Low-bit Quantization Needs Good Distribution

Haibao Yu[1*] Tuopu Wen[2*] Guangliang Cheng[1] Jiankai Sun[3] Qi Han[1] Jianping Shi[1]
[1]SenseTime Research [2]Tsinghua University [3]The Chinese University of Hong Kong

{yuhaibao, chengguangliang, hanqi, shijianping}@sensetime.com
wtp18@mails.tsinghua.edu.cn sj019@ie.cuhk.edu.hk

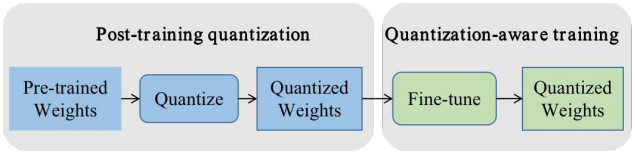
Abstract

Low-bit quantization is challenging to maintain high performance with limited model capacity (e.g., 4-bit for both weights and activations). Naturally, the distribution of both weights and activations in deep neural network are Gaussian-like. Nevertheless, due to the limited bitwidth of low-bit model, uniform-like distributed weights and activations have been proved to be more friendly to quantization while preserving accuracy. Motivated by this, we propose Scale-Clip, a Distribution Reshaping technique that can reshape weights or activations into a uniform-like distribution in a dynamic manner. Furthermore, to increase the model capability for a low-bit model, a novel Group-based Quantization algorithm is proposed to split the filters into several groups. Different groups can learn different quantization parameters, which can be elegantly merged into batch normalization layer without extra computational cost in the inference stage. Finally, we integrate Scale-Clip technique with Group-based Quantization algorithm and propose the Group-based Distribution Reshaping Quantization (GDRQ) framework to further improve the quantization performance. Experiments on various networks (e.g. VGGNet and ResNet) and vision tasks (e.g. classification, detection, and segmentation) demonstrate that our framework achieves much better performance than state-of-the-art quantization methods. Specifically, the ResNet-50 model with 2-bit weights and 4-bit activations obtained by our framework achieves less than 1% accuracy drop on ImageNet classification task, which is a new state-of-the-art to our best knowledge.

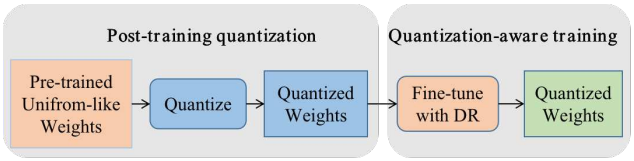
1. Introduction

In recent years, convolutional neural networks (CNNs) have achieved significant breakthroughs in a variety of computer vision tasks, such as image classification [5, 11], object detection [23, 8, 22], and semantic segmentation [31, 14], etc. These deep neural networks are usually computational-intensive and resource-consuming, which restricts them to be deployment on resource-limited devices (*e.g.*, ARM and FPGA). To improve the hardware efficiency, many researchers have proposed to quantize the weights and activations into low-bit [10, 36], especially in a linear quantization way.

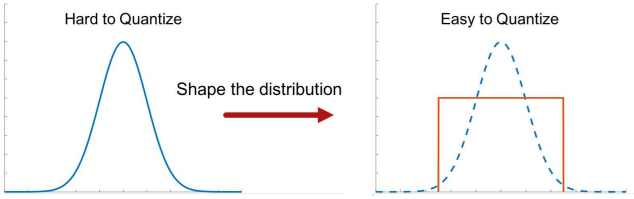
Nevertheless, quantization means that we need to represent floating-point models with fewer linear discrete val-



(a) traditional quantization pipeline.



(b) our distribution-reshaping quantization pipeline



(c) uniform distribution for linear quantization

Figure 1. Model quantization pipelines. (a) Traditional model quantization focus on quantization strategy to determine the quantization bins and fine-tuning based on the given pre-trained weights. (b) Our distribution-reshaping quantization optimizes both pre-trained weights and quantization strategy, so as to jointly reduce the quantization loss while improve the perfromance. (c) We reshape the weights into uniformly-distributed to adapt the linear quantization.

*equal contribution

ues. Thus quantization could results in performance degradation inevitably because of the indifferentiability and limited expression capacity of deep neural networks. To alleviate the performance degradation, traditioanl quantization pipelines adopt post-training quantization and quantization-aware training to recovery the performance. These traditioanl quantization pipelines focus on the strategy to determine the quantization bins to adapt the pre-trained model, such as minimizing the KL-divergence between the original weights and quantized weights when training [34, 9, 32, 6, 16, 27, 4].

Actually, we observe that pre-trained model with different distribution also play different role in quantization, while they are ignored in previous methods. Therefore, we propose to optimize both the pre-trained model and the quantization bins together. We theoretically analyze that uniformly-distributed pre-trained models result in less quantized-loss and is more friendly to linear quantization. Subsequently we propose a simple but effective technique named scale-clip technique to reshape the pre-trained models into uniformly-distributed, and optimize the quntization bins to quntize the pre-trained models. Experiments shows uniformly-distributed pre-trained model with scale-clip technique can improve the quantization performance significantly. Furtherly, to better utilize the low-bit expression capacity, we adopt group-based quantization, that is to cluster the filters into groups for quantizing.

