An Empirical study of Data-Free Quantization's Tuning Robustness

Hong Chen [*1], Yuxuan Wen[*1], Yifu Ding[*1], Zhen Yang[2], Yufei Guo[3], Haotong Qin[1†]
[1]Beihang University [2]Shanghai Aerospace Electronic Technology Institute
[3]The Second Academy of China Aerospace Science and Industry Corporation

Abstract

Deep convolutional neural networks are now performing increasingly superior in various fields, while the network parameters are getting massive as the advanced neural networks tend to be deeper. Among various model compression methods, quantization is one of the most potent approaches to compress neural networks by compacting model weights and activations to lower bit-width. The data-free quantization method is also proposed, which is specialized for some privacy and security scenarios and enables quantization without access to real data. In this work, we find that the tuning robustness of existing data-free quantization is flawed, progressing an empirical study and determining some hyperparameter settings that can converge the model stably in the data-free quantization process. Our study aims to evaluate the overall tuning robustness of the current data-free quantization system, which is existing methods are significantly affected by parameter fluctuations in tuning. We also expect data-free quantification methods with tuning robustness to appear in the future.

1. Introduction

Recently, deep convolutional neural networks are performing increasingly superior in various of applications, such as image classification [7, 13–17, 19, 21], object detection [4, 5, 8, 11, 12, 18], semantic segmentation [3, 23], *etc*. However, with the continuously improved performance, the network parameters is getting massive as the advanced neural networks tend to be deeper. Therefore, deploying state-of-the-art models on resource-constrained devices becomes more challenging, which has to strike a balance between computational complexity and inference latency. Among various model compression methods, such as distillation, pruning, parameter sharing, lightweight architecture design and so on, quantization is one of the most potential approaches to compress neural networks by compacting model weights and activations to lower bit-widths, which enables the model to accelerate during inference by integer operations and reduce model size.

Since the full-precision parameters of the model are projected to low-bit representations, it is almost inevitable for quantized models to lose information. Therefore, models may suffer an accuracy degeneration after quantization, especially when quantized to ultra-low bit-width. Thus, the general methods to obtain accurate quantized networks are calibrating or fine-tuning the quantized model to mitigate the accuracy loss, namely Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT), respectively. In contrast to QAT which requires a large amount of data for training, PTQ only takes a small amount of unlabeled data and directly calibrate a well pretrained model to be quantized. It is more efficient for time and energy, which is now widespread in real applications in industry. It is notable that in some certain scenarios, especially in high privacy situations such as medical, personal information, *etc.*, it is always difficult to access the original data for security concerns. Therefore, data-free quantization method is proposed which enables quantization without access to real data. The mainstream of data-free quantification methods is based on generative approaches, *i.e.*, generating a batch of synthetic data for further fine-tuning or calibration.

However, we notice that it is worth a more profound exploration of the robustness of data-free quantization since the performance strongly fluctuates when the calibration strategy or hyperparameters are modified. For example, for some data-free PTQ quantization methods (such as ZeroQ [1]), if we change the hyperparameters such as the learning rate in a particular range, the results will change severely and even occur partial overfitting. Thus, to explore the tuning robustness of existing data-free quantization methods, we investigate the quantized models produced under different hyperparameter settings and calibration strategies of data-free quantization to present their tuning robustness.

We first empirically select the sensitive hyperparameters for two types of mainstream data-free quantization pipelines. Our comprehensive experiments make it possible

[*]Equal contribution.
[†]Corresponding author

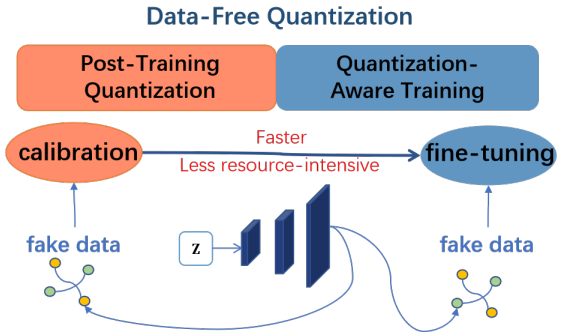

Figure 1. As can be seen in the figure, data-free quantization consists of two basic frameworks, PTQ and QAT. Generators are the most important part of them, as we use fake data in many of our methods.

to present optimal hyperparameter settings that can help the model converge stably in the data-free quantization process. Finally, we conclude and categorize the hyperparameters broadly into two levels. We recommend paying more attention to those hyperparameters that would impact the performance while designing and applying data-free quantization approaches. While for those that have neglectable impact, we provide a set of generally functional hyperparameters. Our study aims to evaluate the overall tuning robustness of the current data-free quantization system, which will significantly fluctuate with different hyperparameter settings. While giving recommended settings and analysis, we also expect data-free quantification methods with tuning robustness to appear in the future.

