


# Towards Accurate Network Quantization with Equivalent Smooth Regularizer

Kirill Solodskikh<sup>1\*</sup>, Vladimir Chikin<sup>1\*</sup>, Ruslan Aydarkhanov<sup>1\*</sup>, Dehua Song<sup>1</sup>,  
Irina Zhelavskaya<sup>2</sup> , and Jiansheng Wei<sup>1</sup>

<sup>1</sup> Huawei Noah's Ark Lab

{solodskikh.kirill1, vladimir.chikin, ruslan.aydarkhanov,  
dehua.song, weijiansheng}@huawei.com

<sup>2</sup> Skolkovo Institute of Science and Technology (Skoltech)  
irina.zhelavskaya@skolkovotech.ru

**Abstract.** Neural network quantization techniques have been a prevailing way to reduce the inference time and storage cost of full-precision models for mobile devices. However, they still suffer from accuracy degradation due to inappropriate gradients in the optimization phase, especially for low-bit precision network and low-level vision tasks. To alleviate this issue, this paper defines a family of equivalent smooth regularizers for neural network quantization, named as SQR, which represents the equivalent of actual quantization error. Based on the definition, we propose a novel QSin regularizer as an instance to evaluate the performance of SQR, and also build up an algorithm to train the network for integer weight and activation. Extensive experimental results on classification and SR tasks reveal that the proposed method achieves higher accuracy than other prominent quantization approaches. Especially for SR task, our method alleviates the plaid artifacts effectively for quantized networks in terms of visual quality.

**Keywords:** network quantization, smooth regularizer, equivalence, gradient, low-level vision task

## 1 Introduction

Deep Neural Network (DNN) has dramatically boosted the performance of various practical tasks due to its strong representation capacity, for example, image classification [19], image translation [7] and speech recognition [14]. Along with the requirements of deploying DNN into mobile devices increasing, it has been necessary to develop low-latency, efficient and compact networks. Recently, large amounts of approaches have been proposed to solve this problem, including network pruning [25, 11], quantization [18, 9] and adder neural network [2].

Network quantization is one of the most appealing way to reduce the inference latency, energy consumption and memory cost of neural networks. Since low-bit integer tensors (weight/activation) and integer arithmetics are employed

---

\* These authors contributed equally to this work.

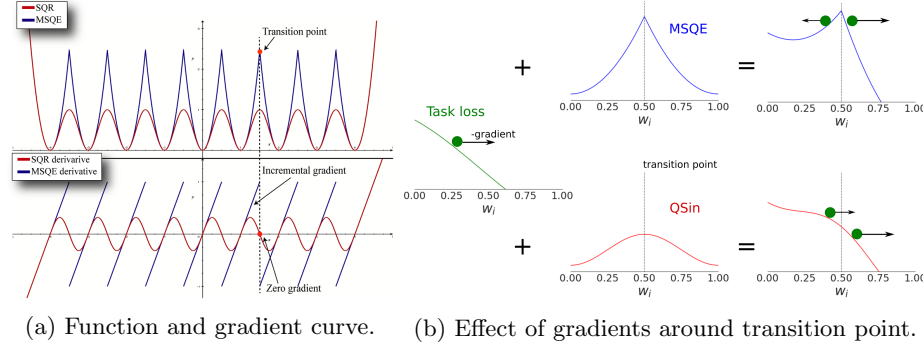


Fig. 1: The comparison of SQR and MSQE [4] regularizer. SQR is smooth everywhere instead of the unsmoothness of MSQE in each transition point, and represents the equivalent of actual quantization error, which allows to obtain better gradient behavior in neighborhood of transition points.

in quantized network, the model size and computation consumption could be decreased significantly. The advantages of quantization network on low precision hardware has been demonstrated with multiple systems [10,18], but it still suffers from accuracy degradation due to inappropriate gradients in the optimization phase, especially for low-bit precision network and low-level vision tasks.

Minimization of objective function for quantized neural networks in general case is a hard optimization problem since the gradient is either zero or undefined. The prominent Quantization Aware Training (QAT) algorithms [16,5,31] usually adopted the Straight-Through Estimator (STE) [1,18,3] strategy to solve this gradient issue, which approximates the gradient of the rounding operator as 1. Although several further approaches [28,22] have been proposed to refine the gradient approximation, such kind of algorithms still suffer from the gradient error, especially for lower-bit quantization. Another alternative way is to train the network with regularizer [4,8] of quantization error to generate the quantized model, where gradients from accuracy loss could be propagated effectively.