In this paper, with the integration of the distribution reshaping method and group-based quantization, we propoe the Group-based Distribution Reshaping Quantization frameworks, that reshapes the pre-trained models into more uniformly-distributed for better quantization. Our GDRQ framework has the following advantages. (1) Models directly use uniform quantization expression, which is easy to be deployed on resource-limited devices. (2) Our proposed Distribution Reshaping method can optimize the original distribution of weights and activations more quantized uniform, which fully utilizes the capacity of low-bit representation while retains performance. (3) Group-based quantization can enhance the low-bit model's capacity while not impact the deployment. (4) Our framework is generally useful to all vision tasks with different network complexity. The main contributions of this paper can be summarized as follows:

1. **Good Distribution for Linear Quantization:** We theoretically and experimentally proved that uniformly-distributed pre-trained model can help the quantized models to achieve higher accuracy.

2. **Scale-Clip for Distribution Reshaping:** We propose a simple yet effective technique named scale-clip to reshape the pre-trained floating-point model to be uniformly-distributed, which not affect the performance of float-point model.

3. **GDRQ framework:** We incorporate the Distribution Reshaping method and Group-based quantization into our quantization framework, and validate that our framework outperforms state-of-art methods in a variety of networks and tasks.

2. Related Work

Convolution neural networks have achieved remarkable performance and have been widely used in a variety of computer vision tasks. To deploy the CNN models on resource-limited devices (*e.g.*, mobile phones or self-driving cars), many model compression algorithms [3, 24] have been proposed to reduce the model's storage as well as to accelerate inference.

Quantization Quantization can be used for reducing the number of bits required to represent weights and activations. Quantization techniques can be roughly categorized into non-uniform quantization and uniform quantization. Non-uniform quantization usually contains scalar and vector quantization. [17, 32] quantize the network as logarithmic numbers. BalanceQ [35] selects the quantization bins by Histogram Equalization while [20] takes weighted entropy as the measurement. [29] regards convolution and full-connected layers as inter product operations and thus transfer the product quantization into the network quantization. FFN [26] approximates weight matrices using the weighted sum of the outer product of several vector pairs with ternary entries vectors, which facilitates the network deployment on fixed-point computation architectures.

Most of the above methods require more bits to represent numbers during arithmetic computation, making it inconvenient to be deployed on resource-limited devices. Uniform quantization is more hardware friendly. Researches are focusing on designing an effective quantization training framework to deal with the indifferentiability of quantization. Previous works (*e.g.*,DoReFa-Net [34]) utilize straight-through estimator (STE) to estimate the quantization gradient. Ristretto [9] proposes to calculate gradients with quantized parameters while updating the gradients on the latent floating-point weights. Some works [4, 21, 15, 27] focus on extremely low-bit quantization training strategy and obtain quantization levels by minimizing reconstruction error. ELQ [33] adopts an incremental training strategy, which fixes part of weights and updates the rest to compensate for the degradation of performance. HWGQ [2] introduces clipped and long-tailed ReLU versions to remove outliers and utilizes Half Wave Gaussian Quantizer to optimize the quantization intervals of activations.

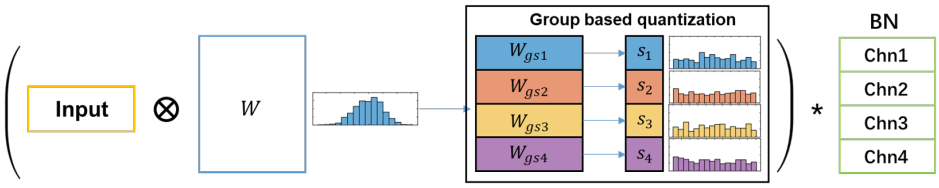


Figure 2. *

(a) Group-based Distribution Reshaping Quantization

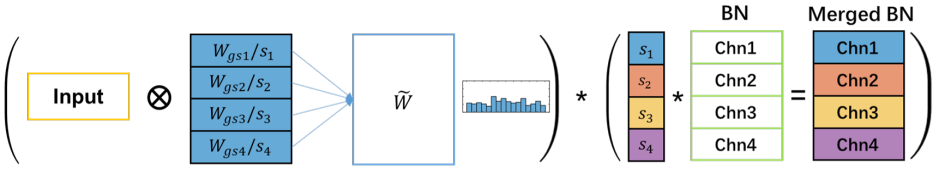


Figure 3. *

(b) Inference for Group-based Distribution Reshaping Quantization

Figure 4. Overview of our quantization framework. (a) illustrates the main flow of the distribution reshaping for group-based quantization. Weights are divided into several groups and their distributions are respectively clipped with different thresholds T_i^w and reshaped into uniformly-distributed. Then the linear quantization is performed on the reshaped distribution of each weight group. In (b), during the test phase, different clipping thresholds for each group can be merged into following batch normalized layer.

