DFFG: Fast Gradient Iteration for Data-free Quantization

Huixing Leng[1*]
lenghuixing@buaa.edu.cn
Shuangkan Fang[1*]
skfang@buaa.edu.cn
Yufeng Wang[2†]
wyfeng@buaa.edu.cn
Zehao Zhang[1]
zhangzehao@buaa.edu.cn
Dacheng Qi[1]
dc_qi@buaa.edu.cn
Wenrui Ding[2]
ding@buaa.edu.cn

[1] The School of Electrical and Information Engineering, Beihang University, Beijing 100191, China

[2] Institute of Unmanned System, Beihang University, Beijing 100191, China

Abstract

Model quantization is a technique that optimizes neural network computation by converting weight parameters and activation values from floating-point numbers to low-bit integers or fixed-point representations. This reduces storage and computational cost and improves computational efficiency. Currently, common quantization methods, such as QAT and PTQ, optimize quantization parameters using training data to achieve the best performance. However, in practical applications, there may be little or no data available for downstream model quantization due to restrictions such as privacy and security. Therefore, researching how to perform model quantization without data is essential. This article proposes a data-free quantization technique called DFFG, based on fast gradient iteration, which uses information learned from the full-precision model, such as the BN layer, to recover the distribution of the original training data. We propose, for the first time, using a momentum-assisted variant of the FGSM gradient iteration strategy to update the generated data. This approach enables quick perturbation of the optimized data while maintaining the diversity of the generated data through the manipulation of gradient variability. We also propose using intermediate data generated during the iteration process as a part of data for subsequent model quantization, greatly improving the speed of data generation. We have demonstrated the effectiveness of our proposed method through empirical evaluations. Our method generates data that not only ensures model quantization performance but also significantly surpasses other similar data generation techniques in terms of speed. Specifically, our approach is 10X faster than ZeroQ.

* Equal contribution. † Corresponding author.

Figure 1: these images are generated by our method (DFFG) given a pretrained ResNet-50 model, classes top to bottom: goldfish, orange, castle, hotpot.

1 Introduction

Deep neural networks(DNNs) have shown tremendous success in various domains such as image classification [16, 20, 25, 46, 48], object detection [42, 43], image super-resolution [1, 44, 56], 3D reconstruction [10, 11, 39, 53] and others. However, deploying these models on resource-constrained devices, such as mobile phones and embedded systems, remains a challenge due to their high computational and memory requirements [12, 27, 29, 52]. To overcome this challenge, various methods have been investigated to reduce model complexity, such as knowledge distillation [19], pruning [18, 30], and model quantization[12, 23, 34], which is one of the widely adopted techniques.

Model quantization is the process of reducing the precision of the weights and activations of a deep learning model. By quantizing the model, we can reduce its memory footprint and computational complexity, making it feasible to deploy on low-power devices. In recent years, research on model quantization has gained significant attention due to its practical importance [2, 4, 12, 29, 34, 34, 52, 57].

However, traditional model quantization techniques require access to the training dataset, which is known as data-dependent quantization. This approach is not always practical since obtaining and storing the entire dataset may be costly or impossible due to privacy concerns. One solution is data-independent quantization, or data-free quantization, which aims to perform model quantization without accessing the training data.

In recent years, significant progress has been made in data-free model quantization, with various techniques proposed for quantizing deep neural networks [4, 5, 6, 15, 32, 50, 55]. For example, ZeroQ [4] achieves zero-shot / data-free post-quantization by reconstructing data impressions via BNS and supports mixed precision quantization with a Pareto frontier-based determination. DSG [55] enhances the diversity of data, Qimera [5] put forward a method that uses superposed latent embeddings to generate synthetic boundary supporting samples, IntraQ [57] is proposed to well retain the intra-class heterogeneity in the synthetic images, due to Generative Adversarial Networks(GAN) [13] have received wide attention in the image synthesis field for their potential to learn high-dimensional and complex real data distribution, GDFQ [50] ploys a generator to synthesize training data, ZAQ [31] drives a generator to synthesize informative and diverse data examples to optimize the quantized model in an adversarial learning fashion.

