

BASQ: Branch-wise Activation-clipping Search Quantization for Sub-4-bit Neural Networks

Han-Byul Kim^{1,2}, Eunhyeok Park^{3,4}, and Sungjoo Yoo^{1,2}

¹ Department of Computer Science and Engineering

² Neural Processing Research Center (NPRC)

^{1,2} Seoul National University, Seoul, Korea

³ Department of Computer Science and Engineering

⁴ Graduate School of Artificial Intelligence

^{3,4} POSTECH, Pohang, Korea

shinestarhb@gmail.com, canusglow@gmail.com, and sungjoo.yoo@gmail.com

Abstract. In this paper, we propose Branch-wise Activation-clipping Search Quantization (BASQ), which is a novel quantization method for low-bit activation. BASQ optimizes clip value in continuous search space while simultaneously searching L2 decay weight factor for updating clip value in discrete search space. We also propose a novel block structure for low precision that works properly on both MobileNet and ResNet structures with branch-wise searching. We evaluate the proposed methods by quantizing both weights and activations to 4-bit or lower. Contrary to the existing methods which are effective only for redundant networks, e.g., ResNet-18, or highly optimized networks, e.g., MobileNet-v2, our proposed method offers constant competitiveness on both types of networks across low precisions from 2 to 4-bits. Specifically, our 2-bit MobileNet-v2 offers top-1 accuracy of 64.71% on ImageNet, outperforming the existing method by a large margin (2.8%), and our 4-bit MobileNet-v2 gives 71.98% which is comparable to the full-precision accuracy 71.88% while our uniform quantization method offers comparable accuracy of 2-bit ResNet-18 to the state-of-the-art non-uniform quantization method. Source code is on <https://github.com/HanByulKim/BASQ>.

Keywords: Mobile network, quantization, neural architecture search

1 Introduction

Neural network optimization is becoming more and more important with the increasing demand for efficient computation for both mobile and server applications. When we reduce the data bit-width via quantization, the memory footprint is reduced significantly, and the computation performance could be improved when the hardware acceleration is available. However, one major drawback of quantization is the output quality degradation due to the limited number of available values. In particular, when we apply sub-4-bit quantization to optimized networks, e.g., MobileNet-v2 [36], because the backbone structure is already highly optimized and has limited capacity, the accuracy drop is significant

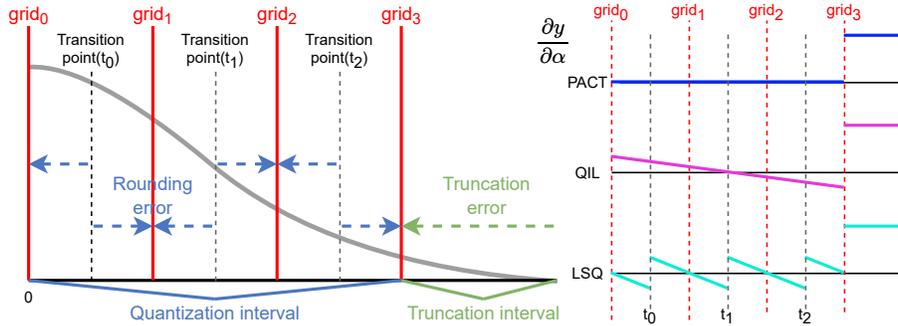


Fig. 1: Error components of quantization for activation distribution (left) and gradient of clipping threshold (right)

compared to the sub-4-bit quantization of the conventional, well-known redundant network, e.g., ResNet-18 [16]. However, because the computation efficiency of the optimized network is much higher than that of the redundant network, it is highly desirable to be able to quantize the advanced networks.

To preserve the quality of output or minimize the accuracy loss as much as possible, diverse quantization schemes have been proposed. Differentiable quantization is the representative method that determines the hyper-parameter of the quantization operator via back-propagation. For instance, in the case of PACT [8], the range of activation quantization is determined by an activation clipping parameter α , which denotes the value of the largest quantization level. In the case of QIL [21], the interval of quantization is parameterized by two learnable parameters for the center and size of the quantization interval. On the contrary, LSQ [11] learns the step size, which denotes the difference of quantization levels. PACT and QIL consider the single component of quantization error, truncation error (truncation and clipping are interchangeably used in this paper), and rounding error, respectively, while LSQ tries to optimize both components at the same time, as shown in Figure 1. Due to this characteristic, LSQ often outperforms other quantization methods and exhibits state-of-the-art results.