2. Related Work

2.1. Data-Free Quantization

The existing data-free quantization mainly includes two major approaches, PTQ and QAT. The main difference between them is whether the quantized model is further trained or not. The PTQ method only calibrates the quantizer, which is often faster and requires less computational resources, while the QAT method fine-tunes the quantized model to achieve higher accuracy. The frameworks are shown in Fig. 1. Previous works have been devoted to effective data-free quantization methods. DFQ [10], which improves the calibration process through cross-layer equalization, offset absorption, and offset correction; ZeroQ, a zero-shot scheme that allows mixed-precision quantization by generating and optimizing a set of synthetic data. DSG [22] relaxes the constraints on the BN layer and uses layer-level sample enhancement with different loss functions to jointly diversify the synthetic samples. GDFQ [20] proposes a knowledge-matching generator that can mine the distribution information and boundary knowledge of the data by Gaussian input sampling noise, and it also applies knowledge distillation to calibrate the quantized model better. ZAQ [9] is a data-free quantization method based on adversarial learning, and quantization is accomplished by "adversarial" between the generator and the quantization model. Qimera [2], which mainly adds the previously ignored boundary information to the generated data, generates synthetic boundary supporting [6] samples by using superimposed latent embeddings, while proposing to understand the mapping layer and extracting information from the full precision model.

2.2. Data-Free Quantization's Tuning

While calibrating a quantized model using data-free quantization methods, the performance is highly affected by calibration hyperparameters and strategies. To analyse the tuning robustness of data-free quantization methods, we study the following hyperparameters:

Learning rate. For data-free quantization, on the one hand, we need to train the generator to get suitable fake data, and on the other hand, when it comes to a QAT method, the quantized model needs to be retrained, and that's why we need to choose the appropriate learning rate to get the best outputs of our quantized models.

Epochs. For QAT, we set a larger number of training epochs to find better results when fine-tuning. For PTQ, we generally generate a batch of data and then calibrate the quantized model just one time. However, when we repeat the training during the data generation phase it is impossible for us to know exactly which result is better, more epochs perhaps yield worse results by overfitting. Moreover, for model quantization, we have the full-precision model as input, and there is no process of initializing the classification model. As for generative Data-Free Quantization method, we also need to train the generator, and the generator needs to be initialized. If the initialized generator is directly used to fine-tune the quantization model at the beginning, it may be counterproductive.The common solution is to warm up the generator before it can produce helpful samples. Therefore, the warm-up steps refers to the number of rounds set by training the generator separately at the beginning of quantization. Here, warm-up can be regarded as an initialization process for the generator.

Weight decay. We often experience overfitting of the training dataset, and weight decay can prevent such overfitting. This hyperparameter is involved in the fine-tuning phase of the quantized model in the QAT method.

Batch size. If we don't use batch training in Data-Free Quantization, we need to input all the data to the network at once during training, and then calculate the loss and update the weights. When increasing the batch size, the gradient of the Data-Free Quantization models would be more stable, and this is because the variance of the samples will be reduced and the gradient will be more accurate as a result. For instance, in ZeroQ scheme, We believe that the critical batch size is the batch size of the fake data. On one hand it determines the amount of fake data, on the other hand it

may affect the distribution characteristics of the fake data.

3. The Study of Turing Methodologies in Existing Data-Free Quantization

To study the tuning robustness of Data-Free Quantization, we should first reproduce its low robustness performance and then explore it in depth.

