

One-Step Forward and Backtrack: Overcoming Zig-Zagging in Loss-Aware Quantization Training

Lianbo Ma¹, Yuee Zhou¹, Jianlun Ma¹, Guo Yu^{2*}, Qing Li³

¹Software College, Northeastern University, Shenyang, China

²Institute of Intelligent Manufacturing, NanJing Tech University, Nanjing, China

³Peng Cheng Laboratory, Shenzhen, China

malb@swc.neu.edu.cn, {zhouyuee, jianlunma}@stumail.neu.edu.cn, guo.yu@njtech.edu.cn, liq@pcl.ac.cn

Abstract

Weight quantization is an effective technique to compress deep neural networks for their deployment on edge devices with limited resources. Traditional loss-aware quantization methods commonly use the quantized gradient to replace the full-precision gradient. However, we discover that the gradient error will lead to an unexpected zig-zagging-like issue in the gradient descent learning procedures, where the gradient directions rapidly oscillate or zig-zag, and such issue seriously slows down the model convergence. Accordingly, this paper proposes a one-step forward and backtrack way for loss-aware quantization to get more accurate and stable gradient direction to defy this issue. During the gradient descent learning, a one-step forward search is designed to find the trial gradient of the next-step, which is adopted to adjust the gradient of current step towards the direction of fast convergence. After that, we backtrack the current step to update the full-precision and quantized weights through the current-step gradient and the trial gradient. A series of theoretical analysis and experiments on benchmark deep models have demonstrated the effectiveness and competitiveness of the proposed method, and our method especially outperforms others on the convergence performance.

Introduction

With the increase of computation and storage consumption in various cloud and edge computing applications, it is challenging to deploy deep neural networks (DNNs) on computation-constrained devices (Liu and Mattina 2019; Jain et al. 2018). To enhance the efficiency of DNNs (e.g., low-power inference), it is critical to compress deep models with comparable performance (Xu et al. 2018; Dai et al. 2019). Typical compression approaches include low-rank decomposition (Denton et al. 2014; Yu et al. 2017), model pruning (He, Zhang, and Sun 2017; Lin et al. 2017; Luo, Wu, and Lin 2017; Peng et al. 2019; Ye et al. 2018), knowledge distillation (Hinton, Vinyals, and Dean 2015; Romero et al. 2014; Zhang, Chen, and Sun 2021; Shen et al. 2021), and low-bit quantization (Choi, El-Khamy, and Lee 2016; Han, Mao, and Dally 2015; Lin et al. 2019; Peng and Chen 2019; Rastegari et al. 2016).

In this paper, we focus on the compression approaches of low-bit quantization, which are to reduce the model size with no or minor performance degradation. Specifically, the low-bit quantization is to quantize full-precision weights (or activations) to low bitwidth ones (Rastegari et al. 2016; Howard et al. 2017). To this end, pioneer quantization approaches strive to find the closest low-precision approximation of the full-precision weights during the model learning process (Peng et al. 2021; Hinton et al. 2012). However, these approaches neglect the effect of quantization on the final loss, so that they may suffer from the severe accuracy degradation problem (Fu et al. 2020; Zhang et al. 2018c). This problem becomes even more serious if the extremely low bitwidth fixed point (e.g., 1-bit) representation is used in the quantization (Bai, Wang, and Liberty 2018). Therefore, many studies on loss-aware quantization (LAQ) have proposed to solve the above issue (Hou and Kwok 2018; Hou et al. 2017; Zhou et al. 2018; Qu et al. 2020; Saito and Mitsui 1996), and the main idea of LAQ is to optimize the quantized weights rather than the full-precision ones during the minimization of the loss function.

However, we discover that existing LAQ methods (Hou et al. 2017; Zhou et al. 2018) may suffer from the zig-zagging-like issue that the gradient directions rapidly oscillate or zig-zag during the gradient descent learning procedures, and such issue seriously slows down the model convergence. An example is shown in Fig.1 (a) that the quantized gradient direction at each step tends to be zig-zag or rapidly oscillated and the search with quantized weights needs more epochs to get converged in comparison with the search with full-precision weights. In some extreme cases, such issue even makes the search fail to converge and end up with oscillation. Unfortunately, most recently published studies do not realize this issue but compensates for the gradient quantization error by enhancing quantization representations (Lin et al. 2019; Peng et al. 2021; Zhou et al. 2018; Qu et al. 2020). Due to the big gap between extremely low-bit (e.g., 1-bit) representations and full-precision (e.g., 32-bit) ones, it is hard to make full compensation of quantization error.

Inspired by the numerical analysis theory (Saito and Mitsui 1996) that iteratively using the trial results backtracked from next-step search to update next-step items can contribute to the numerical stability, we try to tackle the above

*Corresponding author.

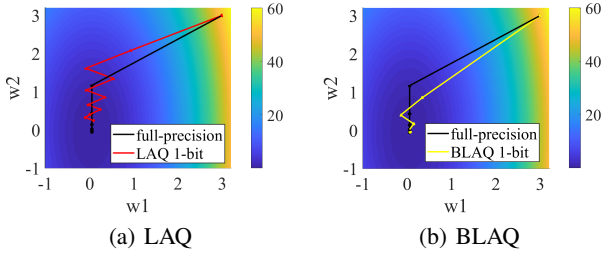


Figure 1: Trajectories of weight updating obtained by LAQ and BLAQ in 1-bit quantization on the loss function $\ell(\omega) = 5(\omega_1 - 0.054)^2 + (\omega_2 + 0.055)^2$. The black line shows the evolution of full-precision weight values via iterations during training, while the red and yellow lines are the evolution of 1-bit quantization case.

zig-zagging-like issue in a different way, i.e., improving the quantization updating rules to get more accurate and stable gradient direction. In other words, the key to reduce the zig-zagging issue in this study is to explore trial information from the next-step exploratory search and take them as extra gradient compensation to enhance the stability of weight updating.

