Unified Data-Free Compression: Pruning and Quantization without Fine-Tuning

Shipeng Bai[*]   Jun Chen[*]   Xintian Shen   Yixuan Qian   Yong Liu[†]

College of Control Science and Engineering, Zhejiang University

[shipengbai, junc, 22132133, 22260066]@zju.edu.cn, yongliu@iipc.zju.edu.cn

Abstract

Structured pruning and quantization are promising approaches for reducing the inference time and memory footprint of neural networks. However, most existing methods require the original training dataset to fine-tune the model. This not only brings heavy resource consumption but also is not possible for applications with sensitive or proprietary data due to privacy and security concerns. Therefore, a few data-free methods are proposed to address this problem, but they perform data-free pruning and quantization separately, which does not explore the complementarity of pruning and quantization. In this paper, we propose a novel framework named Unified Data-Free Compression(UDFC), which performs pruning and quantization simultaneously without any data and fine-tuning process. Specifically, UDFC starts with the assumption that the partial information of a damaged(e.g., pruned or quantized) channel can be preserved by a linear combination of other channels, and then derives the reconstruction form from the assumption to restore the information loss due to compression. Finally, we formulate the reconstruction error between the original network and its compressed network, and theoretically deduce the closed-form solution. We evaluate the UDFC on the large-scale image classification task and obtain significant improvements over various network architectures and compression methods. For example, we achieve a 20.54% accuracy improvement on ImageNet dataset compared to SOTA method with 30% pruning ratio and 6-bit quantization on ResNet-34. Code will be available at here.

1. Introduction

Model compression is the most common way to reduce the memory footprint and computational costs of the model, and it mainly includes two methods: pruning[22, 28, 4] and quantization[8, 17, 43, 5, 3]. Among the pruning domain, structured pruning[41, 36] is more actively studied than unstructured pruning[20, 34] since it eliminates the whole channel or even the layer of the model while not requiring any special hardware or libraries for acceleration. Under such conditions, we also focus our attention on structured pruning in this paper. Quantization methods attempt to reduce the precision of the parameters and/or activations from 32-bit floating point to low-precision representations. Thus the storage requirement for the model can be diminished substantially, as well as the power consumption.

Although the existing compression methods achieve a satisfactory compression ratio with a reasonable final performance, most of them require the original training data for a tedious fine-tuning process. The fine-tuning process is not only data-dependent but also computationally expensive, while users may not have access to sufficient or complete data in many real-world applications, such as medical data and user data. Therefore, data-free compression methods are proposed, which don't require any real data and fine-tuning process. For instance, Data-free parameter pruning[38] first introduces the data-independent technique to remove the redundant neurons, and Neuron Merging[18] extends the data-free method from fully connected layers to convolutional layers. Meanwhile, there exist some methods using the synthetic samples to perform the fine-tuning process, such as Dream[44]. In the field of quantization, recent works propose post-training quantization methods[32, 2, 45, 24, 6, 46] that use the synthetic data to replace the real data for quantization and achieve the SOTA results. For instance, ZeroQ [2] uses the distilled data that matches the statistics of batch normalization layers to perform post-training quantization. DSG[45] proposes a novel Diverse Sample Generation scheme to enhance the diversity of synthetic samples, resulting in better performance.

However, some problems still hinder the deployment of data-free compression. On the one hand, the latest data-free quantization approaches focus on improving the quality of synthetic samples rather than releasing the quantization from its dependence on data. In this case, generating the synthetic samples introduces extra computational costs. On the other hand, current approaches perform data-free pruning and quantization separately, which does not explore the complementarity of weight pruning and quantization.

---

[*]Equal contribution.
[†]Corresponding author.
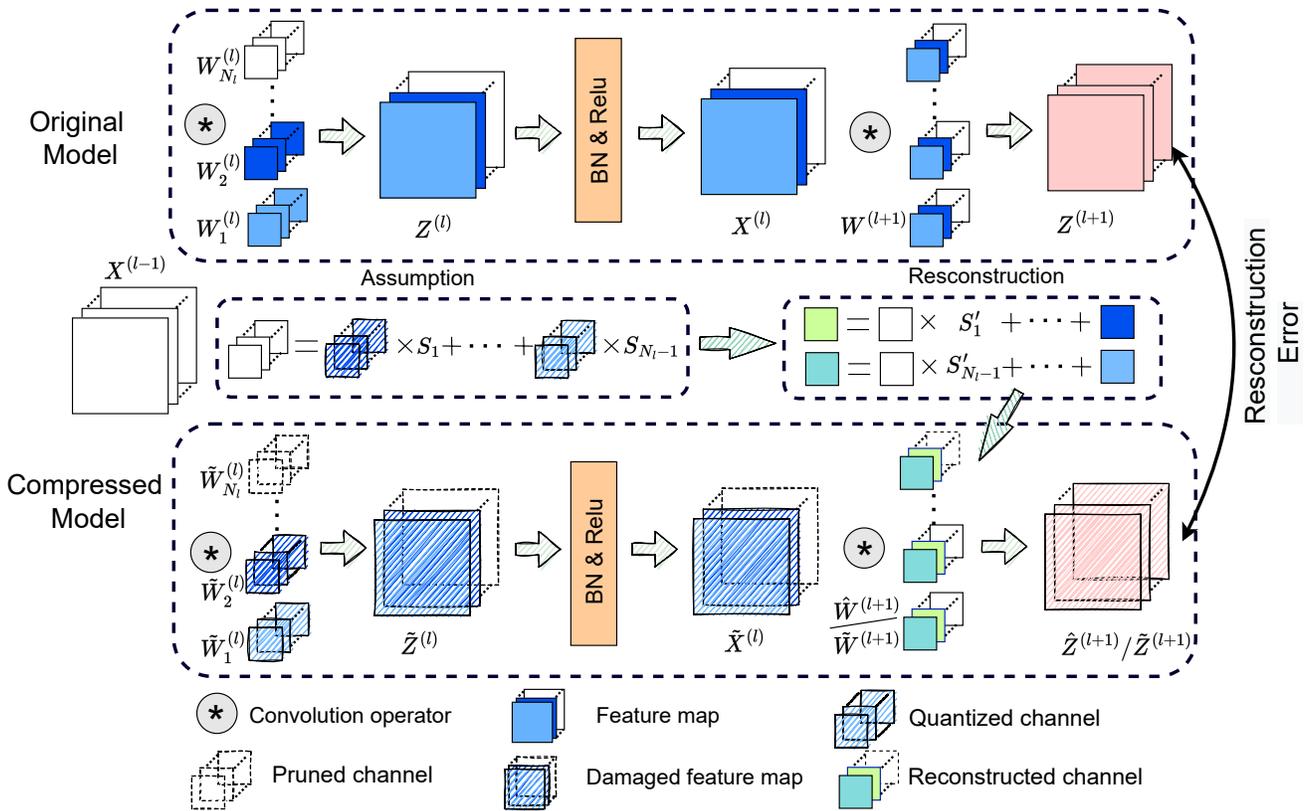
Figure 1. The general overview of UDFC, which performs the pruning and quantization simultaneously. S is the scale factor and $S \neq S'$. See Section 3 for details of S. After the output channels of l-th layer are pruned or quantized, our goal is to maintain the feature map $Z^{(l+1)}$ of $(l+1)$-th layer. We first deduce the reconstruction form based on our assumption and then reconstruct the input channels of $(l+1)$-th layer to restore the information loss caused by compression of l-th layer. Finally, we formulate the reconstruction error between the feature map $Z^{(l+1)}$ and $\hat{Z}^{(l+1)}/\tilde{Z}^{(l+1)}$.