3.1. Robustness Observation for the Tuning Settings

We noticed that the data-free quantization method is sometimes sensitive to specific hyperparameters. It means that some hyperparameters are poorly robust. It is possible that a small change can drastically affect the accuracy of the quantized model. For this reason, we want to use a systematic approach to find the hyperparameters of low robustness. The two mainstream methods correspond to different types of hyperparameters. We can select these essential hyperparameters separately to study their commonalities and differences. For example, the generated data's batch size is involved in both PTQ and QAT methods. For these all hyperparameters, we make small perturbations to explore their impact on the final results to find the critical sensitive hyperparameters. Also, we can set different quantization bits and backbone networks to find some acute settings. According to the theoretical analysis, we believe that lower bits are more sensitive relative to higher bits. The backbone network with a more significant number of parameters is more sensitive relative to smaller backbone networks.

3.2. Tuning of Post-Training Quantization

The ZeroQ scheme is one of the most typical PTQ methods, with high accuracy and fast quantization speed. We selected the ZeroQ schemes as representatives of the PTQ methods. For the PTQ method, the critical phase is the one that generates the fake data, and the hyperparameters are mostly concentrated in this part. For example, the batch size(batch_size) and the number of batches(num_batch) represent the size of each batch of generated data and the number of samples, respectively. They are two crucial hyperparameters that represent the generated data. The generated data are obtained from training, and hyperparameters such as the learning rate(opt_lr) during training are also particularly important. There are also hyperparameters for the ZeroQ scheme, including the minimum learning rate(sch_min_lr) and scheduler patience(sch_patience), which represent the lower bound on the learning rate of all param groups and the number of epochs with no improvement after which learning rate will be reduced. On the other hand, the epochs at distillation(distillation epochs) represent the extent to which the fake data are trained and may also impact the results. The wrong settings of all the above hyperparameters may lead to model failure, one of the key factors determining its robustness.

3.3. Tuning of Quantization-Aware Training

For the QAT method, the most representative one is the GDFQ scheme. It has higher accuracy and can perform the quantization task well, even in the low-bit case. In addition to generating fake data, there is fine-tuning of the quantized model in the QAT method. Therefore this part also involves hyperparameters like the learning rate, but for a different purpose. So there are more types of hyperparameters in the QAT method. The QAT method generally uses the generated data to continuously fine-tune the quantized network and optimizes the generator simultaneously, unlike the PTQ method, which generates fake data in a single pass. So the batch size(batch_size) may not have the same impact on quantization as PTQ. Compared to the distillation epochs and learning rate mentioned above, there are also corresponding warm-up epochs(warmup_epochs) and learning rate of generator(lr_G) here. In addition, there is also the learning rate of the optimizer(lr_S), which refers to the learning rate when fine-tuning the quantized model. What's more, we noted that the weight decay(weight_Decay) and decay rate of generator(decay_Rate_G) regarding the learning rate scheduler and optimizer might also impact quantization. Those as mentioned above are the key hyperparameters for the QAT method and may affect the robustness of the scheme.

After the above evaluation, we can observe some key sensitive hyperparameters. Next, we should investigate how the hyperparameters affect the final quantization accuracy for certain sensitive or generally significant cases, which can give us more comprehensive information about its robustness when tuning. We can extensively adjust the values of crucial hyperparameters and explore trends in the variation of results. In addition, we can find some tuning strategies for data-free quantization.

4. Experiments and Analysis

In past studies, we would generally only refer to the hyperparameter settings given. However, we can perturb these hyperparameters to evaluate the hyperparameters' sensitivity and the model's robustness.

4.1. Experiments Settings

In this section, we will conduct an experimental analysis to give researchers a clearer understanding of the hyperparameters applied to the data-free quantization and a trace when tuning the model. We choose two representative methods: one is ZeroQ for PTQ, using ResNet18 and ResNet50 as backbone networks, for experiments on the ImageNet dataset; the other is GDFQ for QAT, for which we use ResNet20 as backbone networks, tuned on the CIFAR-100 dataset.

