proportion of Eq. (16), which can be represented as:

$$\min_{c_j}\left\|\left(\frac{c_j\hat{\gamma}_j\cdot\hat{\mathcal{W}}_j^l}{\hat{\sigma}_j}-\frac{\gamma_j\cdot\mathcal{W}_j^l}{\sigma_j}\right)\otimes\mathcal{A}^{l-1}\right\|_2^2.\tag{17}$$

Since $\mathcal{A}^{l-1}$ cannot be accessed without data, we can only minimize the other part of the above equation, i.e.,

$$\min_{c_j}\left\|\left(\frac{c_j\hat{\gamma}_j\cdot\hat{\mathcal{W}}_j^l}{\hat{\sigma}_j}-\frac{\gamma_j\cdot\mathcal{W}_j^l}{\sigma_j}\right)\right\|_2^2.\tag{18}$$

Furthermore, we also introduce the activation function to consider the complete compensation process, i.e., $\hat{\mathcal{A}}_j^l=f(\mathrm{BN}(\hat{\mathcal{X}}_j^l))$ and $\mathcal{A}_j^l=f(\mathrm{BN}(\mathcal{X}_j^l))$. Note that the activation function is generally ReLU.

Lemma 2. *If there are both batch normalization and a* ReLU *function between a feature map and its activation map, the reconstruction loss is the same as in Lemma 1 where the upper bound of $c_j\cdot\hat{\mathcal{A}}_j^l-\mathcal{A}_j^l$ is given by Eq. (16), i.e.,*

$$|c_j\cdot\hat{\mathcal{A}}_j^l-\mathcal{A}_j^l|\le|c_j\cdot\mathrm{BN}(\hat{\mathcal{X}}_j^l)-\mathrm{BN}(\mathcal{X}_j^l)|\tag{19}$$

Proof. See Appendix 4.4. □

For brevity, let us use some variable substitution based on Eq. (16):

$$\begin{cases}\Gamma=\frac{c_j\hat{\gamma}_j\cdot\hat{\mathcal{W}}_j^l}{\hat{\sigma}_j}-\frac{\gamma_j\cdot\mathcal{W}_j^l}{\sigma_j}\\\Theta=\left(\frac{\gamma_j}{\sigma_j}\mu_j-\frac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\hat{\mu}_j\right)+(c_j\hat{\beta}_j-\beta_j)\end{cases}.\tag{20}$$

Consequently, the reconstruction loss of Eq. (19), we need to minimize is

$$\|c_j\cdot\hat{\mathcal{A}}_j^l-\mathcal{A}_j^l\|_2^2=\|\Gamma\otimes\mathcal{A}^{l-1}+\Theta\|_2^2.\tag{21}$$

Recall that the final reconstruction loss $\|\tilde{\mathcal{X}}_t^{l+1}-\mathcal{X}_t^{l+1}\|_2^2$ also requires minimizing the term $\|c_j\cdot\left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right)-\mathcal{W}_{t,j}^{l+1}\right)\|_2^2$ in addition to minimizing $\|c_j\cdot\hat{\mathcal{A}}_j^l-\mathcal{A}_j^l\|_2^2$. Therefore, we introduce a regularization term $\|\mathbf{c}\|_2^2$ for the purpose of restricting the term $\|c_j\cdot\left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right)-\mathcal{W}_{t,j}^{l+1}\right)\|_2^2$. And we can give the data-free compensation loss function to minimize the final reconstruction loss

$$\mathcal{L}=\|\Gamma\|_2^2+\lambda_1\|\Theta\|_2^2+\lambda_2\|\mathbf{c}\|_2^2,\tag{22}$$

where λ_1 and λ_2 are the regularization coefficients.

4.3. Method implementation

First of all, we need to make some clarifications about our proposed method. On the one hand, the coefficient vector $\mathbf{c}$ is defined every two layers, whose size is equal to the output channel of the lth layer and the input channel of the $(l+1)$th layer. Note that the two channel sizes are matched. On the other hand, for the high-bitwidth compensation, we can achieve parallel computation of each input channel of $\tilde{\mathcal{W}}_j^{l+1}$, and different channels will not affect each other. In other words, we can get $\tilde{\mathcal{W}}^{l+1}$ directly.