Following the above idea, we propose an effective backtracking-search loss-aware quantization (BLAQ) method, which optimizes weights in a one-step forward and backtrack way to get more accurate and stable gradient direction to defy the zig-zagging-like issue. During the gradient descent learning, the one-step forward search is to find a trial gradient of the next step to drive the gradient of current step towards the direction of fast convergence. After that, we backtrack the current step to update the full-precision and quantized weights through the current-step gradient and the trial gradient. As a result, the estimation of gradient direction is more accurate and stable than the traditional methods (Lin et al. 2019; Peng et al. 2021; Zhou et al. 2018; Qu et al. 2020). A pilot study is shown in Fig.1 (b) that BLAQ is able to effectively and efficiently reduce redundant zig-zagging steps and converged faster in comparison with LAQ.

The main contributions of this paper include:

- We discover the zig-zagging-like issue in the search of quantized weights via LAQ, and find that the issue can seriously slow down the model convergence.
- We propose a novel loss-aware quantization method to defy the zig-zagging-like issue, which generates low-bit quantized network in a one-step forward and backtrack way. Therefore, a new quantization framework with good convergence property is provided in this study.
- We provide theoretical analysis to show that the proposed quantization is mathematically better than other counterparts in convergence properties, and the effectiveness of our method is also verified on a set of deep models.

Preliminaries

Notations for Weight Quantization

In this paper, we use W to represent a full-precision neural network, and \hat{W} to represent a quantization neural network. For a convolution neural network, its full-precision weights of all L layers are defined as $W=(W_1, W_2, \dots, W_L)$, where $\omega_l = \text{vec}(W_l)$ and $W_l \in R^{h \times h \times c_i \times c_o}$ is the weight tensor at layer l . Here, h , c_i and c_o are the size of convolution kernel, the numbers of input and output channels, respectively. The corresponding quantized weights are denoted by $\hat{W}=(\hat{W}_1, \hat{W}_2, \dots, \hat{W}_L)$, where $\hat{\omega}_l = \text{vec}(\hat{W}_l)$ and \hat{W}_l is a quantization weight tensor at layer l . Here, $\text{vec}(X)$ outputs a vector by stacking the columns of a matrix X , and $\text{diag}(X)$ a diagonal matrix, where the diagonal elements are constructed by the diagonal of X . We refer to D_l and \hat{D}_l as the approximated diagonal Hessian matrix for W_l and \hat{W}_l , respectively. Q is a set of $2k$ quantized values, where $Q = \{-1, \dots, -\frac{1}{2^{k-1}}, -\frac{1}{2^k}, \frac{1}{2^k}, \frac{1}{2^{k-1}}, \dots, 1\}$. When $k = 1$, $Q = \{-1, 1\}$. “ \oslash ” is used as elementwise division and “ \odot ” as elementwise multiplication. $\langle x, y \rangle$ is the inner product of vectors x and y . Besides, $\|\omega\|_D^2 = \omega^T D \omega$, and $[\omega]_Q$ denotes the rounding entries of ω to the closed fixed point in Q .

Loss-Aware Weight Quantization

In the forward propagation, LAQ (Hou et al. 2017) utilizes the second-order Taylor expansion to quantize the full-precision weights, and then uses the proximal Newton method to solve the optimization problem (Rakotomamonjy, Flamary, and Gasso 2015). At iteration t , the objective function of LAQ is replaced by the second-order series expanded at $\hat{\omega}^t = \alpha^t \beta^t$

$$\begin{aligned} \min \ell(\hat{\omega}^{t-1}) + \hat{g}^{T^{t-1}}(\hat{\omega}^t - \hat{\omega}^{t-1}) \\ + \frac{1}{2}(\hat{\omega}^t - \hat{\omega}^{t-1})^T \hat{H}^{t-1}(\hat{\omega}^t - \hat{\omega}^{t-1}), \quad (1) \\ \text{s.t. } \hat{\omega}^t = \alpha^t \beta^t, \alpha^t = \frac{\|\omega^t\|}{n}, \beta^t = \text{sign}(\omega^t), \end{aligned}$$

where \hat{g}^{t-1} and \hat{H}^{t-1} are the first and second derivatives of the loss function $\ell(\hat{\omega}^{t-1})$ with respect to quantized weight $\hat{\omega}^{t-1}$, respectively.

For DNNs, the Hessian matrix H^{t-1} is rarely positive semi-definite and intractable to calculate. Hence, the diagonal Hessian matrix D^{t-1} is suggested to approximate H^{t-1} (i.e., $D = \text{diag}(H)$) (Kingma and Ba 2014). Therefore, Eq. 1 can be rewritten as (The derivation process is provided in the Section 2 of the Appendix ¹).

$$\arg \min_{\hat{\omega}^t} \frac{1}{2} \|\omega^t - \hat{\omega}^t\|_{\hat{D}^{t-1}}^2, \quad (2)$$

where

$$\omega^t = \hat{\omega}^{t-1} - \hat{g}^{t-1} \oslash \hat{D}^{t-1}. \quad (3)$$

Eq. 2 and Eq. 3 are used to update the quantized and full-precision weights, respectively.

¹You can refer to our appendix on the following website: <https://github.com/paperProof24/Appendix.BLAQ>

Zig-zagging-like Issue of Weight Updating

We discover that the zig-zagging-like issue easily occurs in LAQ methods that the gradient directions rapidly oscillate or zig-zag during the gradient descent learning procedures and the gradient quantization error seriously slow down the model convergence, especially for extremely low bit-width quantization. A pilot experiment is given in Fig. 1 (a) that the gradient direction is sharply oscillated and evolution takes many steps to get converged.

