Learning Non-Uniform Step Sizes for Neural Network Quantization

Shinya Gongyo[1], Jinrong Liang[2], Mitsuru Ambai[1],
Rei Kawakami[2], and Ikuro Sato[1,2]

[1] Denso IT Laboratory, Tokyo, Japan
[2] Institute of Science Tokyo, Japan
Corresponding author: gongyo.shinya@core.d-itlab.co.jp

Abstract. Quantization of neural networks enables faster inference, reduced memory usage, and lower energy consumption, all of which are crucial for deploying AI algorithms on devices. However, quantization may degrade performance compared to full-precision models as precision decreases. While prior research has primarily focused on uniformly quantizing network weights and activations, capturing the long-tail distributions of these quantities imposes a challenge. To address this issue, this paper introduces a non-uniform learned step-size quantization (nuLSQ) approach. It optimizes individual step sizes for quantizing weights and activations. Evaluations on CIFAR-10/100 and ImageNet datasets, using ResNet, MobileNetV2, Swin-T, and ConvNeXT with 2-, 3-, and 4-bit precisions, demonstrate that nuLSQ outperforms other quantization methods. The code is available at https://github.com/DensoITLab/nuLSQ.

Keywords: Quantization · Compression · Real-time inference

1 Introduction

Deep learning has improved model performance across a range of applications such as image classification [8, 14, 51], object detection [47, 48], speech synthesis [41], and language translation [52]. In these applications, large over-parameterized models can generally achieve high performance [60], but they require huge computational costs during inference. Compact models that solve the same tasks with higher inference speed, lower memory footprint and lower energy consumption are in high demand especially for low-end edge devices.

Aiming at speeding up inference with low memory consumption and little performance loss, methods of neural network compression and acceleration have been introduced, such as knowledge distillation [3,16,24,29,56,59], network pruning [15, 19, 35], and network quantization [5, 6, 9, 10, 30, 32, 45]. This work focuses on neural network quantization given its capability of reducing the number of floating-point operations and memory footprint by replacing real-valued weights and activations with integer-valued ones. Although a quantized network has an inherent problem of quantization error, training of quantized networks with little performance loss is becoming feasible [6, 9, 10, 22, 61].

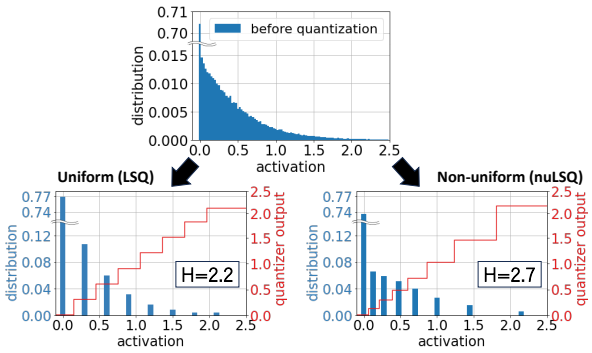


Fig. 1: Distributions of activation in ResNet-20 before quantization (upper) and after uniform quantization by LSQ (lower left) and non-uniform quantization by our method, nuLSQ (lower right). Our nuLSQ, which has multiple learnable step sizes, acquires finer/coarser step sizes in the activation range having higher/lower distribution, which leads to higher information entropy H.

A high-performing method referred to as Learned Step-Size Quantization (LSQ) [9] utilizes back-propagation with approximate gradients using the straight-through estimator (STE) [4] to optimize both model weights and step sizes, which define different quantization levels for weights and activations. In LSQ, weights in the same layer share a common step size, and activations do likewise. For example, with 4-bit quantization in LSQ, there are 16 discrete levels that are equally spaced by the learned step size within a layer. However, this uniform-step quantization imposes limitations on model expressivity, given that the distributions of weights and activations in neural networks are typically highly non-uniform [12, 22, 39, 53, 55, 61].

In this regard, researchers have been exploring non-uniform quantization techniques to address the limitations of uniform quantization [28, 45, 61]. As illustrated in Fig. 1, non-uniform quantization schemes optimize multiple step sizes within a layer, allowing quantization levels to be unevenly spaced. This enables the intervals to better adapt to the distribution of data, resulting in greater expressive capabilities. This approach potentially maintains a higher information entropy because quantization patterns are less biased. Due to its superior expressivity, non-uniform quantization for neural networks compression tends to achieve better performance than its uniform counterpart. However, model performance does depend on non-uniform quantization settings, and finding the best practice is an open issue.

To this end, we propose a method called non-uniform Learned Step-Size Quantization (nuLSQ), where multiple step sizes within a layer are individually optimized according to their step-size gradients based on the straight-through estimator (STE) [4]. In contrast to LSQ, nuLSQ optimizes step sizes non-uniformly to better preserve the original long-tailed data distribution observed in weights and activations. Different from existing non-uniform quan-

tization methods, nuLSQ has no explicit or implicit constraints in designing piecewise-constant increasing non-uniform quantizers. Evaluations of nuLSQ compared to various baseline methods on CIFAR-10, CIFAR-100, and ImageNet [7] provide supportive evidence that the proposed scheme yields high generalization performance. Our contributions are summarized as follows:

1. We propose nuLSQ for training non-uniform step sizes to quantize weights and activations with the desired precision. Unlike existing non-uniform quantization methods, nuLSQ optimizes individual step sizes independently, without imposing constraints on the quantizers. With this construction, nuLSQ can potentially generate arbitrary quantization levels to leverage network expressivity.
2. We found that our method becomes more pronounced as the bit-width decreases. Particularly in MobileNetV2, this feature is significant, resulting in an 8% improvement in state-of-the-art top-1 accuracy at 2 bit. In addition to MobileNetV2, our method achieves the highest accuracy among the existing quantization-aware training (QAT) methods across 2, 3, and 4-bits for ResNet-20 and ResNet-56 on CIFAR10.
3. Furthermore, in fair comparisons with LSQ in ResNet-20 on CIFAR100, and Swin-T and ConvNeXT on ImageNet as well as other existing methods including LSQ on ResNet-18, our method consistently achieves the highest accuracy.