In this paper, we propose a novel joint compression framework named Unified Data-Free Compression(UDFC), which overcomes abovementioned issues without any original/synthetic data and fine-tuning process, as shown in Figure 1. Our contributions can be summarized as follows:

- We propose the assumption that the partial information of a damaged(e.g., pruned or quantized) channel can be preserved by a linear combination of other channels. Based on this assumption, we derive that the information loss caused by pruning and quantization of the l-th layer can be restored by reconstructing the channels of the $(l+1)$-th layer. The assumption and reconstruction form are described in Section 3.2.

- Based on the reconstruction form, we formulate the reconstruction error between the original network and its compressed network. The reconstruction error is described in Section 3.3.

- Based on the reconstruction error, we prove that reconstruction error can be minimized and theoretically deduce the closed form solution in Section 4.

Furthermore, extensive experiments on CIFAR-10[19] and ImageNet[35] with various popular architectures demonstrate the effectiveness and generality of UDFC. For example, UDFC on VGG-16 yields around 70% FLOPS reduction and 28× memory footprint reduction, with only a 0.4% drop in accuracy compared to the uncompressed baseline on CIFAR-10 dataset.

2. Related Work

2.1. Model Compression

Researchers have proposed various methods to accelerate the model inference, mainly including network pruning[12, 26] and network quantization[11]. The early pruning methods concentrate around unstructured pruning, which removes single parameters from networks[30]. These approaches, though theoretically interesting, are more difficult to implement within current hardware and software settings. Therefore, much recent work has focused on structured pruning[1], where network channels can be removed, and the models can be practically compressed and accelerated. Weight quantization refers to the process of

discretizing the range of weight values so that each weight can be represented using fewer bits. [11] first quantizes the network weights to reduce the model size by grouping the weights using k-means. [16, 33] then introduce the binary network, in which weights are quantized to 1-bit. However, these aforementioned methods require access to data for fine-tuning to recover the performance. Fine-Tuning is often not possible for applications with sensitive or proprietary data due to privacy and security concerns. Therefore, we focus on pruning and quantization without any data and fine-tuning process in this paper.

2.2. Data-Free Pruning

Some pruning methods attempt to eliminate the dependency on the entire dataset and expensive fine-tuning process. [29] formally establishes channel pruning as an optimization problem and solves this problem without using the entire dataset. [31] introduces a novel pruning algorithm, which can be interpreted as preserving the total flow of synaptic strengths through the network at initialization subject to a sparsity constraint. Meanwhile, there is a branch of data-free pruning methods [39, 40, 44] that still fine-tune the pruned model with limited or synthetically generated data. Although the above approaches propose effective methods for channel or neuron selection, several epochs of fine-tuning process and some training data are unavoidable to enable adequate recovery of the pruned network.

In fact, there are only two methods to prune the model without any data and fine-tuning process. Data-free parameter pruning[38] shows how similar neurons are redundant and proposes a systematic way to remove them. Then Neuron Merging[18] extends the data-free method from fully connected layers to convolutional layers based on the assumption that a pruned kernel can be replaced by another similar kernel.

2.3. Data-Free Quantization

Data Free Quantization [32] suffers a non-negligible performance drop when quantized to 6-bit or lower bit-width. Therefore, more recent studies employ generator architectures similar to GAN [9] to generate synthetic samples that replace the original data. Such as, ZeroQ [2] generates samples that match the real-data statistics stored in the full-precision batch normalization layer to perform the post-training quantization resulting in better performance. IntraQ[46] propose a local object reinforcement that locates the target objects at different scales and positions of synthetic images, aiming to enhance the intra-class heterogeneity in synthetic images. DSG[45] slackens the batch normalization matching constraint and assigns different attention to specific layers for different samples to ensure diverse sample generation. However, using the generated samples to improve the accuracy of quantized models is

time-consuming and complex. In this paper, we do not use any data to quantize the network.

3. Formualtion of Reconstruction Error

In this section, we first illustrate how to reconstruct the channels based on our assumption after pruning and quantization, and then mathematically formulate the reconstruction error.