Unfortunately, the gradient of the most conventional regularizer for quantization, mean square quantization error (MSQE) [4], is undefined in each transition point which is illustrated in Fig. 1. It hinders the quantization error from being propagated to the weights of each layer. What's worse, steep gradients around transition points would dominate the direction of update step for the joint objective, which is prone to reach the closest grid point instead of the optimal point of the accuracy loss (see Fig. 1(b)). SinReQ [8] explored a smooth regularizer for quantization to alleviate the gradient issue. However, its variation trend outside of quantization segment is quite different from the actual quantization error, which results in high clamping error and significant accuracy degradation. To reduce the quantization error, the prime regularizer should not only be smooth everywhere but also represent the equivalent of actual quantization error. Hence,

this paper defined a family of equivalent smooth regularizers for neural network quantization, called SQR. Based on this definition, we proposed a novel QSin regularizer as an instance to evaluate the performance of SQR, and also built up an algorithm to train the network for integer weight and activation. The quantization error could be reflected effectively and propagated to weights smoothly. To evaluate the performance and generality of our approach, extensive experiments on classification and SR tasks were conducted. The results reveal that the proposed method achieves higher accuracy than other prominent approaches. Especially for SR task, our method alleviates the plaid artifacts effectively for quantized networks in terms of visual quality, since the pixel value regression is more easily affected by the quantization error.

The main contributions of this paper are threefold:

1. We defined a novel family of equivalent smooth regularizer for quantization and analyzed its properties theoretically.
2. We proposed a novel QSin regularizer belonging to SQR and built up a general algorithm to train the network with weight and activation quantization for any bit-width. It is important to note that our regularizer allows to train quantized network without weights rounding comparing with the most quantization algorithms.
3. Our method works stable and achieves state-of-the-art results on wide spectra of computer vision tasks, including image classification and super-resolution tasks. What's more, our method could alleviate the plaid artifacts effectively for quantized networks.

## 2 Related Works

Quantization is one of the most important technique for model compression, which attracts many researchers to investigate it. In the last decade, many quantization approaches were proposed to improve the performance of quantization network. According to the criterion of whether training the quantized network or not, the quantization methods could be roughly divided into two categories: Post Training Quantization (PTQ) [23] and Quantization Aware Training (QAT) [16].

*Post training quantization.* Post training quantization algorithms aim at quantizing the trained full precision network into low-precision one with compact unlabeled calibration set or even without any data. Nowadays such algorithms have achieved significant progresses in quantization of classification networks. Nagel *et al.* [23] employed the minimum and maximum values of weights to define weight quantization parameters and moving average of minimum and maximum values to define activations quantization parameters, respectively. Based on this method, Hubara *et al.* [17] further explored tuning batch normalization layer and boosted the performance of quantized network significantly. Such kind of methods are attractive because of quick implementation, setup and application but usually lead to accuracy drop comparing with full precision networks.

*Quantization aware training* To reduce the accuracy drop of quantized network, numerous quantization aware training algorithms [5,31,3] have been proposed, which utilize stochastic gradient descent technique with quantized weights and activations on forward pass stage but full precision weights on backward pass procedure. Since the gradient of round function is zero or undefined everywhere, Straight Through Estimator [1] has been proposed to propagate the derivative. LSQ [9] method was proposed to further improve the accuracy via learnable step size. More quantization parameters were suggested to learn with end-to-end optimization manner [30]. To alleviate the gradient error problem, various approaches (DSQ [12], PACT[3], QuantNoise [28], *etc.* ) were introduced with progressive way to train the quantization network. Unfortunately, these methods still cannot solve this issue thoroughly.

*Quantization through regularization* Another alternative way of generating quantized network is to train the network with regularizer [4,8,24] of quantization error, where gradients from accuracy loss could be propagated smoothly. Choi *et al.* [4] firstly proposed regularizer term of mean squared quantization error (MSQE) for weight and activation quantization. However, the gradients of MSQE regularizer in transition points are undefined, which prevents the quantization error from propagating. SinReQ [8] explored periodic functions as regularizer for weight quantization. Unfortunately, its variation trend outside of quantization segment is quite different from the actual quantization error, which results in high clamping error and significant accuracy degradation.

### 3 Preliminaries and Motivation

Here we firstly briefly introduce the basic principles of neural network quantization, and then discuss the difficulty of quantization network training.

We consider neural network  $\mathcal{F}(\mathbf{W}, \mathbf{X})$  as an ordered graph with  $n$  layers, and each layer corresponds to the function  $\mathcal{F}_i(\mathbf{W}_i, \mathbf{A}_i)$ , where  $\mathbf{W}_i$  and  $\mathbf{A}_i$  denote the parameter tensor and input features of the  $i$ -th layer, respectively. For convenience, we denote the set  $\{\mathbf{W}_i\}_{i=1}^N$  as  $\mathbf{W}$  and an input data tensor with  $\mathbf{X}$ . The  $\mathbf{X}$  could be simply modeled by continuous distribution  $\xi$ .