Theoretically, the full-precision weights ω^t and quantized weights $\hat{\omega}^t$ in Eqs. 2-3 are alternatively and iteratively computed. Note that \hat{g}^{t-1} related to the quantized gradient is used to update ω^t in Eq. 3. However, the loss of information from the low-bit quantization will make the quantization error about \hat{g}^{t-1} unavoidable accumulated over the iterations. Once the accumulated error reaches a threshold, the zig-zagging-like phenomenon² will occur. Even worse, Eqs. 2-3 may make the model fail to converge. Given a simple toy loss function $f(\omega) = c\omega^{\frac{3}{2}}$, if Eqs. 2-3 are used to calculate quantized weights, for any initial weights ω_0 , the optimization of $f(\omega)$ cannot converge and end in oscillating. The proof is provided in the Section 3 of the Appendix.

Notably, the zig-zagging phenomenon in our study is essentially different from the oscillation (Nagel et al. 2022). The zig-zagging-like issue mentioned in our work is essentially caused by the zig-zag fluctuation of the weights due to errors in the quantization of the gradient during training. Due to the inaccuracy of the quantized gradient, the weights are updated in such a way that they deviate from the original trajectory during training using the quantized gradient, leading to the zig-zagging-like issue. And the cause of the oscillation problem proposed by (Nagel et al. 2022) is the inaccurate calculation of full-precision weights due to truncation error and rounding error when the full-precision weights are close to the quantization decision boundaries, leading to inaccurate quantization results and recurrent oscillations of the quantization weights. The external manifestations of the two oscillation phenomena may be somewhat similar, but the internal causes are different, so different solutions need to be taken.

Zig-zagging-like Issue in Practice

The zig-zagging-like issue exists not only in the toy example presented in Fig. 1, but also in the quantization of large neural networks. An example is shown in Fig. 2 that a clear zig-zagging-like phenomenon during the quantization of ResNet18 on ImageNet using the LAQ method. In addition, as shown in Fig. 3, the zig-zagging-like phenomenon is significantly improved during the quantization of ResNet18 on ImageNet using our proposed BLAQ method.

Convergence Result Analysis

We also find that the lower the bitwidth of the quantization uses, the more intense the zig-zagging phenomenon of weights will be. The corresponding experimental results are

²In fact, the landscape feature of optimization problem is also a factor to generate zig-zagging phenomenon but this work only focuses on the gradient errors.

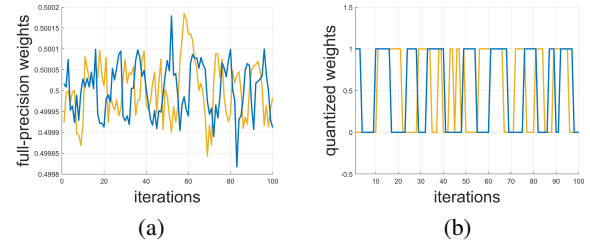


Figure 2: Regression of randomly selected weights during the last 100 epochs of training ResNet18 on ILSVRC12 (ImageNet) during LAQ 2-bit quantization. The (a) show the full-precision weights, and the (b) plot the corresponding quantized weights.

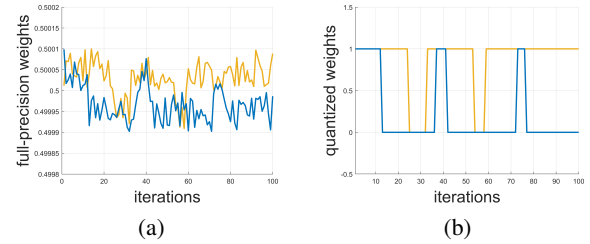


Figure 3: Regression with the same weights as Fig. 2 selected during the last 100 epochs of training ResNet18 on ILSVRC12 (ImageNet) during BLAQ 2-bit quantization. The (a) show the full-precision weights, and the (b) plot the corresponding quantized weights.

presented in Table 1 which shows that the maximum oscillation amplitude of the full-precision weights and oscillation frequencies of the quantized weights of the last 150 epochs during the training of ResNet18 on ILSVRC12 (ImageNet) using different quantization bitwidths. We can observe that when we use the lower bitwidth for quantization, the maximum oscillation amplitude of the full-precision weight becomes more intense, and the oscillation on the quantization weight becomes more frequent. In summary, the zig-zagging phenomenon is an important issue in the quantization, which can slow down or even prohibit the convergence of the model, especially in the case of extremely low-bit quantization.

Proposed Method

Loss-Aware Quantization via Two-stage Weight Updating

In order to defy the above issue, we follow the procedure of LAQ (Hou et al. 2017) and come up with a novel one-step forward and backtrack quantization framework, which is terms as BLAQ for convenience.

The principal process of BLAQ is presented in Fig. 4. In stage 1, the one-step forward search is performed. Given the real full-precision weights ω^t and quantized weights $\hat{\omega}^t$, we firstly use Eq. 6 to obtain the trial full-precision weights

Method	bitwidth	amplitude	frequency
LAQ	1-bit	0.000486	15
	2-bit	0.000342	11
	4-bit	0.000261	8
BLAQ	1-bit	0.000143	4
	2-bit	0.000102	3
	4-bit	0.000067	1

Table 1: The maximum oscillation amplitude of the full-precision weight and oscillation frequencies of the quantized weight for the same locked weight under the LAQ and BLAQ methods.

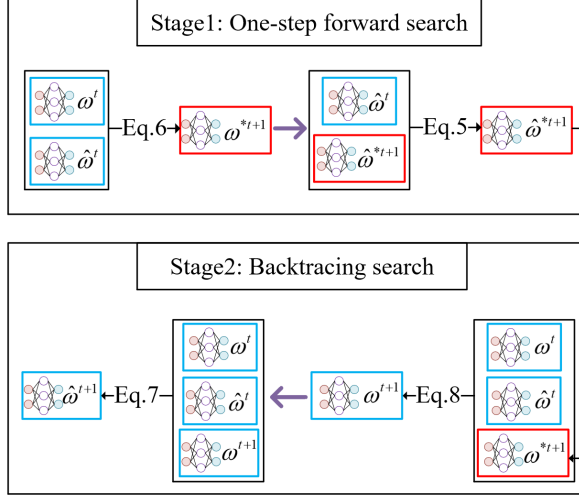


Figure 4: Two-stage updating process: the blue rectangles denote the real weights, and the red ones are the trial weights.