For ZeroQ, we chose six tuning directions, namely

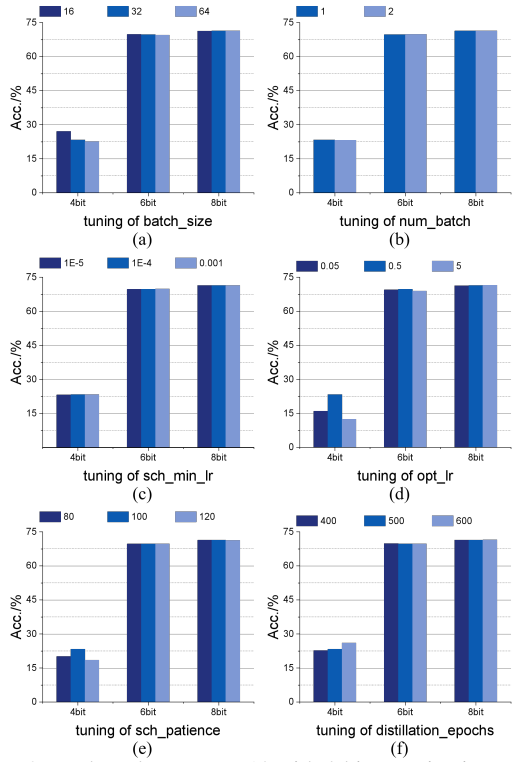

Figure 2. We based on ResNet18 with 4-bit quantization and tuned corresponding ranges of batch_size, num_batch, opt_lr, sch_min_lr, sch_patience, and distillation_epoch on ZeroQ, and repeated the experiments on 6-bit and 8-bit to obtain the model results.

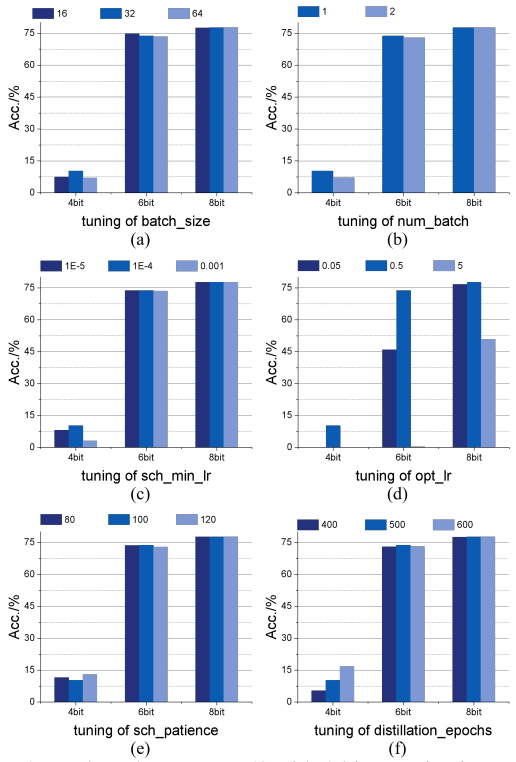

Figure 3. We based on ResNet50 with 4-bit quantization and tuned corresponding ranges of batch_size, num_batch, opt_lr, sch_min_lr, sch_patience, and distillation_epoch on ZeroQ, and repeated the experiments on 6-bit and 8-bit to obtain the model results.

batch_size, num_batch, opt_lr, sch_min_lr, sch_patience, and distillation_epochs, for which we applied certain perturbations respectively, and the specific hyperparameters are shown in Tab. 1. The corresponding results will be shown in Fig. 2 and Fig. 4.

For GDFQ, we also choose six hyperparameters, namely batch_size, warmup_epochs, lr_S, lr_G, decay_Rate_G, and weight_Decay, to which we applied specific perturbations shown in Tab. 1 respectively. The corresponding results are shown in Tab. 2.

From the results, we can see that in ZeroQ scheme, opt_lr and distillation_epochs have more effects on the model; in GDFQ scheme, the perturbations of lr_S, lr_G, and warmup_epochs have more impact on the model. In addition, we found that the tuning robustness of the lower bit quantization was poor. For the ZeroQ scheme, 4-bit quantization is challenging to achieve more satisfactory results, and tuning robustness is naturally insufficient. At the same time, the fluctuations in results in the QAT method are also more pronounced for the 4-bit case. The following section will conduct further experiments on these hyperparameters that significantly impact data-free quantization. The following section will be: first, we will do further experiments to show the effect of the most significant learning rate on the model. Secondly, we will show the impact of epochs. Finally, we will conclude with the experimental results and effects of other hyperparameters more commonly used in data-free quantization.

4.2. Impact of Learning Rate

4.2.1 Impact on Quantization-Aware Training

In the QAT method, we fine-tune the quantized model simultaneously to train the generator to serve the fine-tuning process better. We noticed that the training of the two parts is relatively independent. With independent parameters update procedures, different learning rates can be set. The impact of lr_S is similar to training a convolutional neural network for classification. It controls the speed of parameters update and also affects the result. What is specific is the setting of lr_G, which may affect whether the generator can successfully become ”warm” in the warm-up session and whether the update of the quantized model and the update of the generator can be synchronized in the further joint training process. In the first phase of the experiments, we also saw that the effect of lr_G is more significant.

