DFQF: Data Free Quantization-aware Fine-tuning

Bowen Li LIBW@ZJU.EDU.CN

Kai Huang HUANGK@VLSI.ZJU.EDU.CN

Siang Chen CHENSIANG@ZJU.EDU.CN

Dongliang Xiong XIONGDL@ZJU.EDU.CN

Haitian Jiang JIANGHAITIAN@ZJU.EDU.CN

Institute of VLSI Design, Zhejiang University, Hangzhou, China

Luc Claesen LUC.CLAESEN@UHASSELT.BE

Engineering Technology - Electronics-ICT Dept, Hasselt University, 3590 Diepenbeek, Belgium

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

Data free deep neural network quantization is a practical challenge, since the original training data is often unavailable due to some privacy, proprietary or transmission issues. The existing methods implicitly equate data-free with training-free and quantize model manually through analyzing the weights' distribution. It leads to a significant accuracy drop in lower than 6-bit quantization. In this work, we propose the data free quantization-aware fine-tuning (DFQF), wherein no real training data is required, and the quantized network is fine-tuned with generated images. Specifically, we start with training a generator from the pre-trained full-precision network with inception score loss, batch-normalization statistics loss and adversarial loss to synthesize a fake image set. Then we fine-tune the quantized student network with the full-precision teacher network and the generated images by utilizing knowledge distillation (KD). The proposed DFQF outperforms state-of-the-art post-train quantization methods, and achieve W4A4 quantization of ResNet20 on the CIFAR10 dataset within 1% accuracy drop.

Keywords: Data-free, Quantization, Adversarial

1. Introduction

The state-of-the-art DNNs play a decisive role in various artificial intelligence applications such as computer vision Krizhevsky et al. (2012) and natural language processing Kim (2014). The rising practical value drives the DNNs to move from server-level GPUs into mobile phones, Internet of Things and edge devices. The large memory footprint, high power consumption and long inference latency make it hard to directly apply full-size DNN models into those embedded devices. Compressing and alleviating the excessive burden of the heavy deep models are highly desirable for real-time applications or resource-limited devices. To achieve this, Denton et al. (2014) utilize low-rank approximation to speed up the bottleneck convolution operations. Han et al. (2016) propose a deep compression pipeline employing pruning, trained quantization and Huffman coding. Hinton et al. (2015) propose the knowledge distillation aiming at training an alternative small network with the dark knowledge

distilled from the original large network. Among all, neural network quantization, convert-

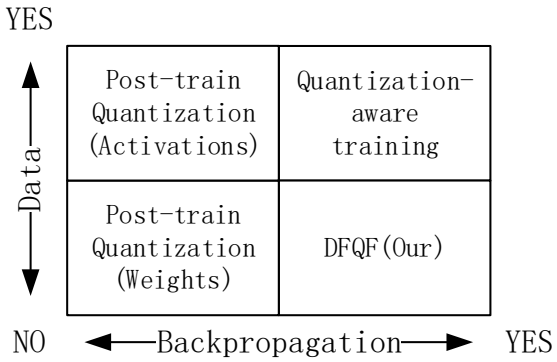


Figure 1: The state-of-the-art quantization solutions can be quartered by data-free or not and backpropagation-free or not. Our method fills in the blanks of requiring backpropagation but no data.

ing 32-bit floating-point (FP32) data to integers (INT), is an effective compression method and it does not change the network architecture. However, the quantization operation introduces noise and results in accuracy drop. Various types of algorithms are proposed to narrow the performance gap, which can be roughly divided into post-training quantization (PTQ) and quantization-aware training (QAT) dependent on requiring back-propagation or not, as illustrated in Figure 1. QAT has high compressibility and high fidelity, but requires full-size training data. PTQ is time-saving and data-free or only requires a small amount of data, but the accuracy drop is non-negligible in low-bit length representations. In many cases, we can only get the pre-trained FP32 model and allow no access to the training data. The previous studies equate data-free quantization with back-propagation free quantization and solve it with PTQ methods. Since they have no data for back-propagation and weights are quantized linearly, the performance is normally worse than the algorithms with fine-tuning or re-training on the quantization domain. Therefore, an effective knowledge extraction approach from a pre-trained model for data-free quantization is needed.

To address this, we regard the pre-trained model as an encoded format of training data and try to extract more information from it. Chen et al. (2019) propose DAFL which rebuilds the training dataset from a pre-trained model with the help of a generator network. Inspired by DAFL, our data-free quantization approach falls into two sub-steps: generating a fake training set and training the quantized model. We firstly establish a data-free knowledge distillation process including a training generator and a student. Then we fine-tune the quantized network with the generated image set. We select fine-turning instead of training from scratch because it has a higher accuracy in our data-free quantization cases. In summary, the contributions of this paper are as follows:

- We propose a data-free knowledge distillation method by optimizing inception score loss, batch-normalization statistic loss and adversarial loss, which fasten the convergence and improve the accuracy of the student network.