Based on clipped ReLU, PACT [12] further adaptively learns and determines the clipping parameter α during training for uniform quantization. There are other recent work [19, 13] that theoretically reveals the advantages of clipped ReLU in training quantized models. Currently, there are also some interesting works like HAQ and MPQ [28, 25] focusing on how to search the proper bit for the weights and activations with the help of reinforcement learning. Although great progress has been made in uniform quantization approaches, the non-negligible performance decrease in large scale datasets still exists.

3. Method

In this paper, we model the linear quantization task as a quantized-loss optimization problem. Our basic quantization pipeline compose of post-training quantization and quantization-aware training. Traditional quantization researches focus on the quantization strategy to determine the proper quantization bins to quantize the pre-trained model. However, they actually ignore the role that the pre-trained model play in optimizing the quantized-loss. Therefore, quantization often causes significant performance drop, even fine-tuning cannot recovery the drop. To address this issue, we propose to optimize both the pre-trained model and the quantization bins. We theoretically and experimentally prove that uniformly-distributed pre-trained weights is more friendly to linear quantization and fine-tuning. Then we propose a simple technique named scale-clip to reshape the weights into uniformly-distributed weights while not affect the pre-trained model performance. Finally, we propose to incorporate the group-based quantization into our distribution-reshaping pipelines. In the following, we first describe the linear quantization formulation (Section 3.1) and then detail our solution.

3.1. Linear Quantization

Before presenting the detailed framework, some preliminary knowledge of linear quantization are introduced. We denote the convolutional weights as $\mathbf{W} = \{\mathbf{W}_i | i = 1, \cdots, n\}$. For each weight atom $w \in \mathbf{W}_i$, linear quantization linearly discretizes it as Eq. 1.

$$Q(w;\alpha) = [\frac{\text{clamp}(w,\alpha)}{s}] \cdot s, \tag{1}$$

where $\text{clamp}(\cdot,\alpha)$ is to truncate the values into $[-\alpha,\alpha]$, $[\cdot]$ is the rounding operation and α is the clipping value. The scaling factor s is defined as Eq. 2.

$$s(\alpha) = \frac{\alpha}{2^{n_w-1}-1} \tag{2}$$

For activations, the linear quantization truncates the values into the range $[0,\alpha]$ since the activation values are non-negative after the ReLU layer. For brevity, we respectively denote the weight quantization and activation quantization as $Q(\mathbf{W};\alpha)$ and $Q(\mathbf{A};\alpha)$.

3.2. Good Distribution for Linear Quantization

Quantization means that we need to represent floating-point models with fewer linear discrete values to achieve similar accuracy. And quantization often cause the quantized weights to have significant quantized-loss, resulting in the accuracy drop. Like signal-to-noise ratio (*SNR*), here

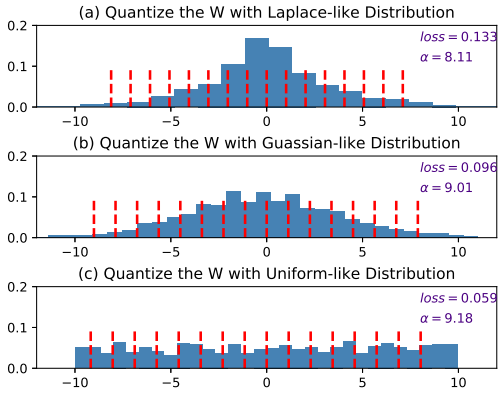


Figure 5. We respectively quantize the $\mathbf{W}$s which obey: (a) Laplace-like distribution, (b) Gaussian-like distribution and (c) Uniform-like distribution into $Q(\mathbf{W})$ as Eq. 12 with $n_w = 4$ and $||\cdot||_p = ||\cdot||_1$, where red lines means the quantization bins. We calculate *quantized-loss* as Eq. 3. Comparatively, uniform distribution better fits the uniform quantization.

we define the quantized-loss as Eq. 3

$$QL(\mathbf{W}, Q(\mathbf{W};\alpha)) = \frac{||\mathbf{W} - Q(\mathbf{W};\alpha)||_p}{||\mathbf{W}||_p}, \tag{3}$$

where $||\cdot||_p$ denotes the p-norm and here we take $||\cdot||_p$ as $||\cdot||_1$. Note that we can also rewrite Eq. 3 in KL divergence.

Given the pre-trained float-point model, traditional quantization methods focus on the strategy to determine the quantization bins to adapt the pre-trained model with the formulas

$$\alpha^* = \min_{\alpha} \frac{||\mathbf{W} - Q(\mathbf{W};\alpha)||_p}{||\mathbf{W}||_p} \tag{4}$$

However, we observe that pre-trained model with different distribution also play different roles in quantized-loss optimization. Most of the weights in convolutional layers distribute near-zero areas, (*e.g.*, Laplace distribution or Gaussian distribution). These distributions often produce large quantized-loss compared to uniform distribution. We respectively generate three data distribution examples ((a) Laplace distribution, (b) Gaussian distribution and (c) uniform distribution) composed of 1000 samples, illustrated in Fig. 5. The optimal α^* for (a), (b) and (c) are calculated as $\alpha_a^* = 8.11$, $\alpha_b^* = 9.01$ and $\alpha_c^* = 9.18$ according to Eq. 4. The corresponding quantized-loss of uniform distribution is 0.059 while quantized-loss about Laplace distribution reaches 0.133. Therefore, uniform-like distribution is more friendly to linear quantization to reduce the quantized-loss.