Despite the remarkable progress has been made in data-free model quantization, there are still several challenges and limitations that need to be addressed. One of the major limitations is that the generation of high-quality synthetic data which can follow real data distribution requires a significant amount of time and effort, and this will slow down the quantization process. Another limitation of data-free model quantization is that some techniques will lead to a huge loss of accuracy when the model is quantized to a lower bit such as 4 bit.

To address the aforementioned issue, we propose a novel data generation technique for model quantization in this paper, called DFFG. We introduce a fast gradient iterative strategy with momentum to update the generated data, which is different from the methods that focus solely on designing corresponding constraints to generate higher-quality images. Moreover, our method achieves significant breakthroughs in the speed of image generation. Specifically, our contributions are as follows:

- We present a novel data generation technique for model quantization, called DFFG, which utilizes a momentum term in the fast gradient iteration process to recover model training data from the inherent information in the full-precision model, assisting in low-bit quantization of models in the absence of data.

- We leverage the fast and diverse image generation capabilities of DFFG to simultaneously consider both image generation speed and quality. By extracting intermediate-generated images during a full iteration process, we further reduce image generation time.

- We validate the effectiveness of our approach on benchmark datasets. The images generated using our technique can be directly applied in PTQ, QAT, and distillation quantization strategies. Our generated data significantly surpasses other similar data generation techniques in terms of generation speed, as evidenced by our experiments, with our method being 10X faster than ZeroQ.

2 Related works

2.1 Quantization with data

The process of network quantization can effectively compress the size of a model and accelerate inference by representing the full-precision model (FP-32) using low-bit integers, such as 8-bit, 6-bit, 4-bit, etc [2, 4, 12, 14, 21, 28, 29, 49, 52, 57]. One of the most widely used techniques is weight quantization, which involves reducing the precision of the weights in the model to lower the memory footprint and improve inference speed. Various methods have been proposed for weight quantization, including Post-Training Quantization (PTQ) and Quantization-Aware training (QAT). There are different approaches to PTQ, including uniform quantization [3, 47], where all values are quantized to a fixed number of bits, and non-uniform quantization [28, 41, 47], where the number of bits used for each value varies based on its importance. QAT involves simulating the quantization process during training by using lower precision weights and activations, and minimizing the loss function accordingly. The objective of quantization-aware training is to train a model that is both accurate and resilient to the reduced precision of quantization [33, 35, 36, 45]. However, both PTQ and QAT typically require real data to quantize the model. PTQ relies on real training data to approximate the optimal activation clipping value [2], whereas QAT requires training data to retrain the quantization model, focusing on the design of quantizers [9, 28], training strategies [26, 58], and dynamic quantization [21, 54], which enables competitive results with lower bit quantization.

2.2 Quantization without data

There are two main ways to address the issue of model quantization without relying on real data. The first approach involves analyzing and utilizing the structure information of the model, such as DFQ [34], which proposes weight equalization and bias correction without fine-tuning. However, such method may result in significant performance degradation when quantization methods with ultra-low precision are employed. The second approach involves synthesizing alternatives to the original training data, which can be classified into three categories based on the synthesis algorithms: noise optimization [4, 5, 51, 57], generative reconstruction [17, 50], and adversarial exploration [7, 31]. Noise optimization samples noise from a Gaussian distribution as input and optimizes it iteratively with gradient descent until certain constraints are met. ZeroQ [4] and IntraQ [57] are typical methods in this category. Generative reconstruction aims to design a generator to synthesize images. GDFQ [50] adopts generative models guided by both the batch normalization statistics and extra category label information to synthesize samples. Adversarial exploration provides an adversarial learning perspective where the generator aims to maximize the model discrepancy, while the quantization model is trained to minimize it. ZAQ [31] generates adversarial samples via a generator, by maximizing the discrepancy between full-precision model and quantization model, and minimizing their gap to benefit quantization model for calibration. Despite these synthesis algorithms achieve considerable performance gain, a performance gap still exists between fine-tuning with synthetic and real data.