Based on the hyperparameter settings of the GDFQ source code, we choose CIFAR-100 as the original dataset, ResNet20 and ResNet56 as the backbone network, and quantize the weights and activations to 4 bits. The original lr_S is set to 0.0001, lr_G is set to 0.001, and the value of lr_G is adjusted to the original 0.01, 0.05, 0.1, 0.2, 0.5, 2,

PTQ	**batch_size**	**num_batch**	**opt_lr**	**sch_min_lr**	**sch_patience**	**distillation_epochs**
tuning	[160, 240]	[1, 2]	[0.05, 5]	[1e-5, 1e-4]	[80, 120]	[400, 600]
QAT	**batch_size**	**warmup_epochs**	**lr_S**	**lr_G**	**decay_Rate_G**	**weight_Decay**
tuning	[160, 240]	[2, 8]	[1e-5, 1e-3]	[1e-4, 1e-2]	[0.08, 0.12]	[1e-5, 1e-3]

Table 1. We specify a range for batch_size, warmup_epochs, lr_S, lr_G, decay_Rate_G, weight_Decay, batch_size, num_batch, opt_lr, sch_min_lr, sch_patience, and distillation_epochs, and then adjust them. Suppose the accuracy is affected even in such a small range. In that case, it is necessary to focus on the hyperparameters that significantly impact the model and investigate them further.

#Bits	**Baseline**	**batch_size**		**warmup_epochs**		**lr_S**		**lr_G**		**decay_Rate_G**		**weight_Decay**	
		160	240	2	8	1e-5	1e-3	1e-4	0.01	0.08	0.12	1e-5	1e-3
4	63.59	63.16	63.52	63.26	63.36	**61.62**	62.29	**54.73**	**61.6**	63.7	63.58	62.87	62.67
6	69.07	69.12	69.04	69.14	69.06	69.36	69.04	69.01	69.2	69.15	68.91	69.02	69.08
8	70.43	70.37	70.32	70.4	70.46	70.29	70.43	**70.18**	70.31	70.32	70.36	70.26	70.29

Table 2. We separately adjusted the hyperparameters of the data-free quantization according to the hyperparameter perturbation range shown in Tab. 1. We obtained the corresponding accuracy of GDFQ, and the results are shown in this table. We noted that the model is more susceptible in the case of lower bit-width quantization and more stable in the case of higher bit-width. We have blacked out the data in the table that has changed significantly.

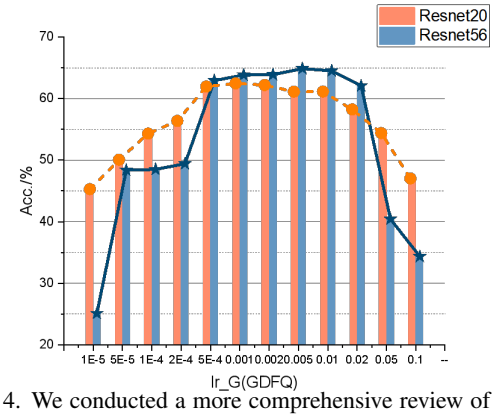

Figure 4. We conducted a more comprehensive review of lr_G to explore further how much its change would affect the model. We adjusted lr_G from 0.01 times of baseline to 100 times of baseline based on ResNet20 and ResNet56, respectively, using 4-bit quantization, and the results are displayed in Fig. As can be seen, although the accuracy of ResNet56 is greater at the highest point, its fluctuation is correspondingly greater, indicating that the robustness of GDFQ for lr_G adjustment is not outstanding under ResNet56. In contrast, its robustness for lr_G adjustment is better under ResNet20.

5, 10, 20, 50, 100 times, and compare the training results, that is shown in Fig. 4.

