Explicit Model Size Control and Relaxation via Smooth Regularization for Mixed-Precision Quantization

Vladimir Chikin[1⋆], Kirill Solodskikh[1⋆], and Irina Zhelavskaya[2]

[1] Huawei Noah's Ark Lab
{vladimir.chikin,solodskikh.kirill1}@huawei.com
[2] Skolkovo Institute of Science and Technology (Skoltech)
irina.zhelavskaya@skolkovotech.ru

Abstract. While Deep Neural Networks (DNNs) quantization leads to a significant reduction in computational and storage costs, it reduces model capacity and therefore, usually leads to an accuracy drop. One of the possible ways to overcome this issue is to use different quantization bit-widths for different layers. The main challenge of the mixed-precision approach is to define the bit-widths for each layer, while staying under memory and latency requirements. Motivated by this challenge, we introduce a novel technique for explicit complexity control of DNNs quantized to mixed-precision, which uses smooth optimization on the surface containing neural networks of constant size. Furthermore, we introduce a family of smooth quantization regularizers, which can be used jointly with our complexity control method for both post-training mixed-precision quantization and quantization-aware training. Our approach can be applied to any neural network architecture. Experiments show that the proposed techniques reach state-of-the-art results.

Keywords: neural network quantization, mixed-precision quantization, regularization for quantization

1 Introduction

Modern DNNs allow solving a variety of practical problems with accuracy comparable to human perception. The commonly used DNN architectures, however, do not take into account the deployment stage. One popular way to optimize neural networks for that stage is quantization. It significantly reduces memory, time and power consumption due to the usage of integer arithmetic that speeds up the addition and multiplication operations.

There are several types of quantization that are commonly used. The fastest in terms of application time and implementation is post-training quantization [1], [7], [17], [3], in which the weights and activations of the full-precision (FP) network are approximated by fixed-point numbers. While being the quickest

⋆ These authors contributed equally to this work.

approach, it usually leads to the decrease in accuracy of a quantized network. To reduce the accuracy drop, quantization-aware training (QAT) algorithms can be used [15], [9], [26], [5]. QAT employs stochastic gradient descent with quantized weights and activations on the forward pass and full-precision weights on the backward pass of training. It usually leads to better results but requires more time and computational resources than PTQ.

One technique to make QAT converge faster to the desired quantization levels is regularization. Periodic functions, such as mean squared quantization error (MSQE) or the sine function, are typically used as regularizers [6], [18], [11]. The high-precision weights are then naturally pushed towards the desired quantization values corresponding to the minima of those periodic functions. The sine function is smooth and therefore has advantages over MSQE, which is not smooth. The sine function has an infinite number of minima, however, which may lead to a high clipping error in quantization, and therefore, can have a negative impact on the model accuracy.

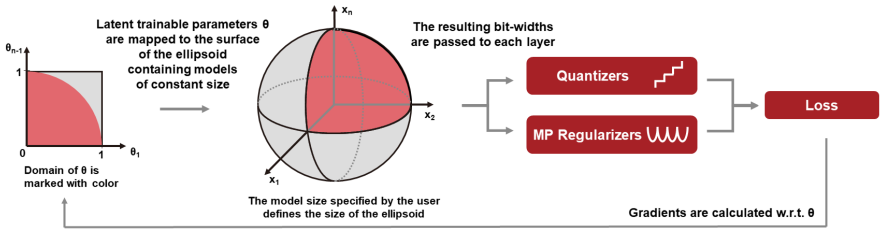

Fig. 1. Illustration of the proposed pipeline. Parameters θ are mapped to the ellipsoid of constant-size DNNs. Each point of this ellipsoid defines the bit-width distribution across different layers in order to obtain the model size pre-defined by the user. Next, the bit-widths are passed to a DNN for the forward pass. Our special smooth mixed-precision (MP) regularizers are computed jointly with the task loss and help reduce the gradient mismatch problem. The DNN parameters, quantization parameters, and θ are updated according to the calculated loss on the backward pass.

If a trained quantized network still does not reach an acceptable quality, mixed-precision quantization may be applied. Such algorithms allocate different bit-widths to different layers and typically perform better than the fixed bit-width counterparts. At the same time, it is also important to keep the memory and latency constraints in mind, so that a quantized model meets specific hardware requirements. Too much compression may lead to a significant loss in accuracy, while not enough compression may not meet the given memory budget. Many existing studies use gradient descent to tune the bit-width distribution, but this approach inherently does not have the ability to explicitly set the required compression ratio of the model. One way to constrain the model size is to use additional regularizers on the size of weights and activations [22]. However, regularizers also do not allow setting the model size, and hence the compression

ratio, explicitly, and multiple experiments may still be needed to obtain a desired model size. Other state-of-the-art methods are based on reinforcement learning [23], [12], or general neural architecture search (NAS) algorithms [24], [4], [27], where the search space is defined by the set of possible bit-widths for each layer. However, such methods imply training of multiple instances of a neural network, which usually require large computational and memory resources.