- We propose a quantization paradigm based on our data-free knowledge distillation architecture and we name it Data Free Quantization-aware Fine-tuning (DFQF). It requires back-propagation but no data, and fills in the blanks of the bottom right corner in Figure 1.
- DFQF can be effectively applied to low-bit quantization (3 to 8 bit), and obtains a higher accuracy than state-of-the-art on the CIFAR dataset.

2. Related Works

2.1. Neural Networks Quantization

Quantization refers to the process of reducing the number of bits. By replacing FP32 data with lower-precision numerical formats, the requirements for bandwidth and computation can be reduced. Several techniques have been introduced to improve networks Sung et al. (2015) and can be roughly divided into two categories: quantization-aware training and post-train quantization.

There are several works focusing on quantization-aware training. Rastegari et al. (2016); Hubara et al. (2016) explore Binary neural networks (BNN), quantizing both weights and activations into binary values {-1, 1}. Zhou et al. (2016) introduce the DoReFa training network with low bitwidth weights, activations and gradients. Zhou et al. (2017) proposes INQ, incrementally quantizing the weight to power-of-two. Choi et al. (2018) present PACT, clipping the activation with a learnable threshold. Jacob et al. (2018) propose a quantization scheme for integer-arithmetic-only inference, which has been applied in Tensorflow-lite.

The state-of-the-art post-train quantization methods mainly concentrate on clipping, and it can be applied to both weights and activations. The weight clip threshold can be obtained off-line. For activation, the existing methods either need a small calibration set to collect activation statistics or dynamically compute the clip threshold at runtime which increases the inference workload. Migacz (2017) selects the clip threshold with the minimum Kullback-Leibler (KL) divergence between the FP32 and INT. Banner et al. (2019) propose ACIQ, which assumes the tensors follow the Laplacian or Gaussian distribution and finds the clip threshold based on minimum mean square error (MSE). Krishnamoorthi (2018) introduces a pre-channel weights and activations quantization scheme whitepaper which exploits the ranges, and tensors can be quantized to 8-bits with almost no loss in accuracy. Nagel et al. (2019) introduce DFQ that maintains close to FP32 performance at 6-bit quantization through weight equalization and bias correction. Cai et al. (2020) propose ZeroQ, distilling input data for activation range calculation and requiring no training data.

2.2. Data-free Knowledge Distillation

Knowledge Distillation (KD) is a model compression method, aiming at transferring knowledge from a pre-trained larger model (teacher network) into a small model (student network) Bucila et al. (2006). Hinton et al. (2015) generalize previous work and introduce the concept of dark knowledge extraction. The student learns distilled knowledge by minimizing the difference between its predicted class probabilities distribution and teacher's. Several techniques employ knowledge distilling to restore the quantization error Polino et al. (2018).

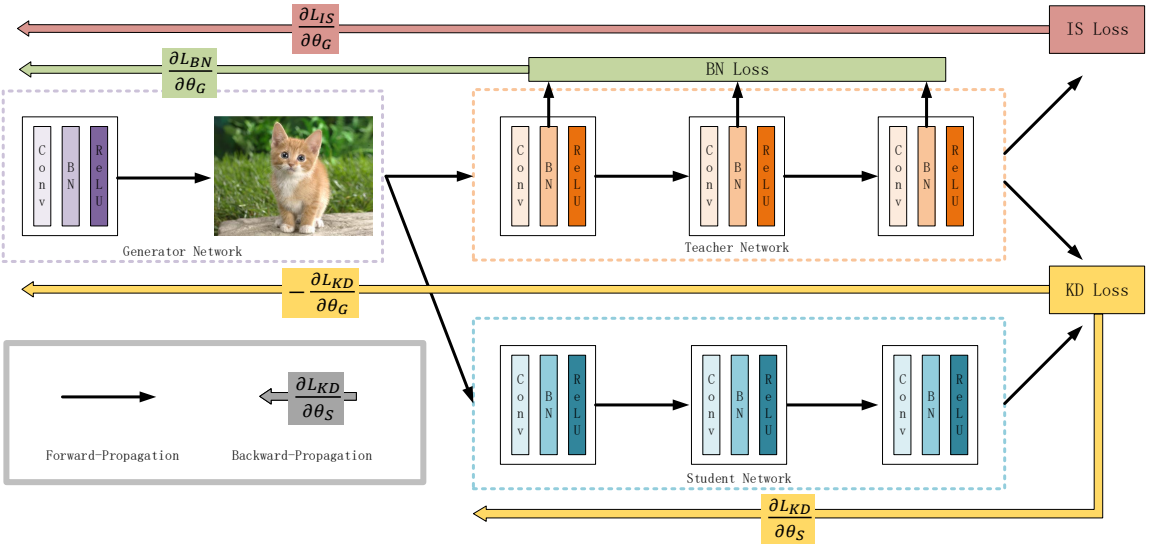


Figure 2: The proposed architecture, includes a generator, a teacher and a student. The Teacher network is fixed. The Generator abstracts knowledge from the Teacher and is trained adversarially against the Student. The Student network learns through imitating the teacher.