3.1. Background Knowledge

CNN architecture. Assuming that a CNN model with L layers, we use N_{l-1} and N_l to represent the number of input channels and the output channels for the l-th convolution layer. The l-th convolution layer transforms the input feature map $X^{(l-1)} \in \mathbb{R}^{N_{l-1} \times H_{l-1} \times W_{l-1}}$ into the output feature map $Z^{(l)} \in \mathbb{R}^{N_l \times H_l \times W_l}$. The convolution weights of the l-th layer are denoted as $W^{(l)} \in \mathbb{R}^{N_l \times N_{l-1} \times K \times K}$. Note that K is the kernel size of each channel, and $H \times W$ is the corresponding feature map size. Therefore,

$$Z^{(l)} = X^{(l-1)} \circledast W^{(l)}, \qquad (1)$$

where $\circledast$ denotes the convolution operator. For CNN architectures, the convolution operation is widely followed by a batch normalization(BN) procedure and an activation function, thus the activation feature map $X^{(l+1)}$ can be formulated as:

$$X^{(l)} = \Theta(B(Z^{(l)})) = \Theta(\frac{\gamma(Z^{(l)} - \mu)}{\sigma} + \beta), \qquad (2)$$

in which $B(\cdot)$ is the BN procedure and $\Theta(\cdot)$ is the activation function. γ, μ, σ and β are the variables of BN.

Pruning criterion. In channel pruning, most methods follow a selecting strategy, i.e., selecting some original channels via the l_2-norm[22] of weight and scaling factors[27] in BN layer. In general, pruning criterion tends to be closely related to model performance. In this paper, we do not focus on proposing a complex criterion but on restoring the performance of networks that are pruned in a simple criterion such as l_1-norm and l_2-norm.

Uniform quantization. Quantization converts the floating-point parameters W in the pretrained full-precision model to low-precision fixed-point values $\tilde{W}$, thus reducing the model complexity. Uniform quantization [47] is the simplest and most hardware-friendly method, which is defined as:

$$\tilde{W} = \frac{2}{2^k - 1}round[(2^k - 1)(\frac{W}{2max|W|} + \frac{1}{2})] - 1, \quad (3)$$

where k is the quantization bit-width. In this case, we use uniform quantization to preform the quantization process in this paper.

3.2. Layer-wise Reconstruction

Assumption. In model compression, the performance of the compressed network is usually worse than original network. To improve the performance of compressed network without any data and fine-tuning process, an ideal idea is to preserve the information of these damaged channels(e.g., pruned channel or quantized channel). We assume that the partial information of the damaged channels can be preserved by a linear combination of other channels. For clarity, we describe the assumptions about pruning and quantization separately. Suppose that convolution weight $W^{(l)}$ is pruned to its damaged versions $\hat{W}^{(l)} \in \mathbb{R}^{\hat{N}_l \times N_{l-1} \times K \times K}$, where $\hat{N}_l$ is the number of unpruned channels. The Assumption of pruning can be formulated as follows:

$$W_j^{(l)} \approx \sum_{i=1}^{\hat{N}_l} \hat{s}_i \times W_i^{(l)}, \quad \forall j \in [\hat{N}_l, N_l], i \in [1, \hat{N}_l] \quad (4)$$

where $\hat{s}$ is a scale factor that measures the degree of association of the i-th channel with the j-th channel under the pruning. We prove that there always exists $\hat{s}_i$ minimizing the MSE error ($\|W_j^{(l)} - \sum_{i=1}^{\hat{N}_l} \hat{s}_i \times W_i^{(l)}\|_2^2$) of Eq.4 in Section 4.

Suppose that the m-th channel of l-th layer is quantized to its damaged versions $\tilde{W}_m^{(l)}$, the assumption of quantization can be formulated as:

$$\tilde{W}_m^{(l)} \approx \tilde{s}_m \times W_m^{(l)}, \quad \forall m \in [1, N_l] \quad (5)$$

where $\tilde{s}$ is a scale factor that measures the degree of association of the m-th channel with its quantized version under the quantization. We prove that there always exists $\tilde{s}_i$ minimizing the MSE error ($\|\tilde{W}_m^{(l)} - \tilde{s}_m \times W_m^{(l)}\|_2^2$) of Eq.5 in Section 4.

Reconstruction after pruning. Our goal is to maintain the output feature map of the $(l+1)$-th layer while pruning the channels of the l-th layer. For brevity, we prune only one channel in the l-th layer to illustrate how the channels of $(l+1)$-th layer are reconstructed, which can easily be generalized to multiple channels. Without loss of generality, the j-th channel of the l-th layer is to be pruned.

As shown in Figure 1, after the j-th output channel of the l-th layer is pruned, the output feature map $Z_j^{(l)}$ is subsequently deleted. Based on Eq.1 and Eq.4, we can deduce that the pruned output feature map $Z_j^{(l)}$ can be replaced by a linear combination of other undamaged feature maps:

$$Z_j^{(l)} = X^{(l-1)} \circledast W_j^{(l)} \approx X^{(l-1)} \circledast \sum_{i=1,i\neq j}^{N_l} \hat{s}_i \times W_i^{(l)}$$
$$= \sum_{i=1,i\neq j}^{N_l} \hat{s}_i \times Z_i^{(l)}, \quad (6)$$

When only considering the BN layer, we have $X^{(l)} = B(Z^{(l)})$. Based on Eq.6, the k-th channel of output feature map $Z^{(l+1)}$ can be represented as:

$$Z_k^{(l+1)} = \sum_{i=1}^{N_l} X_i^{(l)} \circledast W_{k,i}^{(l+1)}$$
$$\approx \sum_{i=1,i\neq j}^{N_l} B(Z_i^{(l)}) \circledast (W_{k,i}^{(l+1)} + \hat{s}_i \times W_{k,j}^{(l+1)}), \quad (7)$$

(More details in **Appendix A**.)
in which $(W_{k,i}^{(l+1)} + s_i \times W_{k,j}^{(l+1)})$ is a reconstructed filter. In this way, we can preserve the information of pruned channels in the l-th layer by adding its corresponding pruned channel to each of the other channels in the next layer. According to the Eq.7, we reconstruct the channels of the $(l+1)$-th layer to restore the information loss caused by pruning the l-th layer in the following form:

$$\hat{Z}_k^{(l+1)} = \sum_{i=1,i\neq j}^{N_l} X_i^{(l)} \circledast (W_{k,i}^{(l+1)} + \hat{s}_i \times W_{k,j}^{(l+1)}), \quad (8)$$

where $\hat{Z}_k^{(l+1)}$ represents the reconstructed version after pruning.