*Quantization* Network quantization aims at reducing the precision of both parameters and activations with minimal impact on the representation ability of full-precision models. Firstly, we need to define a function which can quantize real value set into a finite set. The conventional uniform quantization function  $\mathcal{Q}_U$  is defined as follows:

$$\mathcal{Q}_U(x) = \begin{cases} \lfloor x \rfloor, & \text{if } r_b \leq x \leq r_t, \\ r_b, & \text{if } x < r_b, \\ r_t, & \text{if } x > r_t, \end{cases} \quad (1)$$

where  $x$  is the input of function.  $r_b$  and  $r_t$  denote the minimum and maximum of clipping range, respectively.  $\lfloor \cdot \rfloor$  is the round-to-nearest operator. To quantize

a real value into an integer, we usually need three quantization parameters: scale factor  $s$ , zero-point and bit-width. For convenience, here we employ the symmetric uniform quantization to analyze problems. Then, equation 2 could be utilized to quantize the weights and activations.

$$q = \mathcal{Q}_U\left(\frac{x}{s}\right), \quad (2)$$

where  $x$  is the real value, and  $q$  is the quantized integer. When both weights and activations are quantized into integer, computation could be executed with an integer-arithmetic way, which results in significant acceleration on hardware. The full-precision layer  $\mathcal{F}_i(\mathbf{W}_i, \mathbf{A}_i)$  is replaced by the quantized layer:

$$\mathcal{F}_i^q = s_{w_i} s_{a_i} \mathcal{F}_i\left(\mathcal{Q}_U\left(\frac{\mathbf{W}_i}{s_{w_i}}\right), \mathcal{Q}_U\left(\frac{\mathbf{A}_i}{s_{a_i}}\right)\right), \quad (3)$$

where  $s_{w_i}$  and  $s_{a_i}$  denote the scale factor of weight and activation of the  $i$ -th layer, respectively.

*Mean Squared Quantization Error* Considering network quantization problem as an optimization problem with special constraints, Choi *et al.* [4] proposed the mean-squared-quantization-error (MSQE) as regularization term for weight and activation quantization, which is defined as follows:

$$\text{MSQE}(\mathbf{V}; s) = \frac{1}{K} \sum_{x_j \in \mathbf{V}} |x_j - s \cdot \mathcal{Q}_U\left(\frac{x_j}{s}\right)|^2, \quad (4)$$

where  $V$  denotes the input tensor with  $K$  components. It reflects the error between original full-precision value and its quantized value. To constraint the weights and activations of the whole network, the regularizer term should contains the quantization error of each layer. The complete MSQE regularizer terms for weights ( $\text{MSQE}_w$ ) and activations ( $\text{MSQE}_a$ ) are defined as Eq. 5.

$$\text{MSQE}_w = \frac{1}{N} \sum_{i=1}^N \text{MSQE}(\mathbf{W}_i, s_{w_i}), \quad \text{MSQE}_a = \frac{1}{N} \sum_{i=1}^N \text{MSQE}(\mathbf{A}_i, s_{a_i}). \quad (5)$$

Let  $\mathcal{L}$  is the original objective function of full-precision neural network  $\mathcal{F}(\mathbf{W}, \mathbf{X})$ . Then, we consider the quantization network training issue as a *neural network optimization problem with quantization constraints*:

$$\begin{cases} \mathbb{E}[\mathcal{L}(\mathcal{F}(\mathbf{W}, \mathbf{X}))] \rightarrow \min, \\ \text{MSQE}_w < C_w, \\ \mathbb{E}[\text{MSQE}_a] < C_a. \end{cases} \quad (6)$$

where  $C_w$  and  $C_a$  denote the thresholds restricting the quantization error for weights and activations.  $\mathbb{E}[\cdot]$  is the expectation among the whole database. Theoretically, we can not effectively reach the minimization of Lagrange function 6

since MSQE is not smooth. From Fig. 1, we can see that the gradient of MSQE is undefined in each transition point, which limits the performance of quantization network. To address this issue, we define a class of Smooth Quantization Regularizers (SQR) which represents the equivalent of actual quantization error.

## 4 SQR: Equivalent Smooth Quantization Regularizer

Smooth property of regularizer is friendly to network optimization, which is helpful to solve the gradient problem. Here, we propose a family of smooth quantization regularizer to replace the MSQE regularizer, which represents the equivalent of actual quantization error and allows to obtain better gradient behavior in neighborhood of transition points.

### 4.1 Definition of SQR

From Eq. 4 and Fig. 1, we can observe that the MSQE regularizer is not a smooth function due to the quantization operator. To acquire the smooth regularizer for quantization, we should deal with the transition points carefully. Besides the smooth property, we also hope that the smooth regularizer could effectively reflect the trend of quantization error and preserve the same number of minimum with MSQE. Then, we can define the ideal smooth quantization regularizers (SQR) as follows. Here we abbreviate the quantization regularizer  $MSQE(x; s)$  as  $MSQE(x)$  for simplicity.

**Definition 1.** *With the same constant scale factor  $s$  with  $MSQE(x)$ , function  $\phi(x)$  is a Smooth Quantization Regularizer (SQR) for the uniform grid of integers with the segment  $[r_b, r_t]$  when it satisfies the following three properties:*

1) **Order preserving.** *Function  $\phi(x)$  preserves the order of  $MSQE(x)$ , i.e.:*

$$MSQE(x_1) \leq MSQE(x_2) \Leftrightarrow \phi(x_1) \leq \phi(x_2),$$

$$\forall x_1, x_2 \in [r_b, r_t] \text{ or } x_1, x_2 \in \mathbb{R} \setminus [r_b, r_t].$$

2) **Equivalence.** *There exists  $a, b \in \mathbb{R}$ , and  $0 < a < b$ , such that*

$$a MSQE(x) \leq \phi(x) \leq b MSQE(x), \forall x \in \mathbb{R}. \quad (7)$$

3) **Smoothness.**  $\phi(x) \in \mathcal{C}^2(\mathbb{R})$ .

where  $\mathcal{C}^2(\mathbb{R})$  denotes the twice differentiable function family for the domain of all real numbers.