However, according to our observation, LSQ becomes unstable and converges to sub-optimal points, especially when applying quantization to an optimized structure, e.g., a network with depthwise-separable convolution. In such an optimized network, the activation distribution gets skewed at every iteration depending on the weight quantization, as pointed out in the PROFIT study [33]. Under this circumstance, we empirically observe that PACT with a judicious tuning of weights on L2 decay of activation clipping parameter (called L2 decay weight parameter throughout in this paper) in its loss function shows a potential of lowering the final loss, especially for the quantization of optimized networks.

Since the value range of activation exhibits distinct per-layer characteristics, the L2 decay weight parameter, which determines the clip value, i.e., the value range of activation quantization, should be tuned judiciously in a layerwise man-

ner. In this paper, we propose a novel idea, branch-wise activation-clipping search quantization (BASQ), that automates the L2 decay weight tuning process via the design space exploration technique motivated by neural architecture search studies. BASQ is designed to assume that the quantization operators with different L2 decay weights are selection candidates and decide, in a discrete space, the candidate that minimizes the accuracy degradation under quantization. According to our extensive studies, this scheme stabilizes the overall quantization process in the optimized network, offering state-of-the-art accuracy.

To further improve ultra-low-bit quantization, we adopt the recent advancement in the network structure for binary neural networks (BNN). In these studies [29,47], newly designed block design schemes are proposed to stabilize network training in low precision and accelerate the convergence via a batch normalization arrangement and new activation function. In this paper, motivated by these studies, we propose a novel extended block design for sub-4-bit quantization with search algorithm, contributing to accuracy improvement in low precision.

2 Related Works

2.1 Low-bit Quantization

Recently, various quantization algorithms for low precision, such as 4-bit or less, have been proposed. In uniform quantization, mechanisms such as clip value training [8], quantization interval learning [21], PACT with statistics-aware weight binning [7], and differentiable soft quantization [12] show good results in 2-bit to 4-bit precision. LSQ [11], which applies step size learning, shows good 2-bit to 4-bit accuracy in ResNet [16] networks.

Non-uniform quantizations [26,43] offer outstanding results than uniform quantizations thanks to their capability of fitting with various distributions. However, they have a critical limitation: they need special compute functions and data manipulations to support quantization values mapped to non-uniform levels.

Mixed precision quantizations [37,38,39,31,1] show cost reduction by varying bit precision, e.g., in a layer-wise manner. For instance, they can use high-bits in important layers while using low-bits in non-sensitive layers. But they also have a limitation in that the hardware accelerators must be capable of supporting various bit precisions in their compute units. Also, neural architecture search (NAS) solutions [40,13,14] apply low-bit quantization by constructing search space with operations of diverse quantization bit candidates.

Thus far, existing quantization methods have struggled with low-bit quantization of networks targeting low cost and high performance, such as MobileNet-v1 [19], v2 [36], and v3 [18]. In PROFIT [33], the progressive training schedule is applied to achieve promising 4-bit accuracy from the MobileNet series.

Recently, binary neural networks (BNN) [34,32,3,29,47,28] are also being actively studied. Unlike the low-bit network of 2-bit to 4-bit, there is a difference in BNN by using the sign function as a quantization function. Various works

based on XNOR operation [34], improvements in training schedule [32], NAS solution [3], block structure and activation function [29,47,28] contribute to good results in 1-bit precision.

2.2 Neural Architecture Search

While hand-crafted architectures [16,41,19,36,18,46,30,20] have evolved with good accuracy and low computation, neural architecture search (NAS) has introduced an automated solution to explore the optimal structure, showing better accuracy and lower computation cost than most hand-crafted architectures. Amoebanet [35] and NAS-Net [49] show the potential of architecture search in the early days but require high training costs. Differential NAS and one-shot NAS have emerged with the concept of supernet for efficient NAS. Differential NAS [27,6,4,42] uses shared parameters so that weights and architecture parameters are jointly trained on supernet training, and the architecture decisions are made in the final design stage. One-shot NAS [13,9,44] trains a supernet by continuously sampling subnets on the supernet. Space exploration algorithm such as evolutionary algorithm (EA) is often used to determine the best subnet architecture from the trained supernet.