In this paper, we address the problems mentioned above. First, we show that models of the same size quantized to different mixed-precision lie on the surface of a multi-dimensional ellipsoid. We suggest a parametrization of this ellipsoid by using latent continuous trainable variables, using which the discrete problem of quantized training can be reformulated to a smooth one. This technique imposes almost no computational overhead compared to conventional QAT algorithms and certainly requires much less computational time and resources compared to reinforcement or NAS algorithms. Furthermore, we suggest a universal family of smooth quantization regularizers, which are bounded, and therefore reduce the clipping error and lead to better performance. Our main contributions are the following:

1. We propose a novel method of mixed-precision quantization that strictly controls the model size. It can be applied to both weights and activations. To the best of our knowledge, this is the first method of mixed-precision quantization that allows *explicit* model size control.
2. We construct a family of bounded quantization regularizers for smooth optimization of bit-width parameters. It allows for faster convergence to the discrete values and avoids high clipping error. It also avoids difficulties of the discrete optimization.
3. We validate our approach on image classification and image super-resolution tasks and compare it to the recent methods of mixed-precision quantization. We show that the proposed approach reaches state-of-the-art results.

Notation We use $x, \mathbf{x}, \mathbf{X}$ to denote a scalar, a vector and a matrix (or a tensor, as is clear from the context). $\lfloor . \rfloor, \lceil . \rceil$ and $\lfloor . \rceil$ are the floor, ceiling, and round operators. $\mathbb{E}[\cdot]$ denotes the expectation operator. We call the total model size the following value: $\sum_{i=1}^{n} k_i \cdot b_i + 32 \cdot k_0$, where $\{k_i\}_{i=1}^{n}$ and $\{b_i\}_{i=1}^{n}$ are the sizes of weight tensors of quantized layers and the corresponding bit-widths, and k_0 is the number of non-quantized parameters.

2 Related work

Quantization-aware training Many existing quantization techniques, such as [15], [9], [26], [5] train quantized DNNs using gradient descent-based algorithms. QAT is based on the usage of quantized weights and activations on the forward pass and full-precision weights on the backward pass of training. It is difficult to train the quantized DNNs, however, as the derivative of the round function is zero almost everywhere. To overcome this limitation, straight-through estimators

(STE) were proposed [2]. They approximate the derivative of the round function and allow backpropagating the gradients through it. In that way, the network can be trained using standard gradient descent [15].

Quantization through regularization These methods involve additional regularization for quantized training. [6] used mean squared quantization error (MSQE) regularizer to push the high-precision weights and activations towards the desired quantized values. [18], [11] employed the trigonometric sine function for regularization. Sinusoidal functions are differentiable everywhere and have periodic minima which can be utilized to drive the weights towards the quantized values. However, the sine function has an infinite number of minima points, which may lead to a high clipping error.

Mixed-precision quantization In some cases, quantization of the whole network to low bit-width could produce unacceptable accuracy drop. In such cases, some parts of a network could be quantized to a higher precision. Mixed-precision quantization algorithms are used to obtain a trade-off between quality and acceleration. This task involves searching in a large configuration space. The state-of-the-art methods are based on reinforcement learning [23], [12], or general neural architecture search algorithms [24], [4], [27], where the search space is defined by the set of possible bit-widths for each layer. Such methods imply training of multiple instances of a neural network, which may require large memory resources. Differently in [22], the bit-width is learned from the round function parameterized by quantization scale and range. The derivative of this parametrization is obtained using STE. In all these studies, an additional complexity regularizer is used to control the compression ratio of the model. This technique does not allow to specify the desired compression ratio explicitly, and multiple experiments may be needed to obtain a desired model size.

3 Motivation and preliminaries

In this section, we briefly describe the quantization process of neural networks and the motivation for the proposed methods. Consider a neural network with n layers parameterized by weights $\hat{\mathbf{W}} = \{\mathbf{W}_i\}_{i=1}^n$. Let each layer correspond to some function $\mathcal{F}_i(\mathbf{W}_i, \mathbf{A}_i)$, where $\mathbf{W}_i$ are the weights of layer i, and $\mathbf{A}_i$ are the input activations to a layer.

Quantization Quantization of a neural network implies obtaining a model, whose parameters belong to some finite set. As a rule, this finite set is specified by a function that maps the model parameters to this set. We can define the following *uniform quantization function*:

$$\mathcal{Q}_U^b(x) = clip(\lfloor x \rceil, -2^{b-1}, 2^{b-1}–1) \tag{1}$$

where $[-2^{b-1}, 2^{b-1}-1]$ is the quantization range, b is the quantization bit-width. Under the assumption that layer $\mathcal{F}_i(\mathbf{W}_i, \mathbf{A}_i)$ commutes with scalar multiplication, quantization is reduced to optimizing the quality of a quantized model relative to its parameters $\hat{\mathbf{W}}$ and *quantization scale parameters* $\mathbf{s}_w = \{s_{w_i}\}_{i=1}^n$ and $\mathbf{s}_a = \{s_{a_i}\}_{i=1}^n$. The scale parameters determine the quantization range; if all

FP values of weights are covered, then $s_i = \max|\mathbf{W}_i|/(2^b - 1)$ for the symmetric quantization scheme. Layers $\mathcal{F}_i(\mathbf{W}_i, \mathbf{A}_i)$ are then replaced with *quantized layers*:

$$\mathcal{F}_i^q = s_{w_i} s_{a_i} \mathcal{F}_i \left(\mathcal{Q}_U^{b_w} \left(\frac{\mathbf{W}_i}{s_{w_i}} \right), \mathcal{Q}_U^{b_a} \left(\frac{\mathbf{A}_i}{s_{a_i}} \right) \right), \tag{2}$$

where b_w is the quantization bit-width of the weight tensor and b_a is the quantization bit-width of the activation tensor. The most commonly used layers, such as convolutional and fully-connected layers, satisfy this assumption.