$\omega^{*(t+1)}$. Then, the real quantized weights and the trial full-precision weights are adopted to solve Eq. 5, so that the trial quantized weights $\hat{\omega}^{*(t+1)}$ can be achieved. In stage 2, the backtrack quantization is executed. We firstly apply Eq. 8 to get the full-precision weights ω^{t+1} on the basis of the acquired weights from stage 1. After that, Eq. 7 is solved to obtain the real quantized weights $\hat{\omega}^{t+1}$ by using the weights from the previous steps. The details of the proposed BLAQ is introduced as follows. The corresponding pseudocode of BLAQ is presented in Algorithm 1 of the Appendix.

Stage 1 (One-step forward search): We take one forward step to tentatively obtain the trial quantized weights $\hat{\omega}^{*(t+1)}$. Therefore, the second-order Taylor expansion centered at quantized weights $\hat{\omega}^{*(t+1)}$ is applied, and the loss function is minimized by Eq. 4. For convenience, we omit the subscript l here:

$$\begin{aligned}
 & \arg \min_{\hat{\omega}^{*(t+1)}} \ell(\omega^t) + \hat{g}^{tT} (\hat{\omega}^{*(t+1)} - \omega^t) \\
 & \quad + \frac{1}{2} (\hat{\omega}^{*(t+1)} - \omega^t)^T \hat{H}^t (\hat{\omega}^{*(t+1)} - \omega^t), \\
 & \text{s.t. } \hat{\omega}^{*(t+1)} = \alpha^{t+1} \beta^{t+1}, \\
 & \quad \alpha^{t+1} > 0, \beta^{t+1} = \text{sign}(\omega^{*(t+1)}),
 \end{aligned} \tag{4}$$

where \hat{g}^t and \hat{H}^t are the first and second derivations of the loss function $\ell(\hat{\omega}^t)$ with respect to the $\hat{\omega}^t$, respectively. Besides binarization, our method is also suitable for multi-bit quantization if we reset the value of the β parameter in Eq. 4. If the diagonal Hessian matrix \hat{D}^t is used to approximate \hat{H}^t , Eq. 4 can be reformulated as

$$\arg \min_{\hat{\omega}^{*(t+1)}} \frac{1}{2} \|\omega^{*(t+1)} - \hat{\omega}^{*(t+1)}\|_{\hat{D}^t}^2, \tag{5}$$

where

$$\omega^{*(t+1)} = \omega^t - \hat{g}^t \circ \hat{D}^t. \tag{6}$$

Here α and β are solved by using an alternating update approach, the corresponding pseudocode is presented in Algorithm 2 of the Appendix.

Stage 2 (Backtracking search): After stage 1, we will acquire the corresponding trial quantized weights $\hat{\omega}^{*(t+1)}$, trial quantized gradient $\hat{g}^{*(t+1)}$ and trial diagonal Hessian matrix $\hat{D}^{*(t+1)}$. Then, we backtrack to re-compute the real quantized weights $\hat{\omega}^{t+1}$ as follows:

Following the derivation of original LAQ (i.e., Eqs. 1-3), the optimization formula at stage 2 can be inferred as

$$\arg \min_{\hat{\omega}^{t+1}} \frac{1}{2} \|\omega^{t+1} - \hat{\omega}^{t+1}\|_{\hat{D}^{t+1}}^2, \tag{7}$$

where

$$\omega^{t+1} = \omega^t - \hat{g}^{t+1} \circ \hat{D}^{t+1}. \tag{8}$$

After that, we will compute the real quantized gradients (i.e., \hat{g}^{t+1} and \hat{D}^{t+1}) in Eq. 8 through the real quantized gradients (i.e., \hat{g}^t and \hat{D}^t) at (t) th step, and the trial quantized gradients (i.e., $\hat{g}^{*(t+1)}$ and $\hat{D}^{*(t+1)}$) at $(t+1)$ th step.

$$\hat{g}^{t+1} = a\hat{g}^t + (1-a)\hat{g}^{*(t+1)}, \tag{9}$$

$$\hat{D}^{t+1} = a\hat{D}^t + (1-a)\hat{D}^{*(t+1)}, \tag{10}$$

where a is a coefficient and determined in the ablation study, (which is presented in the Section 6 of the Appendix). Such backtrack-search approach is able to get a more accurate estimation of the quantized gradient \hat{g}^{t+1} .

According to Algorithm 1 of the Appendix, we can obtain the real quantized weights $\hat{\omega}^{t+1}$ by the proposed one-step forward and backtrack quantization framework, i.e., solving the Eqs. 7-8. In Section 4 of the Appendix, we have also theoretically verified that BLAQ will not oscillate on the simple toy experiment, where the traditional LAQ method does.

Convergence Analysis

Following the common assumptions (Peng et al. 2021), i.e., the loss function $\ell(\omega)$ is convex, twice differentiable, L_1 -smooth and μ -strongly convex with respect to ω , we give the following analysis.

Theorem 1. For the loss function $\ell(\omega)$ with the learning rate η^t , the convergence of our method is as

$$\ell(\omega^{t+1}) - \ell(\omega^*) \leq \frac{L_1 + L_1^3 (\eta^{t+1})^2 - 2\mu^2 \eta^{t+1}}{2} \Delta^2. \tag{11}$$

Theorem 2. When $\frac{2}{L_1 \eta} - 1 < a < 1$ holds, the convergence of our method is better than that of LAQ (Hou et al. 2017).

The corresponding proof is presented in the Section 5 of the Appendix.