3 Method

3.1 Preliminaries

Quantizer. In this study, we consider a commonly used and straightforward quantizer design, which uses an asymmetric uniform quantizer to implement network quantization, following previous works such as [50, 57]. The quantizer quantizes the weight of a full-precision model P, the weight is denoted as θ, and the lower and upper bounds of θ is denoted as l and u. The quantizer produces the quantized integer θ^q by restricting the range of θ into n bit as follows:

$$\theta^q = round\left(\theta \times S - Z\right),\ S = \frac{2^n - 1}{u - l},\ Z = S \times l + 2^{n-1} \tag{1}$$

where S is the scale factor that converts the range of θ to n-bit, Z is zero-point, and $round\left(\cdot\right)$ denotes that round the number to the nearest integer. To evaluate the performance of a neural network on a quantized device, the quantized behavior is often simulated during the training

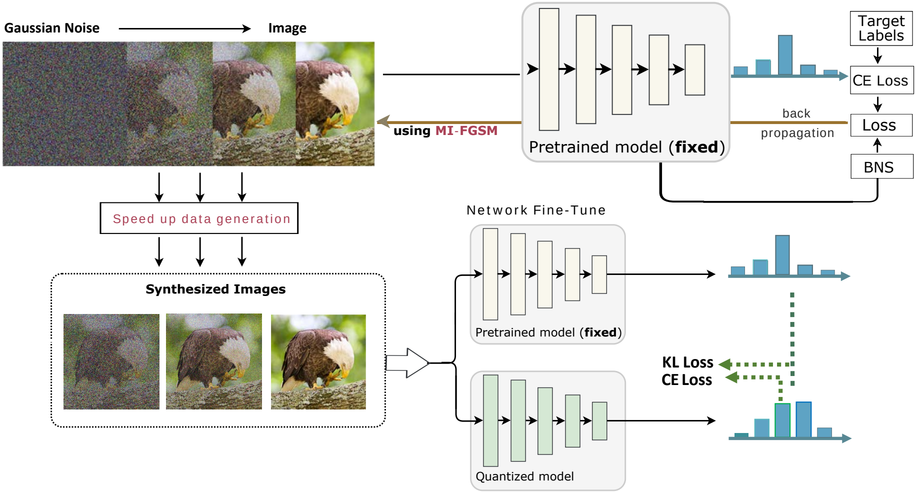

Figure 2: Given a noise sample with a random label, our method can gradually recover the original image from the noise by reducing the loss in Eq.6. We utilize a strategy of fast gradient iteration to achieve this optimizing process, in which we extract the intermediate images as part of the final images, greatly accelerating the speed of image generation. After getting the synthesized images, we can use them to obtain a quantized model by reducing the CE loss and KL loss in Eq.12.

process, which is known as quantization simulation. The corresponding de-quantized value can be calculated as follows:

$$\theta' = \frac{\theta^q + Z}{S}. \tag{2}$$

Using low-bit integers to represent the weight of full-precision models is made possible by the quantizer. However, there may be a gap between θ and θ', which can lead to performance degradation when using the dequantized parameter θ' for inference.

3.2 DFFG

Our quantization process involves two steps as shown in Fig.2. In the first step, we leverage the knowledge learned by the full precision model and design an appropriate loss function to synthesize image data. Specifically, we utilize the momentum iterative fast gradient sign method (MI-FGSM) [8] to ensure rapid convergence in the data generation process, leveraging its fast iterative nature. We also utilize the directional variability of its gradients to promote diversity in the generated data. We implement data extraction during the intermediate iteration process to further accelerate the data generation speed, maintaining diversity of image. In the second step, we fine-tune the model using these synthetic images, which adequately capture the essence of the authentic data. This method allows us to obtain a closer approximation to the original model at a lower bit quantization.