Backpropagation through quantization To train such a quantized network, we need to pass the gradients through the quantization function on the backward pass. Since the derivative of the round function is zero almost everywhere, straight-through estimators (STE) are used to approximate it. The following STE was proposed in [2] and is commonly used (we will use it in this paper as well):

$$\frac{d\mathcal{Q}_U(x)}{dx} = \begin{cases} 1, \text{ if } -2^{b-1} \leq x \leq 2^{b-1} - 1, \\ 0, \text{ otherwise.} \end{cases} \tag{3}$$

Model complexity control The problem of training a quantized model using STE is reduced to optimization of the loss function $\mathcal{L}_Q$:

$$\mathcal{L}_Q = \mathbb{E}[\mathcal{L}(\mathcal{F}^q(\hat{\mathbf{W}}, \mathbf{X}))]. \tag{4}$$

Some modern quantization methods optimize $\mathcal{L}_Q$ relative to variables $(\hat{\mathbf{W}}, \mathbf{s}_w, \mathbf{s}_a)$, that is, the scale parameters and weights are tuned as trainable variables. In the case of traditional quantization, the bit-width values are fixed during training. In mixed precision quantization, bit-widths of different layers can change during training. A number of modern mixed-precision quantization methods use gradient descent to tune the bit-width distribution, but they do not have the ability to explicitly set the required compression ratio of the model. In this paper, we propose a method for training the mixed-precision quantized models with the explicit, user-defined total model size. In the proposed framework, bit-widths are tuned as trainable variables in addition to the scale parameters and weights. We describe it in detail in the next section.

Soft regularization In training of (4), high-precision weights are used in the backward pass of gradient descent, and low-precision weights and activations are used in the forward pass. To reduce the discrepancy in the backward and forward passes, regularization can be utilized. A possible choice for the regularizer is MSQE [6]. However, it has a non-stable behavior of gradients in the neighborhood of the transition points. It can be seen from Fig. 2 that the derivative of MSQE changes drastically in the transition points and pushes the points in the opposite direction due to the large derivative there. In this paper, we propose to use smooth quantization regularizers instead of MSQE. Fig. 2 also shows an example of the sinusoidal regularizer, which has a more stable gradient behavior

in the neighborhood of the transition points. In particular, the gradient of the main loss (4) has a larger impact than the regularization loss there. It is worth noting that some previous works [11], [18] also employed smooth quantization regularizers. However, they were unbounded (see Fig. 2), which led to accumulation of a clipping error due to the points outside the quantization range. We propose to use bounded smooth regularizers, which we describe in detail next.

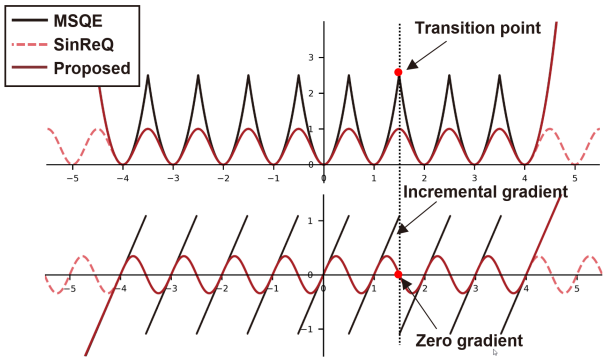

Fig. 2. Comparison of MSQE and smooth bounded (proposed in this work) and unbounded (e.g., SinReQ [11]) regularizers. Smooth quantization regularizers have similar properties to MSQE but different behavior close to the transition points.

4 Methodology

In this section, we first describe the proposed method of mixed-precision QAT with explicit model size control. Then, we describe the proposed family of smooth regularizers that allow faster convergence to quantized values. Finally, we describe an additional technique for bit-widths stabilization during training. We provide a description for the case of symmetric mixed-precision quantization of weights into *int* scheme. The proposed method can be easily generalized to the case of mixed-precision quantization of activations without any additional constraints and other quantization schemes.

4.1 Explicit model size control using surfaces of constant-size neural networks

We propose to use a special parametrization of the bit-width parameters of the model layers. This parametrization imposes restrictions on the size of the quantized model. To do this, we build a surface of neural networks of the same size in the bit-width space. We parameterize this surface using latent independent variables, which are tuned during training. The proposed technique is applicable to any layer of a DNN.

Suppose that we want to quantize n layers of a DNN. For simplicity, we assume that activations of those layers are quantized to fixed bit-width b_a. We denote by $\{k_i\}_{i=1}^n$ and $\{b_i\}_{i=1}^n$ the sizes of weight tensors of those layers and their corresponding quantization bit-widths. The size of the quantized part of the network is $\sum_{i=1}^n k_i b_i$. Our goal is to preserve it during training.