Reconstruction after quantization. The most significant difference between pruning and quantization is whether the channel exists or not. Quantized channels use the low bit-width to save the weights instead of discarding it away. In this case, we compensate for the information loss by adding a scale factor to its corresponding channels on the next layer. For simplicity, let $\tilde{W}_m^{(l)}$ denotes the weight of m-th channel of l-th layer after quantization. Based on Eq.1 and Eq.5, we can deduce the reconstruction version of $\tilde{Z}_k^{(l+1)}$ after quantization, which can be expressed as:

$$\tilde{Z}_k^{(l+1)} = \sum_{i=1,i\neq m}^{N_l} X_i^{(l)} \circledast W_{k,i}^{(l+1)} + \tilde{X}_m^{(l)} \circledast (\tilde{s}_m \times W_{k,m}^{(l+1)}), \quad (9)$$

where $\tilde{X}_m^{(l)}$ denotes the damaged version of $X_m^{(l)}$ after quantification.

3.3. Reconstruction Error

However, the above analyses are all under the assumption, and the reconstruction error is inevitable in fact. After restoring information loss caused by the compression in the l-th layer, we measure the reconstruction error using the feature map $Z_k^{(l+1)}$ of $(l+1)$-th layer before and after compression.

Pruning error. After pruning, the difference e_p between $Z_k^{(l+1)}$ and $\hat{Z}_k^{(l+1)}$ can be expressed as:

$$
\begin{aligned}
e_p &= Z_k{}^{(l+1)} - \hat{Z}_k{}^{(l+1)} \\
&= \{ \frac{\gamma_j}{\sigma_j} \{ X^{(l-1)} \circledast (W_j^{(l)} - \sum_{i=1,i\neq j}^{N_l} \hat{s}_i \frac{\gamma_i \sigma_j}{\sigma_i \gamma_j} W_i^{(l)}) \} + \\
&\quad (\beta_j - \frac{\gamma_j \mu_j}{\sigma_j}) - (\sum_{i=1,i\neq j}^{N_l} \hat{s}_i (\beta_i - \frac{\gamma_i \mu_i}{\sigma_i})) \} \circledast W_{k,j}^{(l+1)}
\end{aligned}
$$
$$(10)$$

(More details in **Appendix B**.)

Influence of Activation Function. The Relu activation function is widely used in CNN architectures. Since we cannot obtain the feature map after the activation function in a data-free way, we qualitatively analyze the effects of the Relu function on our pruning error e_p. In this case, the difference e_p can be re-expressed as:

$$
\begin{aligned}
e_p &= \Theta(B(Z_j^{(l)})) - \sum_{i=1,i\neq j}^{N_l} \hat{s}_i \times \Theta(B(Z_i^{(l)})) \\
&\leqslant \frac{1}{2}(A + |A|),
\end{aligned}
$$
$$(11)$$

(More details in **Appendix C**.)
where $A = B(Z_j^{(l)}) - \sum_{i=1,i\neq j}^{N_l} \hat{s}_i \times B(Z_i^{(l)})$ and we omit the $W_{k,j}^{(l+1)}$ as it doesn't change with pruning. The term $\frac{1}{2}(A + |A|)$ determine the upper boundary of e_p and the form of $(B(Z_j^{(l)}) - \sum_{i=1,i\neq j}^{N_l} \hat{s}_i \times B(Z_i^{(l)}) \circledast W_{k,j}^{(l+1)}$ is the same as Eq.10, so the difference e_p of pruning we obtained is equal whether the Relu activation function is considered or not.

Note that $X^{(l-1)}$ and $W_{k,j}^{(l+1)}$ are not changed with pruning. Therefore, we define the reconstruction error ℓ_p of pruning as:

$$
\begin{aligned}
\ell_p &= \| W_j^{(l)} - \sum_{i=1,i\neq j}^{N_l} \hat{s}_i \frac{\gamma_i \sigma_j}{\sigma_i \gamma_j} W_i^{(l)} \|_2^2 \\
&\quad + \alpha_1 \| (\beta_j - \frac{\gamma_j \mu_j}{\sigma_j}) - \sum_{i=1,i\neq j}^{N_l} \hat{s}_i (\beta_i - \frac{\gamma_i \mu_i}{\sigma_i}) \|_2^2,
\end{aligned}
$$
$$(12)$$

in which, we introduce a hyperparameter α_1 to adjust the proportion of different parts.

Quantization error. After quantization, the difference e_q of $Z_k^{(l+1)}$ can be expressed as:

$$
\begin{aligned}
e_q &= Z_k{}^{(l+1)} - \tilde{Z}_k^{(l+1)} \\
&= \{ (\frac{\gamma_m W_m^{(l)}}{\sigma_m} - \tilde{s}_m \frac{\gamma_m \tilde{W}_m^{(l)}}{\sigma_m}) \circledast X^{(l-1)} + \tilde{s}_m \frac{\gamma_m \mu_m}{\sigma_m} \\
&\quad - \frac{\gamma_m \mu_m}{\sigma_m} + \beta_m - \tilde{s}_m \beta_m \} \circledast W_{k,m}^{(l+1)},
\end{aligned}
$$
$$(13)$$

(More details in **Appendix D**.)