Furtherly, we experimentally prove that uniformly-distributed pre-trained model can help the quantized models to achieve higher accuracy in Section 4.1. Therefore, different from the traditional works [25, 7], we focus on optimize both the pre-trained model and quantization bins. We divide the quantized-loss optimization into two steps, that is, optimizing the pre-trained model firstly, then optimizing the quantization bins.

3.3. Scale-Clip for Distribution Reshaping

In this part, we introduce the scale-clip technique to train the pre-trained model, to reshape the model into uniformly-distributed while not affect its perormance.

To start with, we explore the relationship between the two statistical measures of the uniform distribution: $\max(|\mathbf{W}|)$ and $\text{mean}(|\mathbf{W}|)$. The density function of the uniform-distribution is defined as Eq. 5.

$$p(w) = \begin{cases} C, & w \in [-T, T] \\ 0, & else \end{cases} \tag{5}$$

where $C = \frac{1}{2T}$. Suppose $\mathbf{W}$ follows uniform distribution in $[-T, T]$, $\max(|\mathbf{W}|) = T$. Then $\text{mean}(|\mathbf{W}|)$ can be approximated as Eq. 6.

$$\begin{aligned} \text{mean}(|\mathbf{W}|) &\approx \int p(w)|w|dw \\ &= \int_{-T}^{T} \frac{1}{2T}|w|dw = \frac{T}{2} \end{aligned} \tag{6}$$

Thus we have T as Eq. 7.

$$T = \max(|\mathbf{W}|) \approx 2 \cdot \text{mean}(|\mathbf{W}|) \tag{7}$$

Based on this relationship, we provide a simple yet effective scale-clip technique, to reshape the distribution of a floating-point model into uniform distribution dynamically during training stage, which has the formulation as Eq. 8:

$$\text{clip}(w) = \begin{cases} T^w, & w \geq T^w \\ w, & w \in (-T^w, T^w) \\ -T^w, & w \leq -T^w \end{cases} \tag{8}$$

where

$$T^w = k \cdot \text{mean}(|\mathbf{W}|). \tag{9}$$

The clipping benefits from the proposed Distribution Reshaping method with the following intuitive analysis: when k is near 2, to compensate the lost energy from clipping outliers, more values around zero tend to become larger values. Eventually, the $\mathbf{W}$ reaches the limiting case, that is the distribution of $\mathbf{W}$ tends to be uniform. However, when $k \ll 2$, more outliers will be clipped while there are not enough shifted values to compensate for the lost energy, resulting in the $\mathbf{W}$ converging to zero. When $k \gg 2$, the distribution gradually becomes Gaussian-like and eventually the proposed method will have little impact on distribution reshaping.

Activation $\mathbf{A}$ can also adopt scale-clip technique. Nevertheless, the statistical measures of $\mathbf{A}$ are dependent on the data and unstable in the training process. Thus, we can not directly employ Eq. 9 on activation quantization. To handle this, a large k should be chosen to adapt to the changeable statistical measures mean($\mathbf{A}$). In addition, to achieve stable quantization, we introduce a new update strategy of T^a in the training process as Eq. 10 to dynamically satisfy Eq. 11.

$$\begin{aligned} T^a &= T^a + \lambda \nabla T^a \\ &= T^a + \lambda(T^a - k \cdot \text{mean}(|\mathbf{A}|)). \end{aligned} \tag{10}$$

$$T^a = \arg\min_T \frac{1}{2}||T - k \cdot \text{mean}(|\mathbf{A}|)||_2^2. \tag{11}$$

Therefore, the distribution reshaping method can reshape the distribution of activation as a uniform-like distribution while maintaining the performance.

Note that clipped method has already been widely used in training the deep neural network, such as gradient clipping [1, 18] for avoiding exploding gradients and activation clipping for training quantization model [10, 12]. In our work, we just took advantage of the clipped method as part of our optimization and compared with activation clipping [10, 12], we further analyze the reasons for the advantages of the chipped method for uniform quantization and theoretically analyze how to set a reasonable threshold.

3.4. Group-based Quantization

In this part, to increase representative capacity of the low-bit model, we further adopt the group-based quantization that splits the filters $\mathbf{W}$ into several groups, then quantizing the grouped filters by search different α to determine different quantization bins.

Actually, to achieve better performance, the intuitive solution is that different filters should adopt different α and scaling factor $s(\alpha)$ in the quantization process. For instance, we split the trained weight filters from the first convolutional layer in ResNet-18 on CIFAR-100 into 8 groups, and calculate the optimal α for each group filters, respectively. Fig. 6 illustrates that the optimal α (the blue bar) for all the filters is not always consistent with the optimal α_l (the orange bar) for each group filters. In addition, the α_ls for group filters provide strong diversity for quantization, which will enhance the network capacity without extra bit width. As Fig. 4 illustrates, the scaling factor $s(\alpha_l)$ for each group filters can be gracefully merged into BN layer. Compared to convolution layer, linear operations in BN layer cost negligible resources in resource-limited devices.