2 Related Work

Extensive research has been undertaken to enhance the efficiency of neural network models in terms of reducing latency, memory footprint, and energy consumption with little sacrifice in the generalization ability. There are largely four types of approaches: efficient architecture search [17,18,49,51] equipped with automated machine learning (AutoML) [11,44,54], knowledge distillation [3,16,24, 29,56,57,59], network pruning [15,19,35], and network quantization [5,6,9,10,45]. Among these, we describe the related work of network quantization below. It is also worth mentioning that neural network quantization synergizes well with other compression techniques: Previous studies such as QKD [24], LSQ [9], PROFIT [42], and N2UQ [34] have highlighted the effectiveness of combining knowledge distillation with quantization. This collaborative approach boosts the performance of compressed models.

Neural network quantization replaces the entire or a large portion of real-valued weights and activations by coarsely discretized weights and activations. By drastically reducing the number of floating-point variables and operations, a quantized network yields fast inference with reduced memory usage. Unlike other network compression techniques mentioned earlier, network quantization can compress the original model significantly without any architectural modifications. Quantization of a trained model inevitably shifts the model parameters, thereby causing deviation from the (semi-)convergence point attained during training with floating-point precision. Hence, fine-tuning network parameters

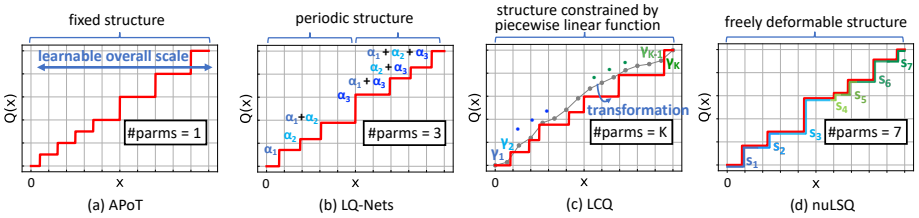


Fig. 2: Behaviors of quantizers in existing non-uniform methods and our method at 3-bit.

through quantization-aware training (QAT) is generally required for achieving desired model performance under low-bit precision.

Despite the effectiveness of the QAT, optimization of the quantization parameters remains challenging, particularly with extremely low-bit precision. The primary challenge comes from the discretization of the parameters or activations by its quantizer. A quantizer, which is essentially a piece-wise constant function, has zero derivatives almost everywhere; therefore, the adoption of gradient-based optimization is not straightforward.

Straight-Through Estimator (STE). A naive way to avoid the problem is to relax the quantizer to a continuous function and gradually revert to the original one [10, 23]. A simpler way is to replace the derivative of the quantizer with that of the identity function with a clipping threshold in the backward pass only, known as STE [6, 9, 63]. Despite the issue of the gradient mismatch, employing STE as a proxy of its gradient has been shown to be an effective approach empirically [6, 9, 22, 55, 61, 63, 64] and theoretically [50, 58]. Using STE, many prior studies have attempted to optimize the uniform/non-uniform quantizer itself.

Uniform Quantization. Uniform quantization involves the mapping of floating-point values to quantized values that are uniformly spaced. PACT [6] first attempted to train the uniform quantization function in activations characterized by the clipping threshold. LSQ [9] achieves higher accuracy for models in which both weights and activations are quantized by effectively utilizing STE and training uniform step sizes. LSQ is further improved by LSQ+ [5] that can be applied to the quantization of activations other than ReLU function like swish [46] and h-swish [17] functions. QIL [22] and N2UQ [34] perform uniform quantization following some non-linear transformation. QIL utilizes a trainable power function, while N2UQ achieves this directly by employing a quantization function that equalizes non-uniform intervals.

Non-Uniform Quantization. Non-uniform quantization involves the mapping of floating-point values to quantized values that are non-uniformly spaced. LQ-Nets [61] has non-uniform quantization levels, where the weights and activations are decomposed into a quantizer basis and binarized matrix by minimizing the quantization error. Due to the property of its decomposition, the quantization

levels are constrained to have a periodic structure (Fig. 2b). APoT [28], a generalization of the log quantization [39], achieves efficient and hardware-friendly non-uniform quantization levels by constraining its base structure to a sum of powers of two and training only the clipping threshold (Fig. 2a). LCQ [55] gives less-constrained quantization levels by transforming uniform step-sizes to non-uniform ones with learnable piecewise-linear monotonically-increasing functions (Fig. 2c). Despite its seemingly higher number of parameters, the deformability of its quantizer is practically limited by implicit constraints embedded in the piecewise linear functions with the specific parametrization and number of joints. On the other hand, our method nuLSQ, a non-uniform extension of LSQ, trains non-uniform step sizes, or equivalently non-uniform quantization levels, directly without any transformations or constraints. Therefore, our method has the potential to achieve arbitrary quantization levels within a given precision through optimizing the individual step sizes in an independent fashion (Fig. 2d).

Simluated Quantization. When it comes to quantized neural network deployment, there are generally two widely adopted approaches: integer-only inference [20, 30, 31] and fake quantization. Integer-only inference performs all computations under low-bit precision, while fake quantization simulates its effects by quantizing only weights and activations while floating-point operations partially remain. Fake quantization we adopted is widely investigated in the field of the model quantization because of its simple implementation [9, 28, 32, 45].