Consider the n-dimensional *ellipsoid equation*:

$$\sum_{i=1}^{n} k_i x_i^2 = C. \tag{5}$$

We can parameterize the surface of an ellipsoid in the first orthant ($x_i > 0$) using $n-1$ independent variables $\theta = \{\theta_i\}_{i=1}^{n-1}$, each from segment $[0, 1]$:

$$\begin{cases} x_1 = \sqrt{\frac{C}{k_1}}\theta_1, \\ \dots, \\ x_{n-1} = \sqrt{\frac{C}{k_{n-1}}}\theta_{n-1}, \\ x_n = \sqrt{\frac{C - \sum_{i=1}^{n-1} k_i x_i^2}{k_n}}. \end{cases} \tag{6}$$

Fig. 1 illustrates the process of mapping trainable parameters θ_i to the ellipsoid of models with the same size. We do not consider other orthants for parametrization as they are redundant and increase the search space during the bit-width tuning. It is important to note that such parametrization does not satisfy the ellipsoid equation for all possible θ in $[0, 1]^{n-1}$ (see Fig. 1). In particular, during training we clamp the outliers. From our empirical evaluation, such outliers appear with almost zero probability.

Additionally, let us define variables $t_i = 2^{x_i^2 - 1}$. We can use x_i and t_i for weight quantization of the i-th layer by replacing the bit-width parameter b_w by x_i^2 and the maximum absolute value of the quantization range $2^{b_w - 1}$ by t_i in eq. (2):

$$\mathcal{F}_i^q = \frac{\tilde{s}_{w_i}}{t_i} \frac{\tilde{s}_{a_i}}{2^{b_a - 1}} \mathcal{F}_i \left(\mathcal{Q}_U^{x_i^2} \left(t_i \cdot \frac{\mathbf{W}_i}{\tilde{s}_{w_i}} \right), \mathcal{Q}_U^{b_a} \left(2^{b_a - 1} \cdot \frac{\mathbf{A}_i}{\tilde{s}_{a_i}} \right) \right), \tag{7}$$

where $\tilde{s}$ corresponds to absolute maximum values of FP weights/activations (see (3)).

As a result, $\mathbf{W}_i$ is quantized using a grid of integers from the range $[-\lfloor t_i \rceil, \lfloor t_i \rceil - 1]$ consisting of $2\lfloor t_i \rceil$ integers. If x_i^2 is an integer, the value of $2\lfloor t_i \rceil$ equals to $2^{x_i^2}$ – an integer power of two, which corresponds to quantization to x_i^2 bits. For example, if $x_i^2 = 8$, then $t_i = 128$ and the corresponding tensor $\mathbf{W}_i$ is quantized using a grid of integers consisting of 256 elements, or to 8 bits. Thus, x_i^2 serves as a continuous equivalent of the bit-width of the weights of the i-th layer. To estimate the real bit-width of the i-th layer during the training, we use the smallest sufficient integer value, namely:

$$b_i = \lceil \log_2(2 \cdot \lfloor 2^{x_i^2 - 1} \rceil) \rceil \approx x_i^2. \tag{8}$$

We propose to train the quantized model by minimizing loss $\mathcal{L}_Q$ (4) relative to variables $(\mathbf{W}, \mathbf{s}_w, \mathbf{s}_a, \theta)$. Thus, we can train quantized models, whose mixed-precision bit-widths are tuned during model training in a continuous space. Despite the fact that parameters θ, and hence the bit-widths of the layers, change during training, it follows from the definition of the ellipsoid parametrization that (5) is always satisfied, which means that the size of a model is preserved during training. Due to the error of rounding the bit-widths to integer values (8), the size of the quantized model may differ from C, but not significantly.

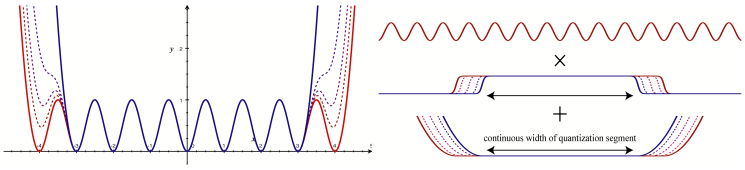

Fig. 3. The proposed family of smooth quantization regularizers. It can be implemented as a multiplication of the periodic function by the hat function, and then adding a function, which is square outside and zero inside the quantization range.

4.2 Smooth bounded regularization as a booster for quantized training

Additional regularization is often used as a special technique for training quantized models, see (2) and (3). We propose to use special regularizers that allow improving the quality of bit-width tuning and training of mixed-precision quantized models. We consider a family of smooth bounded regularizers $\phi(x, t)$ that smoothly depend on parameter t determining the width of the quantization range. We require the following from those regularizers:

- For fixed t, integers from the range $[-\lfloor t \rfloor, \lfloor t \rfloor - 1]$ are the roots and minima of function $\phi(\cdot, t)$, in particular, functions ϕ and ϕ' are zero in these points.
- If $t \in \mathbb{Z}$, then there are no other minima.
- If $t \notin \mathbb{Z}$, then a maximum of 2 more local minima outside $[-\lfloor t \rfloor, \lfloor t \rfloor - 1]$ are possible.

