IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 February 2023, accepted 16 March 2023, date of publication 29 March 2023, date of current version 3 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3259959

APPLIED RESEARCH

Non-Zero Grid for Accurate 2-Bit Additive Power-of-Two CNN Quantization

YOUNG MIN KIM[1], KYUNGHYUN HAN[2], WAI-KONG LEE[3], (Member, IEEE), HYUNG JIN CHANG[4], (Member, IEEE), AND SEONG OUN HWANG[3], (Senior Member, IEEE)

[1]Department of IT Convergence Engineering, Gachon University, Seongnam 13120, South Korea
[2]Department of Electronics and Computer Engineering, Hongik University, Sejong 30016, South Korea
[3]Department of Computer Engineering, Gachon University, Seongnam 13120, South Korea
[4]School of Computer Science, University of Birmingham, B15 2TT Birmingham, U.K.

Corresponding author: Seong Oun Hwang (sohwang@gachon.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) and its High-Potential Individuals Global Training Program under Grant 2021-0-01532 (50%), in part by the National Research Foundation of Korea (NRF) funded by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant 2021-0-00540 (25%), and in part by the Gachon University Research Fund under Grant GCU-202008460009 (25%).

ABSTRACT Quantization is an effective technique to reduce the memory and computational complexity of CNNs. Recent advances utilize additive powers-of-two to perform non-uniform quantization, which resembles a normal distribution and shows better performance than uniform quantization. With powers-of-two quantization, the computational complexity is also largely reduced because the slow multiplication operations are replaced with lightweight *shift* operations. However, there are serious problems in the previously proposed grid formulation for 2-bit quantization. In particular, these powers-of-two schemes produce zero values, generating significant training error and causing low accuracy. In addition, due to improper grid formulation, they also fallback to uniform quantization when the quantization level reaches 2-bit. Due to these reasons, on large CNN like ResNet-110, these powers-of-two schemes may not even train properly. To resolve these issues, we propose a new non-zero grid formulation that enables 2-bit non-uniform quantization and allow the CNN to be trained successfully in every attempt, even for a large network. The proposed technique quantizes weight as power-of-two values and projects it close to the mean area through a simple constant product on the exponential part. This allows our quantization scheme to closely resemble a non-uniform quantization at 2-bit, enabling successful training at 2-bit quantization, which is not found in the previous work. The proposed technique achieves 70.57% accuracy on the CIFAR-100 dataset trained with ResNet-110. This result is 6.24% higher than the additive powers-of-two scheme which only achieves 64.33% accuracy. Beside achieving higher accuracy, our work also maintains the same memory and computational efficiency with the original additive powers-of-two scheme.

INDEX TERMS Quantization, deep learning, convolutional neural network, Internet of Things.

I. INTRODUCTION

It is challenging to implement deep learning on mobile and Internet of Things (IoT) devices, due to the excessive memory and computational costs [24]. Therefore, various techniques have been proposed to reduce the deployment cost of CNNs on constrained devices [3], [10], [11]. Quantization is one of the representative techniques that resolve these two aspects in CNN implementations. In a nutshell, quantization maps continuous distribution of weight and activation to a discrete value with a fixed number of bits. Consider that when a 32-bit floating point (FP) is mapped to an 8-bit discrete value, the memory size is reduced to 25% of the original [4]. Both [24] and [4] show that quantization can be useful for small devices such as IoT with less memory. However, the lower the bit size, the lesser the value a model can express, which eventually

The associate editor coordinating the review of this manuscript and approving it for publication was Rute C. Sofia.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

decreases the accuracy. Recently, many researchers have proposed techniques to perform quantization at the sub-byte level (e.g., 4-bit and 2-bit) [7], [15], [19], [22], [23] with the aim of getting an extremely small CNN model that does not degrade the accuracy too much. There are two types of quantization on CNN: uniform quantization and non-uniform quantization. These two quantization techniques projects the discrete values following the interval between quantization (resolution), e.g., 4-bit and 2-bit, in a different way. From previous works [19], [20], it is observed that the weights in CNN after training are usually in a normal distribution, which is generally bell-shaped. In other words, the distribution of weights is concentrated near the peak where the mean area is zero, and it is ideal that there is also a proper distribution on the tail of wide range [8]. Therefore, it is important for a quantization scheme to closely resemble a proper normal distribution in order to achieve a good accuracy. In particular, non-uniform quantization assigns more values near the mean area and expresses them to be very similar to a normal distributions. Due to this reason, it is generally more accurate than uniform quantization.

Power-of-two (PoT) [31] is considered one of the most efficient non-uniform quantization techniques. After applying PoT [31] quantization, all multiplications can be replaced by *shift* operations, which are very efficient on many modern computer architectures. Zhou et al. proposed a PoT [31] quantization technique that assigns closer quantization value to the mean area but ignores the other parts. Li et al. [20] proposed additive powers-of-two (APoT) to improve the PoT [31], wherein they constrain all quantization values to the sum of the PoT [31], which can better adapt to normal distribution. APoT [20] adopts non-uniform quantization and achieves good performance at a low-bit width. Since PoT [31] and APoT [20] store the power-of-two in the grid weight or activation in advance and use it as a quantization value, there is no need to perform additional calculations for quantization projection. Due to this reason, PoT [31] and APoT [20] can achieve good accuracy and are computational lightweight. However, in 2-bit quantization, the grid formulation in APoT [20] treated the -0 and +0 as the same value (zero), causing it to fallback to a uniform quantization. Due to this reason, APoT [20] is unable to train successfully at 2-bit quantization in many cases, when the CNN model is large. Even if it trains successfully in some attempts, the accuracy of APoT [20] is severely limited. We observed that the same problem also occurs in PoT [31] scheme that shares the similar grid formulation that generates zeros.

To resolve this issue, we attempt to improve the APoT [20] scheme at 2-bit level. Our contributions are summarized as follows:

1) A new grid formulation is proposed for APoT [20] at 2-bit level. In particular, our method stores quantization values in grid as small values instead of zero (-0 or +0). This allows the distribution of 2-bit quantization to be non-uniform. With the proposed grid formulation, it is possible for APoT [20] to train a large CNN model successfully in every attempt including large CNN model, which was impossible for the original APoT [20] scheme.

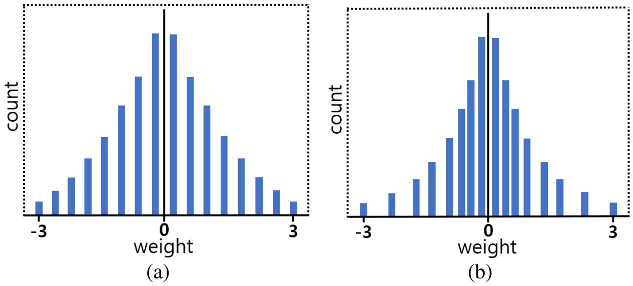


FIGURE 1. These two graphs show distributions of uniform (a) and non-uniform (b) quantization when projected at 4-bit. For uniform quantization, the quantization interval is evenly distributed. In contrast, the interval in non-uniform quantization is small and it is excessively concentrated on the mean area.