Hardware implementation in non-uniform quantization We consider two ways to deploy our non-uniform quantization: (i) mapping to step sizes hardware-friendly utilized in log quantization [39] and APoT [28], and (ii) using Look-Up Tables (LUTs) [55]. (i) For example, when considering 4 bits and mapping the step sizes learned by our method into 2 additive power-of-two terms in the multiplier, the non-uniform quantization is approximately twice as fast as the uniform quantization in multiplication [28]. (ii) When targeting FPGAs, LUTs are preferred: LUTs offer superior parallelism compared to DSPs for multipliers even for uniform quantization below 8 bits [26]. Hence, LUTs are the preferred choice for both uniform quantization and non-uniform quantization on FPGA. The sizes of the LUTs for non-uniform quantization are estimated to be reasonably small [55].

Distributions of weights and activations. It has been empirically observed that weights and activations of a layer in a trained real-valued DNN tend to have a long-tailed distribution [12,39,61]. Their distributions are far from the uniform distribution and have a peak near the origin. Particularly, the distribution of the activation has a strong peak at zero due to the ReLU function, and decays rapidly as it moves away from the origin, while it continues up to large values as illustrated in the top figure of Fig. 1. When quantizing the activation with uniform quantizers such as LSQ, the quantized values near zero tend to appear more frequently, and the values near the largest value tend to appear less frequently. This strong bias likely decreases the information entropy of the quantized patterns, potentially restricting the network's descriptive ability.

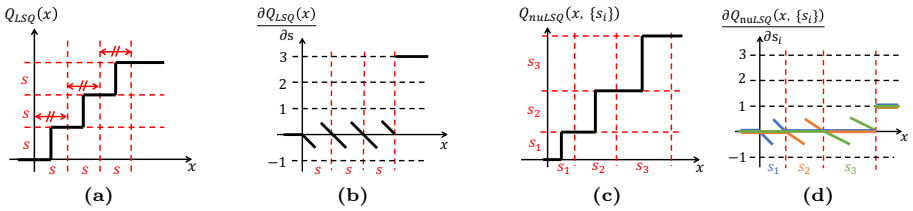


Fig. 3: The 2-bit quantizers for activations used in LSQ and nuLSQ. Note that $Q_n = 0$, $Q_p = 3$. (a) Uniform quantizer with shared step size s. (b) Approximate gradients of the uniform quantizer with respect to s. (c) Non-uniform quantizer with individual step sizes s_1, s_2, and s_3. (d) Approximate gradients of the non-uniform quantizer with respect to s_1 (blue), s_2 (orange), and s_3 (green).

3 Proposed Method

3.1 Preliminaries: overview and reformulation of LSQ

Since our proposed method (nuLSQ) is closely related to LSQ, we start with an overview of LSQ. Its uniform quantizer $Q_{LSQ}(\cdot)$ depicted in Fig. 3a is given by

$$Q_{LSQ}(x, s) = \lfloor \mathrm{clip}\left(\frac{x}{s}, -Q_n, Q_p\right) \rceil s = \begin{cases} -Q_n s, & x/s \le -Q_n \\ \lfloor x/s \rceil s, & -Q_n < x/s < Q_p \\ Q_p s, & x/s \ge Q_p \end{cases} \tag{1}$$

where x is either the input value or the weight, s denotes the step size, and Q_p and Q_n are non-negative integers specifying the upper and lower bounds of the quantization level, respectively. $\lfloor \cdot \rceil$ is the round operator and $\mathrm{clip}(x, n, p)$ returns x if x is within $[n, p]$, n if x is below n, and p if x is above p. For b-bit weight quantization, $Q_n = 2^{b-1}$ and $Q_p = 2^{b-1} - 1$. For b-bit ReLU activation quantization, $Q_n = 0$ and $Q_p = 2^b - 1$.

In a strict sense, one cannot optimize a quantized network with gradient descent because the gradient of the rounding operator is zero everywhere except for non-differentiable points. To overcome this issue, STE is used to replace the gradient of $\lfloor x/s \rceil$ with that of x/s in the backward pass. According to the above procedure, the step-size gradient depicted in Fig. 3b is obtained as

$$\frac{\partial Q_{LSQ}}{\partial s}(x, s) = \begin{cases} -Q_n, & x/s \le -Q_n \\ \lfloor x/s \rceil - x/s, & -Q_n < x/s < Q_p \\ Q_p, & x/s \ge Q_p. \end{cases} \tag{2}$$

The uniform quantizer in Eq. (1) can be rewritten by using a linear combination of the unit step function $\sigma(\cdot)$,

$$Q_{LSQ}(x, s) = \begin{cases} -\sum_{n=1}^{Q_n} s\sigma\left(-x - ns + \frac{s}{2}\right), & x < 0 \\ \sum_{n=1}^{Q_p} s\sigma\left(x - ns + \frac{s}{2}\right). & x \ge 0 \end{cases}, \tag{3}$$

where Q_n, Q_p are assumed to be positive integers. For $Q_n = 0$ and $Q_p > 0$, (e.g., ReLU activation quantization), $Q_{LSQ}(x, s)$ is defined as 0 for $x < 0$, while the above equation applies for $x \geq 0$. The derivative with respect to the step size using STE for each step function coincides with Eq. (2) as shown in detail in the supplementary material. This reformulation is part of our contribution, addressing the original derivation's lack of capacity for extension to unequal intervals.