Mishra and Marr (2018) presented three schemes of improving low-precision networks with knowledge distillation techniques.

Recently, several works study the data-free learning. Lopes et al. (2017) compress original training data leveraging some extra metadata and train from the reconstructed training data through knowledge distillation. Chen et al. (2019) propose DAFL, a data-free knowledge distillation learning with GANs, and portable networks can be learned effectively. But they regard the pre-trained teacher networks as fixed discriminator and use a specific target (eg. one-hot output of classification network) when training the generator. Fixed discriminator means there is no adversarial relationship between the generator and discriminator. It is more like ordinary supervised learning. Yin et al. (2019); Haroush et al. (2019) also propose data-free compression methods, and they generate synthetic samples by directly optimizing the input images from trained models. Fang et al. (2019) introduce adversarial distillation to the generator for data-free learning.

3. Method

3.1. Date-free Knowledge Distillation

The training data of the supervised learning task consists of input-output pairs $\mathcal{D} = (x_i, y_i)$. In classification task, x_i and y_i are images and its labels respectively. In this section, we address the labels first, then the images.

3.1.1. Training without Labels

Hinton et al. (2015) distilled knowledge by using a temperature parameterized "softmax" that converts the logits into a probability. The loss function of the knowledge distillation is calculated as:

$$L_{KD}(x, W_s) = \lambda \mathcal{H}_{CE}(\sigma(y_s, \tau), \sigma(y_t, \tau)) + (1 - \lambda)\mathcal{H}_{CE}(\sigma(y_s, 1), y). \tag{1}$$

Where x is the input image, W_s are the student network parameters, y is the ground truth label, $\mathcal{H}_{CE}$ is the cross-entropy loss function, σ is the "softmax temperature" function, y_s and y_t are the respectively logits of the student and teacher. λ is the coefficient deciding the contribution ratio of the distillation loss. We do not have the real label, and neglect the temperature. Our student loss can be simplified to:

$$L_S(x, W_s) = \mathcal{H}_{CE}(\sigma(y_s), \sigma(y_t)). \tag{2}$$

Given a pre-trained teacher network $\mathcal{N}_T$ and a dataset without label $\mathcal{D} = (x_i)$, we can generate a new labeled dataset $\hat{\mathcal{D}} = (x_i, \mathcal{N}_T(x_i))$, and train the student network $\mathcal{N}_S$ with KD loss.

3.1.2. Image Generation

Training images are the premise of knowledge distillation. The simplest way is using random images generated from a uniform distribution. However, the output of the teacher network is uncontrollable, for example the teacher network trained from the CIFAR10 dataset considers random images as either birds or frogs. Such results make sense because the protective color of those two animals looks like a random texture. But it is almost impossible to train a student network well with such an unbalanced training set. Consequently, we need to create a fake image set whose distribution is similar to real images. The image set can also be generated from a generator neural network, which is normally trained together with a discriminator in the GAN Goodfellow et al. (2014).

We introduce a data-free training framework. The framework consists of three parts: teacher (pre-trained network), questioner (image generator network) and student (compressed network). The teacher can answer all the questions but can not ask one itself. Therefore, we need a questioner to generate not only clear but also comprehensive questions from teacher's feedback. Clear means the teacher's answers can not be ambiguous and comprehensive means the questions should cover all the knowledge points. Then the student learns knowledge from the question-answer pairs. In order to fasten the learning progress, the questioner updates its question base and stresses the questions for which teacher and student have different answers. The questioner helps the student fill in gaps by searching out its weaknesses. Akin to GANs, during the competition, both questioner and student improve their ability until the student masters all the knowledge from the teacher. The whole architecture is shown in Figure 2. We train the generator with the following loss functions.

Inception Score Loss: The inception score Salimans et al. (2016) is a widely used metric for evaluating generative models. It takes both image quality and diversity into

consideration and correlates with human judgment. The inception score is acting on the output of an inception network and is formulated as:

$$IS(G) = exp(E_{x \in p_g} D_{KL}(p(y|x)||p(y))). \tag{3}$$

Where $p(y|x)$ indicates the conditional label distribution, and $p(y) = \int_x p(y|x)p_G(x)$ is the marginal class distribution, D_{KL} is the KullbackLeibler(KL)-divergence between two distributions, the exponential operation is used only for comparison. And the inception score is related to entropies of x and y (for proof see Barratt and Sharma (2018)) :

$$ln(IS(G)) = I(y; x) = H(y) - H(y|x). \tag{4}$$

Where $I(y; x)$ is mutual information, $H(y|x)$ is conditional entropy. The smaller $H(y|x)$ indicates the generated samples are more likely to belong to a certain category. $H(y)$ is information entropy. The larger $H(y)$ indicates the generated samples are more evenly distributed on all classes. The inception score is similar to a combination of the one-hot loss and the information entropy loss in DAFL. Here, we treat the pre-trained teacher as the inception network and optimize the inception score of the generator. We use a hyper-parameter α to balance two entropies. Our inception score loss function can be formulated as:

$$L_{is}(\alpha) = -\frac{1}{N} \sum_i p(y|x^{(i)}) log \frac{p(y|x^{(i)})}{\hat{p}(y)^{\alpha}}. \tag{5}$$

Optimizing the metric normally used in the objective evaluation stage dose not seem a proper way. Barratt and Sharma (2018) also list many suboptimalities of the metric and warn that directly optimizing the inception score will lead to the generation of adversarial examples. It is a no-other-alternative move without real training data, and the goal of it is impelling the generator to generate realistic and varied images. FID (Fréchet Inception Distance) Dowson and Landau (1982) is another better evaluation metric of generator, but the calculation of it needs real data which is unavailable.