2) The proposed non-zero grid formulation closely resembles the ideal normal distribution, eventually improving the accuracy of APoT [20]. The proposed technique was evaluated on ResNet-32, ResNet-56 and ResNet-110 and compared against the APoT [20] and other state-of-the-art quantization scheme. Experimental results show that our work is able to achieve higher accuracy compared to existing techniques.
3) A thorough analysis was performed to pin-point the problems in APoT [20] grid formulation, why it failed to train successfully at 2-bit quantization, and how the proposed grid formulation can solve this issue. Our observations show that the presence of zeros after the quantization process can produce extremely large gradient, which greatly reduces the accuracy in CNN. The proposed grid solved this issue by using non-zero values, thus achieving a better accuracy. These observations can be helpful in analyzing other quantization schemes that shows the similar problem at 2-bit quantization level. The code of our implementation can be found at https://github.com/as705d/Enhanced-Quantization.git.

Note that both PoT [31] and APoT [20] show the same problems at 2-bit quantization. Hence, the non-zero grid proposed in this paper is applicable to both schemes.

II. RELATED WORK

A. UNIFORM QUANTIZATION

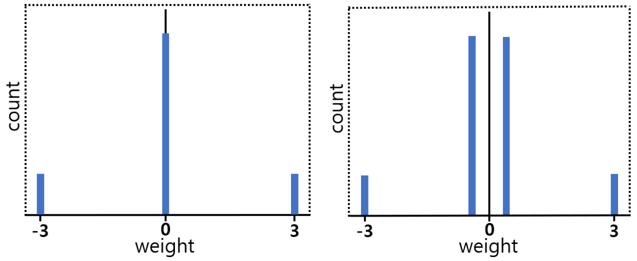
Uniform quantization projects the weights as discrete values with a constant continuous distribution range [17]; this is illustrated in Figure 1a. From previous work, [32] and [5] are trained and tested based on uniform quantization. They also use different methods to optimize the quantization. In the case of [5], new parameters are added to the existing activation function to find the optimal values for each training, while [32] applies quantization to gradient values to improve computational efficiency. Jacob et al. [13] and Wu et al. [26] utilized uniform quantization in their works because it is easy to implement, and consumes small memory and computational efficiency in inference. Kim et al. [16] optimized the student network using knowledge distillation techniques for quantization training. Recently, Lee et al. [19] pointed out the gradient mismatch problem for quantization in backpropagation process and proposed to solve this problem with a gradient scaling method. Note that the weights of the trained CNN model follow a normal distribution with a mean of 0. On the other hand, the uniform quantization eventually produces uniform distribution, which clearly fails to resemble the ideal normal distribution. This does not achieve good accuracy of the CNN model.

B. NON-UNIFORM QUANTIZATION

In contrast to uniform quantization, non-uniform quantization projects the weights with more values concentrated on the mean area; [1], [21], [30] this is illustrated in Figure 1b. In particular, quantization in [1] is performed using a lookup table, and [21] uses random matrix theory (RMT) to search for optimal quantization values. In addition, [30] proposed a vector segmentation scheme for bit operations of quantization. Yang et al. [29] presents a method to accurately retrieve discrete-weighted values in quantized neural networks using differential methods. Yamamoto et al. [27] uses *Compander* technology to reduce the bit width of the input value. These non-uniform quantization techniques introduce significant computational overhead because they dynamically explore suitable quantization values during inference process. Recently, some researchers proposed to represent the quantization values as power-of-two, which converts the expensive multiplication operations to a much cheaper shift operations. PoT [31] and APoT [20] are two representative examples in this research direction. However, both PoT [31] and APoT [20] fallback to uniform quantization at 2-bit, seriously affecting its accuracy.

C. ROUNDING APPROXIMATION FUNCTION

Rounding approximation quantization is a technique that can be used when conventional quantization is implemented as a simple rounding function. Quantization through rounding functions is applied during the training process. This results in gradient vanishing because the gradient for the rounding function in backpropagation becomes zero. Therefore, training with a function that approximates the rounding function during training can compensate for the accuracy loss when using the rounding function in inference. Prior works [6], [14] perform quantization based on the round function. In particular, they approach the gradient problem that the round function has, and properly adjusts the quantization interval during training to produce the optimal gradient. Yang et al. [28] proposed to train a CNN model by approximating the *Sigmoid* function as a rounding function. Gong et al. [7] uses the *Tanh* function instead of the *Sigmoid* function to closely approximate the rounding function. Kim et al. [15] trains a CNN model by computing the distance between the existing value and the quantization value to create a various rounding approximation. Although these approaches can provide good results to CNN inference, it is not applicable to our case because we do not use rounding functions in quantization.



(a) 2-bit uniform qauntization [20], [31]

(b) Non-zero grid

FIGURE 2. Comparing the distribution in 2-bit quantization. The proposed grid closely resembles a non-uniform quantization.

III. METHOD

A. BACKGROUND

This section briefly discusses the problem of 2-bit quantization in PoT [31] and APoT [20]. First, before performing weight quantization, a clipping function is used to keep the range constant for all layers:

$$W_{norm} = \frac{W - \mu}{\sigma} \tag{1}$$

$$\hat{W} = clip(\frac{W_{norm}}{\alpha}, -1, 1) \tag{2}$$

In Eq. 1, W is the floating point weight of the model and is normalized using the mean (μ) and standard deviation (σ) of each weight [12]. After that, According to clipping value (α), W_{norm} is fixed within the [-1,1] range for all layers. The following equations generate a grid that stores the power-of-two for quantization.

- PoT [31]:

$$Q^{w}(\alpha, b) = \alpha \times g \tag{3}$$

where $g \in \left\{0, 2^{-i}, \dots, 1\right\}, 0 \leq i < 2^{b} - 1$

- APoT [20]:

$$Q^{w}(\alpha, kn) = \alpha \times \left\{ \sum_{i=0}^{n-1} g_i \right\} \tag{4}$$

where $g_i \in \left\{0, \frac{1}{2^i}, \frac{1}{2^{i+n}}, \dots, \frac{1}{2^{i+(2^k-2)n}} \right\}$