3.2.1 Loss design

During the generation of synthetic images, our method progresses from random noise to visually coherent pictures through an iterative optimization process. To design the loss function, we utilize the commonly used batch normalization (BN) layer in neural network, adopting the batch normalization statistics (BNS) loss. Specifically, we optimize a set of noise samples $\{\mathbf{x}_i\}_{i=1}^N$ to match the BNS of the pre-trained model, resulting in the synthesis of high-quality images:

$$\begin{aligned}\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{L}_{BNS} = \frac{1}{L}\sum_{l=1}^{L}(||\mu^l(\theta) - \mu^l(\theta, \{\mathbf{x}_i\}_{i=1}^N)|| \\ +||\sigma^l(\theta) - \sigma^l(\theta, \{\mathbf{x}_i\}_{i=1}^N)||),\end{aligned} \tag{3}$$

The mean/variance parameters: $\mu^l(\theta)/\sigma^l(\theta)$, are stored in the l-th BN layer of the full-precision model after trained with real sample.To synthesize images, we calculate $\mu^l(\theta, \{\mathbf{x}_i\}_{i=1}^N)$ and $\sigma^l(\theta, \{\mathbf{x}_i\}_{i=1}^N)$ by feeding the noise samples into the full-precision model with parameter θ. To ensure consistency with the distribution of real data, we add a classification loss (e.g. cross-entropy) using random Gaussian noise and a random target label. We calculate the cross-entropy loss between the given target label and the output of the full-precision network:

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{L}_{CE} = \frac{1}{N}\sum_{i=1}^{N} CE\left(P\left(\mathbf{x}_i;\theta\right), \mathbf{y}_i\right) \tag{4}$$

where $P(\cdot;\theta)$ stands for the output of full-precision model parameterized with θ, $CE(\cdot,\cdot)$ represents the cross-entropy loss, and $\mathbf{y}_i$ is the label assigned to $\mathbf{x}_i$ as a prior classification knowledge. Typically images with better visual quality are closer to the real training samples. To incorporate this idea, we introduce an image regularization term: $\mathcal{R}(\cdot)$, which consists of the total variation (TV) loss and a L_2 regularization term to avoid overly large pixel values in the generated images:

$$\mathcal{R}_{\text{prior}}(\mathbf{x}) = \alpha_{\text{tv}}\mathcal{R}_{\text{TV}}(\mathbf{x}) + \alpha_{\ell_2}\mathcal{R}_{\ell_2}(\mathbf{x}), \tag{5}$$

where R_{TV} and R_{ℓ_2} penalize the total variance and ℓ_2 norm of $\mathbf{x}$, respectively, with scaling factors α_{tv}, α_{ℓ_2}. The image-prior regularization provides a more stable convergence toward valid images. Consequently, the total loss used during the data generation process is summarized as follows:

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{L}_{total} = \mathcal{L}_{BNS} + \alpha_{\text{tv}}\mathcal{R}_{\text{TV}}(\mathbf{x}_i) + \alpha_{\ell_2}\mathcal{R}_{\ell_2}(\mathbf{x}_i) + \beta\mathcal{L}_{CE}, \tag{6}$$

where α_{tv}, α_{ℓ_2}, β are hyper-parameters balancing the importance of these four terms.

3.2.2 Optimizer

When choosing an optimizer, many noise-optimized image generation methods, such as IntraQ [57], use the Adam [22]optimizer. However, our experiment results have shown that the use of Adam optimizer often results in a lower loss of image generation, which can lead to poor performance of generating images on classification boundary. To address this issue, we adopt the fast gradient sign method (FGSM), which has a greater gradient variability during the image optimization process, maintaining the diversity of images. The iterative process of FGSM can be represented as follows:

$$x^* = x + \varepsilon \cdot \operatorname{sign}\left(\nabla_x J(x,y)\right) \tag{7}$$

Where x is the object of iteration, y is the label of x , and $J(x,y)$ is the loss function. The ε control the magnitude of disturbance. In order to better stabilize the updating direction and get rid of the bad local maximum in the iterative process, the momentum iterative gradient MI-FGSM [8] was put forward. Here we further take MI-FGSM to refine the noise with label information. The iterative process of MI-FGSM we use in the method can be represented as follows:

$$g_{t+1} = \mu \cdot g_t + \frac{J\left(x_t^*, y^*\right)}{\left\|\nabla_x J\left(x_t^*, y^*\right)\right\|_1} \tag{8}$$

$$x_{t+1}^* = x_t^* - \varepsilon \cdot \operatorname{sign}\left(g_{t+1}\right) \tag{9}$$

Here introduce a momentum term for a more stable iterative process.