Same as pruning, $X^{(l-1)}$ and $W_{k,m}^{(l+1)}$ are not changed with quantization, while the activation function does not influence the form of the reconstruction error. Therefore, we define the reconstruction error ℓ_q of quantization as:

$$
\begin{aligned}
\ell_q &= \| \frac{\gamma_m W_m^{(l)}}{\sigma_m} - \tilde{s}_m \frac{\gamma_m \tilde{W}_m^{(l)}}{\sigma_m}) \|_2^2 + \\
&\quad \alpha_2 \| (\beta_m - \frac{\gamma_m \mu_m}{\sigma_m}) - \tilde{s}_m (\beta_m - \frac{\gamma_m \mu_m}{\sigma_m}) \|_2^2
\end{aligned}
$$
$$(14)$$

in which, we introduce a hyperparameter α_2 to adjust the proportion of different parts.

Reconstruction error Previously, we analyzed the errors caused by pruning and quantization separately. When pruning and quantization are performed simultaneously, the reconstruction error ℓ_{re} can be expressed as:

$$\ell_{re} = \ell_p + \ell_q \qquad (15)$$

4. Solutions for Reconstruction Error

In this section, we prove the existence of the optimal solution s by minimizing the reconstruction error. The j-th channel is pruned and the m-th channel is quantized in l-th layer, and we get the reconstruction error ℓ_{re}:

$$
\begin{aligned}
\ell_{re} &= \| W_j^{(l)} - \hat{s}_i \sum_{i=1,i\neq j}^{N_l} \frac{\gamma_i \sigma_j}{\sigma_i \gamma_j} W_i^{(l)} \|_2^2 \\
&\quad + \alpha_1 \| (\beta_j - \frac{\gamma_j \mu_j}{\sigma_j}) - (\sum_{i=1,i\neq j}^{N_l} \hat{s}_i (\beta_i - \frac{\gamma_i \mu_i}{\sigma_i})) \|_2^2 \\
&\quad + \| \frac{\gamma_m W_m^{(l)}}{\sigma_m} - \tilde{s}_m \frac{\gamma_m \tilde{W}_m^{(l)}}{\sigma_m}) \|_2^2 \\
&\quad + \alpha_2 \| (\beta_m - \frac{\gamma_m \mu_m}{\sigma_m}) - \tilde{s}_m (\beta_m - \frac{\gamma_m \mu_m}{\sigma_m}) \|_2^2,
\end{aligned}
$$
$$(16)$$

For simplicity, we have:

$$\begin{cases} \mathbf{G}_i = \frac{\gamma_i \sigma_j}{\sigma_i \gamma_j} \mathbf{W}_i^{(l)}, \quad \mathbf{V} = W_j^{(l)}, \\ \mathbf{Q} = [\mathbf{G}_1, \cdots, \mathbf{G}_{j-1}, \mathbf{G}_{j+1}, \cdots, \mathbf{G}_{N_l}], \\ \mathbf{K}_i = \beta_i - \frac{\gamma_i \mu_i}{\sigma_i}, \\ \mathbf{P} = [\mathbf{K}_1, \cdots, \mathbf{K}_{j-1}, \mathbf{K}_{j+1}, \cdots, \mathbf{K}_{N_l}], \\ \mathbf{R}_i = \frac{\gamma_i \mathbf{W}_i^{(l)}}{\sigma_i}, \\ \hat{\mathbf{s}} = [\hat{s}_1, \cdots, \hat{s}_{j-1}, \hat{s}_{j+1}, \cdots, \hat{s}_{N_l}], \end{cases} \quad (17)$$

where $\mathbf{V}$ is the vectorized $W_i^{(l)}$, $\mathbf{K}_i$ is the vectorized $\beta_i - \frac{\gamma_i \mu_i}{\sigma_i}$ and $\mathbf{R}_i$ is the vectorized $\frac{\gamma_i \mathbf{W}_i^{(l)}}{\sigma_i}$. Then the loss can be simplified as follows:

$$\begin{aligned} \ell_{re} &= (\mathbf{V} - \hat{\mathbf{s}}\mathbf{Q})^T (\mathbf{V} - \hat{\mathbf{s}}\mathbf{Q}) + \alpha_1 (\mathbf{K}_j - \hat{\mathbf{s}}\mathbf{P})^T (\mathbf{K}_j - \hat{\mathbf{s}}\mathbf{P}) \\ &+ (\mathbf{R}_m - \tilde{s}_m \mathbf{R}_m)^T (\mathbf{V} - \tilde{s}_m \mathbf{R}_m) \\ &+ \alpha_2 (\mathbf{K}_m - \tilde{s}_m \mathbf{K}_m)^T (\mathbf{K}_m - \tilde{s}_m \mathbf{K}_m) \end{aligned} \quad (18)$$

The first and second derivative of the $\hat{\mathbf{s}}$ is:

$$\frac{\partial \ell_{re}}{\partial \hat{\mathbf{s}}} = -2\mathbf{Q}^T \mathbf{V} + 2\hat{\mathbf{s}}\mathbf{Q}^T \mathbf{Q} + \alpha_1(-2\mathbf{P}^T \mathbf{K}_j + 2\hat{\mathbf{s}}\mathbf{P}^T \mathbf{P})$$

$$\frac{\partial^2 \ell_{re}}{\partial \hat{\mathbf{s}}^2} = 2\mathbf{Q}^T \mathbf{Q} + 2\alpha_1 \mathbf{P}^T \mathbf{P} \quad (19)$$

ℓ_{re} is a convex function and thus there exists an unique optimal solution s such that $\frac{\partial \ell_{re}}{\partial \hat{\mathbf{s}}} = 0$, so we get the optimal solution as follows:

$$\hat{\mathbf{s}} = (\mathbf{Q}^T \mathbf{V} + \alpha_1 \mathbf{P}^T \mathbf{K}_j)(\mathbf{Q}^T \mathbf{Q} + \alpha_1 \mathbf{P}^T \mathbf{P})^{-1} \quad (20)$$

Similarly, we get the optimal solution of $\tilde{s}_m$:

$$\tilde{s}_m = (\mathbf{R}_m^T \mathbf{R}_m + \alpha_2 \mathbf{K}_m^T \mathbf{K}_m)(\mathbf{R}_m^T \mathbf{R}_m + \alpha_2 \mathbf{K}_m^T \mathbf{K}_m)^{-1} \quad (21)$$