In this paper, we adopt the one-shot NAS method to conduct our search-based quantization. Specifically, we propose to find the L2 decay weight of the clip value by considering the multiple L2 decay weights as selection candidates.

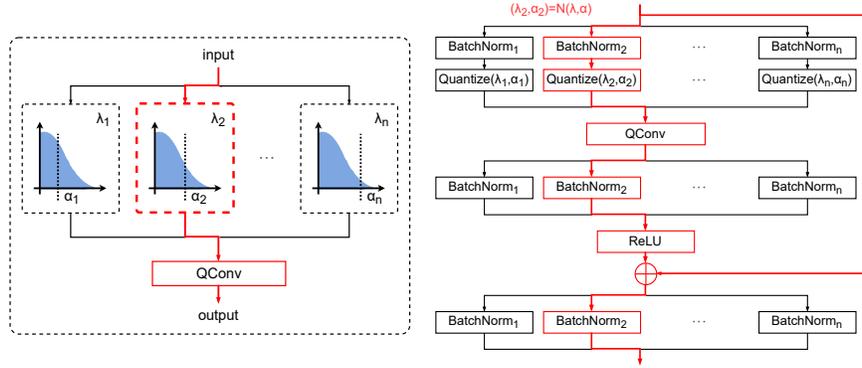
3 Preliminary

Considering that, under quantization, the number of available levels is highly restricted to 2^{bit} , to maintain the quality of output after quantization, hyperparameters for the quantization should be selected carefully. In this work, our quantization method for activation is based on the well-known differentiable quantization method, PACT [8]. Before explaining the details of our proposed methods, BASQ and novel structure design, we will briefly introduce PACT.

As a conventional quantization method, PACT is also implemented based on the straight-through estimator [2]. The rounding operation in the quantization interval is ignored through back-propagation and bypasses the gradient. The output function of the PACT activation quantizer is designed as in Equation 1,

$$y = \frac{1}{2}(|x| - |x - \alpha| + \alpha), \quad \frac{\partial y}{\partial \alpha} = \begin{cases} 0, & \text{if } x < \alpha \\ 1, & \text{if } x \geq \alpha, \end{cases} \quad (1)$$

where α is the learnable parameter for the activation clipping threshold. The input values larger than the truncation interval are clamped to the clipping threshold, and the input values smaller than the threshold are linearly quantized. According to Equation 1, the gradient of the clipping threshold comes from the values in the truncation interval; thereby, the clipping threshold tends to become



(a) BASQ on activation quantization (b) BASQ search block

Fig. 2: BASQ: Branch-wise activation-clipping search quantization (execution path by selected branch from uniform sampling is highlighted.)

larger in order to minimize the truncation error. In order to prevent the clipping threshold from exploding, L2 decay weight is introduced to guide the convergence of the clipping threshold, balancing the rounding error and truncation error. The updating process of the clipping threshold works as follows:

$$\alpha_{new} = \alpha - \eta \left(\sum_i^N \frac{\partial \mathcal{L}}{\partial y_{q,i}} \frac{\partial y_i}{\partial \alpha} + \lambda |\alpha| \right). \quad (2)$$

4 Branch-wise Activation-clipping Search Quantization

Branch-wise activation-clipping search quantization (BASQ) is designed to search the optimal L2 decay weight for the activation clipping threshold in the PACT algorithm. When the optimal value of the L2 decay weight hyper-parameter is determined in a layer-wise manner, an exhaustive search of candidates is impractical. For instance, in case of ResNet-18 having eight available L2 decay weights per block, the number of candidates is as much as 17 Million. Searching for such a large space is prohibitively expensive; thus, we adopt an alternative approach motivated by NAS. In this algorithm, we introduce multiple branches having different L2 decay weights (λ) as selection candidates, as shown in Figure 2. There are multiple branches of quantization operators with different L2 decay weights for the activation clipping. One of the branches could be selected to determine the L2 decay weight used to learn the clipping threshold for quantizing the input activation. To minimize quality degradation after quantization, BASQ searches the best configurations of L2 decay weights among candidates.