Experiment

Experimental Setup

Four widely-used datasets (CIFAR10, MNIST, SVHN and ILSVRC12 (ImageNet)) are applied to validate the performance of BLAQ. The baseline methods for comparison include: 1) BC (Courbariaux, Bengio, and David 2015), 2) BWN (Rastegari et al. 2016), 3) LAQ (Hou et al. 2017), 4) TRQ (Li et al. 2021), 5) AQ (Khoram and Li 2018), 6) ALQ (Qu et al. 2020), 7) DC (Han, Mao, and Dally 2015), 8) ADMM (Zhang et al. 2018b), 9) LR (Shayer, Levi, and Fetaya 2018), 10) DSQ (Gong et al. 2019), 11) TWN (Li et al. 2016), 12) LQ-Net (Zhang et al. 2018a), 13) QIL (Jung et al. 2019), and 14) OCTAV (Sakr et al. 2022). For deep network architectures, we conduct experiments with VGG (Hou et al. 2017) on CIFAR10, with LeNet5 (Qu et al. 2020) and 4-layer Model (Hou et al. 2017) on MNIST, with SVHN-Net (Hou et al. 2017) on SVHN, and with ResNet18 (Qu et al. 2020) on ILSVRC12. See Section 7 of the Appendix for more details. In fact, although the optimal values of the hyperparameters a and m vary across different datasets and models, we conducted extensive ablation experiments and ultimately concluded that for small datasets and models, the optimal values of a and m are typically 0.6 and 5. For large datasets and models, the optimal values of a and m are usually 0.9 and 10. See Section 7 of the Appendix for more details. Therefore, in BLAQ, we set the hyperparameters a and m to 0.6 and 5 in the datasets CIFAR10, MNIST, and SVHN, and set the hyperparameters a and m to 0.9 and 10 in the dataset ILSVRC12.

Effectiveness of BLAQ

Fig. 5 presents the training loss of different methods with the increase of the number of epochs. The results indicate that there is an obvious gap between the baseline (LAQ) and backtracking-search updating approach (BLAQ). Specifically, the training loss curve of LAQ fluctuates more drastically, and also shows slower convergence velocity than BLAQ. It may result from the quantization error from \hat{g}^{t-1} in Eq. 3, which is the fundamental cause of zig-zagging-like issue. By contrast, BLAQ achieves a more accurate gradient estimation during the weight training, which not only accelerates the training convergence, but also benefits the test accuracy as shown in Tables 3-6.

Convergence Result Analysis

Fig. 6 shows the test accuracy versus training epochs on three datasets. The results show that BLAQ not only converges more stable and faster, but also gets a higher accuracy than LAQ. For example, BLAQ reaches to a high level of accuracy in few epochs on CIFAR10, but LAQ has to take more epochs to achieve similar accuracy. Similar observations can be found on MNIST and SVHN. Such encouraging results are consistent with Theorem 2. Therefore, it can be concluded that the competitive performance of BLAQ may be attributed to the backtracking-based two-stage updating, which can reduce the error caused by the gradient approximation. To ensure fair comparisons, all our comparisons are based on the results published by the original au-

Methods		Top-1 accuracy
Full-precision	32-bit	92.8%
BC	1-bit	90.1%
BWN	1-bit	89.5%
LAQ	1-bit	89.5%
TRQ*	ternary	91.2%
AQ	multi-bit	90.9%
ALQ	multi-bit	90.9%
BLAQ (ours)	1-bit	91.5%

Table 2: Top-1 accuracy(VGG on CIFAR10).

thor of LAQ, while the author of LAQ didn’t conduct experiments on ILSVRC12 (ImageNet). Our BLAQ method is an improvement of the LAQ method. Fig. 5 and Fig. 6 show a comparison of the BLAQ and LAQ training processes. Since LAQ has no experimental data on ILSVRC12 (ImageNet), Fig. 5 and Fig. 6 don’t show the comparison of the BLAQ and LAQ training processes on ILSVRC12 (ImageNet).

Experimental Results on CIFAR10

In this section, we firstly compare the BLAQ with the state-of-the-art binarization baselines on CIFAR10. Table 2 has demonstrated the performance of the proposed method against the baselines on CIFAR10 in terms of Top-1 accuracy. From this table, we find that our model achieves significantly better results than these weight binarization methods (BC, BWN and LAQ). For the popular 1-bit weight quantization approaches, our method obtains at least 1.4% improvement over BC, BWN and LAQ. Compared with the latest weight ternary method (TRQ), our method achieves better accuracy with almost 0.3% improvement. These encouraging results validate the effectiveness of the proposed one-step forward and backtrack way in handling network binarization task.

Furthermore, we also compare our method with the adaptive multi-bit quantizers (AQ and ALQ), which assign different bitwidth for each parameter and prune these parameters according to their importance. The results in Table 2 show that our method obtains similar or even better accuracy without any expense of additional operations like pruning and multi-bit computation. Besides, our method outperforms them with almost 0.6% improvement, especially. The results reveal that our method without any additional operations is able to get competitive performance.

Experimental Results on MNIST

Here, the proposed method is compared with two groups of baselines on MNIST: 1) Non-pruning binarization methods (i.e., BC, BWN and LAQ); 2) Pruning-based quantization methods (i.e., ALQ, DC, and ADMM). For the first group, all methods are based on a 4-layer full-connected model; For the second group, all methods are based on a modified LeNet5 model.

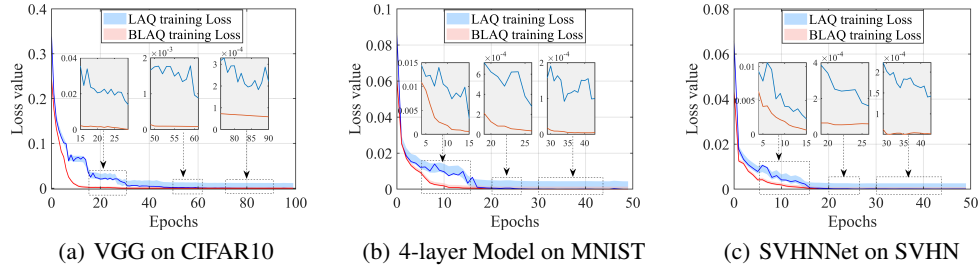


Figure 5: Training loss trained with LAQ/baseline and BLAQ. The red curve shows the evolution of mean loss values obtained by BLAQ, while the blue one by LAQ. The shaded areas indicate the fluctuation (error) range of the mean loss curve.