There are various types of regularizers satisfying the above conditions. In our quantization experiments, we use an easy-to-implement function (shown in Fig. 3):

$$\begin{aligned}\phi(x,t) = \sin^2(\pi x) \cdot \sigma(x+t) \cdot \sigma(t-1-x) + \\ + \begin{cases} \pi^2(x+t)^2, \; x \le -t, \\ 0, \; x \in [-t, t-1], \\ \pi^2(x-t+1)^2, \; x \ge t-1, \end{cases}\end{aligned} \tag{9}$$

where σ is the sigmoid function. For quantization of weights of the i-th layer, scaled tensor $\mathbf{W}_i$ must be passed to regularizer ϕ as argument x, and parameter $t_i = t_i(\theta)$ as argument t. For quantization of activations of the i-th layer, scaled tensor A_i must be passed to the regularizer as argument x, and the maximum absolute value of the activation quantization grid 2^{b_a-1} as argument t.

Analysis of the Taylor series expansion of function $\frac{s^2}{t^2}\phi\big(t \cdot \frac{x}{s}, t\big)$ shows that this function is a good estimate of the mean squared quantization error in the neighborhood of integer points of the range $[-\lfloor t \rfloor, \lfloor t \rfloor - 1]$ (see Appendix A for a detailed proof). Therefore, we propose to use the following functions as regularizers:

$$\mathcal{L}_w = \sum_i \frac{s_{w_i}^2}{t_i^2(\theta)} \phi\left(t_i(\theta) \cdot \frac{\mathbf{W}_i}{s_{w_i}}, t_i(\theta)\right) \tag{10}$$

$$\mathcal{L}_a = \sum_i \mathbb{E}\left[\frac{s_{a_i}^2}{2^{b_a}} \phi\left(2^{b_a-1} \cdot \frac{\mathbf{A}_i}{s_{a_i}}, 2^{b_a-1}\right)\right] \tag{11}$$

As a result, training of a mixed-precision quantized model can be done by optimizing the following loss function relative to parameters (W, s_w, s_a, θ):

$$\mathcal{L}_Q = \mathbb{E}\big[\,\mathcal{L}\left(\mathcal{F}^q(\mathbf{W}, \mathbf{X})\right)\big] + \lambda_w \mathcal{L}_w + \lambda_a \mathcal{L}_a\,. \tag{12}$$

The proposed family of smooth regularizers also allows tuning the bit-widths as a post-training algorithm without involving other parameters in the training procedure (see section 5.1 for an example).

4.3 Regularizers for bit-width stabilization

During training, parameters t_i can converge to values that are not degree of 2, corresponding to non-integer bit-widths. For example, if $x_i^2 = 3.2$, then the scaled tensor is quantized to a grid of integers consisting of $2 \cdot \lceil 2^{3.2-1} \rceil = 10$ elements. In this case, 3 bits are not sufficient to perform the calculations (for this, there should be not more than 8 elements), and we do not fully use 4 bits (only 10 from 16 possible elements are used). In order for the bit-widths to converge to integer values, we add a sinusoidal regularizer for bit-width values $\mathcal{L}_b = \sum_{i=1}^{n} \sin^2(\pi x_i^2)$ to loss $\mathcal{L}_Q$ (12).

In some tasks, we may need the ability to use a specific set of bit-width values for mixed-precision quantization, for example, 4, 8 or 16 bits. In this case, we add a special regularizer for bit-widths values to the loss $\mathcal{L}_Q$ (12), which is aimed at contracting them to the required set. We propose smooth regularizers that have local minima at the points from the required set of bit-widths and have no other local minima. Using such regularizers during training makes the bit-width parameters converge to a given set of values. We normalize these regularizers using regularization parameter λ_b.

4.4 Algorithm overview

Here, we describe the overall algorithm combining the proposed techniques. A pseudo code for the algorithm and more specifics are given in Appendix B. We minimize loss function $\mathcal{L}_Q$ (12) relative to trainable variables $(\mathbf{W}, \mathbf{s}_w, \mathbf{s}_a, \theta)$ using stochastic gradient descent. We quantize weights and activations of a model on the forward pass of training. We use a straight-through estimator (3) to propagate through round function $\mathcal{Q}_U$ on the backward pass.

We initialize the scale parameters of the weight tensors with their maximum absolute values. To initialize the scale parameters of the activation tensors, we pass several data samples through the model and estimate the average maximum absolute values of inputs to all quantized layers. To set the required model size, a user must specify parameter b_{init} – an initial value of the continuous bit-widths x_i^2 of the quantized layers. Using equations (6), we initialize parameters θ_i so that continuous bit-widths x_i^2 of all quantized layers are equal to b_{init}. Parameter b_{init} determines the total size of the quantized part of the model using equation (5), which means that it determines the ellipsoid that is used for training.

Table 1. Influence of the proposed techniques. Weights only quantization of ResNet-20 on CIFAR-10. MC – model compression ratio (times), Top 1 – Top-1 quantized accuracy in %, MP – mixed precision.