3.2.3 Speed up data generation

Commonly used quantization methods for data generation save the last batch of images after iteration, which has two drawbacks: a longer iteration period resulting in fewer images generated, and a fixed number of iterations reducing the diversity of generated samples. To address these issues, we propose a new strategy that involves saving some of the images during the intermediate iteration process. This approach has two benefits: first, saving images multiple times during a complete iteration cycle greatly speeds up data generation, and second, the extracted images in the intermediate iteration process access a soft label, increasing the likelihood of their appearance on the classification boundary.

3.3 Network Fine-Tuning

After obtaining the generated images I by our method, we adopt a similar training strategy as IntraQ. We use the full-precision model as the teacher to distill the quantized model. The designed loss contains two parts. The first one is as follows:

$$\mathcal{L}_{\mathrm{CE}}^{Q} = CE(Q(I), y) \tag{10}$$

This is the cross-entropy loss between the label and the output of quantized network Q. And the second one is:

$$\mathcal{L}_{\mathrm{KD}}^{Q} = KL(Q(I), P(I)) \tag{11}$$

This is the KL divergence between the output of the full-precision model P and the output of the quantized model Q. And the total loss for network fine-tuning can be summarized as:

$$\mathcal{L}^{Q} = \mathcal{L}_{\mathrm{CE}}^{Q} + \alpha \cdot \mathcal{L}_{\mathrm{KD}}^{Q} \tag{12}$$

where α balances the importance of $\mathcal{L}_{\mathrm{CE}}^{Q}$ and $\mathcal{L}_{\mathrm{KD}}^{Q}$.

Dataset	Model	Bit width	Real Data	ZeroQ	DSG	ZAQ	Qimera	GDFQ	IntraQ	DFFG (ours)
CIFAR-10	ResNet-20 (93.89)	3w3a	87.94	69.53	48.99	-	-	71.1	77.07	**84.68**
		4w4a	91.52	89.66	88.93	**92.13**	91.26	90.25	91.49	91.63
		5w5a	-	-	-	93.36	**93.46**	93.38	-	93.30
CIFAR-100	ResNet-20 (70.33)	3w3a	56.26	26.35	43.42	-	-	43.87	48.25	**52.13**
		4w4a	66.8	63.97	62.62	60.42	65.1	63.58	64.98	**66.30**
		5w5a	-	-	-	68.7	69.02	67.52	-	**69.30**
ImageNet	ResNet-18 (71.59)	4w4a	67.89	63.38	63.11	52.64	63.84	60.6	66.47	**66.69**
		5w5a	70.31	69.72	69.53	64.54	69.29	66.82	69.94	**70.03**
	MobileNetV2 (73.08)	4w4a	67.9	60.15	60.45	0.1	61.62	51.3	65.10	**65.63**
		5w5a	72.01	70.95	70.87	62.35	70.45	68.14	71.28	**71.53**
	ResNet-50 (77.76)	4w4a	-	-	-	53.02	66.25	54.16	-	**69.47**
		5w5a	-	-	-	73.38	75.32	71.63	-	**75.83**

Table 1: Comparison with other data-free quantization methods. -: no results are reported in the given paper. nwna indicates the weights and activations are quantized to n bit. The data below the model represent the accuracy at full precision.

4 Experiments

4.1 Implementation details

We verify the final experimental effect on three typical image classification datasets, which are CIFAR-10 [24], CIFAR-100 [24], and ImageNet [25]. The networks we used to evaluate our method include ResNet-20 [16] for CIFAR, ResNet-18 [16], ResNet-50 [16], and MobileNetV2 [46] for ImageNet. We record the top-1 accuracy on validation sets. All pretrained models are from the PytorchCV library, and all experiments are implemented with Pytorch [38]. In order to generate the image, we optimize the loss function using the MI-FGSM with a step size of 0.1, and a momentum of 0.9, and the final images we generated are showed in Figure1.