3.2 nuLSQ: Non-Uniform Learned Step-Size Quantization

We propose nuLSQ that jointly optimizes weights and multiple step sizes of quantized weights and activations as illustrated in Fig. 3c. When approximating a long-tailed distribution with samples having quantized patterns, the non-uniform quantization scheme generates less redundant quantized patterns than the uniform counterpart. Therefore, the expressivity is improved. By replacing the common step size s in Eq. (3) with sets of step sizes that can be moved individually in the positive range $\{s_i |\ i = 1, \ldots, Q_p\}$, and in the negative range $\{s'_i |\ i = 1, \ldots, Q_n\}$, the non-uniform quantization function can be obtained as

$$Q_{\mathrm{nuLSQ}}(x, \{s_i\}, \{s'_i\}) = \begin{cases} -\sum_{n=1}^{Q_n} s'_n \sigma\left(-x - \sum_{m=1}^{n-1} s'_m + \frac{s'_n}{2}\right), & x < 0 \\ \sum_{n=1}^{Q_p} s_n \sigma\left(x - \sum_{m=1}^{n-1} s_m + \frac{s_n}{2}\right). & x \geq 0 \end{cases} \tag{4}$$

As in the uniform quantization case, $Q_{\mathrm{nuLSQ}}(x, \{s_i\}, \{s'_i\})$ for $x < 0$ is replaced with 0 for the unsigned integer range ($Q_n = 0$ and $Q_p = 2^b - 1$).

Their approximate gradients are obtained using STE:

$$\frac{\partial Q_{\mathrm{nuLSQ}}(x, \{s_i\}, \{s'_i\})}{\partial s_k} = \begin{cases} 0, & x < \sum_{m=1}^{k-1} s_m \\ D_k(x, \{s_i\}), & \sum_{m=1}^{k-1} s_m \leq x < \sum_{m=1}^{k} s_m \\ 0, & \sum_{m=1}^{k} s_m \leq x < \sum_{m=1}^{Q_p} s_m \\ 1, & \sum_{m=1}^{Q_p} s_m \leq x \end{cases} \tag{5}$$

$$\frac{\partial Q_{\mathrm{nuLSQ}}(x, \{s_i\}, \{s'_i\})}{\partial s'_k} = \begin{cases} -1, & x < -\sum_{m=1}^{Q_n} s'_m \\ 0, & -\sum_{m=1}^{Q_n} s'_m \leq x < -\sum_{m=1}^{k} s'_m \\ B_k(x, \{s'_i\}), & -\sum_{m=1}^{k} s'_m \leq x < -\sum_{m=1}^{k-1} s'_m \\ 0, & -\sum_{m=1}^{k-1} s'_m \leq x \end{cases} \tag{6}$$

with $D_k(x, \{s_i\}) = \sigma\left(x - \sum_{m=1}^{k} s_m + \frac{s_k}{2}\right) - \frac{x - \sum_{m=1}^{k-1} s_m}{s_k}$ and $B_k(x, \{s'_i\}) = -\sigma\left(-x - \sum_{m=1}^{k-1} s'_m + \frac{s'_k}{2}\right) - \frac{x + \sum_{m=1}^{k-1} s'_m}{s'_k}$. Using the gradients, all step-sizes $\{s_i\}$ and $\{s'_i\}$ are updated independently.

Note that if all the step sizes $\{s_i\}$ and $\{s'_i\}$ are set to be equal, the gradient in Eq. (5) leads to the uniform step-size gradient used in LSQ. This implies that

the estimated gradient of nuLSQ (see in Fig. 3d) inherits the characteristic of the one in LSQ [9].

Similar derivations have been discussed in N2UQ [34] for achieving a sort of uniform quantization in activations, which equalize the non-uniform intervals. Our derivation, however, is not restricted to uniform outputs, thereby enhancing the flexibility of the quantizer, and increasing its expressive capacity. Moreover, our derivation can be applied to both weights and activations.

4 Experiments

4.1 Experimental Settings

Datasets. All experiments were conducted on CIFAR-10/100 [25] and ImageNet [7] datasets. The CIFAR-10/100 dataset consists of 60K 32×32 color images, with 6K/600 images in each of the 10/100 different classes. There are 50K training images and 10K testing images. The ImageNet dataset consists of over 1.2M images for training from 1K classes and 50K images for validation. For training, we applied the common data augmentation techniques found in [9, 61] under the following order: random image crop, image resize to 224×224, and random horizontal flip. For testing, we limited the data augmentation techniques to only center crop. All transformed images were normalized by the mean and the standard deviation.

Implementation details. Our proposed method was implemented in PyTorch. We tested nuLSQ by quantizing the architecutes of pre-activation and original versions of ResNet [14], MobileNetV2 [49], Swin-tiny [33], and ConvNeXt-tiny [36]. As summarized in Table. 1, we quantized both weights and input activations uniformly or non-uniformly. Both weights and activations were quantized into in 2-, 3-, or 4-bits for all the convolution and fully connected layers except for the first and last layers, whose low-bit quantization is empirically known to lead to significant performance degradation. Both the first and last layers were set to be 8-bit to balance hardware overhead and the accuracy degradation [9, 28, 45]. In all experiments, we applied layer-wise quantization for both activations and weights. Signed and unsigned quantization ranges were adopted for weight and activation quantization, respectively. The pre-trained models were taken from PytorchCV [1] and timm library [2]. We used the cosine learning rate decay without restart [37].

Table 1: Configurations of quantization.

Configuration	Weights	Activations
LSQ	Uniform	Uniform
nuLSQ-A	Uniform	Non-uniform
nuLSQ-W	Non-uniform	Uniform
nuLSQ-WA	Non-uniform	Non-uniform

For CIFAR10/100, we adopted the pre-activation version of ResNet-20 and -56. For ImageNet, we adopted the pre-activation and original versions of ResNet-18, MobileNetV2, Swin-T, and ConvNeXt. After hyperparameters were selected

in 15 epochs via validation data (10% of the original training dataset), we evaluated accuracy for 15 or 90 epochs with test data. More details of the implementation, along with hyper-parameter settings, are given in the supplementary material.