According to the definition, we could further infer that SQRs are periodic within the domain of quantization segment  $[r_b, r_t]$ . In addition, SQRs could not only preserve the same minima points with MSQE, but also acquire the close asymptotic around the quantization grid points and at infinity. In other words,

for arbitrary SQR  $\phi$  and  $s > 0$ , there exists  $B > 0$  such that the following relation holds for  $MSQE(x; s) \rightarrow 0$ :

$$s^2 \phi\left(\frac{x}{s}\right) = B \cdot MSQE(x; s) + o(MSQE(x)). \quad (8)$$

These admirable characteristics guarantee that we could employ SQRs to replace the conventional MSQE with negligible relaxation.

Following the notation of MSQE, we extend the SQR for tensor  $\mathbf{X}$  with the average of all the components'  $\phi(x)$  values. Then, Lagrange function minimization of quantization network in the definition domain of parameters  $(\mathbf{W}, \mathbf{s}_w, \mathbf{s}_a)$  could be rewritten as follows:

$$\mathcal{L}_Q = \mathbb{E}[\mathcal{L}(\mathcal{F}(\mathbf{W}, \mathbf{X}))] + \lambda_w \mathcal{L}_w + \lambda_a \mathcal{L}_a \quad (9)$$

$$\mathcal{L}_w(\mathbf{W}; \mathbf{s}_w) = \frac{1}{N} \sum_{i=1}^N s_{w_i}^2 \phi(\mathbf{W}_i, s_{w_i}), \quad \mathcal{L}_a(\mathbf{A}; \mathbf{s}_a) = \frac{1}{N} \sum_{i=1}^N \mathbb{E}[s_{a_i}^2 \phi(\mathbf{A}_i, s_{a_i})]. \quad (10)$$

This objective function becomes smooth and amenable to optimize. According to the property defined in Eq. 7, it also effectively constrains the solution of network in the compact domain which belongs to the solution domain with MSQE regularizer. Indeed, if SQR  $\phi(x)$  is less than  $c \in \mathbb{R}$  in some domain  $x \in \Omega$ , therefore MSQE is less than  $ac$  for some  $a \in \mathbb{R}$ . This means that while we minimize SQR to zero then MSQE also converges to zero. More details and proofs could be seen in Appendix A.

## 4.2 QSin Regularizer

According to the definition of SQR, we proposed a novel smooth regularizer, QSin, to improve the performance of quantization network. The definition of QSin is showed in Eq. 11. QSin is a sinusoidal periodic function within the domain of quantization segment  $[r_b, r_t]$ , while it is a quadratic function beyond the quantization segment domain.

$$QSin(\mathbf{V}; s) = \frac{s^2}{K} \sum_{x_j \in \mathbf{V}} QSin_{on}\left(\frac{x_j}{s}\right), \quad (11)$$

where the  $QSin_{on}$  is defined as follows:

$$QSin_{on}(x) = \begin{cases} \sin^2(\pi x), & \text{if } r_b \leq x \leq r_t, \\ \pi^2(x - r_b)^2, & \text{if } x < r_b, \\ \pi^2(x - r_t)^2, & \text{if } x > r_t. \end{cases} \quad (12)$$

QSin is a twice differentiable function for the whole domain of definition. Its function curve and gradient curve with  $s = 1$  for scalar input are illustrated

in Fig. 1. QSin is smooth everywhere since there is no quantization operator, which is quite different from MSQE. As for quantization network optimization, the scale factor  $s$  usually needs to be optimized. Hence, we also compared the best solution of  $s$  for QSin and MSQE regularizer. As for the uniform quantization of random variable  $\xi$ , we randomly sampled  $M$  (*e.g.* 128) values from standard normal distribution, and then computed the quantization error  $MSQE(\xi; s)$  and  $QSin(\xi; s)$  for each scale value. The best solution of  $s$  from QSin is quite close to that from MSQE for various distributions of  $\xi$ . More details could be seen in Appendix B. Therefore, we can employ smooth QSin to replace the MSQE regularizer while preserving sufficient constraint for quantization error.