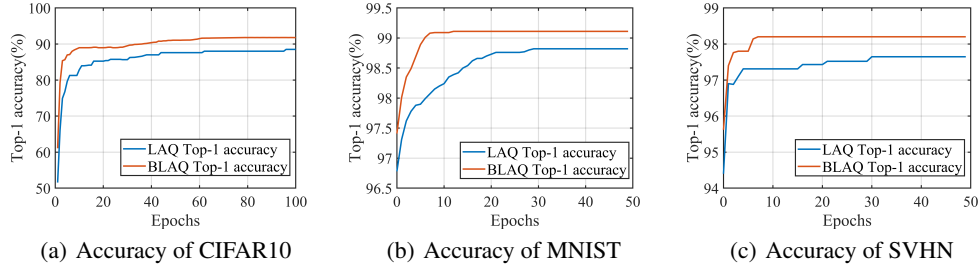


Figure 6: Test convergence obtained by LAQ/baseline and BLAQ.

Methods		Top-1 accuracy
Full-precision	32-bit	98.81%
BC	1-bit	98.72%
BWN	1-bit	98.69%
LAQ	1-bit	98.82%
BLAQ (ours)	1-bit	99.11%

Table 3: Top-1 accuracy (4-layer Model on MNIST)

Methods		Top-1 accuracy
Full-precision	32-bit	99.19%
DC	Multi-bit	99.26%
ADMM	Multi-bit	99.20%
ALQ	Multi-bit	99.12%
BLAQ (ours)	1-bit	99.38%

Table 4: Top-1 accuracy (LeNet on MNIST).

Table 3 shows the comparison results in terms of Top-1 accuracy between BLAQ and the classical non-pruning binarization baselines (BC, BWN and LAQ) on MNIST. The table shows that our method outperforms them with at least 0.29% improvement in terms of the Top-1 accuracy, which may result from the fact BLAQ has reduced the quantization error to large extent.

Table 4 shows the comparison results with three pruning-based binarization methods on MNIST. It can be observed that our method gains consistently best performance over the counterparts although they utilize additional compression-related operations to assist the quantization. For example, DC and ADMM use sparse tensors, which require special li-

braries or hardware for execution, while ALQ needs to prune networks during the multi-bit quantization. By contrast, our method is a pure binarization using no additional pruning-related operations, and more suitable to generic off-the-shelf platform. More importantly, our method achieves better results than the full-precision approaches, which again shows its promising capability in dealing with the low-bit quantization problem.

Experimental Results on SVHN

In this section, we further explore the performance of our method on the well-known model (SVHN) used in LAQ (Hou and Kwok 2018). Table 5 shows the test accuracy on SVHN. Similar to Table 3, BLAQ performs better than other binarization methods, and even better than the full-precision approach. For example, in comparison with BLAQ, the contrast algorithm (BWN) does not quantize the first layer and the last layer, but still gets unsatisfactory test accuracy. In addition, BLAQ achieves better performance than the approach with full-precision weights, which may be attributed to the reduction of the redundant weights in BLAQ. Overall, BLAQ also shows competitive performance on SVHN.

Experimental Results on ILSVRC12 (ImageNet)

Note that a large batch size in large-scale datasets may influence the direction of gradient descent, as large batch size tends to help increase the model stability but decrease the generalization ability. Hence, we conduct an experiment with large batch size and quantize the weights of ResNet18 network model with low-bit (1-bit and 2-bit) on ILSVRC12 dataset, where ResNet18 is widely-used in the quantization

Methods		Top-1 accuracy
Full-precision weights	32-bit	97.73%
BC	1-bit	97.55%
BWN	1-bit	97.46%
LAQ	1-bit	97.64%
BLAQ (ours)	1-bit	98.13%

Table 5: Top-1 accuracy (SVHNNet on SVHN).

Methods		Top-1 accuracy
Full-precision	32-bit	69.8%
BWN	1-bit	60.8%
LR*	1-bit	59.9%
DSQ*	1-bit	63.7%
ALQ	1.01-bit	65.6%
BLAQ(ours)	1-bit	66.73%
TWN	2-bit	61.8%
LR	2-bit	63.5%
LQ-Net*	2-bit	68.0%
QIL*	2-bit	68.1%
ALQ	2-bit	68.9%
OCTAV	4-bit	69.17%
BLAQ(ours)	2-bit	69.62%

Table 6: Top-1 accuracy (ResNet18 on ILSVRC12).

for ILSVRC12 dataset. In this experiment, BLAQ is compared with the state-of-the-art low-bit networks.

Table 6 shows the Top-1 accuracy results of all methods on ILSVRC12. The results show that our method achieves the highest Top-1 accuracy, and outperforms the baselines with at least 1.13% and 0.72% improvements on the cases of 1-bit and 2-bit, respectively. In contrast to the popular schemes like BWN, LR, DSQ, and LQ-Net, which do not quantize the first and last layers to ensure the quantization performance since the quantization of the two layers may cause a huge accuracy degradation, the proposed BLAQ uniformly quantizes the first and last layers with 8-bit but still gets better results. In addition, we also find that the recently published ALQ is not extremely low-bit in the strict sense in the quantization, where ALQ sets the bitwidth of all layers to 8-bit, which is different from our method that has physically realized 1-bit and 2-bit quantization. The above results show that BLAQ is also able to get competitive performance in handling the quantization of deep neural networks on large-scale datasets.