Finally, we provide the implementation details of **Group-based Quantization** as followings:

i) Decomposing convolution filters $\mathbf{W}$ into group $\mathbf{G}_l = \{\mathbf{W}_{(l-1)*gs+1}, \cdots, \mathbf{W}_{l*gs}\}, l = 1, \cdots, \frac{n}{gs}$, where gs is group size.

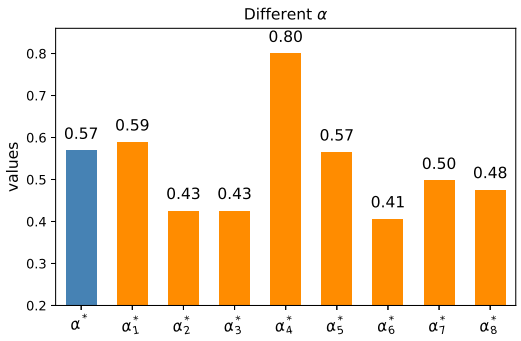


Figure 6. Optimal α for first convolutional layer's weights $\mathbf{W}$ of trained ResNet-18. Blue bar represents the optimal α^* for $\mathbf{W}$ while orange bars correspond to each group filters.

ii) Quantizing the group filters $\mathbf{G}_l$ with α_l calculated by the following form

$$\alpha_l^* = \arg\min_\alpha \frac{||\mathbf{G}_l - Q(\mathbf{G}_l; \alpha)||_1}{||\mathbf{W}||_1}. \tag{12}$$

3.5. Group-based Distribution Reshaping Quantization Framework

In this section, we integrate distribution reshaping and group-based quantization into the Group-based Distribution Reshaping Quantization (GDRQ) framework, and introduce the implementation details of GDRQ framework.

By applying the Distribution Reshaping on each group filters, we clip the $\mathbf{G}_l$ with $\alpha_l = k \cdot mean(|\mathbf{G}_l|)$ and reshape the distribution of each group's filters as uniform-like. The training and inference operation are also demonstrated in Fig. 4. Subsequently, the reshaped group filters are quantized with scaling factor $s(\alpha_l)$. In the inference stage, we merge the scaling factors $s(\alpha_l)$ into BN layers, so that the weights in different groups will share the same uniform quantization range, which is equivalence to traditional quantization setting. Therefore we can also easily deploy the quantized low-bit model into resource-limited devices under our GDRQ framework. The key operations in our quantization framework are illustrated in Alg. 1.

4. Experiment

We conduct experiments to validate our proposed Distribution reshaping method and Group-based Quantization in Section 4.1 and Section 4.2. Extensive experiments on varieties of networks and tasks to demonstrate the effectiveness of our GDRQ framework are shown in Section 4.3, Section 4.4 and Section 4.5.

4.1. Distribution Reshaping method validation

In this part, we conduct experiments in two steps: (1) validating that Distribution Reshaping method can reshape

Algorithm 1 Group-based Distribution Reshaping Quantization Framework

Require: bit width n_w and n_a
Ensure: Low-bit inference model
  Cluster the filters into groups $\mathbf{G}_l$
  while Training **do**
    for each layer **do**
      for each group filters $\mathbf{G}_l$ **do**
        Reshape the $\mathbf{G}_l$ into uniform-like with T_l^w
        Quantize the group filters into n_w-bit
      end for
      Reshape the activations into uniform-like with T^a
      Quantize activations into n_a-bit
    end for
  end while
  for each layer and group filters **do**
    Merge the α_l (that is T_l^w) into BN layer
  end for

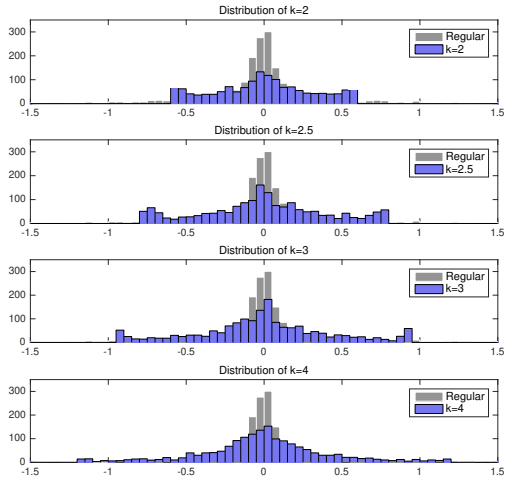
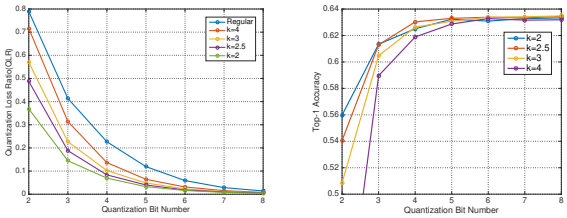


Figure 7. The blue bars show the weight distribution of the first convolutional layer with Distribution Reshaping using different Scale-Clip factors. The gray bars show the weight distribution of the first convolution layer trained without Distribution Reshaping.

the distribution of weights into a different shape, especially when k_w is near 2 the shape is uniform-like, (2) validating that uniform distribution facilitates reducing the *quantized-error* as well as promotes the low-bit model's performance.