In both equations, α is the clipping value, g denotes the grid, and b is the bit size. Both PoT [31] and APoT [20] store quantization values in the grid as a power-of-two, which can be implemented as a *shift* operation in the hardware. In APoT [20], the authors proposed using $k = 2$ when calculating parameters $kn = b$. Therefore, the selected grid is quantized by the following equation:

$$I_i = \arg\min_i \left|\hat{W} - g\right| \tag{5}$$

$$\tilde{W} = \{g[I_0],\ g[I_1],\ \ldots,\ g[I_i]\} \tag{6}$$

$$Q(W) = \alpha \times sign(\tilde{W}) \tag{7}$$

Eq. (5) selects the nearest quantization values by calculating the distance between the weight and the grid, where $0 \leq i < 2^{b-1}$, in which i is the number stored in the grid. Eq. (6) selects the quantization values from the grid following the order where the quantization interval is closest. Eq. (7) applies the *sign* function and multiplies it with α to generate the final quantization value. Note that b-bit quantization can represent 2^b distinct values. Since the *sign* function is used in Eq. (7), the bit size for the weight is smaller by one bit. For example, if 3-bit is used for weight quantization in APoT [20], then the grid generated is $\left\{0, 2^{-1}, 2^{-2}, 1\right\}$ since $k = 2$ and $n = 1$. PoT [31] also generates the same grid. However, for 2-bit quantization, $k = 1$ and $n = 1$, so grid [0, 1] is generated, along with $\alpha \times$ [1, -1, -0, +0]. Note that -0 and +0 are assigned the same value: zero; this implies that PoT [31] and APoT [20] are equivalent to uniform quantization at the 2-bit level. This restricts the accuracy of both quantization techniques becuase the grid does not closely follow a non-uniform quantization. We have access to two techniques, PoT [31] and APoT [20], because we approach a 2-bit weight grid. Thus, uniform quantization expressed in 2-bit can be newly expressed as non-uniform quantization by the following method.

B. THE PROPOSED NEW NON-ZERO GRID FORMULATION

A new grid formulation is proposed to allow non-uniform, PoT weight, 2-bit quantization. The grid we propose is obtained as follows:

$$Q^{w}(\alpha, B) = \alpha\ \times g \tag{8}$$

where $g = \left[1, 2^{-B\times Z}\right], B = b-1, Z > 0$. The α is the clipping value, b is the bit size, and B = b - 1 is satisfied. Z is a parameter consisting of integers. We can also see that the largest value in the grid for APoT [20] and PoT [31] (i.e., 1) is fixed to the clipping value. Therefore, we also use 1 as the clipping value by default in the proposed equation, and use power-of-two for the remaining values. In addition, for 2-bit quantization, the grid stores 2^B values. So, the value of 1 in the grid becomes the clipping range by multiplying it with the clipping value, and the value of the second term is projected as the value nearest 0 through the power-of-two. For example, when α is 3 and Z is 2, then 3 × $[1, 2^{-1\times 2}]$ is generated according to Eq. (8). Therefore, we can see that quantized values $[-3, -0.75, 0.75, 3]$ are non-uniform with clipping range -3 to 3. Referring to the illustration in Figure 2b, we can see that the proposed grid closely resembles a non-uniform quantization, while the existing PoT [31] and APoT [20] follow uniform quantization.

TABLE 1. Check the quantization values projected according to the Z value. In the proposed grid, we can see that the higher Z is, the nearest it is projected to the peak area. In the grid, 1 is fixed as the clipping value.

Z	grid → $[1, 2^{-B\times Z}]$
1	[1, 0.5]
2	[1, 0.25]
4	[1, 0.0625]
10	[1, 9.765625e-04]
20	[1, 9.5367431640625e-07]

C. CONFIGURING PARAMETER Z

In our proposed grid, Z is a configurable parameter. As Z increases, the PoT term $(2^{-B\times Z})$ decreases. Referring to Table 1, we see that when Z is 4, the second term in the grid is expressed as 2^{-4}, and when Z is 10, we see that it is expressed as 2^{-10}. This means that as Z increases, the quantization value becomes much closer to the mean area. If the quantization value is projected too close to the mean area, the gap between the two quantization levels becomes very narrow (see Figure 3). This results in a rigid resolution problem [20] in which the model cannot properly judge the image during training because the resolution is increased but the representation is distorted. Therefore, it is important to choose an appropriate Z value. In this paper, we can select various values of Z and find the optimal Z through experiments.

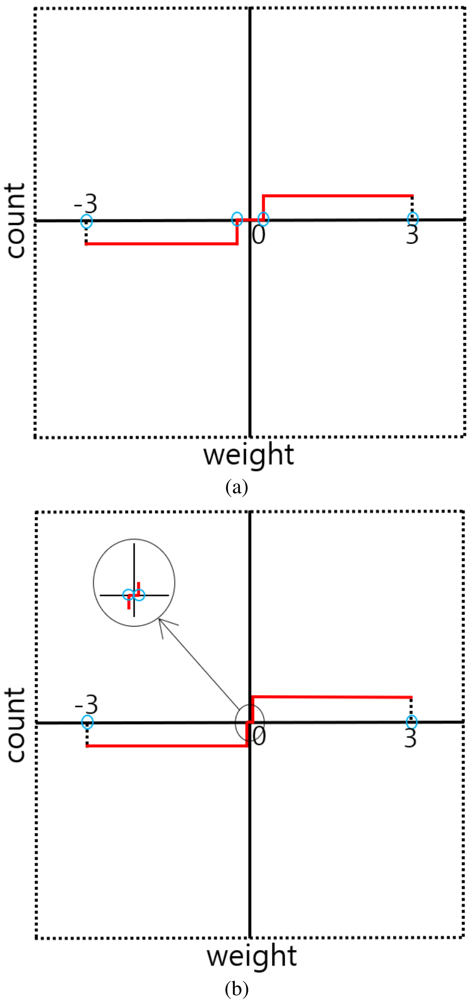
D. TUNING THE CLIPPING VALUE

When applying quantization to a CNN, the appropriate Clipping value can help the model train by selecting quantization values from the optimal range. However, Using a large Clipping value, it cannot select the optimal quantization value because the value is selected over a wide range. And the small Clipping values do not maintain a ideal normal distribution because they only select quantization values around the mean area. In addition, the clipping value used when quantizing each layer is not optimal and can be adjusted appropriately. Therefore, APoT [20] tunes the clipping value with the reparameterized clipping function (RCF) to find the optimal value. RCF calculates new clipping value after calculating gradient in Backpropagation during training. RCF proposed by APoT [20] is as follow:

$$\frac{\partial L}{\partial \alpha_w} = \frac{\partial L}{\partial O_i}\frac{\partial O_i}{\partial Q(W)}\frac{\partial Q(W)}{\partial \alpha_w} \tag{9}$$

$$\tilde{\alpha_w} = \alpha_w - \eta\frac{\partial L}{\partial \alpha_w} \tag{10}$$