Related Work

In the network quantization, early studies (Wang et al. 2018; Deng et al. 2009) use 8-bit fixed point representation, which achieves the state-of-the-art performance on ILSVRC12. Later, more efforts are made to the extremely low-bit quantization for DNNs compression (Wang et al. 2018; Deng et al. 2009; Lin, Zhao, and Pan 2017; Liu et al. 2018; Li et al. 2016), which mainly discretizes activations and weights of DNNs into binary or ternary values (e.g., $\{-1, +1\}$ or $\{-1, 0, +1\}$). However, simply using bit-wise operations to approximate the convolution operations of DNNs may suffer from

performance deterioration. To improve the model accuracy, some quantization improvements are come up with, such as learnable quantizing (Zhang et al. 2018a) and non-uniform logarithmic representation (Miyashita, Lee, and Murmann 2016). In principle, previous studies treat the quantization problem as a straightforward approximation of original full-precision weights (i.e., minimizing the error between low-bit quantized weights and full-precision ones), but ignore the effect of quantization on the loss.

To overcome the above limitation, the research (Hou et al. 2017) proposes a loss-aware quantization approach, directly optimizing binarized weights to minimize the final loss. The following work (Hou and Kwok 2018) extends this binarization scheme to m -bit quantization, but Peng et al. (Peng et al. 2021) have discovered that the above approaches under certain conditions may fail to converge due to the quantization error. Zhou et al. (Zhou et al. 2018) have proposed an explicit loss-aware weight quantization method by integrating the information of loss function with respect to full-precision weights into the quantization. But this method neglects the curvature information of the loss function and quantization errors, when updating the full-precision weights (Peng et al. 2021). In this study, we discover that quantization error in LAQ may lead to severe zig-zagging-like issue and seriously slow down the model convergence, especially for extremely low bit-width quantization. Therefore, our focus is not only to compensate for gradient computation error, but also to defy the zig-zagging-like problem in the loss-aware quantization process. Consequently, our work aims to improve the iterative updating rules for weights or gradient according to the numerical stability theory, so that a new quantization framework using backtracking-based updating principle is proposed.

Conclusion

In this paper, we discover that the quantization error in LAQ may trigger a serious zig-zagging-like issue, which can severely slow down the model convergence. To handle the above issue, we propose a backtracking-search loss-aware quantization method for low-bit quantization. The main idea is to utilize potential information obtained from next-step exploratory search to compensate for the gradient error during the optimization. Specifically, our approach works in a one-step forward and backtrack way: At each iteration, the search first explores one forward step to find the trial gradient at the next-step, which can be adopted to assist in adjusting the gradient at current step towards the direction of fast convergence. Then we backtrack to update the quantized gradient using current gradient and trial gradient. This way is able to effectively and efficiently solve the zig-zagging-like issue. A number of theoretical analysis have validated the effectiveness of our approach in convergence. In addition, the experimental results in Tables 2-6 have also shown that our method performs well across different network models on three datasets, and achieves a consistent performance improvement over its counterparts. These results have demonstrated that our method is more robust than the previous quantization methods to tackle complicated network binarization scenarios with different network models.

Acknowledgments

This work is supported by the NSFC Key Supported Project of the Major Research Plan Grant (No.92267206), the National Natural Science Foundation of China (No. 62032013), the Key Technologies R&D Program of Liaoning Province (2023JH1/10400082, 2023020456-JH/104), the National Natural Science Foundation of China (No. 62103150, 62333010), and China Postdoctoral Science Foundation (No.2021M691012).

References

- Bai, Y.; Wang, Y.-X.; and Liberty, E. 2018. ProxQuant: Quantized Neural Networks via Proximal Operators. In *Proceedings of the 6th International Conference on Learning Representations*.
- Choi, Y.; El-Khomy, M.; and Lee, J. 2016. Towards the limit of network quantization. *arXiv preprint arXiv:1612.01543*.
- Courbariaux, M.; Bengio, Y.; and David, J.-P. 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 28: 3123–3131.
- Dai, X.; Zhang, P.; Wu, B.; Yin, H.; Sun, F.; Wang, Y.; Dukhan, M.; Hu, Y.; Wu, Y.; Jia, Y.; et al. 2019. Chamnet: Towards efficient network design through platform-aware model adaptation. In *Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11398–11407.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 22th Conference on Computer Vision and Pattern Recognition*, 248–255. IEEE.
- Denton, E. L.; Zaremba, W.; Bruna, J.; LeCun, Y.; and Fergus, R. 2014. Exploiting linear structure within convolutional networks for efficient evaluation. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 27.
- Fu, F.; Hu, Y.; He, Y.; Jiang, J.; Shao, Y.; Zhang, C.; and Cui, B. 2020. Don't waste your bits! squeeze activations and gradients for deep neural networks via tinsycript. In *Proceedings of the 37th International Conference on Machine Learning*, 3304–3314. PMLR.
- Gong, R.; Liu, X.; Jiang, S.; Li, T.; Hu, P.; Lin, J.; Yu, F.; and Yan, J. 2019. Differentiable Soft Quantization: Bridging Full-Precision and Low-Bit Neural Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4851–4860.
- Han, S.; Mao, H.; and Dally, W. J. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*.
- He, Y.; Zhang, X.; and Sun, J. 2017. Channel pruning for accelerating very deep neural networks. In *Proceedings of the 16th IEEE International Conference on Computer Vision*, 1389–1397.
- 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. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6): 82–97.
- Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Hou, L.; and Kwok, J. T. 2018. Loss-aware Weight Quantization of Deep Networks. In *Proceedings of the 6th International Conference on Learning Representations*.
- Hou, L.; Yao, Q.; Kwok, J.; and Tin, Y. 2017. Loss-Aware Binarization of Deep Networks. In *Proceedings of the 5th International Conference on Learning Representations*.
- Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Jain, S.; Venkataramani, S.; Srinivasan, V.; Choi, J.; Chuang, P.; and Chang, L. 2018. Compensated-DNN: Energy efficient low-precision deep neural networks by compensating quantization errors. In *Proceedings of the 55th Annual Design Automation Conference*, 1–6.
- Jung, S.; Son, C.; Lee, S.; Son, J.; Han, J.-J.; Kwak, Y.; Hwang, S. J.; and Choi, C. 2019. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4350–4359.
- Khoram, S.; and Li, J. 2018. Adaptive quantization of neural networks. In *Proceedings of the 6th International Conference on Learning Representations*.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Li, F.; Liu, B.; Wang, X.; Zhang, B.; and Yan, J. 2016. Ternary weight networks. *arXiv preprint arXiv:1605.04711*.
- Li, Y.; Ding, W.; Liu, C.; Zhang, B.; and Guo, G. 2021. Trq: Ternary neural networks with residual quantization. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 8538–8546.
- Lin, D.; Annapureddy, V. S. R.; Howard, D. E.; Julian, D. J.; Majumdar, S.; and BELL, I. W. R. 2019. Fixed point neural network based on floating point neural network quantization. US Patent 10,373,050.
- Lin, J.; Rao, Y.; Lu, J.; and Zhou, J. 2017. Runtime neural pruning. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 30.
- Lin, X.; Zhao, C.; and Pan, W. 2017. Towards accurate binary convolutional neural network. *Proceedings of the 31th International Conference on Neural Information Processing Systems*, 30: 344–352.
- Liu, Z.; Wu, B.; Luo, W.; Yang, X.; Liu, W.; and Cheng, K.-T. 2018. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the 15th European Conference on Computer Vision*, 722–737.