Based on the above analysis, we define $\mathbf{w}$ and $\hat{\mathbf{w}}$ as the matrices with respect to the input channel of $\mathcal{W}_j^l$ and $\hat{\mathcal{W}}_j^l$, respectively. Following Eq. (22), then the data-free compensation loss function can be rewritten as

$$\begin{aligned}\mathcal{L}(\mathbf{c})=&\left(\mathbf{c}\cdot\frac{\hat{\gamma}\cdot\hat{\mathbf{w}}}{\hat{\sigma}}-\frac{\gamma\cdot\mathbf{w}}{\sigma}\right)^\top\left(\mathbf{c}\cdot\frac{\hat{\gamma}\cdot\hat{\mathbf{w}}}{\hat{\sigma}}-\frac{\gamma\cdot\mathbf{w}}{\sigma}\right)+\lambda_2\mathbf{c}^\top\mathbf{c}\\&+\lambda_1\left[\mathbf{c}\cdot\left(\hat{\beta}-\frac{\hat{\gamma}\cdot\hat{\mu}}{\hat{\sigma}}\right)-\left(\beta-\frac{\gamma\cdot\mu}{\sigma}\right)\right]^\top\left[\mathbf{c}\cdot\left(\hat{\beta}-\frac{\hat{\gamma}\cdot\hat{\mu}}{\hat{\sigma}}\right)-\left(\beta-\frac{\gamma\cdot\mu}{\sigma}\right)\right].\end{aligned}\tag{23}$$

By taking the derivative of the loss function with respect to $\mathbf{c}$, we have

$$\begin{aligned}\frac{\partial\mathcal{L}(\mathbf{c})}{\partial\mathbf{c}}=&-2\left(\frac{\hat{\gamma}\cdot\hat{\mathbf{w}}}{\hat{\sigma}}\right)^\top\left(\frac{\gamma\cdot\mathbf{w}}{\sigma}\right)+2\left(\frac{\hat{\gamma}\cdot\hat{\mathbf{w}}}{\hat{\sigma}}\right)^\top\left(\frac{\hat{\gamma}\cdot\hat{\mathbf{w}}}{\hat{\sigma}}\right)\cdot\mathbf{c}+2\lambda_2\mathbf{c}\\&-2\lambda_1\left(\hat{\beta}-\frac{\hat{\gamma}\cdot\hat{\mu}}{\hat{\sigma}}\right)^\top\left(\beta-\frac{\gamma\cdot\mu}{\sigma}\right)+2\lambda_1\left(\hat{\beta}-\frac{\hat{\gamma}\cdot\hat{\mu}}{\hat{\sigma}}\right)^\top\left(\hat{\beta}-\frac{\hat{\gamma}\cdot\hat{\mu}}{\hat{\sigma}}\right)\cdot\mathbf{c}.\end{aligned}\tag{24}$$

Furthermore, we have the second derivative of the loss function

$$\frac{\partial^2\mathcal{L}(\mathbf{c})}{\partial\mathbf{c}\partial\mathbf{c}^\top}=2\left(\frac{\hat{\gamma}\cdot\hat{\mathbf{w}}}{\hat{\sigma}}\right)^\top\left(\frac{\hat{\gamma}\cdot\hat{\mathbf{w}}}{\hat{\sigma}}\right)+2\left(\lambda_1\left(\hat{\beta}-\frac{\hat{\gamma}\cdot\hat{\mu}}{\hat{\sigma}}\right)^2+\lambda_2\right)I.\tag{25}$$

Consequently, the loss function is a convex function because $\frac{\partial^2\mathcal{L}(\mathbf{c})}{\partial\mathbf{c}\partial\mathbf{c}^\top}$ is positive definite. For brevity, let us use some variable substitution:

$$\begin{cases}\hat{X}=\left(\frac{\hat{\gamma}\cdot\hat{\mathbf{w}}}{\hat{\sigma}}\right), & X=\left(\frac{\gamma\cdot\mathbf{w}}{\sigma}\right),\\\hat{y}=\left(\hat{\beta}-\frac{\hat{\gamma}\cdot\hat{\mu}}{\hat{\sigma}}\right), & y=\left(\beta-\frac{\gamma\cdot\mu}{\sigma}\right),\end{cases}\tag{26}$$

and we can deduce the global minimum when $\frac{\partial\mathcal{L}(\mathbf{c})}{\partial\mathbf{c}}=0$, i.e.,

$$\mathbf{c}=\left[\hat{X}^\top\hat{X}+\lambda_1\hat{y}^2I+\lambda_2I\right]^{-1}\left[\hat{X}^\top X+\lambda_1\hat{y}^\top yI\right].\tag{27}$$

In general, we keep the two trainable parameters constant, i.e., $\hat{\gamma}=\gamma$ and $\hat{\beta}=\beta$, which is consistent with the pre-trained full-precision model [17]. And we can complete the solution by re-calibrating the two statistics $\hat{\mu}$ and $\hat{\sigma}$.

For the forward inference, the solved $\mathbf{c}$ can be combined into γ and β such that Eq. (7) can be fully quantized. In conclusion, we present the whole procedure of our data-free mixed-precision compensation method in Algorithm 1.