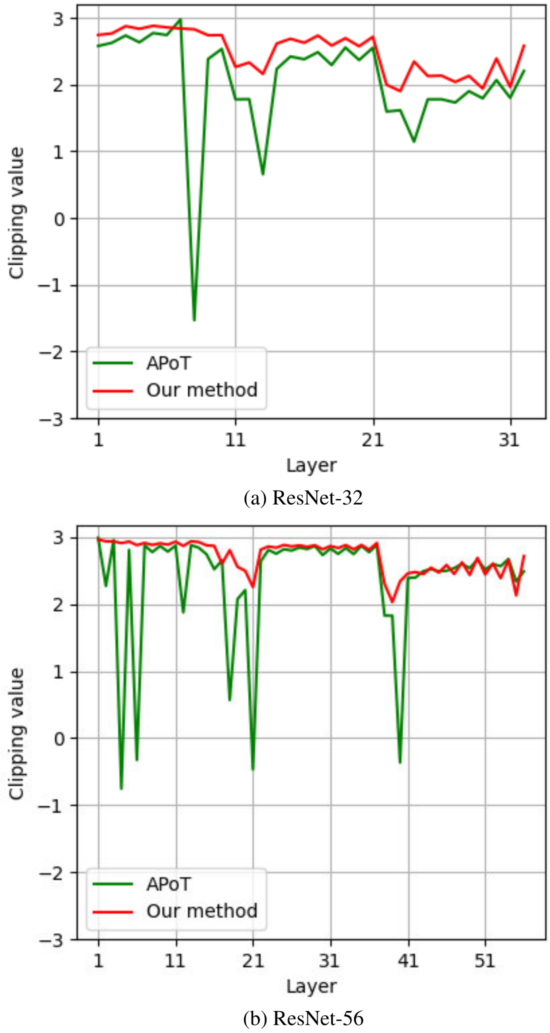
Eq. (9) calculates gradient of clipping value (α_w) for each layer, where L is a loss value, O_i is an output value for each layer, and Q(W) is a quantized weight. And, Eq. (10) updates the new clipping value ($\tilde{\alpha_w}$). Initially, for each layer, select Clipping value from a range that is usually not large. Therefore, the model can select quantization values from the optimal range. However, the clipping value tuned with a RCF in 2-bit level is not optimal. A small value is already selected from the first epoch (see Figure 4) where each layer is projected as a quantization value in a fluctuating range



(a)

(b)

FIGURE 3. The rigid resolution problem [20]: comparing the distances of the two quantization values in the mean area, we see that (b) is much closer than (a). Note that the small blue circles represent the quantization value.



(a) ResNet-32

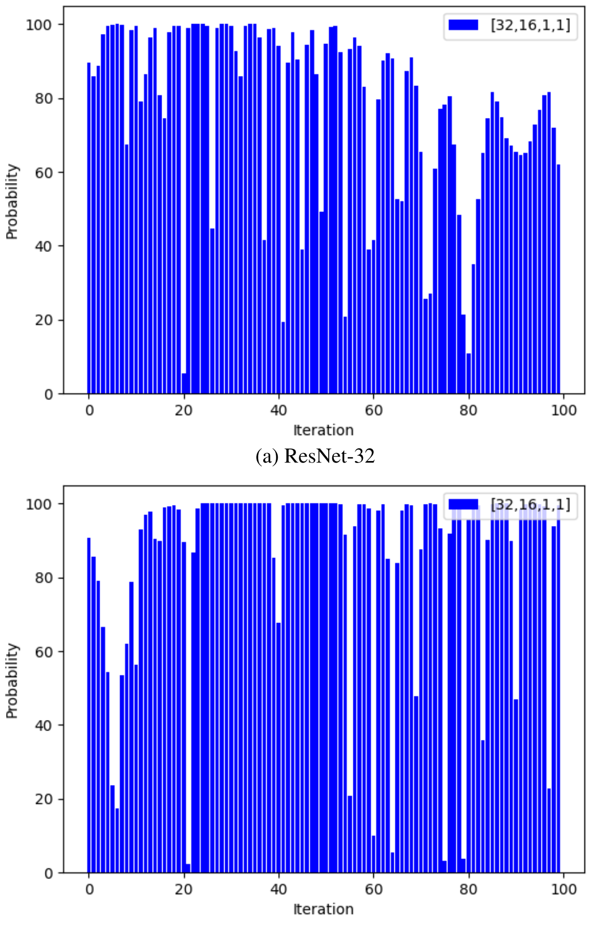
(b) ResNet-56

FIGURE 4. Training ResNet-32 and ResNet-56: (a) 2-bit APoT [20] chooses small and fluctuating clipping values; (b) our grid chooses clipping values with a larger range, and does not fluctuate across different layers.

rather than a smooth one. A small clipping range restricts the tuning process because there could be other unexplored values during the quantization step, resulting in a non-optimal range. In contrast, our proposed grid does not select a small Clipping value and can select an optimal quantization value by selecting a stable range from different layer.

E. THE PROBLEM OF THE PRESENCE OF ZEROS

The existing PoT techniques [20], [31] generate quantized values that may contain a signed zero (+0 and −0). The presence of zero in quantized weights can significantly affect the training process. For instance, 1 × 1 convolution is widely used in various CNN models (e.g., ResNet [9], GoogleNet [25]). The quantization of the weight may generate 0, which is multiplied with the input during the 1 × 1 convolution process. This can affect the training results, because all the output is computed as 0, which is more serious for low-bit quantization (e.g., 2-bit). For example, in 4-bit quantization, 2 out of 2^4 values are zeros after quantization, and the probability is 12.5%. For 3-bit quantization, this becomes 2 out of 2^3, and the probability is 25%, which is still acceptable. However, when we look at 2-bit quantization proposed in APoT [20], it is 2 out of 2^2 (a 50% probability). When many zeros are generated in the 1 × 1 output, the next layer will also be affected, and the model is likely to be trained in the wrong direction. Referring to Figure 5, more than 50% of the distributions in ResNet-32 and ResNet-56 after applying 2-bit APoT [20] quantization are zeros. This problem is also further revealed in Backpropagation's Clipping value selection. The third term $(\frac{\partial Q(W)}{\partial \alpha_w})$ in Eq. (9) is defined in APoT [20], and the gradient is calculated from each output value and the loss value. However, if the output value contains many zeros, the calculation of the gradient is greatly affected. Figure 6 shows the gradients of the quantized weights for each layer's output value. For APoT [20], we can clearly see some extremely large gradients generated due to division by zero. However, our method uses different non-zero values in 2-bit quantization instead of signed zeros, thus



(a) ResNet-32

(b) ResNet-56

FIGURE 5. The distributions of 0 at 1 × 1 weight with at size [1], [1], [16], [32] on ResNet models (e.g., ResNet-32, -56), when trained with mini batches. Note that [output channel, input channel, weight, weight].

avoiding the problem of division by zero; this allows the model to train normally.