It is worth noting that the MSE error of Eq.4 and Eq.5 are the main components of the reconstruction error. Therefore, the optimal solution s not only minimizes the reconstruction error but also satisfies the assumptions.
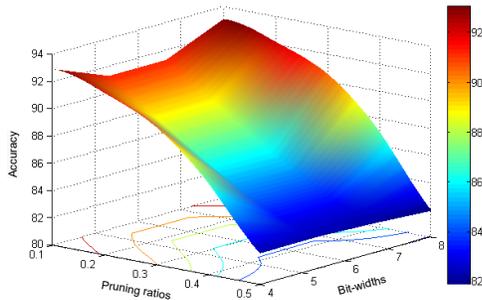
Figure 2. Comparison of the accuracy of ResNet-56 with different pruning ratios and bit widths on CIFAR-10 dataset.
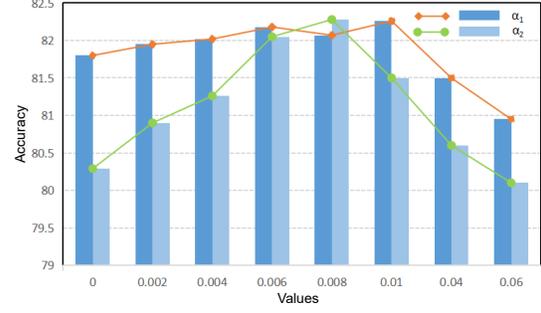
Figure 3. The accuracy comparison of different α values on ResNet-56. As the value α increases, the accuracy curve first rise and then fall.

Implementation of the scale factors. As shown in Eq.8 and Eq.9, we replace the original convolutional layer with the reconstruction form after getting the optimal scales.

5. Experiments

In this section, we conduct experiments with several different widely-used neural networks for image classification task to evaluate the effectiveness of our data-free compression method that do not require any data and fine-tuning process. In all experiments, we quantize the weights of all model layers using uniform quantization and prune the channels with the simple pruning criterion l_1-norm and l_2-norm. In addition, we visualize the weights offset and loss

Table 1. Quantization results on ImageNet dataset. 'No-D' denotes whether to use synthetic samples or calibration sets.

Model	Method	No-D	(W/A)Bit	Size(MB)	Top-1(%)
ResNet-18	Baseline	-	32/32	44.59	71.47
	DFQ[32]	✓	6/6	8.36	66.30
	DSG[45]	×	6/6	8.36	70.46
	SQuant[10]	×	6/6	8.36	70.74
	Ours	✓	6/6	8.36	**72.76**
	DDAQ[24]	×	4/4	5.58	58.44
	DSG[45]	×	4/4	5.58	34.53
	Ours	✓	4/4	5.58	**63.49**
ResNet-50	Baseline	-	32/32	97.49	77.72
	ZeroQ[2]	×	6/6	18.46	75.56
	DSG[45]	×	6/6	18.46	76.07
	SQuant[10]	×	6/6	18.46	77.05
	Ours	✓	6/6	18.46	**77.57**
	OMSE[7]	✓	4/32	12.28	67.36
	GDFQ[42]	×	4/4	12.28	55.65
	SQuant[10]	×	4/4	12.28	70.80
	Ours	✓	4/4	12.28	**72.09**
MobileNetV2	Baseline	-	32/32	13.37	73.03
	DFQ[32]	✓	8/8	3.34	71.20
	DDAQ[24]	×	6/6	2.50	70.30
	ZeroQ[2]	×	6/6	2.50	69.62
	Ours	✓	6/6	2.50	**71.87**
DenseNet	Baseline	-	32/32	31.92	74.36
	OMSE[7]	✓	4/32	6.00	64.40
	Ours	✓	4/32	6.00	**70.15**

Table 2. Results of VGG-16 and ResNet-56 on CIFAR-10 dataset. 'P-R' represents the pruning ratio. 'Ave-Im' denotes the accuracy improvement compared to Prune. 'W-bit' denotes the bit-width of the weights.

| | P-R | Method | Criterion | | Ave-Im($\uparrow$) | W-bit | Size(MB) |
			l_2-norm	l_1-norm			
VGG-16 (Acc.93.70)	60%	Prune	89.14	88.70	0	32	21.6
		NM	93.16	93.16	4.24	32	21.6
		Ours	**93.40**	**93.40**	**4.48**	**6**	**4.04**
	70%	Prune	35.83	35.55	0	32	16.4
		NM	65.77	65.35	29.87	32	16.4
		Ours	**93.32**	**93.20**	**57.31**	**6**	**3.08**
	80%	Prune	18.15	17.56	0	32	11.2
		NM	40.26	39.49	22.02	32	11.2
		Ours	**91.79**	**91.26**	**73.67**	**6**	**2.12**
ResNet-56 (Acc.93.88)	30%	Prune	76.95	74.46	0	32	2.4
		NM	85.22	84.41	9.11	32	2.4
		Ours	**90.33**	**90.28**	**14.6**	**4**	**0.30**
	40%	Prune	46.44	49.68	0	32	2.0
		NM	76.56	77.89	24.16	32	2.0
		Ours	**86.99**	**87.29**	**39.08**	**4**	**0.24**
	50%	Prune	24.34	25.58	0	32	2.15
		NM	56.18	56.45	31.36	32	2.15
		Ours	**81.90**	**81.60**	**56.79**	**4**	**0.20**

landscape [23] to further illustrate the validity of UDFC, and the results are shown in **Appendix E**.

5.1. Ablation Study

Our proposed method consists of two compression techniques, quantization and pruning. Meanwhile, there exist hyperparameters α_1 and α_2 in the reconstruction error that impacts the compressed network performance. We perform the following ablation studies to evaluate the effects of different parts of our framework.