4.4. Appendix and proof

Proof of Theorem 1.

$$\begin{aligned}&\tilde{\mathcal{X}}_t^{l+1}-\mathcal{X}_t^{l+1}\\&=c_j\cdot\left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right)-\mathcal{W}_{t,j}^{l+1}\right)\otimes\hat{\mathcal{A}}_j^l+\mathcal{W}_{t,j}^{l+1}\otimes(c_j\cdot\hat{\mathcal{A}}_j^l-\mathcal{A}_j^l)\\&=c_j\cdot\left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right)-\mathcal{W}_{t,j}^{l+1}\right)\otimes\left(\sum_{m=1}^{i}\hat{\mathcal{W}}_{j,m}^l\otimes\mathcal{A}_m^{l-1}\right)\\&+\mathcal{W}_{t,j}^{l+1}\otimes\left[c_j\cdot\sum_{m=1}^{i}\hat{\mathcal{W}}_{j,m}^l\otimes\mathcal{A}_m^{l-1}-\sum_{m=1}^{i}\mathcal{W}_{j,m}^l\otimes\mathcal{A}_m^{l-1}\right]\\&=c_j\cdot\left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right)-\mathcal{W}_{t,j}^{l+1}\right)\otimes\left(\sum_{m=1}^{i}\hat{\mathcal{W}}_{j,m}^l\otimes\mathcal{A}_m^{l-1}\right)\\&+\mathcal{W}_{t,j}^{l+1}\otimes\left[\sum_{m=1}^{i}\left(c_j\cdot\hat{\mathcal{W}}_{j,m}^l-\mathcal{W}_{j,m}^l\right)\otimes\mathcal{A}_m^{l-1}\right].\end{aligned}$$

□

Algorithm 1 Data-free mixed-precision compensation method.

Input: Pre-trained full-precision model $[\mathcal{W}^1, \ldots, \mathcal{W}^L]$.
Output: Mixed low-precision model $[\ldots, \tilde{\mathcal{W}}^{l-1}, \hat{\mathcal{W}}^l, \tilde{\mathcal{W}}^{l+1}, \hat{\mathcal{W}}^{l+2}, \ldots]$.

for each two layers $n \in [1, L/2]$ **do**
  At odd layer $l = 2n-1$, ternary weight filter $\hat{\mathcal{W}}^{2n-1}$ based on Eq. (3);
  for each input channel $j \in [1, i]$ **do**
    Compute the coefficient $c_j \leftarrow \arg\min \|\Gamma\|_2^2 + \lambda_1 \|\Theta\|_2^2 + \lambda_2 \|c_j\|_2^2$ based on Eq. (20) and Eq. (22);
    At even layer $l = 2n$, quantized weight filter $\tilde{\mathcal{W}}_j^{2n} = c_j \cdot {}^k\mathcal{Q}\left(\mathcal{W}_j^{2n}\right)$ based on Eq. (6) and Eq. (7);
  end for
end for

Proof of Lemma 1. Since the proposed method does not depend on a fine-tuning process, we substitute Eq. (14) into Eq. (10)

$$\begin{aligned}
&\tilde{\mathcal{X}}_t^{l+1} - \mathcal{X}_t^{l+1} \\
&= c_j \cdot \left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) - \mathcal{W}_{t,j}^{l+1}\right) \otimes \hat{\mathcal{A}}_j^l + \mathcal{W}_{t,j}^{l+1} \otimes (c_j \cdot \hat{\mathcal{A}}_j^l - \mathcal{A}_j^l) \\
&= c_j \cdot \left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) - \mathcal{W}_{t,j}^{l+1}\right) \otimes \hat{\mathcal{A}}_j^l \\
&\quad + \mathcal{W}_{t,j}^{l+1} \otimes \left[c_j \cdot \left(\hat{\gamma}_j \tfrac{\hat{\mathcal{X}}_j^l - \hat{\mu}_j}{\hat{\sigma}_j} + \hat{\beta}_j\right) - \gamma_j \tfrac{\mathcal{X}_j^l - \mu_j}{\sigma_j} + \beta_j\right] \\
&= c_j \cdot \left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) - \mathcal{W}_{t,j}^{l+1}\right) \otimes \hat{\mathcal{A}}_j^l \\
&\quad + \mathcal{W}_{t,j}^{l+1} \otimes \left[\left(\tfrac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\hat{\mathcal{X}}_j^l - \tfrac{\gamma_j}{\sigma_j}\mathcal{X}_j^l\right) - \left(\tfrac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\hat{\mu}_j - \tfrac{\gamma_j}{\sigma_j}\mu_j\right) + (c_j\hat{\beta}_j - \beta_j)\right].
\end{aligned}$$