F. TRAINING PROCESS

Algorithm 1 shows the application of 2-bit non-uniform quantization during training using the PoT system. Note that our method applies only to weights during training. First, In forward, the normalized weight obtained by the mean (μ) and standard (σ) deviation of weights is clipped in a certain range (lines 3 and 4). We then use our method to obtain quantization the values. Since our method is used in 2-bit, it is fixed to the grid with two values. In addition, the distance from the mean area varies depending on Z, and the grid can be set by selecting the appropriate Z (line 5). The closest distance can be expressed as an index by calculating the distance between grid and weight (line 6). The index is expressed as 0 and 1. Finally, a quantization value may be selected from the grid for each index, and a final quantization value may be selected through a sign function (line 7 and 8). This calculated quantization value is multiply with Input (line 9 and 10), also the quantization value calculated using our method is also composed of power-of-two, so fast computation speed can be achieved. Backward updates the weight and clipping value through gradient every training (line 13 and 14). In particular, since loss is affected by quantization weights and clipping values, gradients can be calculated with each other (line 11 and 12). Here, weight propagates the gradient (e.g., $\frac{\partial Q(W)}{\partial \hat{W}} = 1$) using the Straight-Through-Estimator (STE) [2] because the gradient for the function that converts it to the quantization value is zero. Gradient calculations for the clipping value are described in Section III-D.

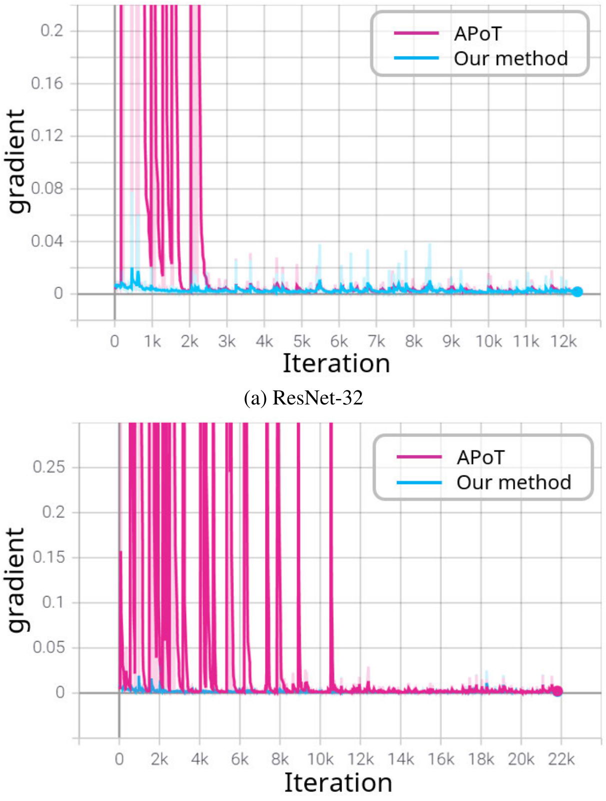
Algorithm 1 Forward and Backward Algorithm With the Proposed Method

```
Forward:
1: The bit-width for weight is applied one bit lower.
2: Input data: $x$,  Floating Point Weight: $W$,  bit: $b$, learning rate: $\eta$
3: Weight Normalized: $W_{norm} = \frac{W-\mu}{\sigma}$
4: Weight clipping: $\hat{W} = clip(\frac{W_{norm}}{\alpha}, -1, 1)$
5: Non-zero grid(Ours): $Q^{w}(\alpha, B) = \alpha \times g$ where $g = \left[1, 2^{-B\times Z}\right], B = b-1, Z > 0$
6: Select nearest quantization value index: $Idx_i = min \left|\hat{W} - g\right|_{idx}$ where $\left(0 \leq i < 2^{b-1}\right)$
7: Select quantization value from grid: $\tilde{W} = \{g[Idx_0],\ g[Idx_1],\ \ldots,\ g[Idx_i]\}$
8: Final quantization value: $Q(W) = \alpha \times sign(\tilde{W})$
9: Convolution: $Conv(x, Q(W)) = x * Q(W)$
10: Output: $F(Conv(x, Q(W)))$ where $F = Relu$ activation function
Backward:
11: Compute gradient using STE: $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Q(W)} \frac{\partial Q(W)}{\partial \hat{W}} \frac{\partial \hat{W}}{\partial W}$
12: Compute gradient of clipping value: $\frac{\partial L}{\partial \alpha_w} = \frac{\partial L}{\partial O_i} \frac{\partial O_i}{\partial Q(W)} \frac{\partial Q(W)}{\partial \alpha_w}$
13: Update the weight parameter: $W_{new} = W - \eta \frac{\partial L}{\partial W}$
14: Update the clipping value: $\tilde{\alpha_w} = \alpha_w - \eta \frac{\partial L}{\partial \alpha_w}$
```

IV. EXPERIMENTS

A. EXPERIMENT DETAILS

The proposed idea was experimentally verified with the CIFAR-100, and CIFAR-10 [18] datasets on four CNN models: ResNet-20, ResNet-32, ResNet-56 and ResNet-110 [9]. We trained the models with 2-bit quantization by applying the proposed grid to the weight. Quantization of the activation function followed the method in [20]. If the quantization bit size is 2 then b for the weight is set to 1. On the other hand, the sign function in Eq. (7) applies only to weights, so quantization b for activation function is applied as is. In addition, parameter Z in Eq. (8) was configured at $Z = 1, 2, 4, 10, 20$ in order to observe its effect on training accuracy. The clipping value was set to weight = 3.0 and activation = 8.0, following [20]. CIFAR-100 was trained over 300 epochs using ResNet-32, ResNet-56, and ResNet-110; CIFAR-10 was trained over 300 epochs using ResNet-20 [9].



(a) ResNet-32

(b) ResNet-56

FIGURE 6. Gradient graphs of quantized weights for APoT [20] and the proposed method when $Z = 2$ in ResNet-32 and ResNet-56. (Note that the y-axis is $\frac{\partial O_i}{\partial Q(W)}$).

TABLE 2. Performance comparison: CIFAR-100.

Method	Precision	Accuracy(%)	FixOPS
	(W/A)	Top-1	
FP.(ResNet-32)	32/32	70.03	70.07M
LSQ [6]	2/2	65.92	5.21M
QKD [16]	2/2	66.40	5.21M
DAQ [15]	2/2	65.81	5.21M
EWGS [19]	2/2	66.77	5.21M
APoT [20]	2/2	67.41	3.24M
(Ours) $Z = 1$	2/2	67.64	3.24M
(Ours) $Z = 2$	2/2	**68.13**	3.24M
(Ours) $Z = 4$	2/2	67.85	3.24M
(Ours) $Z = 10$	2/2	67.67	3.24M
(Ours) $Z = 20$	2/2	67.24	3.24M
FP.(ResNet-56)	32/32	71.77	127.38M
LSQ [6]	2/2	66.60	9.40M
DAQ [15]	2/2	67.49	9.40M
EWGS [19]	2/2	68.58	9.40M
APoT [20]	2/2	69.06	5.47M
(Ours) $Z = 1$	2/2	69.64	5.47M
(Ours) $Z = 2$	2/2	**69.87**	5.47M
(Ours) $Z = 4$	2/2	69.75	5.47M
(Ours) $Z = 10$	2/2	69.51	5.47M
(Ours) $Z = 20$	2/2	68.81	5.47M

We set the batch size to 128 and the learning rate to 4e-2, with weight decay set to 1e-4 for both 32-bit and 2-bit. The proposed method is compared with APoT [20], EWGS [19], DAQ [15], LSQ [6] and QKD [16].