Reshaping Effect The experiments are performed on CIFAR-100 dataset. As our focus is on the validation of Distribution Reshaping method, we set different $k_w \in \{2, 2.5, 3, 4, \infty\}$ shown in Eq. 9 and impose the Distribution Reshaping method on the convolutional weights to train five floating-point ResNet-18 where $k_w =$ means there is no reshaping.

In Fig. 7, we present the first convolutional weights' distribution of the five floating-point models. As k_w decreases, the distribution becomes flatter with little outliers, especially when $k_w = 2$, the distribution becomes almost uniform. This phenomenon corresponds to the effectiveness of our Distribution Reshaping method.



(a) Quantized-loss (b) Top-1 acurracy

Figure 8. (a) Quantized-loss of the first convolutional layer's weights. The regular curve is the QL trained without Scale-Clip. (b) Top-1 accuracy of different k_w with different bit.

Performance Comparison We quantize all convolutional layers' weights of above five floating-point ResNet18 into n_w-bit (from 2-bit to 8-bit), and compute the quantized-loss of first convolutional layers' weights as Eq. 3.

In Fig. 8(a), the quantized-loss tends to decrease when bitwidth increases, and for same bitwidth, larger k_w tends to have smaller quantized-loss, green curves ($k_w = 2$) is lower than other curves($k_w > 2$) This results shows that uniform quantization indeed reduce the quantized-loss.

Fig. 8(b) presents the Top-1 accuracy of the low-bit models after finetuning with 50 epochs. The results are consistent with the results shown in Fig. 8(b), that $k_w = 2$ promotes the low-bit model to achieve better final performance. Thus we can conclude that restricting weight to be uniform-like outperforms those with Gaussian-like or Laplace-like distribution.

4.2. Group-based Quantization Validation

In this part, we conduct experiments to validate the consistent effectiveness of our Group-based Quantization and GDRQ framework. The experiments are also performed on ResNet-18(stride is 1 in the first block) and CIFAR-100. Based on the trained floating-point ResNet-18, we use Group-based Quantization to cluster the convolutional filters into groups by group sizes $gs = [1, 4, 16, -1]$, where $gs = -1$ is the special case of layer quantization. Then we respectively quantize all convolutional layer's weights into 2-bit and 3-bit with Group-based Quantization and fine-tune these low-bit models for 50 epochs.

The overall performance of our Group-based Quantization is shown in Table 1. For 2-bit weights, the floating-point model just obtains less than 3% accuracy drop by Group-based Quantization with $gs = 1$, while other group sizes obtain much accuracy drop, even the 2-bit model fails

Table 1. Top-1 Accuracy (%) of ResNet-18 with $n_w = 2$

Fine tune	float	1	4	16	-1
without finetuning	73	69.3	50	20	1
After 50 epochs	-	71.3	69.5	68.1	64.9

Table 2. Top-1 Accuracy (%) of ResNet-18 on CIFAR-100 with GDRQ framework

group size	1	4	16	-1
float	73	73	73	73
2-bit	0.1	0.1	0.1	0.1
Binary	-1.6	-1.4	-1.4	-1.7

with quantization by layer. The result of 3-bit is consistent with 2-bit.

After recovering the accuracy drop with finetuning, 2-bit ResNet-18 by Group-based Quantization with $gs = 1$ achieves 71.3 with less than 1% accuracy drop, while 2-bit quantized ResNet-18 by layer, that is $gs = -1$ obtains more than 7% accuracy drop. The curves in Fig. 9 also shows that $gs = 1$ always achieves better performance than quantization by layer and other group size. Table 1 and Fig. 9 both show that low-bit models achieve better performance with group-size decreasing. Thus Group-based Quantization can reduce the low-bit model's quantized-loss as well as promote the final performance. We believe this is caused by Group-based Quantization increasing the low-bit model's capacity.

Further, we impose the Distribution Reshaping on each group filters and quantize ResNet-18 into low-bit model. The overall performance of the GDRQ framework is shown in Table 2.

The floating-point models trained with different group size achieve similar performance. This result shows that applying the Distribution Reshaping on group filters doesn't affect the model's performance. and even all 2-bit ResNet-18 have little accuracy drop. To compare the performance with different group sizes, we further binarize the weight. Group-based quantization also achieves better performance in binarized model. However, small group size is not always better, since when group size is equal to 1, the binarized model doesn't outperform other models. We think this is because that when imposing the Distribution Reshaping on a too-small number of weights, the distribution will be unstable. Thus, we suggest we should choose the proper group size between increasing low-bit model's capacity and keeping the stable statistical measures for Distribution Reshaping.

4.3. VGG-16 & ResNet-50 on ImageNet

We quantize two typical CNN models using our Group-based Distribution Reshaping Quantization framework: VGG-16 and ResNet-50, which represents two different CNN architectures respectively. Both models are fine-tuned on the ImageNet dataset (ILSVRC-12). Top-1 and Top-5 classification performance are reported on the 50k validation set.