Considering the second term of the above equation, we combine with Eq. (11) to give

$$\begin{aligned}
&c_j \cdot \hat{\mathcal{A}}_j^l - \mathcal{A}_j^l \\
&= \left(\tfrac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\hat{\mathcal{X}}_j^l - \tfrac{\gamma_j}{\sigma_j}\mathcal{X}_j^l\right) - \left(\tfrac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\hat{\mu}_j - \tfrac{\gamma_j}{\sigma_j}\mu_j\right) + (c_j\hat{\beta}_j - \beta_j) \\
&= \left(\tfrac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\sum_{m=1}^{i}\hat{\mathcal{W}}_{j,m}^l \otimes \mathcal{A}_m^{l-1} - \tfrac{\gamma_j}{\sigma_j}\sum_{m=1}^{i}\mathcal{W}_{j,m}^l \otimes \mathcal{A}_m^{l-1}\right) \\
&\quad - \left(\tfrac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\hat{\mu}_j - \tfrac{\gamma_j}{\sigma_j}\mu_j\right) + (c_j\hat{\beta}_j - \beta_j) \\
&= \sum_{m=1}^{i}\left(\tfrac{c_j\hat{\gamma}_j\cdot\hat{\mathcal{W}}_{j,m}^l}{\hat{\sigma}_j} - \tfrac{\gamma_j\cdot\mathcal{W}_{j,m}^l}{\sigma_j}\right) \otimes \mathcal{A}_m^{l-1} + \left(\tfrac{\gamma_j}{\sigma_j}\mu_j - \tfrac{c_j\hat{\gamma}_j}{\hat{\sigma}_j}\hat{\mu}_j\right) + (c_j\hat{\beta}_j - \beta_j).
\end{aligned}$$

□

Proof of Lemma 2. In this case, the reconstruction loss of $|c_j \cdot \hat{\mathcal{A}}_j^l - \mathcal{A}_j^l|$ can be formulated as follow

$$\begin{aligned}
&|c_j \cdot \hat{\mathcal{A}}_j^l - \mathcal{A}_j^l| \\
&= |c_j \cdot \max(\mathrm{BN}(\hat{\mathcal{X}}_j^l), 0) - \max(\mathrm{BN}(\mathcal{X}_j^l), 0)| \\
&= \left|c_j \tfrac{\mathrm{BN}(\hat{\mathcal{X}}_j^l) + |\,\mathrm{BN}(\hat{\mathcal{X}}_j^l)|}{2} - \tfrac{\mathrm{BN}(\mathcal{X}_j^l) + |\,\mathrm{BN}(\mathcal{X}_j^l)|}{2}\right| \\
&= \left|\tfrac{c_j\cdot\mathrm{BN}(\hat{\mathcal{X}}_j^l) - \mathrm{BN}(\mathcal{X}_j^l)}{2} + \tfrac{c_j\cdot|\,\mathrm{BN}(\hat{\mathcal{X}}_j^l)| - |\,\mathrm{BN}(\mathcal{X}_j^l)|}{2}\right| \\
&= \left|\tfrac{c_j\cdot\mathrm{BN}(\hat{\mathcal{X}}_j^l) - \mathrm{BN}(\mathcal{X}_j^l)}{2} + \tfrac{|c_j\cdot\mathrm{BN}(\hat{\mathcal{X}}_j^l)| - |\,\mathrm{BN}(\mathcal{X}_j^l)|}{2}\right| \\
&\le \tfrac{1}{2}\left|c_j \cdot \mathrm{BN}(\hat{\mathcal{X}}_j^l) - \mathrm{BN}(\mathcal{X}_j^l)\right| + \tfrac{1}{2}\left|\,|c_j \cdot \mathrm{BN}(\hat{\mathcal{X}}_j^l)| - |\,\mathrm{BN}(\mathcal{X}_j^l)|\right| \\
&\le \left|c_j \cdot \mathrm{BN}(\hat{\mathcal{X}}_j^l) - \mathrm{BN}(\mathcal{X}_j^l)\right|,
\end{aligned}$$

where we have $c_j \cdot |\,\mathrm{BN}(\hat{\mathcal{X}}_j^l)| = |c_j \cdot \mathrm{BN}(\hat{\mathcal{X}}_j^l)|$ as $c_j \ge 0$. □

5. Experiments

In this section, we evaluate our method on CIFAR10/CIFAR100 [34] and ImageNet [35] datasets, which are well-known datasets for evaluating the performance on the image classification.

Table 1
Top-1 classification accuracy results on CIFAR10 dataset with ResNet18, ResNet56, and VGG16. FP32 denotes the full-precision weights. MP2/6 denotes the layer-wise 2 bit and 6 bit mixed-precision weights.

Model	Method	FP32 (%)	MP2/6 (%)
ResNet18	Original	92.61	10.78
	DF-MPC	92.61	89.12
ResNet56	Original	93.88	38.03
	DF-MPC	93.88	91.05
VGG16	Original	93.70	10.00
	DF-MPC	93.70	90.48

Table 2
Top-1 classification accuracy results on CIFAR100 dataset with ResNet18 and VGG16. FP32 denotes the full-precision weights. MP2/6 denotes the layer-wise 2 bit and 6 bit mixed-precision weights.