Study on pruning ratio and bit-width. UDFC performs pruning and quantization, the appropriate variables(i.e., pruning ratio and bit-width) become critical to compression ratio and performance of compressed model. Therefore, we compress ResNet-56 with different pruning ratios and bit-widths on CIFAR-10[21] dataset to explore the optimal trade-off between pruning and quantization.

As shown in Figure 2, the model performance decreases as the pruning ratio gradually increases. Similarly, the model performance also decreases as the weight bit width decreases, but at 4-bit the accuracy does not drop but rises. This peculiar phenomenon indicates that our method can maximize the restoration of information when quantizing ResNet-56 with 4-bit. We do not present quantified results for lower bits(i.e., 3-bit and 2-bit) because their accuracy drops sharply. At lower bits quantization, the loss of information is so great that our method cannot effectively restore the information.

Study on hyperparameters α_1 and α_2. To explore the effect of hyperparameters α_1 and α_2 on compressed network, we prune and quantize ResNet-56 separately on CIFAR-10 dataset. During the pruning, we use different α_1 values to prune 50% channels of ResNet-56. During the quantization, we use different α_2 values for 2-bit quantization of ResNet-56.

As shown in Figure 3, when α_1 increases from 0 to 0.01, the final performance of the pruned model increase steadily. However, when α_1 is set to 0.04, the accuracy suffers a huge drop. The curve of α_2 is similar to that of α_1, with the maximum performance at 0.008. This phenomenon confirms that the hyperparameters we introduced have improved the performance of the compressed model to some extent.

5.2. Quantization

We quantize ResNet-18[13], ResNet-50[13], MobileNetV2[14] and DenseNet[15] on ImageNet[35] dataset using the uniform quantization. In order to demonstrate the effectiveness of our method on quantization, we compare our method with DFQ[32], OMSE[7] and SQuant[10], which do not require any data and fine-tuning process. In addition, we also compare our method with some Post Training Quantization(PTQ) methods including ZeroQ[2], DDAQ[24], DSG[45] and ZAQ[25], which use the synthetic samples or calibration sets to improve the performance of quantized model.

Table 1 shows that our method has significant advantages

Table 3. Results of ResNet-101 and ResNet-34 on ImageNet dataset. 'P-R' represents the pruning ration. 'Ave-Im' denotes the accuracy improvement compared to Prune. 'W-bit' denotes the bit-width of weights.

	P-R	Method	Criterion		Ave-Im (↑)	W-bit	Size(MB)	FLOPS(G)
			l_2-norm	l_1-norm				
ResNet-101 (Acc.77.31%)	10%	Prune	69.10	68.52	0	32	154.4	6.84
		NM	72.46	71.95	3.40	32	154.4	6.84
		Ours	**74.69**	**74.61**	**5.84**	**6**	**28.8**	**6.84**
	20%	Prune	45.60	44.45	0	32	132.4	6.08
		NM	62.41	60.57	16.46	32	132.4	6.08
		Ours	**71.36**	**71.00**	**26.16**	**6**	**24.8**	**6.08**
	30%	Prune	10.10	9.560	0	32	112.4	5.3
		NM	38.44	37.68	28.23	32	112.4	5.3
		Ours	**65.76**	**65.22**	**55.66**	**6**	**21.2**	**5.3**
ResNet-34 (Acc.73.27%)	10%	Prune	63.51	61.95	0	32	78.8	3.24
		NM	67.10	66.50	4.35	32	78.8	3.24
		Ours	**69.86**	**69.39**	**6.89**	**6**	**14.8**	**3.24**
	20%	Prune	42.80	40.62	0	32	70.0	2.88
		NM	55.70	54.20	13.24	32	70.0	2.88
		Ours	**65.44**	**64.68**	**23.35**	**6**	**13.2**	**2.88**
	30%	Prune	16.80	12.60	0	32	61.6	2.52
		NM	39.40	36.34	23.17	32	61.6	2.52
		Ours	**59.25**	**57.57**	**43.71**	**6**	**11.6**	**2.52**

compared to DFQ, OMSE and other PTQ methods for various models. For instance, when quantizing the weights of ResNet-18 with 6-bit, our method achieves 72.76% accuracy that is 6.46% higher than DFQ and 1.9% higher than DSG. Our method remains robust to low-bit quantization of the lightweight model MobileNetV2(71.87%) and DenseNet(70.15%). In addition, our method has a tremendous advantage in time consumption. ZeroQ takes 29 seconds to quantize ResNet50 on an 8 Tesla V100 GPUs, while UDFC only takes 2 seconds on a RTX 1080Ti GPU.

5.3. Unified Compression.

In this section, we compress the ResNet-56 and VGG-16[37] on CIFAR-10[19] dataset, ResNet-34 and ResNet-101 on ImageNet dataset to demonstrate the effectiveness of our method. Since no data-free method can perform both pruning and quantization simultaneously, we mainly compare our method with data-free pruning methods. In the field of pruning, our direct competitor is Neuron Merging(NM)[18], which is a one-to-one compensation method. Same as Neuron Merging, we do not perform any compensation after pruning as a way to obtain the baseline performance, called $Prune$.

Experiments on CIFAR-10. For the CIFAR-10 dataset, we test UDFC on ResNet-56 and VGG-16 with different pruning rates: 30%-80%. In addition, we quantize the unpruned channels to 4-bit and 6-bit respectively, further reducing the memory footprint of parameters.

As shown in Table 2, UDFC achieves state-of-the-art performance. For example, with about $28\times$ parameters drop(0.53M) and 80% FLOPS reduction, VGG-16 still has excellent classification accuracy (91.26%), which is 51% average accuracy higher than NM at a 80% pruning ratio.

Experiments on ImageNet. For the ImageNet dataset, we test UDFC on ResNet-34 and ResNet-101 with pruning rates: 10%, 20% and 30%. In addition, we quantize the unpruned channels to 6-bit, further reducing the memory footprint of parameters.