Table 3. Top-1 Accuracy (%) of VGG-16-BN in different bit width. [2,4] denotes 2-bit for weights and 4-bit for activations.

Model	float	[2,2]	[2,4]	[4,4]	[2,8]
Ours	72.6	69.8	71.7	72.5	72.3

VGG-16 on ImageNet As described previously, we impose Distribution Reshaping method by group to train the floating-point model. To shape the distribution of activation layer, we also add the Distribution Reshaping in ReLU layers similar to [2]. We use SGD with mini-batch size of 512, and other parameters are kept as the original VGG paper.

As Table 3, we compare the bitwidth of n_a and n_w with $[2, 4]$ bit, $[4, 4]$bit, $[2, 8]$ bit, $[4, 8]$ bit, since these bitwidths are more practical. In Table 3, compared to floating-point model, the low-bit VGG-16-BN with 4-bit weights and 4-bit activations has little accuracy drop. This demonstrates that under our GDRQ framework the 4-bit VGG-16-BN can fully hold the performance. With lower bitwidth such as 2-bit weights, the VGG-16 gets less than 1% accuracy drop. Even 2-bit weights and 2-bit activation could almost reaches 70%.

ResNet-50 on ImageNet The proposed quantization framework is also effective to compress the ResNet-50 architecture, which achieves state-of-art classification accuracy on ImageNet. During the process of training floating-point ResNet-50, Distribution Reshaping is also implemented by groups.

The overall performance of our quantization framework on quantized ResNet-50 is shown in Table 4. There should note that the Top-1 accuracy of floating-point ResNet is less than 76%, however we implement the ResNet-50 without adding Distribution Reshaping but get similar performance. The reason may be that we adopt improper training hyper-parameters on multi-GPUs. Even so, the low-bit ResNet-50 outperforms other methods such as SYQ and FGQ. For example, $[4, 4]$-bit and $[2, 8]$-bit ResNet-50 with our quantization framework obtains 0.3% accuracy drop, while SYQ obtains more than 3% accuracy drop. Compared to floating-point ResNet-50, $[2, 4]$-bit still has less than 1% accuracy drop.

4.4. Comparison on PASCAL-VOC Detection

In this section, we conduct our GDRQ framework in detection task with Faster-RCNN on PASCAL-VOC. Note that we use ResNet-50 as backbone with pretrained model on ImageNet for Faster-RCNN. **Results:** From Table 5, we notice that the mAP of low-bit fixed point models have lit-

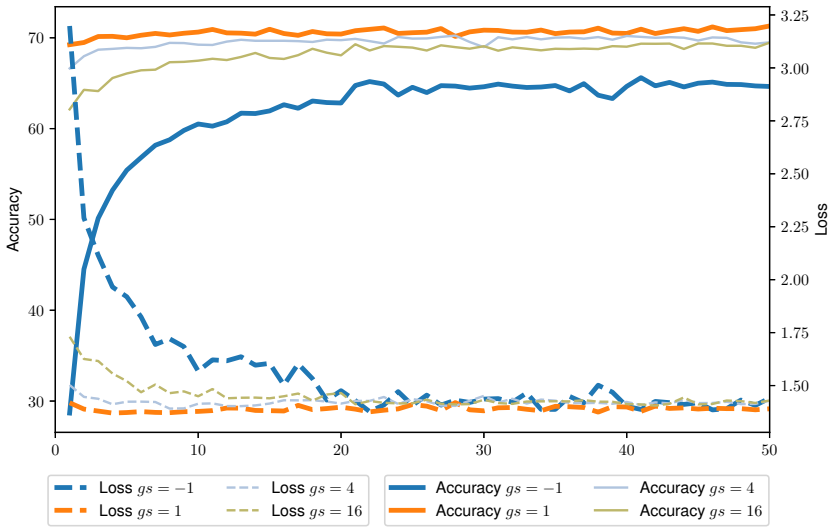


Figure 9. Loss and Accuracy with different group size gs during the fine-tuning stage. Orange curves denote the loss (dash line) and the accuracy of $gs = 1$ varies with epochs. Blue curves denote the loss (dash line) and the accuracy of $gs = -1$ changing with epochs.

Table 4. Top-1 Accuracy (%) of ResNet-50 with three different models in different bit on ImageNet.

Model	float	[2,2]	[2,4]	[4,4]	[2,8]
SYQ	76	-	70.9	-	72.3
FGQ	-	-	68.4	-	70.8
DoreFa-Net	-	-	-	71.4	-
Ours	74.8	70.6	73.9	74.5	74.5

Table 5. mAP of PASCAL-VOC.'*' denotes that activations are not quantized.

Model	float	[5,8]	[4,8]	[4,4]	[2,4]
Park et al.	77.61	77.1*	77*	72.9	66
Yin et al.	77.46	76.99*	74.4*	-	-
Ours	79.0	79.0	78.8	78.5	78.3

tle degradation compared to the floating-point models, even $[2, 4]$ bit models, even models with 2-bit weights and 4-bit activations only drop 0.7% compared to the floating-point model. We compare our quantized detection results with [20] and [30]. Note that the networks in [20] and [30] are modified version of Faster-RCNN as R-FCN. And although they adopt an non-uniform quantization scheme which takes non-uniform discrete values and has more expressive ability, our method is much better than their results since our $[2, 4]$ bit model has no decline.