Model	Method	FP32 (%)	MP2/6 (%)
ResNet18	Original	73.62	1.05
	DF-MPC	73.62	64.90
VGG16	Original	70.09	3.80
	DF-MPC	70.09	64.95

Dataset CIFAR10/CIFAR100 datasets consist of 50k training sets and 10k validation sets, which are natural color images with 32×32 for small-scale experiments. CIFAR10 dataset is organized into 10 classes and CIFAR100 dataset into 100 classes, respectively. ImageNet dataset consists of 1.2 million training sets and 50k validation sets, which are high-resolution natural images for large-scale experiments. These images are organized into 1000 categories.

Model We choose ResNet [36] (including ResNet18, ResNet50, ResNet56, ResNet101), DenseNet121 [37], VGG16 [38] and MobileNetV2 [39] for evaluation. All the model and pre-trained full-precision weights are from pytorchcv library https://pypi.org/project/pytorchcv/.

Setting We implement our method using PyTorch [40] and run the experiments using GTX 1080Ti.

5.1. Ablation study on CIFAR

We first conduct a series of ablation studies on CIFAR datasets to investigate the effect of components of the proposed DF-MPC scheme. We evaluate our method on MP2/6 weights and FP32 activations.

Based on Eq. (27), our method has two regularization coefficients λ_1 and λ_2 that affect the effect of compensation directly. Specifically, we adjust these two hyper-parameters to find the optimal solution, as shown in Fig. 3. On the one hand, as λ_1 varies from 0.1 to 0.5, the final accuracy of the quantized model increase steadily. But it suffers a significant drop when λ_1 is set to 0.6. On the other hand, the final performance is mainly on the decline when λ_2 varies from 0 to 0.01. In summary, the compensation combination of $\lambda_1 = 0.5$ and $\lambda_2 = 0$ is the optimal solution for ResNet56 on CIFAR10 dataset.

For $\lambda_2 = 0$, we also verify from this ablation study that the constraint $\|\mathbf{c}\|_2^2$ in Eq. (22) does not work, which is consistent with our theoretical analysis, i.e., the term $\left({}^k\mathcal{Q}\left(\mathcal{W}_{t,j}^{l+1}\right) - \mathcal{W}_{t,j}^{l+1}\right)$ has very little effect. For $\lambda_1 = 0.5$, we know that in order of importance, $\|\Gamma\|_2^2$ is greater than $\|\Theta\|_2^2$.

Tables 1 and 2 show the performance before and after compensation on CIFAR10 and CIFAR100 datasets, respectively. If the full-precision model is quantized to a mixed-precision of 2-bit and 6-bit directly, its accuracy will become no different from random

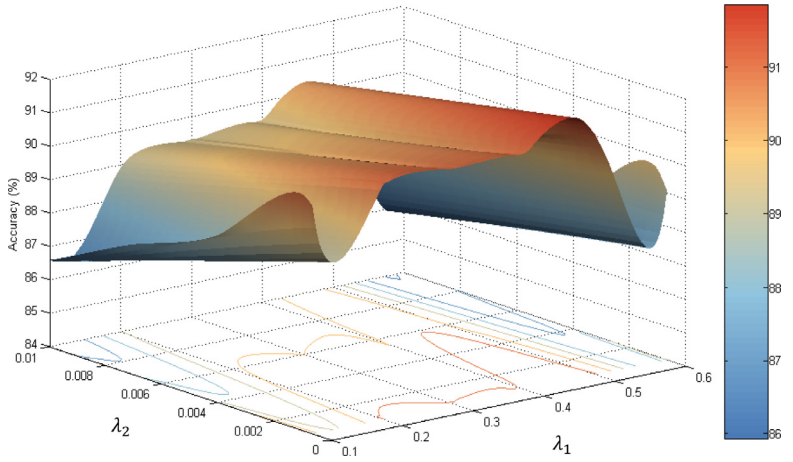
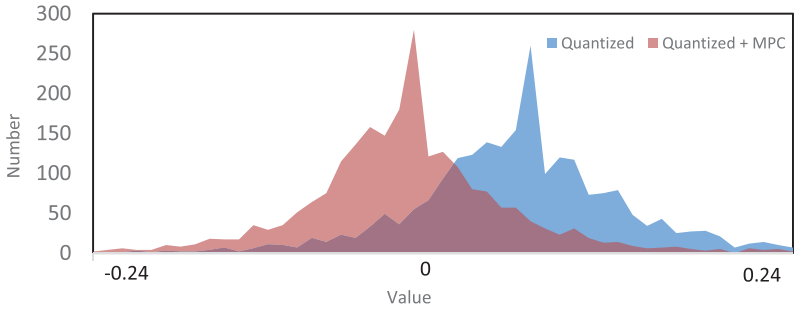
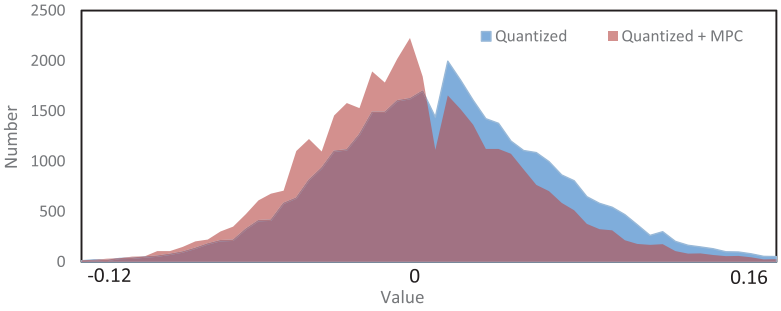