- Liu, Z.-G.; and Mattina, M. 2019. Learning low-precision neural networks without straight-through estimator (STE). In *Proceedings of the 28th International Conference on International Joint Conferences on Artificial Intelligence*, 3066–3072.
- Luo, J.-H.; Wu, J.; and Lin, W. 2017. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the 16th IEEE International Conference on Computer Vision*, 5058–5066.
- Miyashita, D.; Lee, E. H.; and Murmann, B. 2016. Convolutional neural networks using logarithmic data representation. *arXiv preprint arXiv:1603.01025*.
- Nagel, M.; Fournarakis, M.; Bondarenko, Y.; and Blankevoort, T. 2022. Overcoming oscillations in quantization-aware training. In *Proceedings of the 39th International Conference on Machine Learning*, 16318–16330. PMLR.
- Peng, H.; and Chen, S. 2019. BDNN: Binary convolution neural networks for fast object detection. *Pattern Recognition Letters*, 125: 91–97.
- Peng, H.; Wu, J.; Chen, S.; and Huang, J. 2019. Collaborative channel pruning for deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, 5113–5122.
- Peng, H.; Wu, J.; Zhang, Z.; Chen, S.; and Zhang, H.-T. 2021. Deep network quantization via error compensation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9): 4960–4970.
- Qu, Z.; Zhou, Z.; Cheng, Y.; and Thiele, L. 2020. Adaptive loss-aware quantization for multi-bit networks. In *Proceedings of the 33th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7988–7997.
- Rakotomamonjy, A.; Flamary, R.; and Gasso, G. 2015. DC proximal Newton for nonconvex optimization problems. *IEEE Transactions on Neural Networks and Learning Systems*, 27(3): 636–647.
- Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proceedings of the 14th European Conference on Computer Vision*, 525–542. Springer.
- Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.
- Saito, Y.; and Mitsui, T. 1996. Stability analysis of numerical schemes for stochastic differential equations. *SIAM Journal on Numerical Analysis*, 33(6): 2254–2267.
- Sakr, C.; Dai, S.; Venkatesan, R.; Zimmer, B.; Dally, W.; and Khailany, B. 2022. Optimal clipping and magnitude-aware differentiation for improved quantization-aware training. In *Proceedings of the 39th International Conference on Machine Learning*, 19123–19138. PMLR.
- Shayer, O.; Levi, D.; and Fetaya, E. 2018. Learning Discrete Weights Using the Local Reparameterization Trick. In *Proceedings of the 6th International Conference on Learning Representations*.
- Shen, C.; Wang, X.; Yin, Y.; Song, J.; Luo, S.; and Song, M. 2021. Progressive network grafting for few-shot knowledge distillation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2541–2549.
- Wang, N.; Choi, J.; Brand, D.; Chen, C.-Y.; and Gopalakrishnan, K. 2018. Training deep neural networks with 8-bit floating point numbers. In *Proceedings of the 32th International Conference on Neural Information Processing Systems*, 7686–7695.
- Xu, Y.; Wang, Y.; Zhou, A.; Lin, W.; and Xiong, H. 2018. Deep neural network compression with single and multiple level quantization. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, volume 32.
- Ye, J.; Lu, X.; Lin, Z.; and Wang, J. Z. 2018. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers. In *Proceedings of the 6th International Conference on Learning Representations*.
- Yu, X.; Liu, T.; Wang, X.; and Tao, D. 2017. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7370–7379.
- Zhang, D.; Yang, J.; Ye, D.; and Hua, G. 2018a. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *Proceedings of the 15th European Conference on Computer Vision*, 365–382.
- Zhang, T.; Ye, S.; Zhang, K.; Tang, J.; Wen, W.; Fardad, M.; and Wang, Y. 2018b. A systematic dnn weight pruning framework using alternating direction method of multipliers. In *Proceedings of the 15th European Conference on Computer Vision*, 184–199.
- Zhang, X.; Zhou, X.; Lin, M.; and Sun, J. 2018c. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6848–6856.
- Zhang, Z.; Chen, S.; and Sun, L. 2021. P-KDGAN: progressive knowledge distillation with GANs for one-class novelty detection. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 3237–3243.
- Zhou, A.; Yao, A.; Wang, K.; and Chen, Y. 2018. Explicit loss-error-aware quantization for low-bit deep neural networks. In *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9426–9435.