4.5. Comparison on Cityscape Segmentation

In this part, we conduct our Scale-Clip method in segmentation tasks with PSPNet on Cityscapes. Note that we also use ResNet-50 as backbone for PSPNet.

Results: From Table 6, we can also observe that the mIoU of low-bit fixed point models have little degradation,

Table 6. mIoU of Cityscapes.

Model	float	[8,8]	[4,8]	[4,4]	[2,4]
Ours	75.6	75.66	75.29	75.62	74.7

even models with 2-bit weights and 4-bit activations only drops 0.9% co mpared to the floating-point model. As for segmentation, up to our knowledge, there is no open quantization result on large datasets reported, especially in low-bit quantization, so we don't compare our segmentation results.

5. Conclusion

In this paper, we develop a group-based distribution reshaping quantization framework by incorporating our Distribution Reshaping method and Group-based Quantization for uniform quantization. We elaborate experiments in CIFAR-100, ImageNet, COCO, VOC, and network in ResNet-18, ResNet-50, VGG demonstrates our method generalize well to various dataset, tasks, and backbone network. We also make a new record for ImageNet low-bit quantization state-of-the-art. Our uniform quantization can easily support FPGA deployment.

References

[1] Goodfellow I et al. Abadi M, Chu A. Deep learning with differential privacy. pages 308–318, 2016. 5

[2] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. *arXiv preprint arXiv:1702.00953*, 2017. 2, 7

[3] Jian Cheng, Pei-song Wang, Gang Li, Qing-hao Hu, and Han-qing Lu. Recent advances in efficient computation of deep convolutional neural networks. *Frontiers of Infor-*

mation Technology & Electronic Engineering, 19(1):64–77, 2018. 2

[4] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 1

[6] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in neural information processing systems*, pages 1269–1277, 2014. 2

[7] Julian Faraone, Nicholas Fraser, Michaela Blott, and Philip HW Leong. Syq: Learning symmetric quantization for efficient deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4300–4309, 2018. 4

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[9] Philipp Gysel. Ristretto: Hardware-oriented approximation of convolutional neural networks. *arXiv preprint arXiv:1605.06402*, 2016. 2

[10] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 1, 5

[11] 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. 1

[12] Choi J., Wang Z., S. Venkataramani, Chuang P. I., Srinivasan V., and Gopalakrishnan K. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 3, 5

[13] Hou L., Zhang R., and J. T. Kwok. Analysis of quantized models. 2019. 3

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[15] Naveen Mellempudi, Abhisek Kundu, Dheevatsa Mudigere, Dipankar Das, Bharat Kaul, and Pradeep Dubey. Ternary neural networks with fine-grained quantization. *arXiv preprint arXiv:1705.01462*, 2017. 2

[16] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852*, 2017. 2

[17] Daisuke Miyashita, Edward H Lee, and Boris Murmann. Convolutional neural networks using logarithmic data representation. *arXiv preprint arXiv:1603.01025*, 2016. 2

[18] Le Q V et al. Neelakantan A, Vilnis L. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015. 5

[19] Yin P., Lyu J., S. Zhang, Osher S., Qi Y., and Xin J. Understanding straight-through estimator in training activation quantized neural nets. 2019. 3

[20] Eunhyeok Park, Junwhan Ahn, and Sungjoo Yoo. Weighted-entropy-based quantization for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 8

[21] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016. 2

[22] 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. 1

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[24] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017. 2

[25] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization. *arXiv preprint arXiv:1811.08886*, 2018. 3, 4

[26] Peisong Wang and Jian Cheng. Fixed-point factorized networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 3966–3974. IEEE, 2017. 2

[27] Peisong Wang, Qinghao Hu, Yifan Zhang, Chunjie Zhang, Yang Liu, Jian Cheng, et al. Two-step quantization for low-bit neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4376–4384, 2018. 2

[28] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv preprint arXiv:1812.00090*, 2018. 3

[29] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4820–4828, 2016. 2

[30] Penghang Yin, Shuai Zhang, Yingyong Qi, and Jack Xin. Quantization and training of low bit-width convolutional neural networks for object detection. *arXiv preprint arXiv:1612.06052*, 2016. 8

[31] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017. 1

[32] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. *arXiv preprint arXiv:1702.03044*, 2017. 2

[33] Aojun Zhou, Anbang Yao, Kuan Wang, and Yurong Chen. Explicit loss-error-aware quantization for low-bit deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9426–9435, 2018. 2

[34] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. 2

[35] Shu-Chang Zhou, Yu-Zhi Wang, He Wen, Qin-Yao He, and Yu-Heng Zou. Balanced quantization: An effective and efficient approach to quantized neural networks. *Journal of Computer Science and Technology*, 32(4):667–682, 2017. 2

[36] Chenzhuo Zhu, Song Han, Huizi Mao, and William J. Dally. Trained ternary quantization. 2016. 1