4.2 Mitigation strategies for the emergence of negative step sizes

As we move into the extremely low-precision regime with gradient-based learnable quantization approaches, the final performance after quantization training is notably influenced by the initialization of step sizes [5,45] and the handling of their gradients [9]. The extreme sensitivity often results in encountering negative step sizes or being trapped in a local minimum. Empirically, several factors contribute to this unwanted situation: (i) Bad initialization, diverging from the original distribution's shape and (ii) the large and varied magnitude of the step-size gradients, stemming from the STE approximation and the large difference in the number of parameters for weights/inputs and step sizes at each layer. To mitigate this problem, we employ MSE-based initialization and AdamW optimizer for step sizes. In the supplementary material, we also discuss the progressive fune-tuning method [64].

MSE-based initialization and AdamW optimizer. The MSE-based initialization [5,45] and Adam optimizer for step sizes [21,34,40] were seen to show good performance in different previous studies. The MSE-based initialization brings the distributions of the initial quantized weights and inputs closer to the original one. The Adam optimizer normalizes large and varied magnitude of the step-size gradients, thereby facilitating training with a common learning rate for the step sizes. Here, we utilize both methods jointly to obtain good performance while avoiding the emergence of negative step size or being trapped in local minima. To treat the weight decay for the large magnitude of step-sizes' gradient correctly, we use AdamW optimizer instead of Adam optimizer. We performed an ablation study on initialization and optimizer by comparing with LSQ initialization [9] and SGD optimizer. This comparison was conducted using 2-bit nuLSQ-WA applied to the pre-activation version of Resnet-18 network on ImageNet. As shown in Table 2, MSE initialization and AdamW optimizer jointly yield 0.7% enhancement in accuracy.

Table 2: Effects of initialization and optimizer for step sizes at 2-bit pre-activation version of Resnet-18 for 90 epochs

Method	Initialization	optimizer	Top-1 Acc
nuLSQ-WA	LSQ init [9]	SGD	67.11%
	LSQ init [9]	AdamW	67.34%
	MSE init	AdamW	**67.79%**

4.3 Comparison of existing methods with MobilenetV2 architecture

Effect of BN re-estimation Like LSQ, nuLSQ shows significant accuracy degradation on MobileNetV2 due to weight oscillations caused by quantization. As discussed in [40], this can be recovered by re-estimating the batch normalization statistics after training [38, 43]. From Table. 3, we can see that BN re-estimation is effective in nuLSQ as well.

Table 3: Accuracy (%) on 2-bit MobileNetV2 for 90 epochs before and after BN re-estimation

methods	pre-BN	post-BN
nuLSQ-W	53.48	58.26
nuLSQ-A	53.19	58.43
nuLSQ-WA	49.57	58.72

Comparison to other QAT methods We compare our method with other well-established QAT methods without knowledge distillation for MobileNetV2 on ImageNet. From Table 4, we can see that our method outperforms all of the existing methods. In particular, our method shows a significant improvement of 8% in accuracy compared to the existing methods at 2-bit. Remarkably, with the incorporation of oscillation dampening (Dp) and iterative weight freezing (Fz) methods proposed in [40] for LSQ into nuLSQ, it is expected that the accuracy is further improved over the present results using the re-estimation of the batch normalization.

Table 4: Comparison with state-of-the-art QAT methods without and with knowledge distillation (KD) on MobileNetV2. "FP" represents "Full Precision", while the "W/A" values denotes the bit-widths of weights/activations, respectively. The results of existing methods are referenced from their corresponding papers.

Methods without knowledge distillation	Bit-width (W/A) 2/2	3/3	4/4
MobileNetV2 (FP: 72.91)			
PACT [6]	-	-	61.4
LSQ [9, 13]	46.7	65.3	69.5
LSQ + BR [13]	50.6	67.4	70.4
LSQ + Dp/Fz [40]	-	67.8	70.6
UniQ [45]	50.5	65.0	68.2
DSQ [10]	-	-	64.8
LCQ [55]	-	-	70.8
LLSQ [62]	-	-	67.4
EWGS [27]	-	-	70.3
nuLSQ-W	**58.3**	**67.8**	**71.0**
nuLSQ-A	**58.4**	**67.9**	**71.1**
nuLSQ-WA	**58.7**	**68.3**	**70.9**

Methods with knowledge distillation	Bit-width (W/A) 4/4
MobileNetV2 (FP: 72.91)	
QKD [24]	67.4
PROFIT [42]	71.56
nuLSQ-A + KD	**71.89**

Effect of knowledge distillation We have conducted the evaluation with knowledge distillation. We adopted the same distillation approach as discussed in [9]: using the standard and distillation losses at temperature of 1 in the same ratio, and employing a pre-trained FP model with frozen weights as the teacher network. We trained the networks for 128 epochs. We compared our results with the previous studies that utilized knowledge distillation on 4-bit MobileNetV2 in

Table 4. For a fair comparison, N2UQ [34] was excluded despite utilizing knowledge distillation, because it used additional procedures: replacing the activation function with RPReLU and incorporating learnable bias terms.

Our method reduces the accuracy drop to nearly 1%. This further highlights the practical significance of 4-bit quantization.

4.4 Fair comparison of existing methods with ResNet architecture on ImageNet

We conducted a fair evaluation of existing methods using identical setup for various configurations, including optimizers, schedulers, epochs, and choice of pre-trained models. We present a comparison between our methods and some of existing methods in Table 5. We trained 2-bit models with original version of ResNet-18 for 15 epochs. We utilized AdamW optimizer with MSE initialization for quantizer parameters such as step sizes (LSQ/nuLSQ-A) and clipping threshold (PACT/APoT/LQ-Nets), while weights were optimized using SGD. Hyperparameters were selected via validation data in all of the existing methods as well. Notably, nuLSQ-A outperforms the existing methods. In the supplemantary material, we also show the comparison in the pre-activation version of ResNet-18 and -34 with more various existing methods, demonstrating that nuLSQ outperforms them.