BN statistics Loss: If we expect that our generated image set is similar to the original training set, then the statistics distributions of intermediate feature maps also need to be similar. The DAFL introduces activation loss based on the law that feature maps get a higher activation value from real input images than from random vectors. This law is reasonable but not very strict. The running mean μ and variance σ^2 in the Batch Normalization (BN) layer can better reflect its distribution. Haroush et al. (2019) also utilize BN statistics for data samples generation, and they directly optimize the input images instead of generator network. We assume that the feature maps follow the Gaussian distribution. The KL divergence of two Gaussian distribution original feature maps $G(\mu, \sigma^2)$ and generated feature maps $\hat{G}(\hat{\mu}, \hat{\sigma}^2)$ can be formulated as:

$$KL(\hat{G}||G) = log \frac{\sigma}{\hat{\sigma}} + \frac{\hat{\sigma}^2 + (\hat{\mu} - \mu)^2}{2\sigma^2} - \frac{1}{2} \tag{6}$$

And we remove the extraneous and accumulation over all the batch normalization layers to get our BN statistics loss:

Algorithm 1 Data-free knowledge distillation

Input:pre-trained teacher network $\mathcal{N}_T$.
Output: student network $\mathcal{N}_S$, generator network $\mathcal{N}_G$.
repeat
  Sample a batch of m noise samples $\hat{z} = \{z_0, ..., z_m\}$;
  Update the generator network $\mathcal{N}_G$ by descending its stochastic gradient: $\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m}(L_G)$ (Equ. 9);
  Sample a batch of m noise samples $\hat{z} = \{z_0, ..., z_m\}$;
  Update the student network $\mathcal{N}_S$ by descending its stochastic gradient: $\nabla_{\theta_S} \frac{1}{m} \sum_{i=1}^{m}(L_S)$ (Equ. 2);
until convergence

$$L_{bn} = \sum_{l}(\hat{\sigma}_l^2 - \sigma_l^2 log\hat{\sigma}_l^2 + (\hat{\mu}_l - \mu_l)^2) \tag{7}$$

Adversarial Loss: The total framework has three networks: generator, teacher and student. The teacher network is fixed, the generator and the student networks are trained separately without any interactions. We believe that the difficulty of a different category to be classified by the network is different, and we do not know the order of difficulties beforehand, therefore the quantitative balance of each class in the training set is necessary. When the image set is generated dynamically during training, the equilibrium in number is not so important. The image set should pay more attention to the failings of the student and try to stump it. Pertinence education is universally considered better than naive education. In implementation, we build an adversarial relationship between the generator and the student. We impel the generator to widen the divergence between teacher's and student's output, simply through ascending the gradient of student's KD loss as:

$$L_{adv} = -L_S \tag{8}$$

To sum up, the final generator loss function with hyper parameters is formulated as follows and more details can be referred in Algorithm 1:

$$L_G = L_{is}(\alpha) + \beta L_{bn} + \gamma L_{adv}. \tag{9}$$

3.2. Data-free Quantization

We can use the above strategy to generate a fake training dataset, and simplify the data-free quantization problem to a normal quantization problem. Then we can solve it with existing quantization-aware training (QAT) methods. However, from intuitive perception and the experimental results in DAFL, the student always performs no better than the teacher even if they have the same network architecture and bit precision, let alone quantization. That is because the student is trained without a real training set and only guided by the teacher. In the opposite case, the state-of-the-art post-train quantization (PTQ) methods are able to guarantee almost no accuracy drop in the above 8-bits quantization cases. But if we treat the generated images as a calibration set and apply PTQ methods to it, the accuracy drop is

Algorithm 2 Data-free quantization-aware fine-tuning

Input: pre-trained full-precision network $\mathcal{N}_{FP}$.
Output: quantized limited-precision network $\mathcal{N}_{LP}$.
Warm up generator $\mathcal{N}_G$ through Algorithm 1;
Sample a small calibration set $\hat{x}_c$ from $\mathcal{N}_G$;
Run forward $\mathcal{N}_{FP}(\hat{x}_c)$ to collect activation statistics;
Initialize weight and activation clip threshold for $\mathcal{N}_{LP}$;
Fine-tune $\mathcal{N}_{LP}$ with $\mathcal{N}_{FP}$ and $\mathcal{N}_G$ through Algorithm 1;

non-negligible in quantization below 6-bits. Consequently, the performance of quantization by utilizing only one of PTQ and QAT can not be sufficiently well in all bit-width cases.