### 4.3 Quantization Network Optimization

Considering network quantization problem as an optimization problem with special constraints, we substitute Eq. 11 into Eq. 9 and acquire the final objective function. This method employs the full precision network  $\mathcal{F}(\mathbf{W}, \mathbf{X})$  without quantization operations on the forward pass during training stage, called Round Free (RF). Then, SGD technique could be utilized to optimize this objective function  $\mathcal{L}_Q$  of network directly. Inspired by LSQ [9] method, we set the scale factor of weights and activations as a learnable parameter to improve the performance of quantization network.  $s_a$  and  $w_a$  will be updated during each step of gradient descent to minimize the SQR. More details could be seen in Appendix B. During the validation or test stage, the quantized network will be computed according to Eq. 3 with quantization operations and integer arithmetics.

Although our novel regularizer could constraint the activation effectively, there still exists differences between constrained activations and quantized activations to some degree, especially for low-bit quantization. Hence, as for the extreme low-bit quantization, an optional way is to add the quantization operation Eq. 2 to the activations to alleviate the quantization error accumulation problem. The activations before getting through the quantization operation are utilized to calculate the QSin regularizer. In this scheme, STE [1] should be employed to propagate the gradients through round function  $Q_U$ .

As for the coefficient of regularizer  $\lambda_w$  and  $\lambda_a$ , we set them as a power of 10 to normalize the regularizer loss  $\mathcal{L}_a$  and  $\mathcal{L}_w$  for acquiring the same order with the main loss. Weight quantization is achieved with a progressive way. The weight regularizer coefficient  $\lambda_w$  is adjusted with multiplying by 10 gradually during the training stage. The value of  $\lambda_a$  usually does not change for stable training. The whole quantized network training procedure is summarized in the Algorithm 1.

### 4.4 Discussion

In addition to MSQE, SinReQ [8] explored sinusoidal functions as regularizer for weight quantization. Unfortunately, its variation trend outside of quantization segment is quite different from the actual quantization error (see Fig. 2 and



**Algorithm 1** Quantization with smooth regularizers

---

**Require:**  $\mathbf{W}$ ,  $\mathbf{s}_w$ ,  $\mathbf{s}_a$  – learnable parameters (weights and scale factors).  
 $\lambda_w, \lambda_a$  – regularization coefficients.  
 $lr, N_{train}, N_{init}$  – learning rate, epoch size, initialization batches number.

- 1: Initialize  $\mathbf{s}_a$  by sample statistics evaluating  $\mathcal{F}$  on  $N_{init}$  batches from  $\mathbf{X}$ .
- 2: **for**  $N_{train}$  times **do**
- 3:   Sample random batch from train dataset.
- 4:   Evaluate quantization Lagrange function  $\mathcal{L}_Q$ .
- 5:   Calculate the gradients of learnable parameters:  $\mathbf{G}_w = \frac{\partial \mathcal{L}}{\partial \mathbf{W}} + \lambda_w \frac{\partial \mathcal{L}_w}{\partial \mathbf{W}}$ ,  
 $\mathbf{G}_{s_w} = \lambda_w \frac{\partial \mathcal{L}_w}{\partial \mathbf{s}_w}$ ,
- 6:   **If RF:**  $\mathbf{G}_{s_a} = \lambda_a \frac{\partial \mathcal{L}_a}{\partial \mathbf{s}_a}$ ,
- 7:   **If STE:**  $\mathbf{G}_{s_a} = \frac{\partial (\mathcal{L} + \lambda_a \mathcal{L}_a)}{\partial \mathbf{s}_a}$ ,
- 8:   Update learnable parameters using calculated gradients and learning rate  $lr$ .
- 9: **end for**
- 10: Validate quantized model  $\mathcal{F}^q$ .

**Return:**  $\mathbf{W}, \mathbf{s}_w, \mathbf{s}_a$

---

Fig. 1(a)), which results in high clamping error and significant accuracy degradation. Our QSin regularizer represents the equivalent of the actual quantization error effectively. QSin introduces penalty for clamping values which allows to train the quantization scale and improve the accuracy of quantization network. Besides, QSin also adds multiplier  $\pi^2$  and preserves the twice differentiable property successfully. At last, the original SinReQ is only utilized on weight quantization for model compression. Our QSin method is employed on both weights and activations to acquire a fully quantized network. Hence, the proposed QSin method is quite different from the other quantization regularizers.

## 5 Experiment

Extensive experiments are conducted to demonstrate the effectiveness of the proposed QSin method, including classification task and Super-Resolution task.

### 5.1 Implementation Details

*Database* As for classification task, we employed two popular datasets: CIFAR-10 and ImageNet (ILSVRC12) [6]. The CIFAR-10 database contains 50K training images and 10K test images with the size of  $32 \times 32$ , which belong to 10 classes. ImageNet database consists of about 1.2 million training images and 50K test images belonging to 1000 classes. For Super-Resolution task, we employ DIV2K [29] database to train the standard SR network. DIV2K database consists of 800 training images and 100 validation images. The low-resolution images are generated with bicubic degradation operator. During the test stage,

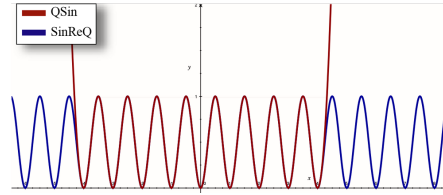


Fig. 2: Comparison of QSin and SinReQ [8] regularizer.