These experiments show that too small or too large lr_G is not conducive to quantization. When lr_G is set too large, the number of times the model is optimized will be significantly reduced during the training process. It will be challenging to improve the accuracy at an earlier stage further. For example, when lr_G is 50 times and 100 times the original value, there will be no accuracy improvement after the ninth and second rounds, respectively. Of course, we do not rule out that the model will reach a new extreme point at a specific time when training continues, but this is not what we hope to get. When lr_G is set too small, we notice that the average time required for every two accuracy improvements increases, which means that the training is slowed down. The final performance is that the quantization accuracy does not meet expectations. Comparing the results of the two backbone networks, we also noticed that for a network with more parameters, the optimal setting value of lr_G is also higher. For ResNet20, the best lr_G is about 0.001, while ResNet56 is about is 0.005.

For lr_S, we noticed in the previous experiments that it is also a sensitive hyperparameter. For further study, we set the quantized bits to 4, the backbone networks to ResNet20, and the original dataset is CIFAR-100. Then, we change the value of lr_s to 0.01, 0.05, 0.1, 0.2, 0.5, 2, 5, 10, 20, 50, 100 times the original. As shown in Fig. 5, although the optimizer's learning rate and the generator's learning rate may differ mechanistically on the final quantization results, there is also an optimal set point for lr_S. Besides, we find that the impact of too large lr_S on the results is more significant compared to too small lr_S.

4.2.2 Impact on Post-Training Quantization

In the PTQ method, we set the quantized bits to 6, the backbone networks to ResNet18 and ResNet50, and the original dataset is IamgeNet. Based on the benchmark-setting, the learning rate is extensively adjusted to 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 4.0, and the results are displayed in Fig. 5. We can see that under the backbone of ResNet50, the learning rate adjustment has a significant impact on the model. Starting from a relatively low learning rate and gradually increasing the learning rate upwards, we can get an increasing and then decreasing accuracy curve like PTQ. A lower learning rate adjusts too slowly at a certain epoch,

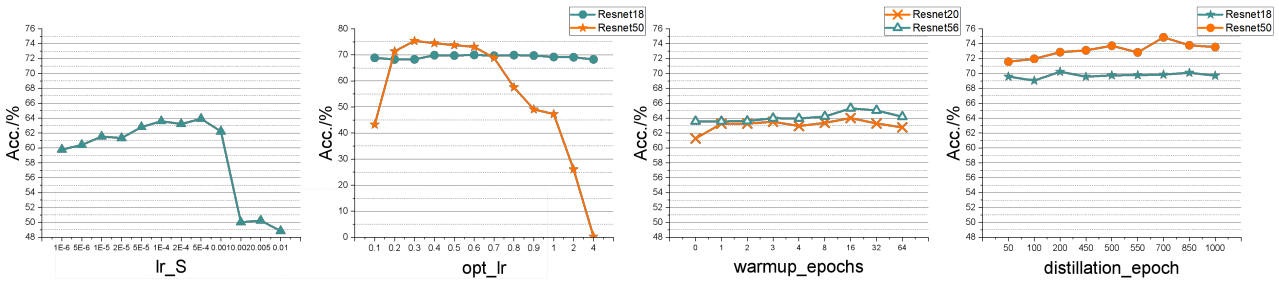

Figure 5. These four figures show our in-depth study of some of the sensitive hyperparameters. In (a), we adjusted lr_S to 0.01x to 100x the original, and GDFQ responded to this series of adjustments. In (b), we changed opt_lr from 0.1 to 4 and conducted ZeroQ's experiments on ResNet18 and ResNet50, respectively. In (c), we increased the warm-up_epochs from 0 to 64 and observed the different responses of GDFQ to this perturbation under ResNet20 and ResNet56, respectively. In (d), we adjusted the distillation_epochs from 50 to 1000 based on ZeroQ under ResNet18 and ResNet50, respectively.

making the final learning very limited. However, a larger learning rate is also undesirable because if the learning rate is set very large at the beginning, it will be difficult to offset the effect when the learning is in the incorrect direction.

However, for ResNet18, we can see that the robustness of the data-free quantized model is better. The change of the learning rate does not have any substantial impact on its accuracy because the network of ResNet18 itself is smaller and has fewer layers. Hence, the number of layers for quantization is also smaller. Therefore, the loss associated with quantization is minor, so when we change the learning rate, the results on ResNet18 show that the accuracy does not decrease. At the same time, since the network is more uncomplicated than ResNet50, the general accuracy is lower, but the stability in most cases makes PTQ (ZeroQ) almost perfectly robust on ResNet18.