To address this, we need to combines the prepotent parts of PTQ and QAT. But, it is hard to combine the two methods directly. On the one hand, the PTQ methods never concern about backward propagation, and many operations' gradient are incalculable. On the other hand, the QAT does not care about the initial state. Some QAT methods change the distribution of weights such as DoReFa-Net Zhou et al. (2016) and INQ Zhou et al. (2017) and need training from scratch.

The state-of-the-art post-train quantization focuses on optimizing the clip threshold. In order to be compatible with PTQ methods, we propose a data-free quantization-aware fine-tuning (DFQF) method. Our method quantize both weight and activation uniformly with learnable step size Δ, and we initialize the learnable step size $\Delta = A/Q_{max}$ using a clipping threshold "A" calculated from PTQ:

$$X_q = clamp\left(\left[\frac{X}{\Delta}\right], Q_{min}, Q_{max}\right) * \Delta. \tag{10}$$

Where [.] is the rounding operation. We dequantize the value back to floating-point format for emulating the effect of quantization, and the clamp function is as following:

$$clamp(x, min, max) = \begin{cases} min, & \text{if } x < min \\ max, & \text{if } x > max \\ x, & \text{otherwise} \end{cases} \tag{11}$$

The distribution of weight is symmetric, $Q_{min} = -2^{n-1}, Q_{max} = 2^{n-1} - 1$. And for activation, ReLU is unilateral, $Q_{min} = 0, Q_{max} = 2^n - 1$. We choose linear quantization because it is hardware-friendly, easy to implement and suits for both QAT and PTQ. And for the same reason, we do not use per-channel or per-filter quantization. The quantization operation is non-differentiable, so we use "Straight-Through Estimator" (STE) Bengio et al. (2013) to approximate the rounding operation's gradient by 1.

Note that we need a generated training set to collect the statistic distribution of activation in initialization. Moreover, since the quantized network can be considered as pre-trained, the generator also requires to be well-trained, otherwise it would be counterproductive during the fine-tuning. As a result, we first warm up the generator, then fine-tune the quantized network. The whole process of data-free quantization is summarized in Algorithm 2. We find that the adversarial loss is still helpful, even if the student network in the

Table 1: Ablation experiments on CIFAR100 with teacher ResNet34 and student ResNet18

IS Loss	BN Loss	Adv Loss	Accuracy Drop(%)
			76.56
√			6.09
	√		60.46
		√	23.96
√	√		4.12
√		√	1.90
√	√	√	**1.66**

warm up stage and the student network in the fine-tune stage are not the same. So, we use a tool student to warm up the generator, and then replace the student with the quantized model in the fine-tune stage. Details and results are listed in Section 4.2.

4. Experiments

4.1. Image Generation Experiments

We conduct experiments on the CIFAR Krizhevsky (2009) and MNIST datasets. The CIFAR dataset consists of a training set of 50,000 and a test set of 10,000 color images with a size of 32 × 32. CIFAR-10 and CIFAR-100 contain 10 and 100 categories, respectively. The MNIST data set of 28 × 28 pixel handwritten digits, has a training set of 60,000 examples and a test set of 10,000 examples. We use DCGAN Radford et al. (2016) as our generator network and normalize the images with a Batch Normalization layer before sending them to the classification network. The generator is trained using Adam with a learning rate of 0.01. For the training of student network, we use SGD optimizer with a multi-step learning rate starting from 0.1 and decayed by 0.1 for every 800 epochs. The student and generator are trained jointly, and same as DAFL Chen et al. (2019), the whole progress are trained for 2000 epochs, where each epoch contains 100 batches of batch size 512. We set $\alpha = 40$, $\beta = 0.1$ and $\gamma = 5$ after a simple grid search.

4.1.1. Ablation Experiments

In this section, we investigate the effectiveness of each term in the loss functions. The results are summarized in Table 1. As we can see, the random noise has no effect on the student's learning. The student is loosely trainable by utilizing inception score loss, but still has a non-negligible accuracy drop (6.09%). Batch-normalization (BN) statistics loss brings a slight improvement, but cannot play a decisive role. Adversarial loss is powerful, but the network may get lost without the guidance of the inception score and spends too much effort on some minor details, which never occur in the condition of real images. The ablation study suggests that each term of the loss function has its own role: the inception score loss L_{is} sets the tone of classification task, the BN statistics loss L_{bn} limits the data

Table 2: Data-free knowledge distillation experiments.