Set5, Set14 and Urban100 [15] database are utilized to evaluate the performance of quantized SR network.

*Settings* In following experiments, the weights and activations of all layers except for the first and last layers are quantized into low-precision integer. We employed Round Free mode for 8-bit quantization and STE mode on activations for 4-bit quantization during the training stage. The quantized network is trained by SGD optimizer with learning rate  $lr = 0.001$  and momentum  $m = 0.9$ . The coefficient of activation regularizer  $\lambda_a$  is setted as a constant value 1. The coefficient of weight regularizer  $\lambda_w$  is initialized as 1 and adjusted each 30 epochs by multiplication on 10. Conventional data augmentation strategies [19,21] are also utilized in these experiments. More details about training configurations could be seen in Appendix D.

## 5.2 Comparison with State-of-the-arts

To compare with other SOTA quantization algorithm, we select four mainstream approaches: MSQE regularization [4], SinReQ [8] regularization, QAT in Tensorflow [18], LSQ [9] and DSQ [12]. By comparing with MSQE and SinReQ methods, we show that QSin achieves better performance since it is smooth and reflects the actual quantization error effectively. The benefits of QSin could be demonstrated by comparing with all these SOTA quantization approaches.

*Image classification* To facilitate comparison, we select conventional classification models including ResNet18 [13] and MibleNet-V2 [26]. We trained the quantized networks with the QSin regularizer on ImageNet database for 4-bit and 8-bit. The experimental results summarized in Table 1 show that QSin method achieves higher top-1 accuracy than other prominent quantization approaches for 4-bit and 8-bit with the architectures considered here. For 8-bit, the quantization networks even achieves slightly better performance than its full-precision model in some cases. Compact neural networks are usually hard to quantize while preserving the accuracy of full-precision model. It is interesting to note that the quantized 8-bit MobileNet-V2 network could achieve close results with full-precision model through QSin approaches. Moreover, QSin results were obtained without weights round during training what lead to higher accuracy on MobileNet-V2 comparing with STE approaches like LSQ.

Table 1: Quantitative results in comparison with state-of-the-art quantization methods on the ImageNet database. The best results are highlighted in bold.

Network	Method	Top-1 Accuracy (%)	
		4-bit	8-bit
ResNet-18 [13]	Full-precision	69.8 (fp32)	
	QAT TF [18]	68.9	69.7
	PACT [30]	69.2	69.8
	DSQ [12]	69.4	69.8
	LSQ [9]	<b>69.8</b>	69.8
	SinReQ [8]	64.63	69.7
	MSQE [4]	67.3	68.1
	<b>QSin</b>	<b>69.7</b>	<b>70.0</b>
MobileNet-V2 [26]	Full-precision	71.8 (fp32)	
	PACT [30]	61.4	71.5
	DSQ [12]	64.8	71.6
	LSQ [9]	68.1	71.6
	SinReQ [8]	61.1	71.2
	MSQE [4]	67.4	71.2
	<b>QSin</b>	<b>68.7</b>	<b>71.9</b>

*Image Super-Resolution* To evaluate the performance of QSin method on low-level vision tasks, we consider the single image super-resolution (SISR) as a conventional task. EDSR [21] and ESPCN [27] are selected typical SR networks to facilitate the comparison. We trained these quantization networks with  $L_1$  loss and QSin regularizer on DIV2K database and summarized the 8-bit quantization results in Table 2. It shows that the proposed QSin method achieves better performance than other quantization approaches, especially for MSQE. Besides, we also illustrate the visual quality of output images from quantized networks in Fig. 3. It is obvious that our QSin method could preserve the texture details of the full-precision model effectively since the smooth property of regularizer. In addition, we also provided comparison with PAMS quantization method [20] which is specially designed for SR network. It could be seen that, without any specific setups and initialization for SISR task, QSin could achieve the same or even slightly better performance as PAMS. More image examples could be seen in Appendix E.

### 5.3 Ablation Study

*Effect of the coefficient of regularizer* The coefficients of regularizer  $\lambda_w$  and  $\lambda_a$  take an important role in the performance of quantized networks. Here we conducted ablation experiments on the CIFAR-10 database to explore the affect of coefficients of regularizer. A series of combinations of  $\lambda_w$  and  $\lambda_a$  were utilized to train the ResNet-20 quantization network. The experimental results summarized in Table 3 reveal that the coefficients has a significant influence on the final performance. Influence of  $\lambda_a$  is not so significant as  $\lambda_w$  but still obvious. The best results from our empirical evaluation shows that multistep scheduling of  $\lambda_w$

Table 2: Quantitative results of SR task in comparison with state-of-the-art quantization methods on benchmark databases. The best 8-bit quantization results are highlighted in bold.