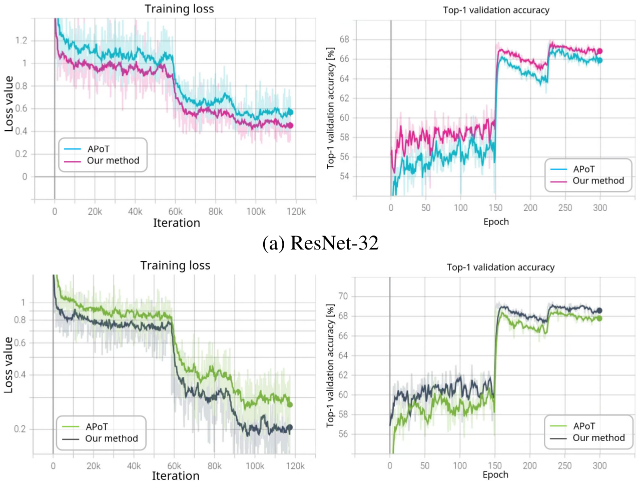
TABLE 3. Performance comparison: CIFAR-10.

Method	Precision	Accuracy(%)	FixOPS
	(W/A)	Top-1	
FP.(ResNet-20)	32/32	91.81	41.41M
LSQ [6]	2/2	88.24	3.39M
QKD [16]	2/2	90.50	3.39M
DAQ [15]	2/2	90.10	3.39M
EWGS [19]	2/2	90.59	3.39M
APoT [20]	2/2	90.24	2.12M
(Ours) $Z = 1$	2/2	90.45	2.12M
(Ours) $Z = 2$	2/2	**90.81**	2.12M
(Ours) $Z = 4$	2/2	90.50	2.12M
(Ours) $Z = 10$	2/2	90.45	2.12M
(Ours) $Z = 20$	2/2	89.59	2.12M

TABLE 4. Performance of ResNet-110 trained on CIFAR-100.

Method	Precision	Accuracy(%)			FixOPS
	(W/A)	Trial 1*	Trial 2	Trial 3	
FP.(ResNet-110)	32/32	72.09	-	-	256.33M
LSQ [6]	2/2	68.36	69.52	69.16	18.44M
DAQ [15]	2/2	70.32	69.90	70.13	18.44M
EWGS [19]	2/2	69.97	69.77	70.19	18.44M
APoT [20]	2/2	1.0	64.33	1.0	10.51M
(Ours) $Z = 1$	2/2	68.00	67.55	68.31	10.51M
(Ours) $Z = 2$	2/2	70.02	70.27	69.64	10.51M
(Ours) $Z = 4$	2/2	70.25	**70.57**	70.34	10.51M
(Ours) $Z = 10$	2/2	58.15	60.57	60.85	10.51M

* Out of 30 trials, only seven are successful. We only report three trials here a better readability.



(a) ResNet-32

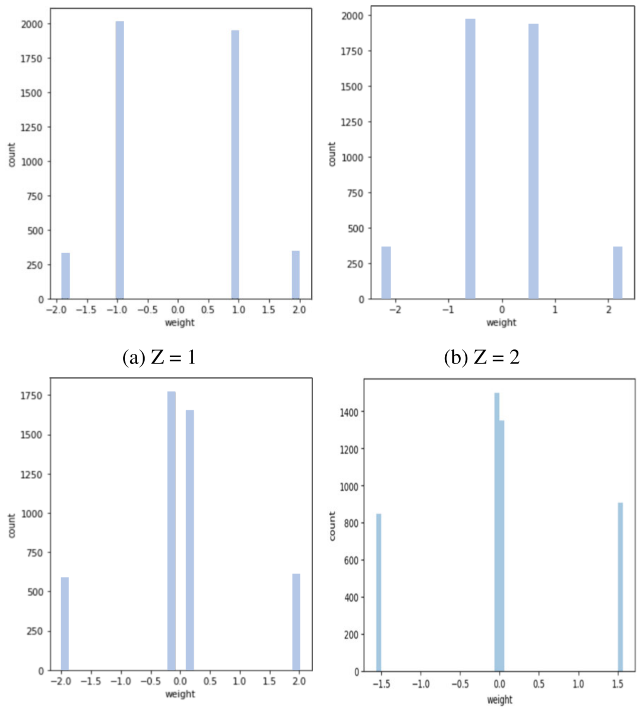
(b) ResNet-56

FIGURE 7. We compare the loss and Top-1 validation accuracy in ResNet-32 and ResNet-56. Our method confirmed that the calculated loss and accuracy are lower than in the existing method.

B. EXPERIMENTAL RESULTS

1) CIFAR-10 AND CIFAR-100

Table 2 compares the performance of 2-bit quantization using the proposed technique and APoT [20] and EWGS [19]. For ResNet-32 and ResNet-56 [9], we can see that $Z = 2([1, 2^{-2}])$ showed the best performance, achieving accuracies 0.72% and 0.81% higher than APoT [20], respectively. Moreover, $Z = 1([1, 2^{-1}])$ and $Z = 4([1, 2^{-4}])$ also showed performance increases of at least 0.23% and 0.44% on ResNet-32 and 0.58% and 0.69% on ResNet-56, respectively, above that of APoT [20]. This shows that we can improve performance with an appropriate Z value. In addition, EWGS [19] is trained



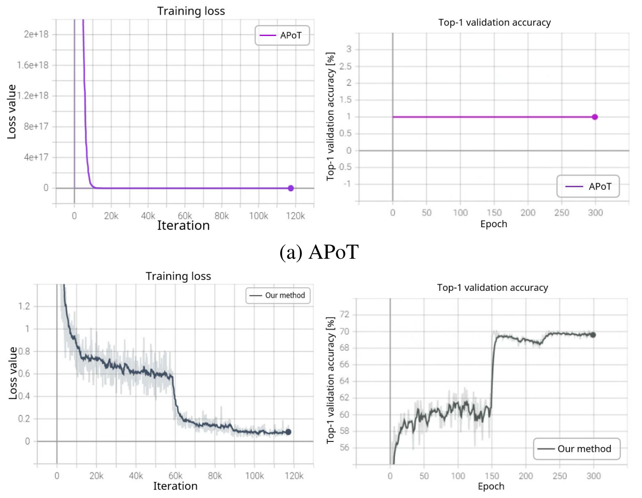
(a) Z = 1

(b) Z = 2

(c) Z = 4

(d) Z = 10

FIGURE 8. The distribution of weight quantization values according to various *Z* at 12th layer of ResNet model. Note that all the ResNet models used in our experiment (ResNet-20, -32, -56 and -110) show the same distribution. We can see that the quantization value becomes very close to the mean area when *Z* is 10.