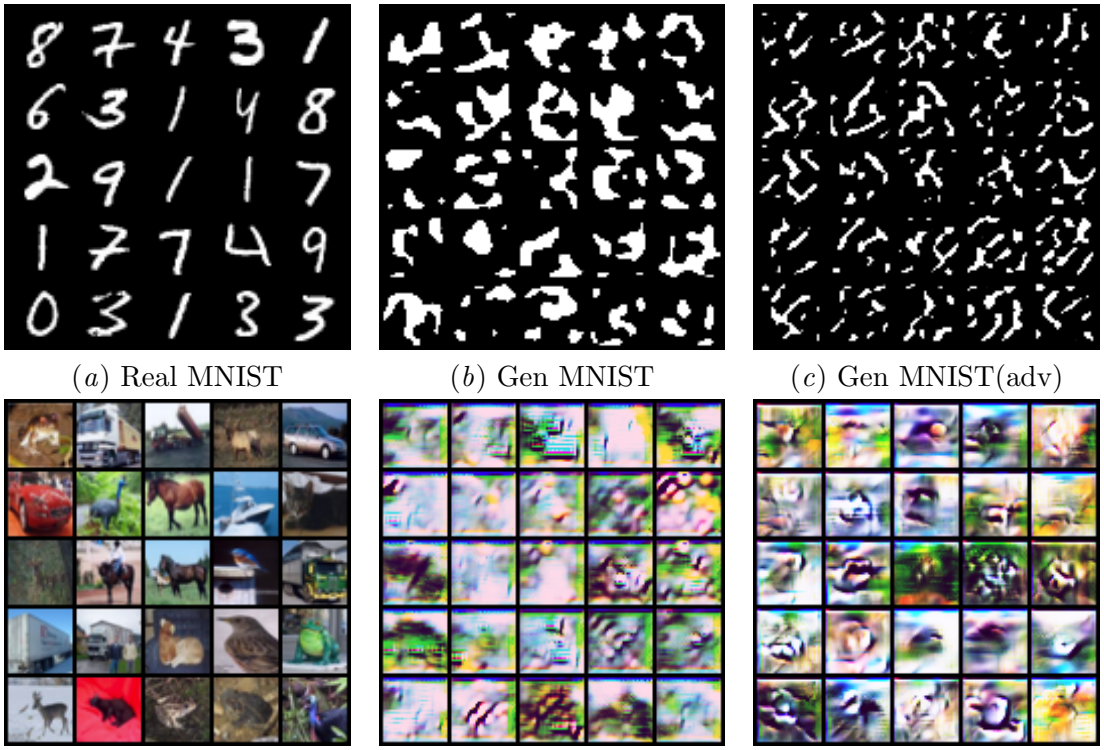
Dateset	Teacher	Student	Method	Acc(%)
MNIST	LeNet5 (98.92%)	LeNet5-H (F:33.0%) (P:25.8%)	KD-RI	98.69
			KD-RIL	98.81
			DAFL	98.20
			DFAD	98.3
			Our	**98.63**
CIFAR10	ResNet34 (95.38%)	ResNet18 (F:46.9%) (P:52.4%)	KD-RI	94.50
			KD-RIL	94.77
			DAFL	92.34
			ADI	93.26
			DFAD	93.3
			Our	**94.59**
	VGG-16 (93.64%)	VGG-13 (F:73.4%) (P:65.1%)	KD-RI	93.44
			KD-RIL	93.59
			DAFL	90.31
			Our	**92.27**
CIFAR100	ResNet34 (77.94%)	ResNet18 (F:46.9%) (P:52.4%)	KD-RI	77.23
			KD-RIL	77.63
			DAFL	74.12
			DFAD	67.7
			Our	**76.28**
	VGG-16 (74.19%)	VGG-13 (F:73.4%) (P:65.1%)	KD-RI	73.29
			KD-RIL	73.68
			DAFL	68.23
			Our	**70.93**

distribution, and the adversarial L_{adv} loss fills in gaps. They are supplementary to each other, and together train a realistic-looking generator for the student.

4.1.2. Data free knowledge distillation

We conduct data-free knowledge distillation experiments on several datasets and networks. The results are shown in Table 2. The value list below the teacher model is its Top-1 test accuracy. And among the information below the student model, 'F' stands for FLOPs and 'P' stands for parameters and the values are the relative percentage of the teacher. KD-RI and KD-RIL refer to the case if the student is trained using knowledge distillation with real images only and image-label pairs, separately. KD-RI is the theoretical upper bound of data-free knowledge distillation, the student learns everything the teacher knows. We further add the labels in KD-RIL, because the teacher is not omniscient, and accuracy of this method is mainly limited by the ability of the student model. When training the KD-RIL mode, we set the coefficient for balancing the soft target and hard target (λ in Equ. 1) the same as the teacher's test accuracy. The results of DAFL are run from open-source code released by the authors [1], and results of ADI Yin et al. (2019) and DFAD Fang et al.

1. https://github.com/huawei-noah/Data-Efficient-Model-Compression/tree/master/DAFL



(a) Real MNIST (b) Gen MNIST (c) Gen MNIST(adv)

(d) Real CIFAR (e) Gen CIFAR (f) Gen CIFAR(adv)

Figure 3: Visualization of generated images

(2019) are edited excerpts from their papers. Our results achieve about 2% better than the DAFL. This is close but there is still a gap in comparison to KD-RI and KD-RIL.