Network	Method	PSNR (dB)		
		Set5	Set14	Urban100
4x EDSR [21]	Full-precision	32.2	28.5	26
	QAT TF [18]	31.9	28.4	25.7
	PACT [30]	31.5	28.2	25.25
	LSQ [20]	32.1	28.5	25.9
	PAMS [20]	32.1	<b>28.6</b>	26
	SinReQ [8]	32.1	28.3	25.3
	MSQE [4]	32.1	28.5	25.9
	<b>QSin</b>	<b>32.2</b>	28.5	<b>26</b>
3x ESPCN [27]	Full-precision	32.5	29	26.1
	QAT TF [18]	32.35	28.8	25.9
	LSQ [9]	32.4	28.9	26
	SinReQ [8]	32.2	28.9	26
	MSQE [4]	32.45	28.95	26
	<b>QSin</b>	<b>32.5</b>	<b>29</b>	<b>26.1</b>

allows significantly improve quality. The intuition behind are follows: less values of  $\lambda_w$  helps to task loss makes more contribution, during training we increase  $\lambda_w$  to slightly decrease quantization error.

Table 3: Experimental results with various coefficients of regularizer on the CIFAR-10 database. Last line shows multistep scheduling on  $\lambda_w$ .

$\lambda_w$ for weights	$\lambda_a$ for activations	Accuracy (%)	
		4-bit	8-bit
0	0	88.9	91.6
0	1	89.2	91.6
0	10	89.2	91.6
1	0	90.0	91.6
1	1	90.2	91.7
10	1	90.6	91.7
100	1	91.1	91.8
(1, 10, 100)	1	91.7	91.9

*Quantization error and convergence analysis* To analyze the effect of Qsin in terms of reducing the quantization error, we explored the variation of the mean square quantization error of activations and weights along with the increasing of

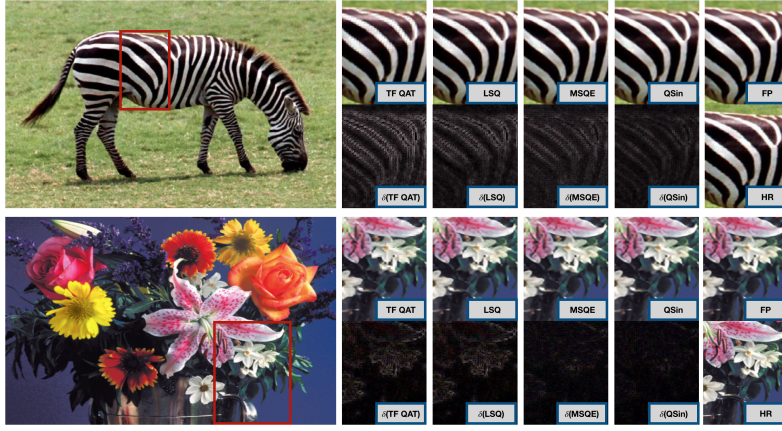


Fig. 3: Visual Comparison of 3x ESPCN with state-of-the-art methods in terms visual quality.  $\delta$  is the residual map of the corresponding image of quantized network and full-precision network.

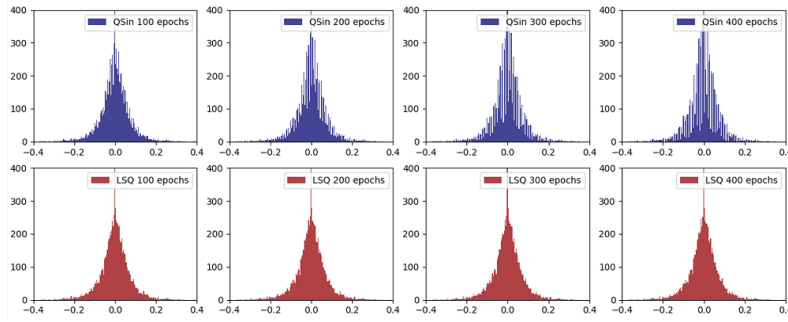


Fig. 4: Histograms of the weights distribution from the third convolution layer of ESPCNN model for SR task. The dynamic evolution of weights distribution from QSin and LSQ approaches are compared here.

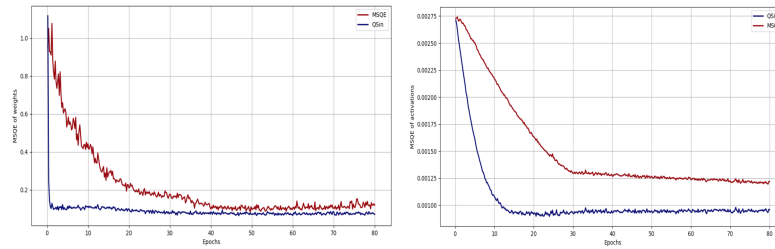


Fig. 5: Quantization error comparison of QSin and MSQE for ResNet-20 on CIFAR-10 in 4-bit quantization.

training iterations. Fig. 5 illustrates the results of 4-bit quantization on ResNet-20 with QSin and MSQE regularizers. It is obvious that QSin leads to much lower quantization error compared with MSQE regularizer since its smooth property. It reveals that QSin not only achieved higher accuracy than MSQE but also acquired more stable convergence. This phenomenon should thanks to the smooth property of QSin regularizer which is more helpful to optimization.