Fig. 3. The accuracy comparison of different λ_1 and λ_2 values in Eq. (27). On CIFAR10 with ResNet56, λ_1 and λ_2 vary from 0.1 to 0.6 and from 0 to 0.01, respectively.



(a) ResNet56 in layer1-2



(b) ResNet56 in layer3-8

Fig. 4. The 6-bit quantized weight distribution before and after compensation on CIFAR10 dataset. The mean of the compensated weight distribution is closer to zero.

initialization. However, after our compensation method, the same quantization mode will result in a fully usable quantized model with great accuracy improvement. Experimentally, this also proves the effectiveness of our DF-MPC.

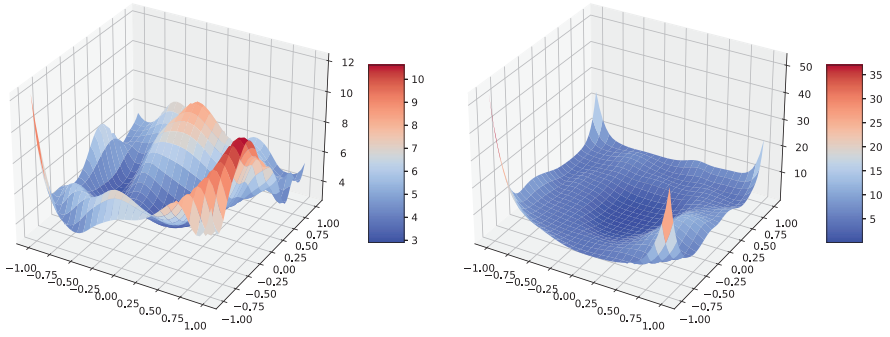
5.2. Experiments on ImageNet

We evaluate our method on ImageNet dataset for the large-scale image classification task, and compare the performance with other data-free quantization methods over various models. Here, GDFQ [25] and GZNQ [42] are the generative methods and they still utilize synthetic data to complete the quantization.

Tables 3 and 4 compare the performance with previous methods, such as OCS [23], DFQ [16], and OMSE [41]. For 2-bit, our DF-MPC uses the ternary representation based on Eq. (3). For 3-bit and 6-bit, our DF-MPC uses the quantized representation based on Eq. (6). Based on layer-wise mixed-precision compensation, we achieve higher accuracy at the smaller model size. In particular, our method with 3/6-bit outperforms DFQ [16] with 6-bit by 0.16% on ResNet18. And our method with 6-bit outperforms DFQ [16] with 8-bit by 0.09% on MobileNetV2. Note that our 6-bit scheme actually implies 6/6-bit mixed-precision quantization.

DF-MPC vs.*ZeroQ* The generative methods need to cost a lot of computation and time due to data synthesis. For example, ZeroQ [17] of ResNet18 takes 12 s on an 8-V100 system. In contrast, DF-MPC of ResNet18 takes only 2 s on a single GTX 1080 Ti, or can even run on CPU only, which makes the deployment of quantized models convenient and fast.

DF-MPC vs.*DFQ* DFQ [16] and DF-MPC have some common ideas. DFQ also considers the relation between the output channel in the lth layer and the input channel in the $(l+1)$th layer. Specifically, DFQ scales the cross-layer factor to equalize the weight ten-



(a) Before Compensation (b) After Compensation

Fig. 5. The loss surfaces of the mixed-precision ResNet56 before and after compensation on CIFAR10 dataset, which reflects the sharpness/flatness of different quantized weights.

Table 3
Top-1 classification accuracy results on ImageNet dataset with ResNet.