(a) APoT

(b) Our Method

FIGURE 9. APoT on ResNet-110 shows that loss explodes during training, keeping the Top-1 validation accuracy at 1%. This prevents normal training.

with uniform quantization, so it is less accurate than the proposed method, non-uniform quantization. However, as *Z* increased (e.g., 10, 20), it gradually got closer to the mean area and the accuracy gradually decreased. Table 3 shows the performance on CIFAR-10 [18] with ResNet-20 [9] where similar performance was observed.

Table 4 shows the results of ResNet-110 trained on the CIFAR-100 dataset. The proposed method with Z = 4 is 0.4% more accurate than EWGS [19]. Note that out of the 30 trials of APoT [20] 2-bit quantization, only seven were successful; the failed trials showed extremely low accuracy (1%). When the training was successful, the proposed technique consistently outperformed APoT [20] by more than 5%. The accuracy achieved by the proposed technique (70.57%) is also very close to the original FP version (72.09%).

Besides accuracy, the number of fixed point operations (FixOPS) was calculated to estimate efficiency during inference. Following APoT [20], FixOPS was calculated as $\frac{m \times n}{64}$ (where m is the number of bits in the weight, and n is the number of bits in the activation). Our proposed technique maintained the same FixOPS as APoT [20] but with higher accuracy. This allows us to apply a CNN to constrained devices that have limited computational capabilities.

2) SELECTION THE OPTIMAL Z

The reduced representation of images due to quantization affects training, but it can be optimized by properly adjusting the distribution of weights. In our method, values of Z can represent different resolutions. Here we raise one question. *How do we find the optimal Z?* According to section III-C, in our method, the quantization value gets very close to the mean area as Z increases. In particular, In order to follow a non-uniform quantization with an ideal normal distribution, it is possible to achieve good performance by projecting the quantization values to be distributed at an appropriate distance from the mean area. We try to analyze various Z values in order to set the optimal Z in model training. In Figure 8, we applied various Z values to the ResNet models (e.g., ResNet-20, -32, -56 and -110) and expressed them graphically. The clipping value searches a wide range according to the RCF, and the distance from the mean area can be known according to Z. We found that when $Z = 1$, the projected quantization value is far from the peak, which does not follow the ideal normal distribution. In addition, it can be seen that as Z increases, it is very close to the peak, so it can cause the rigid resolution problem. In particular, two weight values that are very close to peak can produce very small results in convolution multiplication operations (e.g., $x * Q(W)$). Therefore, the gradient explodes in backpropagation, affecting training, and thus decreasing accuracy. Through our experiments on the selected datasets (CIFAR-10 and CIFAR-100), we found that $Z = 2$ or $Z = 4$ gave the best accuracy. For other datasets, a different optimal value for Z can be determined through experiments. If the two quantization values are projected not too close and not far from the mean area, the expressiveness of the model can be improved, which explains why $Z = 2$ or $Z = 4$ achieved better accuracy than $Z = 1$ or $Z = 10$. In particular, in ResNet-20, ResNet-32 and ResNet-56, when $Z = 2$ was set, the performance was high, and in ResNet-110, $Z = 4$ had the best performance. This allows us to select optimal Z value that are not large and small with an ideal normal distribution for each model.

TABLE 5. Synergy with EWGS [19] on CIFAR-100.

Method	Precision	Accuracy(%)	FixOPS
	(W/A)	Top-1	
FP.(ResNet-32)	32/32	70.03	70.07M
(Uniform) + STE [2]	2/2	66.64	5.21M
(Uniform) + EWGS [19]	2/2	66.77	5.21M
(Ours) + STE [2]	2/2	68.13	3.24M
(Ours) + EWGS [19]	2/2	**68.19**	3.24M
FP.(ResNet-56)	32/32	71.77	127.38M
(Uniform) + STE [2]	2/2	67.92	9.40M
(Uniform) + EWGS [19]	2/2	68.58	9.40M
(Ours) + STE [2]	2/2	69.87	5.47M
(Ours) + EWGS [19]	2/2	**70.13**	5.47M

TABLE 6. Synergy with EWGS [19] on CIFAR-10.

Method	Precision	Accuracy(%)	FixOPS
	(W/A)	Top-1	
FP.(ResNet-20)	32/32	91.81	41.41M
(Uniform) + STE [2]	2/2	90.23	3.39M
(Uniform) + EWGS [19]	2/2	90.59	3.39M
(Ours) + STE [2]	2/2	90.81	2.12M
(Ours) + EWGS [19]	2/2	**90.91**	2.12M

3) ANALYSIS OF TRAINING LOSS AND VALIDATION ACCURACY

This section analyzes the training loss and the Top-1 validation accuracy in each model (i.e., ResNet-32, ResNet-56 and ResNet-110), which is derived from the output of the training result after feedforward. In Section III-E, we showed that 1 × 1 convolution with zero may derive incorrect results in model training; the problem becomes more serious with deeper layers. From Figure 7, we confirmed that our method applied to ResNet-32 and ResNet-56 calculated a lower loss and higher validation accuracy than APoT [20], but the difference is not significant. However, the situation is different with ResNet-110, where the loss in APoT [20] exploded after a few iterations (e.g., Figure 9). We confirm that for ResNet-110, the loss has an accuracy of 1% per epoch when it explodes. In addition, these loss values seriously affects calculation of the gradient in backpropagation, and also results in an incorrect clipping value in Eq. (9). That explains why APoT [20] 2-bit quantization (see Table 4) failed to train correctly. Even for some cases where the model can be trained normally, it introduces a significant accuracy loss (64.33%). In contrast, our method allows the calculation to be derived correctly by removing zero from the grid, thus achieving significantly better accuracy (70.57%).

4) DISCUSSION

Quantization in deep learning models improves performance using various optimization methods. In particular, our method improves performance by quantizing only weights. Therefore, in this section, we would like to discuss that our method is not independent and can be combined with other quantization methods to create synergies. We applied our method only to weights in forward propagation, so other methods can be applied in the activation function or backpropagation process. For example, in most backpropagation process, the STE [2] method is usually used for quantization methods [6], [14], [30]. On the other hand, the EWGS [19] method optimizes by adjusting the gradient in backpropagation to improve performance over STE [2]. Therefore, EWGS [19] is not affected by uniform quantization or non-uniform quantization in forward propagation, and only optimizes with varying gradient, so accuracy improvements can be made. Table 5, 6 shows the results of applying our method to STE and EWGS [19]. First, existing EWGS [19] uses uniform quantization in forward propagation and shows that it achieves higher accuracy than when using STE [2] in each model. In addition, we conducted a test by applying EWGS [19] to the Z value that achieved the highest accuracy of our proposed method. When we applied our method to EWGS [19], we can see that the performance was improved compared to that of STE [2]. It is also better than the accuracy of only using EWGS [19]. Note that we only optimized performance for backpropagation, so FixOPS will not change. Therefore, we can see that our method has a synergistic with other method.