Quantization	# bits W/A	No regularizers		With regularizers	
		Top-1	MC	Top-1	MC
Full precision	32/32	91.73	-	91.73	-
Post-training	2/32	19.39	14.33	-	-
BN tuning	2/32	64.34	14.33	-	-
Bit-widths tuning	MP/32	65.94	14.55	77.05	14.55
Bit-widths and scales tuning	MP/32	79.20	15.02	86.85	14.55
All model parameters tuning	MP/32	**91.27**	14.55	**91.57**	14.55
Post-training	3/32	75.57	9.92	-	-
BN tuning	3/32	89.58	9.92	-	-
Bit-widths tuning	MP/32	89.70	10.11	90.13	10.05
Bit-widths and scales tuning	MP/32	90.65	10.19	90.70	10.19
All model parameters tuning	MP/32	**91.69**	10.16	**91.89**	10.25

5 Experiments

We evaluate the performance of the proposed techniques on several computer vision tasks and models. In section 5.1, we study how each of the proposed techniques influences the quality of mixed precision quantization. In section 5.2, we compare the proposed algorithm to other QAT methods for image classification.

Experimental setup In our experiments, we quantize weights and activations only, and some of the parameters of the models remain non-quantized (for example, biases and batch normalization parameters). The bit-width of the non-quantized parameters is 32. All quantized models use pre-trained float32 networks for initialization. As a model compression metric, we use the compression ratio of the total model size (not just weights), and as an activation compression metric, we use the mean input compression ratio over a set of model layers (same metric as in [22]). We set a specific value for the model compression by setting the corresponding value for the parameter b_{init} in each of the tasks. We normalize the quantization regularizers to have the same order as the main loss $\mathcal{L}_Q$ (4) by using coefficients λ_w, λ_a, which are chosen as powers of 10. After some number of epochs, we adjust λ_w by multiplying it by 10 to reduce the weights quantization error; λ_a does not change during training. The corresponding training strategy for each experiment is described in Appendix C.

5.1 Ablation study

Impact of the proposed techniques The proposed method is based on several techniques described above and involves tuning bit-widths, scale parameters and model parameters (i.e., weights). We investigate the impact of the proposed techniques by applying them separately for quantization of ResNet-20 [14] on the CIFAR-10 dataset [16]. We also compare the obtained results with the results of post-training quantization to fixed bit-width with and without batch-normalization (BN) tuning. Additionally, we explore the influence of smooth regularizers suggested in (4.2). The results are presented in Table 1.

The proposed technique of the bit-width tuning used as a PTQ without training the model parameters and scale parameters obtains a quantized model with a better quality and the same compression ratio as post-training with BN tuning for both 2-bit and 3-bit quantization. The subsequent addition of the proposed methods leads to an increasing improvement in the quality of the quantized model, and the joint training of the bit-widths, scales and model parameters leads to the best accuracy. One can also note that the use of smooth regularizers further improves the quality of the resulting quantized models. Thus, all of the proposed techniques contribute to the increase of quantized model accuracy.

Optimality of the determined mixed precision We further investigate whether the bit-widths of different layers found with our approach are optimal. We show that on the task of 3× image super-resolution for *Efficient Sub-Pixel CNN* [21] with global residual connection. We choose this task to demonstrate the effectiveness of our approach visually as well.

The full precision Efficient Sub-Pixel CNN consists of 6 convolutions. We quantize all layers of the model except for the first layer. The bit-widths of activations are equal to the bit-widths of the corresponding weights of each layer. The last 4 convolutions of the model are almost the same size. We use the proposed method to quantize all layers to 8-bit, except for one of those 4 convolutions, the size of which we set to 4 bits. Our algorithm selects the second of these four convolutions to be quantized to 4 bits. To prove that our

Table 2. Validation PSNR, dB, for ESPCN quantized to mixed-precision. MC – model compression ratio (times).

Dataset	Our	Bit #1	Bit #2	Bit #3
Vimeo-90K	**31.10**	30.93	30.88	30.74
Set5	**30.55**	30.46	30.47	30.02
Set14	**26.93**	26.88	26.88	26.63
MC	**3.37**	3.37	3.37	3.34

Table 3. Validation PSNR, dB, for ESPCN quantized to mixed-precision (for both weights and activations).

Network	Vimeo-90K	Set5	Set14
FP	31.28	30.74	27.06
8 bit	**31.19**	**30.74**	**27.05**
MixPr1	31.10	30.55	30.55
MixPr2	30.95	30.50	26.89
4 bit	30.62	29.84	26.53
Bicubic	29.65	28.92	25.91

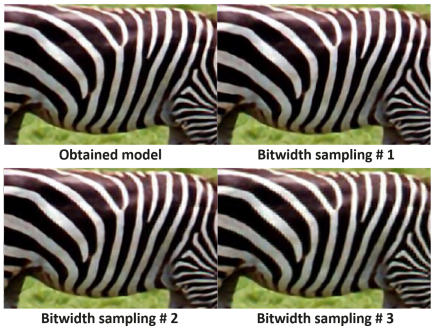

Fig. 4. Ablation study for mixed precision quantization of ESPCN.