Model	Method	W-bit	Size (MB)	Top-1 Acc (%)
ResNet18	Full-precision	32	44.59	71.47
	OMSE [41]	4	5.58	64.03
	GZNQ [42]	4	5.58	64.50
	DFQ [16]	6	8.36	66.30
	DF-MPC	2/6	**5.48**	**66.46**
ResNet50	Full-precision	32	97.49	76.12
	OCS [23]	4	12.28	69.30
	OMSE [41]	4	12.28	70.06
	DF-MPC	2/6	**10.55**	**71.20**
ResNet101	Full-precision	32	170.41	77.31
	OMSE [41]	4	21.30	71.49
	DF-MPC	2/6	**18.36**	**72.59**

Table 4
Top-1 classification accuracy results on ImageNet dataset with DenseNet121 and MobileNetV2.

Model	Method	W-bit	Size (MB)	Top-1 Acc (%)
DenseNet121	Full-precision	32	31.92	74.36
	OCS [23]	4	4.09	63.00
	OMSE [41]	4	4.09	64.40
	DF-MPC	3/6	**3.39**	**70.02**
MobileNetV2	Full-precision	32	13.37	73.03
	GDFQ [25]	6	2.50	70.98
	GZNQ [42]	6	2.50	71.12
	DFQ [16]	8	3.34	71.20
	DF-MPC	6	**2.50**	**71.29**

sor channel ranges. However, DF-MPC scales the cross-layer factor to minimize the output difference of feature maps in the $(l+1)$th layer between the pre-trained full-precision model and its layer-wise mixed-precision quantized model. Theoretically, our method guarantees the minimal quantized error of the layer-wise mixed-precision model.

5.3. Visualization

Figure 4 shows the quantized weight distribution before and after compensation in two different layers of ResNet18. After our DF-MPC method, the mean of the 6-bit quantized weight distribution approaches zero. Moreover, based on the previous work [43], we show the loss surfaces before and after compensation. By analyzing Fig. 5, we find that the loss landscape of the quantized model before compensation is sharp, which shows no noticeable convexity. On the contrary, the loss landscape of the quantized model after compensation is smooth and flat, and shows noticeable convexity, which is consistent with the pre-trained full-precision model.

6. Conclusion

This paper proposed the problem of recovering the accuracy of an ultra-low precision model without any data and fine-tuning, which only relies on the pre-trained full-precision model. By assuming the quantized error caused by a low-precision quantized layer can be restored via the reconstruction of a high-precision quantized layer, we mathematically formulated the reconstruction loss of the feature maps between the pre-trained full-precision model and its mixed-precision quantized model. Based on our formulation, we designed a data-free mixed-precision compensation method along with its closed-form solution.

Since no original/synthetic data is used, we can not access the feature maps, which leads to our method being slightly worse than generative methods with synthetic data. Our future work would extend an expert neural network to estimate the feature maps in the reconstruction loss, which further recovers the performance of the quantized model.

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.

Data availability

Data will be made available on request.

Acknowledgement

This work is funded in part by the Key Research and Development Project of Zhejiang Province under Grant 2021C01035.

References

[1] X. Yu, T. Liu, X. Wang, D. Tao, On compressing deep models by low rank and sparse decomposition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7370–7379.
[2] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, Adv. Neural Inf. Process. Syst. 28 (2015) 1135–1143.
[3] H. Li, A. Kadav, I. Durdanovic, H. Samet, H.P. Graf, Pruning filters for efficient convnets, arXiv preprint arXiv:1608.08710(2016).
[4] H. Wang, C. Qin, Y. Zhang, Y. Fu, Neural pruning via growing regularization, in: International Conference on Learning Representations (ICLR), 2021.
[5] S. Guo, B. Lai, S. Yang, J. Zhao, F. Shen, Sensitivity pruner: filter-level compression algorithm for deep neural networks, Pattern Recognit. 140 (2023) 109508.
[6] W. Chen, J. Wilson, S. Tyree, K. Weinberger, Y. Chen, Compressing neural networks with the hashing trick, in: International Conference on Machine Learning, PMLR, 2015, pp. 2285–2294.
[7] 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.