For network quantization, we employ SGD with Nesterov [37] and set the momentum to 0.9, and the weight decay to 10^{-4} to optimize the quantized models using the loss function described in Eq. (3.3). For CIFAR and ImageNet, we set the batch size to 64 and 4, respectively. The initial learning rate is set to 10^{-5} for CIFAR and 10^{-6} for ImageNet. We decay both learning rates by 0.1 every 100 epochs and a total of 150 epochs are given.

4.2 Quantization performance

In Table 1, we compared the performance of our method with currently available data-free methods. As shown, on the CIFAR dataset, utilizing our method to generate images for subsequent quantization achieves accuracy levels that are close to those achieved with real data when quantizing with 4-bit or 5-bit precision. In the case of lower bit precision, such as 3-bit, our method outperforms other approaches by a significant margin. This demonstrates that our method is capable of handling model quantization tasks with small datasets.

We futher evaluate the performance of ResNet18, ResNet50, and MobileNetV2 on the larger ImageNet dataset. The results demonstrate that our approach outperforms the baseline in terms of classification accuracy. The success of our method is attributed to the multi-variability of gradients in the fast iteration scheme, which enables a more diverse range of samples to be generated. This advantage allows our method to surpass other methods.

Dataset	model	save points	speed	3w3a	4w4a	5w5a
ImageNet	ResNet-18	10	2X	43.06	66.69	70.03
		6,10	4X	42.74	66.36	69.96
		6,7,9,10	8X	41.36	66.46	70.09
		6,7,8,9,10	10X	41.37	66.39	70.10
		baseline ZeroQ:		-	63.38	69.72
	MobileNetV2	10	2X	-	65.63	71.53
		6,10	4X	-	65.79	71.45
		baseline ZeroQ:			60.15	70.95
CIFAR-10	ResNet-20	10	2X	84.68	91.63	93.30
		6,10	4X	83.11	91.62	93.42
		6,7,8,9,10	10X	83.71	91.59	93.34
		baseline ZeroQ:		69.53	89.66	-
CIFAR-100	ResNet-20	10	2X	52.13	66.30	69.30
		6,10	4X	51.92	66.51	69.18
		6,7,8,9,10	10X	51.71	66.12	69.29
		baseline ZeroQ:		26.35	63.97	-

Table 2: Comparison with different save points when generating data for speeding up. Where '10X' indicates that our method is 10 times faster than ZeroQ in generating data.

4.3 Accelerate data generation

In addition to enhancing the performance of quantized models, our method has another advantage of significantly improving the speed of data generation. By extracting and saving intermediate iterated data at different iteration counts, our approach achieves flexible control over the acceleration factor of data generation. We compared our method to ZeroQ under different acceleration factors, and the results are shown in Table 2.

Our method achieves higher accuracy for quantized models compared to ZeroQ, even at a 10X speedup. We achieve this by designing a gradient-rich data iteration strategy and selectively extracting intermediate results, which accomplishes a better balance between data generation quality and speed.

4.4 Ablation studies

model	Bit width	Adam	DFFG	Diff
ResNet-18 **(71.59)**	3w3a	37.68	**43.06**	**+5.38**
	4w4a	66.28	**66.69**	**+0.41**
	5w5a	**70.09**	70.03	-0.06
ResNet-50 **(77.76)**	4w4a	67.46	**69.47**	**+2.01**
	5w5a	75.52	**75.83**	**+0.31**

Table 3: Ablation study of comparison with Adam optimizer.

In this section, we perform ablation studies to evaluate the efficiency of MI-FGSM compared to Adam. The step size of Adam is set to 0.1 and momentum to 0.9, which is the same as MI-FGSM. The experiments are carried out on ResNet-18 and ResNet-50 models, and the results are presented in Tab.3. As observed, our method outperforms Adam in most cases.