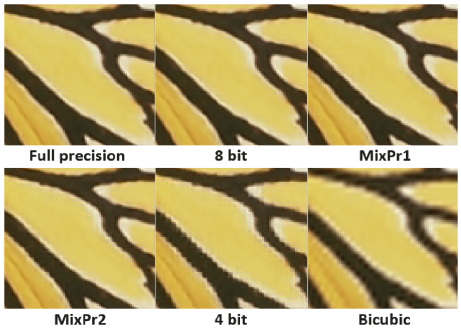

Fig. 5. Quantization of ESPCN: comparison of different bit-widths.

algorithms converges to an optimal configuration, we train other configurations with fixed bit-widths, in which one of the last four layers is quantized to 4 bit. The three possible configurations are denoted by Bit #1, Bit #2, Bit #3, and their performance is shown in Table 2 and Fig. 4. We can see that the model obtained with our method has the best perceptual quality out of other possible model choices. The PSNR of the model obtained with our method is also larger than PSNR of other models. This means that our algorithm determines the best layer for 4-bit quantization while preserving the model compression rate.

Influence of model compression We test the effect of different proportion of 4-bit and 8-bit quantization of a model on its accuracy for the same task as in the previous experiment. To investigate that, we train several models with different compression ratios. First, we train two mixed precision quantized models: *MixPr1*, in which 15.6% of model is quantized to 4 bit and 75.6% to 8 bit, and *MixPr2*, in which 46.8% of model is quantized to 4 bit and 44.4% to 8 bit. We compare these models to the ones quantized to 4 bit and 8 bit only (see Table 3). We train these mixed-precision quantized models using a regularizer for bit-width stabilization for 4 and 8 bits (see section 4.3).

We observe that as the proportion of 8 bit increases, the perceptual quality of the resulting images improves (Fig. 5). The perceptual quality of the 8-bit model produced by our method matches the perceptual quality of the full precision model. More examples are provided in Appendix D.

5.2 Comparison with existing studies

CIFAR-10 We compare our method to several methods for mixed-precision quantization of ResNet-20 in Table 4. In these experiments, we quantize all layers of the models. The first and last layers are quantized to a fixed bit-width, and the rest of the layers are used for mixed-precision quantization. We test cases when activations are quantized to 4 bits and when they are not quantized.

Our method leads to the best compression ratio when activations are quantized to 4 bits and wins over other methods in terms of accuracy except for HAWQ, even though we have used a weaker baseline FP model. Regarding comparison with HAWQ, one can note that the relative differences between the baseline and the resulting quantized accuracies are similar: 0.16% for HAWQ and 0.12% for our method, but the compression ratio for our method is much higher: 15.13 vs. 13.11. The proposed method outperforms other methods when activations are not quantized. The reason for only slight difference of models with and without our regularizers in Table 4 may lie in that both models are very close to the FP accuracy and therefore, are close to saturation in accuracy.
ImageNet We also test our method for quantization of ResNet-18, ResNet-50 [14], and MobileNet-v2 [20] to 4 and 8 bits on ImageNet [19], and compare it with other methods in Table 5. The accuracies of the baseline full precision (FP) models used by all the methods are noted in the table. We used the baseline with the highest accuracy for comparison with other methods. The obtained bit-width distributions are given in the Appendix E. The proposed approach performs significantly better then the other methods both in terms of accuracy and compression ratio. It reaches accuracy larger than the FP model for ResNet-50 and MobileNet-V2, while all other methods do not. It is worth noting that these results were obtained in less than 7 epochs for all the models.

6 Conclusions

In this paper, we propose a novel technique for mixed-precision quantization with explicit model size control, that is, the final model size can be specified by a user unlike in any other mixed-precision quantization method. In particular, we define the mixed-precision quantization problem as a constrained optimization problem and solve it together with soft regularizers, as well as a bit-width regularizer to constrain the quantization bit-widths to a pre-defined set. We validate the effectiveness of the proposed methods by conducting experiments on CIFAR10, ImageNet, and an image super resolution task, and show that the method reaches state-of-the-art results with no significant overhead compared to conventional QAT methods.

Table 4. Quantization of ResNet-20 on CIFAR-10. MC – model compression ratio (times), AC – activation compression ratio (times), FP Top-1 – the baseline FP model accuracy in %, Quant. Top-1 – Top-1 quantized accuracy in %, Difference – difference between quantized and FP Top-1 accuracy in %.

Method	MC	AC	FP Top-1	Quant. Top-1	Difference
MP DNNs [22]	14.97	8	92.71	91.40	-1.31
HAWQ [10]	13.11	8	92.37	**92.22**	-0.15
PDB [8]	11.94	8	91.60	90.54	-1.06
Ours (no regularizers)	**16.17**	8	91.73	91.55	-0.18
Ours (with regularizers)	15.13	8	91.73	91.62	-0.11
MP DNNs [22]	14.97	1	92.71	91.41	-1.3
DoReFa + SinReQ [11]	10.67	1	93.50	88.70	-4.8
Ours (no regularizers)	**16.19**	1	91.73	91.75	+0.02
Ours (with regularizers)	14.73	1	91.73	**91.97**	+0.24

Table 5. Quantization of ResNet-18, ResNet-50 and MobileNet-v2 on ImageNet. MC – model compression ratio (times), AC – activation compression ratio (times), FP Top-1 – the baseline FP model accuracy in %, Quant. Top-1 – Top-1 quantized accuracy in %, Difference – difference between quantized and FP Top-1 accuracy in %.