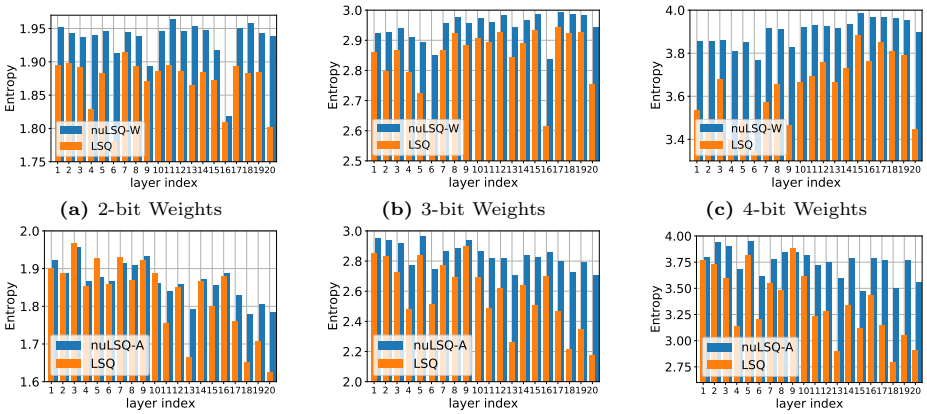
Table 5: Top-1 test accuracy (%) on ImageNet comparisons. ResNet-18 (FP:69.76%) at 2-bit under identical conditions with MSE initialization + AdamW optimizer for quantizers. Results marked with * are obtained from our implentation, while those marked with † are obtained from the original source code after fixing minor bugs.

PACT* [6]	DoReFa* [63]	LSQ* [9]	APoT† [28]	LQ-Nets* [61]	LCQ* [55]	nuLSQ-A
62.48	63.28	64.51	64.41	63.71	64.67	**64.89**

4.5 Detailed comparison with LSQ on various architectures

Since nuLSQ is a natural extension of LSQ, we compared nuLSQ with LSQ in detail. To ensure fairness, we implemented LSQ and conducted the accuracy comparison under identical experimental settings.

Evaluation with Resnet architectures on CIFAR-100. We employed the pre-activation version of ResNet-20 and Resnet-56 on CIFAR-100. We performed the experiment five times and calculated the mean accuracy and the standard deviation σ_{acc}. In Table 6, we can see that nuLSQ outperforms LSQ in all cases except for 4-bit ResNet-20, which shows relatively large standard deviations, 0.33% in LSQ and 0.43% in nuLSQ-A. Particularly, it is shown to be effective at 2-bit and 3-bit with improvements of up to 0.36%. These results indicate that accuracy degradation due to large quantization errors at low bits, especially below



(a) 2-bit Weights **(b)** 3-bit Weights **(c)** 4-bit Weights

(d) 2-bit Activations **(e)** 3-bit Activations **(f)** 4-bit Activations

Fig. 4: Comparison of Shannon Entropy of quantized weights (a)-(c) and quantized activations (d)-(f) using LSQ and nuLSQ-A and -W on ResNet-20. Regardless of bit widths, nuLSQ consistently exhibits larger entropy in most layers.

4-bit, is mitigated with non-uniform quantization. As shown in the supplementary material, nuLSQ-WA, which has non-uniform quantizers for both weight and activation, led to a slight improvement over or nearly the same accuracy as nuLSQ-A or nuLSQ-W.

Table 6: Top-1 accuracy (%) comparison between LSQ and our method with preactivation version of ResNet-20, -56 on CIFAR-100 dataset. Results marked with * are from our implementation.

Network	Methods	Bit-width(W/A)		
		2/2	3/3	4/4
ResNet-20 (FP: 69.8)	*LSQ [9]	65.82 ± 0.23	68.60 ± 0.23	$\mathbf{69.56 \pm 0.33}$
	nuLSQ-A	$\mathbf{66.02 \pm 0.29}$	68.58 ± 0.21	69.42 ± 0.43
	nuLSQ-W	66.00 ± 0.39	$\mathbf{68.70 \pm 0.11}$	69.40 ± 0.14
ResNet-56 (FP: 74.9)	*LSQ [9]	70.50 ± 0.11	72.62 ± 0.15	$\mathbf{73.48 \pm 0.21}$
	nuLSQ-A	70.66 ± 0.29	$\mathbf{72.98 \pm 0.10}$	$\mathbf{73.48 \pm 0.25}$
	nuLSQ-W	$\mathbf{70.82 \pm 0.22}$	72.80 ± 0.18	73.42 ± 0.25

Comparison on information entropy. To examine how nuLSQ affects the data distribution by quantization compared to LSQ, we calculated the Shannon Entropy defined as $H(X) = -\sum_i P(x_i) \log_2 P(x_i)$, where $P(\cdot)$ denotes the probability of observing the value x_i of the set of the quantized output X. The results of the weights and the activations of each quantized layer for ResNet-20 except the first and last layers are shown in Fig. 4. Our results demonstrate that nuLSQ gains more information than LSQ in most layers regardless of bit widths.

This implies that the distributions of the quantized activations and weights in nuLSQ are more diverse and less concentrated around specific values. This fits our motivation described in Section 1: The non-uniform quantization scheme is able to adapt to the original distribution better than the uniform quantization scheme with an increased representation capacity.