4.3. Impact of Quantization Epochs

For the PTQ method, the impact of epochs comes mainly from the distillation stage. While for the QAT method does not have a separate distillation phase but a separate warm-up phase. Therefore, for PTQ and QAT, "the impact of epochs" refers to distillation epochs and warm-up epochs, respectively.

4.3.1 Impact of Warm-up Epochs

We change the setting of warm-up steps(warmup_epochs) to study the necessity of warm-up, and set the warm-up steps to 0, 1, 2, 3, 4, 8, 16, 32, 64. And we compare the training results, shown in Fig. 5.

Compared with other hyperparameters, the number of warm-up steps has less impact on the final quantization accuracy, but it is not entirely without influence. First, it is easy to notice that when no warm-up phase is set, *i.e.*, when the warm-up step is 0, the results are the worst, with a difference of about 2-3 percentage points, which means that a warm-up is necessary.

4.3.2 Impact of Distillation Epochs

For PTQ (ZeroQ), we use the backbone of ResNet18 and ResNet50, respectively, and adjust the distillation_epoch from 50, to 100, 200, 450, 500, 550, 700, 850, 1000, and the results are shown in Fig. 5 . It can be seen that under ResNet50, the fluctuation of the data-free quantization model is larger compared to that under ResNet18. Meanwhile, since the ResNet50 network is larger and more complex, it requires larger training batches to reach the maximum point, which is why the maximum point of ResNet18 appears earlier than that of ResNet50. Meanwhile, same as the warmup_epochs in the above section, since the ResNet18 network is smaller and simpler, it is easier to complete the quantization, which is influenced by the relevant hyperparameters.

5. Conclusion

This work studies the tuning robustness of existing data-free quantization methods by empirically evaluating and analyzing various hyperparameters settings. We find that among settings of tuning, the learning rate, especially those associated with the generation phase, are essential hyperparameters that have a more significant effect on the performance of quantized models. Besides, some hyperparameters can be adjusted within a specific range without substantially impacting the overall performance. We also noted that the robustness of the QAT method is overall better than PTQ. Moreover, for larger models or lower bit-with, the ideal quantization are more challenging to achieve. In this case, its tuning robustness also seems to be lower. The ideal setting for epochs-like hyperparameters tends to get slightly larger as the size increment of the quantized model. Our study aims to evaluate the overall tuning robustness of the current data-free quantization system, which significantly fluctuates with different hyperparameter settings. While giving recommended settings and analysis, we expect data-free quantification methods with tuning robustness to appear in the future.

References

[1] 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

[2] Kanghyun Choi, Deokki Hong, Noseong Park, Youngsok Kim, and Jinho Lee. Qimera: Data-free quantization with synthetic boundary supporting samples. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[3] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge. *IJCV*, 2010. 1

[4] Ross Girshick. Fast r-cnn. In *IEEE ICCV*, 2015. 1

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE CVPR*, 2014. 1

[6] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3771–3778, 2019. 2

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[8] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *IEEE CVPR*, 2019. 1

[9] 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. 2

[10] 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. 2

[11] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *IEEE CVPR*, 2019. 1

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 1

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE CVPR*, 2015. 1

[15] Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world, 2021. 1

[16] Yiru Wang, Weihao Gan, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *IEEE ICCV*, 2019. 1

[17] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *ICCV*, October 2019. 1

[18] Yanlu Wei, Renshuai Tao, Zhangjie Wu, Yuqing Ma, Libo Zhang, and Xianglong Liu. Occluded prohibited items detection: An x-ray security inspection benchmark and de-occlusion attention module. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 138–146, 2020. 1

[19] Yudong Wu, Yichao Wu, Ruihao Gong, Yuanhao Lv, Ken Chen, Ding Liang, Xiaolin Hu, Xianglong Liu, and Junjie Yan. Rotation consistent margin loss for efficient low-bit face recognition. In *CVPR*, June 2020. 1

[20] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 2

[21] Jie Yang, Jiarou Fan, Yiru Wang, Yige Wang, Weihao Gan, Lin Liu, and Wei Wu. Hierarchical feature embedding for attribute recognition. In *CVPR*, June 2020. 1

[22] 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. 2

[23] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Structured binary neural networks for accurate image classification and semantic segmentation. In *IEEE CVPR*, 2019. 1