Table 3 shows that UDFC outperforms the previous method. By varying the ratio of pruning from 10% to 30%, the Ave-Im increases accordingly compared to NM and Prune. That means our method is more robust than one-to-one compensation. For ResNet-101, we get a 55.66% improvement in accuracy compared to Prune and a 27.23% improvement compared to NM at a pruning ratio of 30%. Meanwhile, the parameters are substantially reduced due to the quantization, so that not only do we achieve higher performance but also a lower memory footprint of parameters both in ResNet-34 and ResNet-101.

6. Conclusion

In this paper, we propose a unified data-free compression framework that performs pruning and quantization simultaneously without any data and fine-tuning process. It starts with the assumption that the partial information of a damaged channel can be preserved by a linear combination of

other channels and then gets a fresh approach from the assumption to restore the information loss caused by compression. Extensive experiments on benchmark datasets validate the effectiveness of our proposed method.

7. Acknowledgement

References

[1] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):1–18, 2017. 2

[2] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020. 1, 3, 6, 7

[3] Jun Chen, Shipeng Bai, Tianxin Huang, Mengmeng Wang, Guanzhong Tian, and Yong Liu. Data-free quantization via mixed-precision compensation without fine-tuning. *Pattern Recognition*, page 109780, 2023. 1

[4] Jun Chen, Hanwen Chen, Mengmeng Wang, and Yong Liu. Learning discretized neural networks under ricci flow. *arXiv preprint arXiv:2302.03390*, 2023. 1

[5] Jun Chen, Liang Liu, Yong Liu, and Xianfang Zeng. A learning framework for n-bit quantized neural networks toward fpgas. *IEEE transactions on neural networks and learning systems*, 32(3):1067–1081, 2020. 1

[6] Kanghyun Choi, Deokki Hong, Noseong Park, Youngsok Kim, and Jinho Lee. Qimera: Data-free quantization with synthetic boundary supporting samples. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 1

[7] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019. 6, 7

[8] Allen Gersho and Robert M. Gray. Vector quantization and signal compression. In *The Kluwer international series in engineering and computer science*, 1991. 1

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[10] Cong Guo, Yuxian Qiu, Jingwen Leng, Xiaotian Gao, Chen Zhang, Yunxin Liu, Fan Yang, Yuhao Zhu, and Minyi Guo. SQuant: On-the-fly data-free quantization via diagonal hessian approximation. In *International Conference on Learning Representations*, 2022. 6, 7

[11] Suyog Gupta, Ankur Agrawal, K. Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning*, 2015. 2, 3

[12] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 7

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 7

[16] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016. 3

[17] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. 1

[18] Woojeong Kim, Suhyun Kim, Mincheol Park, and Geunseok Jeon. Neuron merging: Compensating for pruned neurons. *Advances in Neural Information Processing Systems*, 33:585–595, 2020. 1, 3, 8

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 2, 8

[20] César Laurent, Camille Ballas, Thomas George, Nicolas Ballas, and Pascal Vincent. Revisiting loss modelling for unstructured pruning. *ArXiv*, abs/2006.12279, 2020. 1

[21] Vadim Lebedev and Victor Lempitsky. Fast convnets using group-wise brain damage. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2554–2564, 2016. 7

[22] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *ArXiv*, abs/1608.08710, 2017. 1, 3

[23] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems*, 2018. 7

[24] Zhikai Li, Liping Ma, Xianlei Long, Junrui Xiao, and Qingyi Gu. Dual-discriminator adversarial framework for data-free quantization. *Neurocomputing*, 2022. 1, 6, 7

[25] Yuang Liu, Wei Zhang, and Jun Wang. Zero-shot adversarial quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1512–1521, 2021. 7

[26] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017. 2

[27] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2755–2763, 2017. 3

[28] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *ArXiv*, abs/1810.05270, 2019. 1

[29] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017. 3

[30] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pages 2498–2507. PMLR, 2017. 2

[31] Ben Mussay, Margarita Osadchy, Vladimir Braverman, Samson Zhou, and Dan Feldman. Data-independent neural pruning via coresets. *arXiv preprint arXiv:1907.04018*, 2019. 3

[32] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1334, 2019. 1, 3, 6, 7

[33] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016. 3

[34] Masuma Akter Rumi, Xiaolong Ma, Yanzhi Wang, and Peng Jiang. Accelerating sparse cnn inference on gpus with performance-aware weight pruning. *Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques*, 2020. 1

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 7

[36] Yasufumi Sakai, Yusuke Eto, and Yuta Teranishi. Structured pruning for deep neural networks with adaptive pruning rate derivation based on connection sensitivity and loss function. *Journal of Advances in Information Technology*, 2022. 1

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 8

[38] Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015. 1, 3

[39] Jialiang Tang, Mingjin Liu, Ning Jiang, Huan Cai, Wenxin Yu, and Jinjia Zhou. Data-free network pruning for model compression. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2021. 3

[40] Yehui Tang, Shan You, Chang Xu, Jin Han, Chen Qian, Boxin Shi, Chao Xu, and Changshui Zhang. Reborn filters: Pruning convolutional neural networks with limited data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5972–5980, 2020. 3

[41] Ziheng Wang, Jeremy Wohlwend, and Tao Lei. Structured pruning of large language models. In *EMNLP*, 2020. 1

[42] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 1–17. Springer, 2020. 6

[43] Li yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000. 1

[44] Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *The IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[45] Xiangguo Zhang, Haotong Qin, Yifu Ding, Ruihao Gong, Qinghua Yan, Renshuai Tao, Yuhang Li, Fengwei Yu, and Xianglong Liu. Diversifying sample generation for accurate data-free quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15658–15667, 2021. 1, 3, 6, 7

[46] Yunshan Zhong, Mingbao Lin, Gongrui Nan, Jianzhuang Liu, Baochang Zhang, Yonghong Tian, and Rongrong Ji. Intraq: Learning synthetic images with intra-class heterogeneity for zero-shot network quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12339–12348, 2022. 1, 3

[47] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. 3