Method	MC	AC	FP Top-1	Quant. Top-1	Difference
ResNet-18					
MP DNNs [22]	4.24	2.9	70.28	70.66	+0.38
LSQ [13]	4.00	4	70.50	71.10	+0.6
HAWQ-V3 [25]	4.02	4	71.47	71.56	+0.09
Ours	**4.40**	4	71.47	**71.81**	+0.34
DoReFa + WaveQ	7.98	8	70.10	70.00	-0.1
FracBits-SAT	7.61	8	70.20	70.60	+0.4
MP DNNs	8.25	8	70.28	70.08	-0.2
DoReFa + SinReQ	7.61	8	70.50	64.63	-5.87
HAWQ-V3	7.68	8	71.47	68.45	-3.02
Ours	**8.57**	8	71.47	**70.64**	-0.83
ResNet-50					
LSQ [13]	4.00	4	76.90	76.80	-0.1
HAWQ-V3 [25]	3.99	4	77.72	77.58	-0.14
Ours	**4.33**	4	79.23	**79.45**	+0.22
HAWQ-V3	7.47	8	77.72	74.24	-3.48
Ours	**8.85**	8	79.23	**76.38**	-2.85
MobileNet-V2					
MP DNNs [22]	**4.21**	2.9	70.18	70.59	+0.41
HAQ [23]	4.00	4	71.87	71.81	-0.06
Ours	4.13	4	71.87	**71.90**	+0.03

References

1. Banner, R., Nahshan, Y., Soudry, D.: Post training 4-bit quantization of convolutional networks for rapid-deployment. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper/2019/file/c0a62e133894cdce435bcb4a5df1db2d-Paper.pdf
2. Bengio, Y., Léonard, N., Courville, A.C.: Estimating or propagating gradients through stochastic neurons for conditional computation. CoRR **abs/1308.3432** (2013), http://arxiv.org/abs/1308.3432
3. Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Zeroq: A novel zero shot quantization framework. CoRR **abs/2001.00281** (2020), http://arxiv.org/abs/2001.00281
4. Cai, Z., Vasconcelos, N.: Rethinking differentiable search for mixed-precision neural networks (2020)
5. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I., Srinivasan, V., Gopalakrishnan, K.: PACT: parameterized clipping activation for quantized neural networks. CoRR **abs/1805.06085** (2018)
6. Choi, Y., El-Khamy, M., Lee, J.: Learning low precision deep neural networks through regularization (2018)
7. Choukroun, Y., Kravchik, E., Kisilev, P.: Low-bit quantization of neural networks for efficient inference. CoRR **abs/1902.06822** (2019), http://arxiv.org/abs/1902.06822
8. Chu, T., Luo, Q., Yang, J., Huang, X.: Mixed-precision quantized neural network with progressively decreasing bitwidth for image classification and object detection. CoRR **abs/1912.12656** (2019), http://arxiv.org/abs/1912.12656
9. 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)
10. Dong, Z., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: HAWQ: hessian aware quantization of neural networks with mixed-precision. CoRR **abs/1905.03696** (2019), http://arxiv.org/abs/1905.03696
11. Elthakeb, A.T., Pilligundla, P., Esmaeilzadeh, H.: Sinreq: Generalized sinusoidal regularization for low-bitwidth deep quantized training (2019)
12. Elthakeb, A.T., Pilligundla, P., Mireshghallah, F., Yazdanbakhsh, A., Esmaeilzadeh, H.: Releq: A reinforcement learning approach for deep quantization of neural networks (2020)
13. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. CoRR **abs/1902.08153** (2019), http://arxiv.org/abs/1902.08153
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
15. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: Training neural networks with low precision weights and activations (2016)
16. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
17. Nagel, M., van Baalen, M., Blankevoort, T., Welling, M.: Data-free quantization through weight equalization and bias correction. CoRR **abs/1906.04721** (2019), http://arxiv.org/abs/1906.04721

18. Naumov, M., Diril, U., Park, J., Ray, B., Jablonski, J., Tulloch, A.: On periodic functions as regularizers for quantization of neural networks (2018)
19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. CoRR **abs/1409.0575** (2014), http://arxiv.org/abs/1409.0575
20. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks (2018)
21. 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. CoRR **abs/1609.05158** (2016), http://arxiv.org/abs/1609.05158
22. Uhlich, S., Mauch, L., Yoshiyama, K., Cardinaux, F., García, J.A., Tiedemann, S., Kemp, T., Nakamura, A.: Differentiable quantization of deep neural networks. CoRR **abs/1905.11452** (2019), http://arxiv.org/abs/1905.11452
23. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision (2019)
24. Wu, B., Wang, Y., Zhang, P., Tian, Y., Vajda, P., Keutzer, K.: Mixed precision quantization of convnets via differentiable neural architecture search. CoRR **abs/1812.00090** (2018)
25. Yao, Z., Dong, Z., Zheng, Z., Gholami, A., Yu, J., Tan, E., Wang, L., Huang, Q., Wang, Y., Mahoney, M.W., Keutzer, K.: HAWQV3: dyadic neural network quantization. CoRR **abs/2011.10680** (2020), https://arxiv.org/abs/2011.10680
26. 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)
27. Zur, Y., Baskin, C., Zheltonozhskii, E., Chmiel, B., Evron, I., Bronstein, A.M., Mendelson, A.: Towards learning of filter-level heterogeneous compression of convolutional neural networks (2019). https://doi.org/10.48550/ARXIV.1904.09872, https://arxiv.org/abs/1904.09872