[8] J. Chen, H. Chen, M. Wang, Y. Liu, Learning discretized neural networks under Ricci flow, arXiv preprint arXiv:2302.03390(2023).
[9] J.-T. Chien, S.-T. Chang, Bayesian asymmetric quantized neural networks, Pattern Recognit. 139 (2023) 109463.
[10] G. Hinton, O. Vinyals, J. Dean, et al., Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 2(7) (2015).
[11] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, Z. Zhang, Correlation congruence for knowledge distillation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5007–5016.
[12] Y. Cho, G. Ham, J.-H. Lee, D. Kim, Ambiguity-aware robust teacher (art): enhanced self-knowledge distillation framework with pruned teacher network, Pattern Recognit. 140 (2023) 109541.
[13] G. Xu, Z. Liu, C.C. Loy, Computation-efficient knowledge distillation via uncertainty-aware mixup, Pattern Recognit 138 (2023) 109338.
[14] 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(2015).
[15] J. Chen, L. Liu, Y. Liu, X. Zeng, A learning framework for n-bit quantized neural networks toward FPGAs, IEEE Trans. Neural Netw. Learn. Syst. 32 (3) (2020) 1067–1081.
[16] 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.
[17] 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.
[18] 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.
[19] K. Choi, D. Hong, N. Park, Y. Kim, J. Lee, Qimera: data-free quantization with synthetic boundary supporting samples, Adv. Neural Inf. Process. Syst. 34 (2021) 14835–14847.
[20] 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(2016).
[21] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio, Quantized neural networks: training neural networks with low precision weights and activations, J. Mach. Learn. Res. 18 (1) (2017) 6869–6898.
[22] R. Banner, Y. Nahshan, D. Soudry, Post training 4-bit quantization of convolutional networks for rapid-deployment, Adv. Neural Inf. Process. Syst. 32 (2019) 7950–7958.
[23] 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.
[24] 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.
[25] S. Xu, H. Li, B. Zhuang, J. Liu, J. Cao, C. Liang, M. Tan, Generative low-bitwidth data free quantization, in: European Conference on Computer Vision, Springer, 2020, pp. 1–17.
[26] 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.
[27] S. Srinivas, R.V. Babu, Data-free parameter pruning for deep neural networks, arXiv preprint arXiv:1507.06149(2015).
[28] W. Kim, S. Kim, M. Park, G. Jeon, Neuron merging: compensating for pruned neurons, Adv. Neural Inf. Process. Syst. 33 (2020) 585–595.
[29] T. Chu, Q. Luo, J. Yang, X. Huang, Mixed-precision quantized neural networks with progressively decreasing bitwidth, Pattern Recognit. 111 (2021) 107647.
[30] F. Li, B. Zhang, B. Liu, Ternary weight networks, arXiv preprint arXiv:1605.04711(2016).
[31] C. Zhu, S. Han, H. Mao, W.J. Dally, Trained ternary quantization, arXiv preprint arXiv:1612.01064(2016).
[32] Z. Liu, Y. Wang, K. Han, W. Zhang, S. Ma, W. Gao, Post-training quantization for vision transformer, Adv. Neural Inf. Process. Syst. 34 (2021) 28092–28103.
[33] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456.
[34] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
[35] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.
[36] 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.
[37] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556(2014).
[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch(2017).
[41] Y. Choukroun, E. Kravchik, F. Yang, P. Kisilev, Low-bit quantization of neural networks for efficient inference, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE, 2019, pp. 3009–3018.
[42] X. He, J. Lu, W. Xu, Q. Hu, P. Wang, J. Cheng, Generative zero-shot network quantization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3000–3011.
[43] H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the loss landscape of neural nets, Adv. Neural Inf. Process. Syst. 31 (2018).

Jun Chen received the B.S. degree in the department of Mechanical and Electrical Engineering from China Jiliang University, Hangzhou, China, in 2016, and the M.S. degree in from the Zhejiang University, Hangzhou, China, in 2020. He is currently working toward the Ph.D. degree with the Institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include neural network quantization and deep learning.

Shipeng Bai received his B.S. degree in the department of Control and Computer Engineering from North China Electric Power University, Beijing, China, in 2021. He is currently working toward the M.S degree with the institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University. His research interests include neural network compression and deep learning.

Tianxin Huang received his B.S. degree in the department of mechanical engineering from Xi'an Jiao Tong University, Xi'an, China, in 2017. He is currently working for his Ph.D. degree with the Institute of Cyber Systems and Control, Zhejiang University, HangZhou, China. His research interests concentrate on the Learning-based 3D Point Cloud processing.

Mengmeng Wang received the B.S. degree and the M.S. degree in control science and engineering from Zhejiang University, Zhejiang, China, in 2015 and 2018, respectively. She is currently working towards the Ph.D degree with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering, Zhejiang University. Her research interests include visual tracking, action recognition, computer vision, and deep learning.

Guanzhong Tian received the B.S. degree from Harbin Institute of Technology, Harbin, China, in 2010, and the Ph.D. degree in from Zhejiang University, Hangzhou, China, in 2021. He is currently a Research associate with Ningbo Research Institute, Zhejiang University. His research interests include computer vision, model compression, embedded AI.

Yong Liu received his B.S. degree in computer science and engineering from Zhejiang University in 2001, and the Ph.D. degree in computer science from Zhejiang University in 2007. He is currently a professor in the institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University. He has published more than 30 research papers in machine learning, computer vision, information fusion, robotics. His latest research interests include machine learning, robotics vision, information processing and granular computing.