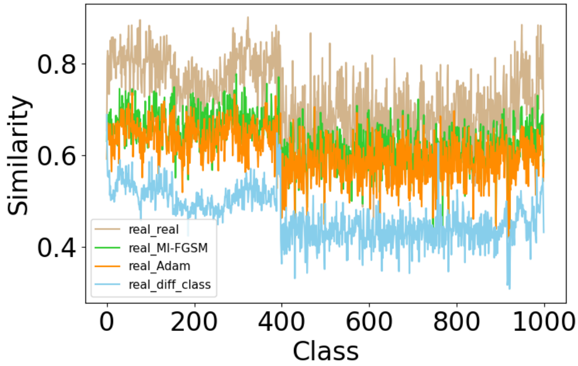

Figure 3: The CLIP similarity between different images.

To measure how similar the images we generated to the images of training. We use the CLIP[40] model to calculate the similarity between the images. We selected 10 images for each category from training data and the data generated from ResNet-18 using MI-FGSM and Adam. We calculated the average CLIP similarity between training data of the same category (real_real), training data of different categories (real_diff_class), our generated data and the training data (real_MI-FGSM), the generated data using Adam and the training data (real_Adam). The results are shown in Figure 3. It can be seen that the CLIP similarity between the data generated by our method and the real data is slightly lower than the similarity between real data of the same category, slightly higher than the similarity using Adam and significantly higher than the similarity between data from different categories, which indicates that our method can effectively simulate real data. Furthermore, we investigated the impact of different save points during the iterations on the quantization performance, and the results are shown in Table 2. It is observed that the performance of our method is basically the same when it is under 4 times faster than the ZeroQ, which shows that saving the data in the middle of the iterative process can not only ensure the quantization performance but also improve the speed of data generation.

5 Conclusions

In this article, we present a novel data-free quantization technique named DFFG, which utilizes a fast gradient iteration strategy and leverages information from the full-precision model, including the BN layer, to recover the distribution of the original training data. We introduce MI-FGSM to update the generated data, which allows for quick perturbation of the optimized data and guarantees the diversity of the generated data by manipulating the gradient variability. Furthermore, we propose utilizing intermediate data generated during the iteration process as data for subsequent model quantization, significantly improving the speed of data generation. Empirical evaluations demonstrate the effectiveness of our approach, which generates data that not only ensures model quantization performance but also significantly outperforms other similar data generation techniques in terms of speed.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant U20B2042 and 62076019.

References

[1] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[2] Ron Banner, Yury Nahshan, Daniel Soudry, et al. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 7950–7958, 2019.

[3] Chaim Baskin, Natan Liss, Eli Schwartz, Evgenii Zheltonozhskii, Raja Giryes, Alex M Bronstein, and Avi Mendelson. Uniq: Uniform noise injection for non-uniform quantization of neural networks. *ACM Transactions on Computer Systems (TOCS)*, 37(1-4): 1–15, 2021.

[4] 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 (CVPR)*, pages 13169–13178, 2020.

[5] Kanghyun Choi, Deokki Hong, Noseong Park, Youngsok Kim, and Jinho Lee. Qimera: Data-free quantization with synthetic boundary supporting samples. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[6] Kanghyun Choi, Hyeyoon Lee, Deokki Hong, Joonsang Yu, Noseong Park, Youngsok Kim, and Jinho Lee. It's all in the teacher: Zero-shot quantization brought closer to the teacher. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[7] Yoojin Choi, Jihwan Choi, Mostafa El-Khamy, and Jungwon Lee. Data-free network quantization with adversarial knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 710–711, 2020.

[8] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[9] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[10] Shuangkang Fang, Weixin Xu, Heng Wang, Yi Yang, Yufeng Wang, and Shuchang Zhou. One is all: Bridging the gap between neural radiance fields architectures with progressive volume distillation. *arXiv preprint arXiv:2211.15977*, 2022.