We also visualize the generated images in Figure 3 and the adversarial loss makes the images more vivid. Compared with the generated images in ADI Yin et al. (2019), it seems that ours look like chaos and unrecognizable for human. That is because the generator is trained from a classification network instead of a real discriminator. However, this does not prevent the student from learning knowledge from the teacher, which means the generated images are similar to the real images in the neural network's view. More realistic pictures do not directly lead to improved performance. The generator can extract the features of the teacher's convolution kernels and is more suitable for knowledge distillation.

4.2. Quantization Experiments

4.2.1. Ablation Experiments

We have further trained the quantized W4A4 ResNet34 with the pre-trained generator from the ablation study (Table 1), and the results are shown in Table 3. The table shows that the learned step size gives better results than the fixed step size. We can also see that the generator trained with adversarial loss performs better, even if the generator is trained with a full-precision ResNet18 student, which means the student in warm-up stage

Table 3: Ablation experiments for quantization

ResNet34(W4A4)	Real	IS	IS+BN	IS+BN+ADV
Fixed Step Size	77.83%	76.70%	76.81%	**77.16%**
Learned Step Size	77.78%	76.82%	76.85%	**77.31%**

and the student in fine-tune stage not have to be the same one. Therefore, we warm up the generator for 200 epochs with a student of the same architecture and precision as the teacher but random initialized. Then we fine-tune the quantized student for 40 epochs using SGD with a learning rate of 0.001. The warmed-up generator can be shared to train the quantized network of arbitrary precision. So, the total epochs for n different bit precisions are $200 + 40 * n$. Empirically we find that 40 epochs fine-tuning is a good trade-off between accuracy and running time, training too much may lead the quantized student model to a different local minimum. Furthermore, 40 epochs are one-fiftieth training epochs of data-free knowledge distillation. The data-free knowledge distillation has to spend a lot of time training from scratch since its teacher and student have different network architectures. The search space of fine-tuning is much smaller than training from scratch and it also results in faster convergence.

During the quantization experiments, we find that if the initial clipping threshold is too small, the learned step size could be negative after some updates, causing the training progress to be unable to proceed. The PTQ method usually has a smaller clipping threshold for the lower bit width, which makes this problem very common at less than 6-bits quantization. The existing QAT methods with clipping generally set the threshold to a large value first, then gradually decreasing it. For example, PACT Choi et al. (2018) starts the training clipping parameter with a large initial value such as 10, and then use the L2-regularizer to force the parameter to converge to a smaller value during the training progress. As a result, we set a lower bound for the initial clipping threshold as the maximum weight or activation multiplied by a ratio. We set the default lower bound ratio as 0.3, and it can be adjusted manually if still too small on different networks or precisions.

4.2.2. CIFAR Results

We evaluate our DFQF on the CIFAR dataset, and the experimental results are shown in the Table 4. In quantization experiments, we use the ResNet20 instead of the ResNet18/34 for the CIFAR dataset. The teacher accuracies on the CIFAR10 and the CIFAR100 are listed under the model name. Quantizing to extreme low precision like W2A2 may brings too much quantization noise and is beyond the capacity of fine-tuning. We use open-source code for ZeroQ (without mixed-precision) [2] and other clip methods [3]. The first and last layers are quantized to 8-bit following the convention. All the clipping thresholds are obtained off-line and fixed once the model is deployed at runtime. The activation clipping value of reference MAX, MSE, ACIQ and KL methods are calculated from a calibration set of 512 random real training images. ZeroQ clip the activation with its distilled images, which are

2. https://github.com/amirgholami/ZeroQ
3. https://github.com/cornell-zhang/dnn-quant-ocs

Table 4: Quantization results, (FG) means we fix the generator during fine-tune stage.