*Weight analysis* To analyze the weight quantization, we provide the histograms of weights distribution from the model which was trained with QSin regularizer in Fig. WeightDistribution. Multiple histograms from various epochs are illustrated together to explore the dynamic evolution of weights distribution. We have compared weights distributions of networks trained through QSin, MSQE and LSQ method [9]. LSQ employs trainable scale factor with straight through estimator to propagate through the round function. From Fig. 4, we can see that the weights distribution of QSin method becomes more and more similar to the categorical distribution along with rising of training epochs. In addition, the histograms of weights distribution from the network trained by QSin are closer to categorical distribution than weights histogram obtained from the LSQ method. During the evaluation and test stage, the quantized network would generate less quantization error on weights for QSin method. This is one reason why QSin method could achieve better performance than LSQ in Table 1 and Table 3. More distributions could be find in Appendix C.

## 6 Conclusion

This paper defined a family of equivalent smooth quantization regularizer to alleviate the accuracy degradation problem in network quantization. Then, we proposed a novel QSin regularizer belonging to SQR which represents the equivalent of actual quantization error and allows to obtain better gradient behavior in the neighborhood of transition points. In addition, we built up the corresponding algorithm to train the quantization network. The extensive experimental results show that the proposed SQR method could achieve much better performance than other prominent quantization approaches on image classification and super-resolution task. What's more, in terms of visual quality, SQR approach would not generate the grid artifact compared with other quantization methods due to its smooth property. The ablation study reveals that SQR could reduce the quantization error significantly and acquire stable convergence. Furthermore, distributions of the learned weights from SQR regularizer are more close to categorical distribution, which is helpful to booting the performance of quantized network.

## References

1. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
2. Chen, H., Wang, Y., Xu, C., Shi, B., Xu, C., Tian, Q., Xu, C.: Addernet: Do we really need multiplications in deep learning? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1468–1477 (2020)
3. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.J., Srinivasan, V., Gopalakrishnan, K.: Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085 (2018)
4. Choi, Y., El-Khamy, M., Lee, J.: Learning low precision deep neural networks through regularization. arXiv preprint arXiv:1809.00095 **2** (2018)
5. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1 (2016)
6. 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)
7. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
8. Elthakeb, A.T., Pilligundla, P., Esmailzadeh, H.: Sinreq: Generalized sinusoidal regularization for low-bitwidth deep quantized training. arXiv preprint arXiv:1905.01416 (2019)
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. Esser, S.K., Merolla, P.A., Arthur, J.V., Cassidy, A.S., Appuswamy, R., Andreopoulos, A., Berg, D.J., McKinstry, J.L., Melano, T., Barch, D.R., et al.: Convolutional networks for fast, energy-efficient neuromorphic computing. Proceedings of the national academy of sciences **113**(41), 11441–11446 (2016)
11. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2018)
12. 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 International Conference on Computer Vision. pp. 4852–4861 (2019)
13. 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)
14. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine **29**(6), 82–97 (2012)
15. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5197–5206 (2015)
16. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: Training neural networks with low precision weights and activations. The Journal of Machine Learning Research **18**(1), 6869–6898 (2017)

17. Hubara, I., Nahshan, Y., Hanani, Y., Banner, R., Soudry, D.: Improving post training neural quantization: Layer-wise calibration and integer programming. arXiv preprint arXiv:2006.10518 (2020)
18. 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)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25**, 1097–1105 (2012)
20. Li, H., Yan, C., Lin, S., Zheng, X., Zhang, B., Yang, F., Ji, R.: Pams: Quantized super-resolution via parameterized max scale. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV* 16. pp. 564–580. Springer (2020)
21. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pp. 136–144 (2017)
22. McKinstry, J.L., Esser, S.K., Appuswamy, R., Bablani, D., Arthur, J.V., Yildiz, I.B., Modha, D.S.: Discovering low-precision networks close to full-precision networks for efficient inference. In: *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing–NeurIPS Edition (EMC2–NIPS)*. pp. 6–9. IEEE (2019)
23. Nagel, M., Fournarakis, M., Amjad, R.A., Bondarenko, Y., van Baalen, M., Blankevoort, T.: A white paper on neural network quantization. arXiv preprint arXiv:2106.08295 (2021)
24. Naumov, M., Diril, U., Park, J., Ray, B., Jablonski, J., Tulloch, A.: On periodic functions as regularizers for quantization of neural networks. arXiv preprint arXiv:1811.09862 (2018)
25. Peng, H., Wu, J., Chen, S., Huang, J.: Collaborative channel pruning for deep networks. In: *International Conference on Machine Learning*. pp. 5113–5122. PMLR (2019)
26. 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)
27. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1874–1883 (2016)
28. Stock, P., Fan, A., Graham, B., Grave, E., Gribonval, R., Jegou, H., Joulin, A.: Training with quantization noise for extreme model compression. In: *International Conference on Learning Representations* (2021)
29. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pp. 114–125 (2017)
30. 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)
31. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients (2016)