[11] Shuangkang Fang, Yufeng Wang, Yi Yang, Weixin Xu, Heng Wang, Wenrui Ding, and Shuchang Zhou. Pvd-al: Progressive volume distillation with active learning for efficient conversion between different nerf architectures. *arXiv preprint arXiv:2304.04012*, 2023.

[12] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[14] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[15] Matan Haroush, Itay Hubara, Elad Hoffer, and Daniel Soudry. The knowledge within: Methods for data-free model compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2020.

[16] 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.

[17] Xiangyu He, Jiahao Lu, Weixiang Xu, Qinghao Hu, Peisong Wang, and Jian Cheng. Generative zero-shot network quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3000–3011, 2021.

[18] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4340–4349, 2019.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[20] 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.

[21] Qing Jin, Linjie Yang, and Zhenyu Liao. Adabits: Neural network quantization with adaptive bit-widths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2146–2156, 2020.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[23] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[26] Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6448–6457, 2021.

[27] Yangyang Li, Shuangkang Fang, Xiaoyu Bai, Licheng Jiao, and Naresh Marturi. Parallel design of sparse deep belief network with multi-objective optimization. *Information Sciences*, 533:24–42, 2020.

[28] Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[29] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461: 370–403, 2021.

[30] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. Hrank: Filter pruning using high-rank feature map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1529–1538, 2020.

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

[32] Yuang Liu, Wei Zhang, Jun Wang, and Jianyong Wang. Data-free knowledge transfer: A survey. *arXiv preprint arXiv:2112.15278*, 2021.

[33] Yuriy Mishchenko, Yusuf Goren, Ming Sun, Chris Beauchene, Spyros Matsoukas, Oleg Rybakov, and Shiv Naga Prasad Vitaladevuni. Low-bit quantization and quantization-aware training for small-footprint keyword spotting. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 706–711. IEEE, 2019.

[34] 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 (ICCV)*, pages 1325–1334, 2019.

[35] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, pages 16318–16330. PMLR, 2022.

[36] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, 110(11-12):3245–3262, 2021.

[37] Yurii Evgen'evich Nesterov. A method of solving a convex programming problem with convergence rate o(kˆ2). In *Proceedings of the Russian Academy of Sciences (RAS)*, pages 543–547, 1983.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 8026–8037, 2019.

[39] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[41] Ali Ramezani-Kebrya, Fartash Faghri, Ilya Markov, Vitalii Aksenov, Dan Alistarh, and Daniel M Roy. Nuqsgd: Provably communication-efficient data-parallel sgd via nonuniform quantization. *The Journal of Machine Learning Research*, 22(1):5074–5116, 2021.

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[44] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[45] Charbel Sakr, Steve Dai, Rangha Venkatesan, Brian Zimmer, William Dally, and Brucek Khailany. Optimal clipping and magnitude-aware differentiation for improved quantization-aware training. In *International Conference on Machine Learning*, pages 19123–19138. PMLR, 2022.

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[47] Sanghyun Seo and Juntae Kim. Efficient weights quantization of convolutional neural networks using kernel density estimation based non-uniform quantizer. *Applied Sciences*, 9(12):2559, 2019.

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[49] Peisong Wang, Qiang Chen, Xiangyu He, and Jian Cheng. Towards accurate post-training network quantization via bit-split and stitching. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 9847–9856, 2020.

[50] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–17, 2020.

[51] Erkun Yang, Dongren Yao, Tongliang Liu, and Cheng Deng. Mutual quantization for cross-modal search with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7551–7560, 2022.

[52] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8715–8724, 2020.

[53] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[54] Haichao Yu, Haoxiang Li, Humphrey Shi, Thomas S Huang, and Gang Hua. Any-precision deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 10763–10771, 2021.

[55] 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 (CVPR)*, pages 15658–15667, 2021.

[56] Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. Data-free knowledge distillation for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7852–7861, 2021.

[57] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[58] Bohan Zhuang, Lingqiao Liu, Mingkui Tan, Chunhua Shen, and Ian Reid. Training quantized neural networks with a full-precision auxiliary module. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1488–1497, 2020.