Evaluation with Modern architectures on ImageNet. We conducted evaluations on more modern networks with self-attention layer and larger kernel sizes. Specifically, we experimented with 2-bit, 3-bit, and 4-bit Swin-T, as well as 3-bit ConvNeXt as shown in Table 7. Across these experiments, nuLSQ-WA consistently outperforms LSQ, confirming the effectiveness of nuLSQ in diverse architecutres.

Table 7: Top-1 accuracy(%) comparison between LSQ and nuLSQ-WA on 2-bit Swin-T and 3-bit ConvNeXt for 15 epochs. Results marked with * are from our implementation.

Methods	Swin-T (FP:81.2)			ConvNeXt (FP:81.87)
	2/2	3/3	4/4	3/3
*LSQ	74.58	77.48	78.33	72.90
nuLSQ-WA(ours)	**74.91**	**77.71**	**78.37**	**73.39**

4.6 Comparison with other non-uniform methods on CIFAR-10

We compared nuLSQ with other well-established non-uniform methods: LQ-Nets, APoT, and LCQ. The accuracy of nuLSQ and those of existing methods on CIFAR-10 are listed in Table 8. For a careful comparison, we implemented LCQ and trained it from the same floating model used in nuLSQ. The detailed hyperparameter settings are summarized in the supplementary material. We performed the experiment five times and calculated the mean accuracy and the standard deviation σ_{acc}. We can see that nuLSQ-A outperforms other existing methods in all cases. This is attributed to the fact that our quantizer was trained to lower the loss due to its high deformability as shown in Fig. 2 and Table 9. Finally, it should also be emphasized that our method has the additional advantage that there are no hyperparameters for the quantizer itself.

5 Conclusion

In this work, we present nuLSQ, a non-uniform version of LSQ (Learned Step-size Quantization), as a novel approach for training quantized networks. Unlike LSQ, where the quantization levels are uniformly structured, nuLSQ offers enhanced flexibility by individually optimizing the quantization levels. The nuLSQ has more flexibility than the existing non-uniform quantization methods. This

Table 8: Top-1 accuracy(%) comparison between nuLSQ-A and other non-uniform methods using ResNet-20, -56 on CIFAR-10 dataset. Results marked with * are from our implementation. † denotes the results from the pre-activation version of ResNet.

Network	Methods	Bit-width(W/A)		
		2/2	3/3	4/4
ResNet-20 (FP: 93.49)	†LQ-Nets [61]	90.2	91.6	-
	APoT [28]	91.0	92.2	92.3
	†*LCQ [55]	90.94 ± 0.38	92.44 ± 0.15	92.94 ± 0.16
	†**nuLSQ-A(ours)**	$\mathbf{91.30 \pm 0.13}$	$\mathbf{92.66 \pm 0.14}$	$\mathbf{93.16 \pm 0.17}$
ResNet-56 (FP: 95.51)	APoT [28]	92.9	93.9	94.0
	†*LCQ [55]	91.82 ± 0.24	94.54 ± 0.18	94.67 ± 0.14
	†**nuLSQ-A(ours)**	$\mathbf{93.84 \pm 0.16}$	$\mathbf{94.66 \pm 0.14}$	$\mathbf{95.34 \pm 0.10}$

Table 9: Hyperparameters and learnable parameters of non-uniform b-bit quantizers. The number in the brackets for learnable parameters denotes the number of them.

methods	hyperparameters	learnable parameters
APoT	#PoT terms	clipping threshold (1)
LQ-Nets	#iterations of QEM alg.	quantizer basis (b)
LCQ	#intervals on piecewise linear function ($= K$)	clipping threshold + slope of intervals ($1 + K$)
nuLSQ	-	step-sizes ($2^b - 1$)

customization allows for a more accurate fitting of the original data distribution, leading to improved network performance.

To validate the effectiveness of nuLSQ, we conducted comprehensive experiments and comparative evaluations. We compared the performance of nuLSQ against state-of-the-art quantization approaches, including LSQ. We utilized three benchmark datasets and four network architectures to assess the generalizability of nuLSQ. We measured and analyzed important performance accuracy to provide a comprehensive evaluation of nuLSQ's advantages over the previous approaches. We highlight its improved performance over LSQ and other state-of-the-art quantization methods, emphasizing its potential for enhancing network training and improving model compression performance.

Acknowledgement. This work is an outcome of a research project, Development of Quality Foundation for Machine-Learning Applications, supported by DENSO IT LAB Recognition and Learning Algorithm Collaborative Research Chair (Science Tokyo). Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.

We would like to thank Teppei Suzuki for helpful feedback on the paper.

References

1. Pytorchcv, `https://pypi.org/project/pytorchcv/`

2. timm, `https://github.com/rwightman/pytorch-image-models`
3. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9163–9171 (2019)
4. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
5. Bhalgat, Y., Lee, J., Nagel, M., Blankevoort, T., Kwak, N.: Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 696–697 (2020)
6. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.J., Srinivasan, V., Gopalakrishnan, K.: Pact: Parameterized clipping activation for quantized neural networks. International Conference on Learning Representation (2018)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
9. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. In: International Conference on Learning Representations (2019)
10. Gong, R., Liu, X., Jiang, S., Li, T., Hu, P., Lin, J., Yu, F., Yan, J.: Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4852–4861 (2019)
11. Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J.: Single path one-shot neural architecture search with uniform sampling. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. pp. 544–560. Springer (2020)
12. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. International Conference on Learning Representations (ICLR) (2016)
13. Han, T., Li, D., Liu, J., Tian, L., Shan, Y.: Improving low-precision network quantization via bin regularization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5261–5270 (2021)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. He, Y., Lin, J., Liu, Z., Wang, H., Li, L.J., Han, S.: Amc: Automl for model compression and acceleration on mobile devices. In: Proceedings of the European conference on computer vision (ECCV). pp. 784–800 (2018)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. stat **1050**, 9 (2015)
17. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1314–1324 (2019)

18. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
19. Huang, Z., Wang, N.: Data-driven sparse structure selection for deep neural networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 304–320 (2018)
20. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2704–2713 (2018)
21. Jain, S., Gural, A., Wu, M., Dick, C.: Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. Proceedings of Machine Learning and Systems **2**, 112–128 (2020)
22. Jung, S., Son, C., Lee, S., Son, J., Han, J.J., Kwak, Y., Hwang, S.J., Choi, C.: Learning to quantize deep networks by optimizing quantization intervals with task loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4350–4359 (2019)
23. Kim, D., Lee, J., Ham, B.: Distance-aware quantization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5271–5280 (2021)
24. Kim, J., Bhalgat, Y., Lee, J., Patel, C., Kwak, N.: Qkd: Quantization-aware knowledge distillation. arXiv preprint arXiv:1911.12491 (2019)
25. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
26. Latotzke, C., Ciesielski, T., Gemmeke, T.: Design of high-throughput mixed-precision cnn accelerators on fpga. In: 2022 32nd International Conference on Field-Programmable Logic and Applications (FPL). pp. 358–365. IEEE Computer Society, Los Alamitos, CA, USA (sep 2022). https://doi.org/10.1109/FPL57034.2022.00061, https://doi.ieeecomputersociety.org/10.1109/FPL57034.2022.00061
27. Lee, J., Kim, D., Ham, B.: Network quantization with element-wise gradient scaling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6448–6457 (2021)
28. Li, Y., Dong, X., Wang, W.: Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=BkgXT24tDS
29. Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., Li, L.J.: Learning from noisy labels with distillation. In: Proceedings of the IEEE international conference on computer vision. pp. 1910–1918 (2017)
30. Li, Z., Gu, Q.: I-vit: integer-only quantization for efficient vision transformer inference. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17065–17075 (2023)
31. Lin, D., Talathi, S., Annapureddy, S.: Fixed point quantization of deep convolutional networks. In: International conference on machine learning. pp. 2849–2858. PMLR (2016)
32. Liu, S.Y., Liu, Z., Cheng, K.T.: Oscillation-free quantization for low-bit vision transformers. arXiv preprint arXiv:2302.02210 (2023)
33. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)

34. Liu, Z., Cheng, K.T., Huang, D., Xing, E.P., Shen, Z.: Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4942–4952 (2022)
35. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE international conference on computer vision. pp. 2736–2744 (2017)
36. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)
37. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. International Conference on Learning Representations (2017), https://openreview.net/forum?id=Skq89Scxx
38. Louizos, C., Reisser, M., Blankevoort, T., Gavves, E., Welling, M.: Relaxed quantization for discretized neural networks. arXiv preprint arXiv:1810.01875 (2018)
39. Miyashita, D., Lee, E.H., Murmann, B.: Convolutional neural networks using logarithmic data representation. arXiv preprint arXiv:1603.01025 (2016)
40. Nagel, M., Fournarakis, M., Bondarenko, Y., Blankevoort, T.: Overcoming oscillations in quantization-aware training. In: Proceedings of the 39th International Conference on Machine Learning. pp. 16318–16330 (2022), https://proceedings.mlr.press/v162/nagel22a.html
41. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. In: 9th ISCA Speech Synthesis Workshop. pp. 125–125 (2016)
42. Park, E., Yoo, S.: Profit: A novel training method for sub-4-bit mobilenet models. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. pp. 430–446. Springer (2020)
43. Peters, J.W., Welling, M.: Probabilistic binary neural networks. arXiv preprint arXiv:1809.03368 (2018)
44. Pham, H., Guan, M., Zoph, B., Le, Q., Dean, J.: Efficient neural architecture search via parameters sharing. In: International conference on machine learning. pp. 4095–4104. PMLR (2018)
45. Pham, P., Abraham, J.A., Chung, J.: Training multi-bit quantized and binarized networks with a learnable symmetric quantizer. IEEE Access **9**, 47194–47203 (2021)
46. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. CoRR **abs/1710.05941** (2017), http://arxiv.org/abs/1710.05941
47. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
48. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
49. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
50. Shekhovtsov, A., Yanush, V., Flach, B.: Path sample-analytic gradient estimators for stochastic binary networks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. pp. 12884–12894 (2020)

51. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)
52. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
53. Wang, L., Dong, X., Wang, Y., Liu, L., An, W., Guo, Y.: Learnable lookup table for neural network quantization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12423–12433 (2022)
54. Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., Keutzer, K.: Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10734–10742 (2019)
55. Yamamoto, K.: Learnable companding quantization for accurate low-bit neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5029–5038 (2021)
56. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4133–4141 (2017)
57. Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8715–8724 (2020)
58. Yin, P., Lyu, J., Zhang, S., Osher, S.J., Qi, Y., Xin, J.: Understanding straight-through estimator in training activation quantized neural nets. In: International Conference on Learning Representations (2019)
59. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: International Conference on Learning Representations (ICLR) (2017), `https://arxiv.org/abs/1612.03928`
60. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021)
61. Zhang, D., Yang, J., Ye, D., Hua, G.: Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 365–382 (2018)
62. Zhao, X., Wang, Y., Cai, X., Liu, C., Zhang, L.: Linear symmetric quantization of neural networks for low-precision integer hardware (2020)
63. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. CoRR **abs/1606.06160** (2016), `http://arxiv.org/abs/1606.06160`
64. Zhuang, B., Shen, C., Tan, M., Liu, L., Reid, I.: Towards effective low-bitwidth convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7920–7928 (2018)