V. CONCLUSION

The PoT [31] and APoT [20] technique performs 2-bit quantization in a uniform manner that is not optimal. Moreover, the existing 2-bit APoT [20] quantization could generate many zeros in 1 × 1 convolution output, degrading training accuracy. The proposed new non-zero grid formulation turns 2-bit APoT [20] into non-uniform quantization, which closely resembles a normal distribution. It also represents the zeros with small quantized values, completely avoiding explosion in the gradient calculation. We conduct experiments on various Z and show that the optimal Z can be found based on the experimental results, and the proposed new grid can achieve higher accuracy than APoT [20]. This technique is also applicable to PoT [31] as it shares the same problems as in APoT [20]. In future, we expect to extend the proposed technique to other 2-bit quantization schemes.

REFERENCES

[1] C. Baskin, N. Liss, E. Schwartz, E. Zheltonozhskii, R. Giryes, A. M. Bronstein, and A. Mendelson, "UNIQ: Uniform noise injection for non-uniform quantization of neural networks," *ACM Trans. Comput. Syst.*, vol. 37, nos. 1–4, pp. 1–15, Nov. 2019.

[2] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.

[3] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 129–146.

[4] A. Capotondi, M. Rusci, M. Fariselli, and L. Benini, "CMix-NN: Mixed low-precision CNN library for memory-constrained edge devices," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 5, pp. 871–875, May 2020.

[5] J. Choi, S. Venkataramani, V. V. Srinivasan, K. Gopalakrishnan, Z. Wang, and P. Chuang, "Accurate and efficient 2-bit quantized neural networks," in *Proc. Mach. Learn. Syst.*, vol. 1, 2019, pp. 348–359.

[6] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," 2019, *arXiv:1902.08153*.

[7] R. Gong, X. Liu, S. Jiang, T. Li, P. Hu, J. Lin, F. Yu, and J. Yan, "Differentiable soft quantization: Bridging full-precision and low-bit neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4852–4861.

[8] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[10] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[11] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[13] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.

[14] S. Jung, C. Son, S. Lee, J. Son, J.-J. Han, Y. Kwak, S. J. Hwang, and C. Choi, "Learning to quantize deep networks by optimizing quantization intervals with task loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4350–4359.

[15] D. Kim, J. Lee, and B. Ham, "Distance-aware quantization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5271–5280.

[16] J. Kim, Y. Bhalgat, J. Lee, C. Patel, and N. Kwak, "QKD: Quantization-aware knowledge distillation," 2019, *arXiv:1911.12491*.

[17] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," 2018, *arXiv:1806.08342*.

[18] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[19] J. Lee, D. Kim, and B. Ham, "Network quantization with element-wise gradient scaling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6448–6457.

[20] Y. Li, X. Dong, and W. Wang, "Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–15.

[21] Z. Liao, R. Couillet, and M. W. Mahoney, "Sparse quantized spectral clustering," 2020, *arXiv:2010.01376*.

[22] E. Park and S. Yoo, "PROFIT: A novel training method for sub-4-bit MobileNet models," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 430–446.

[23] H. Qin, R. Gong, X. Liu, M. Shen, Z. Wei, F. Yu, and J. Song, "Forward and backward information retention for accurate binary neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2250–2259.

[24] J.-C. See, H.-F. Ng, H.-K. Tan, J.-J. Chang, W.-K. Lee, and S. O. Hwang, "DoubleQExt: Hardware and memory efficient CNN through two levels of quantization," *IEEE Access*, vol. 9, pp. 169082–169091, 2021.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[26] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, "Integer quantization for deep learning inference: Principles and empirical evaluation," 2020, *arXiv:2004.09602*.

[27] K. Yamamoto, "Learnable companding quantization for accurate low-bit neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5029–5038.

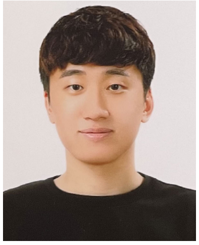
[28] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, and X.-S. Hua, "Quantization networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7308–7316.

[29] Z. Yang, Y. Wang, K. Han, C. Xu, C. Xu, D. Tao, and C. Xu, "Searching for low-bit weights in quantized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4091–4102.

[30] D. Zhang, J. Yang, D. Ye, and G. Hua, "LQ-Nets: Learned quantization for highly accurate and compact deep neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 365–382.

[31] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless CNNs with low-precision weights," 2017, *arXiv:1702.03044*.

[32] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," 2016, *arXiv:1606.06160*.



YOUNG MIN KIM received the B.S. degree in computer engineering from Hongik University, South Korea, in 2021. He is currently pursuing the M.S. degree with Gachon University, South Korea. His research interests include deep learning and cyber security.

KYUNGHYUN HAN received the B.S. and M.S. degrees in computer engineering from Hongik University, South Korea, in 2015 and 2017, respectively. He is currently a Researcher with the Information Security and Machine Learning Laboratory, Hongik University. His research interests include cyber security, machine learning, and blockchain.

WAI-KONG LEE (Member, IEEE) received the B.Eng. degree in electronics and the M.Sc. degree from Multimedia University, in 2006 and 2009, respectively, and the Ph.D. degree in engineering from Universiti Tunku Abdul Rahman, Malaysia, in 2018. He was a Visiting Scholar with Carleton University, Canada, in 2017; Feng Chia University, Taiwan, in 2016 and 2018; and OTH Regensburg, Germany, in 2015, 2018, and 2019. Prior to joining academia, he worked in several multi-national companies, including Agilent Technologies, Malaysia, as a Research and Development Engineer. He is currently a Postdoctoral Researcher with Gachon University, South Korea. His research interests include cryptography, numerical algorithms, GPU computing, the Internet of Things, and energy harvesting.

HYUNG JIN CHANG (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, South Korea. He was a Postdoctoral Researcher with the Imperial Computer Vision and Learning Laboratory and the Personal Robotics Laboratory, Department of Electrical and Electronic Engineering, Imperial College London. He is currently a Lecturer (equivalent to an Assistant Professor) with the School of Computer Science, University of Birmingham. His current research interests include human understanding through visual data analysis, including human/hand pose estimation, eye gaze tracking, articulated structure learning, human–robot interaction, 6-D object pose tracking, human action understanding, and user modeling.

SEONG OUN HWANG (Senior Member, IEEE) received the B.S. degree in mathematics from Seoul National University, in 1993, the M.S. degree in information and communications engineering from the Pohang University of Science and Technology, in 1998, and the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology, South Korea, in 2004. He was a Software Engineer with LG-CNS Systems Inc., from 1994 to 1996. He was a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), from 1998 to 2007, and a Professor with the Department of Software and Communications Engineering, Hongik University, from 2008 to 2019. He is currently a Professor with the Department of Computer Engineering, Gachon University. His research interests include cryptography, cybersecurity, and artificial intelligence. He is an Editor of *ETRI Journal*.