Network	Method	CIFAR10				CIFAR100			
		W8A8	W6A6	W4A4	W3A3	W8A8	W6A6	W4A4	W3A3
ResNet20 (92.27/ 68.87)	MAX	92.19	91.24	83.02	20.39	68.67	67.45	38.91	1.41
	MSE	**92.27**	91.96	82.41	54.35	68.69	68.26	53.31	9.1
	ACIQ	92.21	92.05	89.07	70.99	**68.91**	67.93	55.63	21.8
	KL	89.45	90.99	80.17	45.33	66.27	66.71	46.41	7.05
	ZeroQ	92.24	91.59	86.67	45.44	68.73	67.10	42.78	2.04
	REAL	92.36	92.35	92.03	90.84	69.14	69.08	67.92	65.03
	DFQF(FG)	92.23	**92.10**	**91.28**	**88.94**	**68.91**	68.60	**66.44**	**60.34**
	DFQF	92.26	**92.10**	91.07	87.82	68.79	**68.61**	66.30	60.04
VGG16 (93.64/ 74.19)	MAX	**93.74**	**93.61**	84.14	10.00	74.18	73.64	54.22	1.33
	MSE	93.67	93.43	91.40	76.49	74.16	73.88	67.64	30.74
	ACIQ	**93.74**	93.50	90.74	87.09	74.12	73.82	69.72	60.92
	KL	87.37	82.02	80.53	32.57	73.03	72.33	65.82	15.28
	ZeroQ	93.20	92.98	89.11	17.53	74.04	73.97	70.28	18.56
	REAL	93.86	93.82	93.48	93.00	74.72	74.55	73.84	72.31
	DFQF(FG)	93.57	93.49	93.00	91.58	**74.40**	**74.32**	**73.61**	**71.42**
	DFQF	93.62	93.52	**93.16**	**91.70**	74.38	**74.32**	73.55	71.18
MobileV2 (91.06/ 70.03)	MAX	90.97	90.22	69.99	14.08	69.87	68.55	25.23	2.09
	MSE	**91.03**	90.24	79.00	41.44	69.97	69.57	46.13	4.71
	ACIQ	90.82	89.70	72.44	31.98	69.92	67.62	39.38	3.53
	KL	23.64	10.00	10.00	10.00	1.66	1.01	1.00	1.00
	ZeroQ	90.83	90.38	82.48	27.09	68.50	67.74	43.06	9.22
	REAL	91.41	91.51	90.51	89.12	70.88	70.64	69.67	66.55
	DFQF(FG)	91.02	**91.00**	**89.32**	85.30	70.37	70.14	67.16	60.01
	DFQF	91.01	90.51	89.05	**85.66**	**70.42**	**70.19**	**67.43**	**61.51**

optimized to satisfy the mean and standard deviation in BN layers. Our initial value is calculated by MSE method with lower bound from our generated fake images.

Throughout the post-train quantization methods, MSE clipping maintains a relatively high accuracy in all cases and is consistent with the analysis in Zhao et al. (2019). However, the rounding operation in quantization introduces too much quantization noise in low-bit condition. We can see that the post-training quantization methods are almost invalid in lower than 4-bit quantization. In the opposite, fine-tuning can recover the quantization error and make the network usable at W3A3. Comparing to real image set, our generated set shows very slightly accuracy drop. We can achieve W4A4 quantization of VGG16 on the CIFAR10 dataset within 1% accuracy drop, which even surpasses the full-precision student in Table 2.

5. Conclusion and Discussion

In this paper, we solve the data free quantization problem and propose DFQF. This method requires no training data, and can generate fake training data from the pre-trained full-precision network. We train the generator with inception score loss, BN statistics loss and

adversarial loss. This generator shows promising results in data-free knowledge distillation application. The adversarial learning can improve the images quality universally, and not for a specific student network only. Then we apply it to quantization and fine-tune the quantized model rather than training from scratch, which not only saves training time but also improves the accuracy. Our quantization method has a learnable step size, and is compatible with both post-train quantization and quantization-aware training. The initial method ensures almost no accuracy drop above 8-bit representations, and the following fine-tuning progress recovers error for lower than 8-bit. As a result, DFQF achieves high accuracy from 8-bit to 3-bit quantization without original training set.

Acknowledgments

This work is supported by the National Key R&D Program of China (2020YFB0906000, 2020YFB0906001).

References

Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7948–7956, 2019.

Shane T. Barratt and Rishi Sharma. A note on the inception score. *CoRR*, abs/1801.01973, 2018.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 535–541, 2006.

Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. *CoRR*, abs/2001.00281, 2020.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. *CoRR*, abs/1904.01186, 2019.

Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: parameterized clipping activation for quantized neural networks. *CoRR*, abs/1805.06085, 2018.

Emily L. Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NeurIPS 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1269–1277, 2014.

DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. *CoRR*, abs/1912.11006, 2019.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Matan Haroush, Itay Hubara, Elad Hoffer, and Daniel Soudry. The knowledge within: Methods for data-free model compression. *CoRR*, abs/1912.01274, 2019.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *NeurIPS 2016, December 5-10, 2016, Barcelona, Spain*, pages 4107–4115, 2016.

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. In *CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2704–2713, 2018. doi: 10.1109/CVPR.2018.00286.

Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.

Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *CoRR*, abs/1806.08342, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeruPIS 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *CoRR*, abs/1710.07535, 2017.

Szymon Migacz. 8-bit inference with tensorrt. In *NVIDIA GPU Technology Conference*, 2017.

Asit K. Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. *CoRR*, abs/1906.04721, 2019.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 525–542, 2016. doi: 10.1007/978-3-319-46493-0_32.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016.

Wonyong Sung, Sungho Shin, and Kyuyeon Hwang. Resiliency of deep neural networks under quantization. *CoRR*, abs/1511.06488, 2015.

Hongxu Yin, Pavlo Molchanov, Zhizhong Li, Jose M. Alvarez, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. *CoRR*, abs/1912.08795, 2019.

Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Christopher De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. In *ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 7543–7552, 2019.

Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. In *ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Shuchang Zhou, Zekun Ni, Xinyu Zhou, He Wen, Yuxin Wu, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *CoRR*, abs/1606.06160, 2016.