4.1 Search Space Design

The search candidates of BASQ for each activation quantization layer are defined by a pair of two parameters, L2 decay weight (λ) and clip value (α), (λ, α) . Let the super-set of λ be Λ , and the super-set of α be A . $\Lambda \times A$ is the search space of BASQ with element pair (λ, α) . As in Equation 3, Λ is a finite set by restricting it with L2 decay weight candidates. A is a subset of the real numbers, and each element (α) exists as a learnable parameter. As a result, BASQ searches discrete space with Λ and simultaneously searches continuous space with A .

$$\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_n\} (\lambda_i \in \mathbb{R}), \quad A \subset \mathbb{R} \quad (3)$$

The network model of BASQ consists of multiple quantization branches, as shown in Figure 2a. Each branch has a (λ, α) pair and can be seen as an activation quantization operator following the PACT algorithm. When one of the branches is selected, the forward operation is performed as the left-hand side of Equation 1 and backward operations as the right-hand side of Equation 1 and Equation 2.

In case of BASQ, the training can be considered as the joint training of the continuous learnable parameter α and discrete parameter for the architecture structure λ . The difference from the search structure of existing NAS methods is that only the activation quantization operators are designed to be explored while the computation, e.g., convolution, is shared across different branches. In short, the computation operation is identical across branches while the quantization strategy (for λ) is searched. According to our observation, the accuracy is rather inferior when taking a private computation on each branch. In addition, note that batch normalization layers are private on each quantization branch to solve statistics diversity [5] that may occur from the updated clip value.

The branch structure of BASQ looks similar to that of [45]. Our difference is that we adopt the activation quantization branches to obtain optimal quantization policies for the activation having the same bit-width across branches while the branches in [45] have different bit-widths. In addition, we adopt the space exploration algorithm to determine the best configuration reducing the training loss, while the previous study exploits each path selectively for different bit-width configurations.

4.2 Search Strategy

BASQ is designed to exploit the supernet-based one-shot training as [13], based on the parameter sharing technique. The search process of BASQ is composed of three steps; First, the base network with quantization operators is jointly trained as a supernet covering all possible configurations. Then, we explore the search space to determine the optimal selection among candidates. Finally, we stabilize batch statistics of the optimal selection. The details are provided in the following.

Supernet Training At every iteration, one branch of activation quantization is selected via uniform random sampling. The parameters in the sampled path, or the selected L2 decay weight (λ) and clip value (α), are used to quantize the input activation. The activation quantization process follows the conventional PACT algorithm, as shown in Equations 1 and 2. In this stage, the weight of the network and the quantization parameters are jointly optimized. When the network is trained for long enough, the weights of the supernet are optimized for the possible configurations of multiple branches, which enables us to evaluate the quality of quantization configuration without additional training.

Architecture Search After training the super-network, we obtain the best configuration of truncation parameter trained with L2 decay weight (α^*), based on the evolutionary algorithm. The problem definition is as follows (ACC_{val} is network accuracy of the validation set):

$$\alpha^* = \operatorname{argmax}_{\alpha \in A} ACC_{val}(w, \alpha). \quad (4)$$

Finetuning After selecting the best architecture, we additionally stabilize the batch statistics by performing additional finetuning for the subset of the network, similar to [33]. We freeze all layers except the normalization layers and clip values in the network, and have a finetuning step of 3 epochs.

5 Block Structure for Low-bit Quantization

5.1 New Building Block

Many BNN studies focus on designing an advanced network architecture that allows stable and fast convergence with binary operators. The representative study [29] proposes a ReAct block that allows skipping connection end-to-end across all layers in the network. [47] provides better accuracy by modifying the arrangement of the activation function and batch normalization in the ReAct block. In this paper, we present an innovative building block that can have a good property in low precision. As shown in Figure 3a and 3b, the new building block for BASQ additionally places a batch normalization layer in front of activation quantization in the block structure of [47] including the skip connection. Originally, [47] places batch normalization after the shortcut to obtain the effect of balanced activation distribution through the affine transformation of the batch normalization layer. However, the gradient through the skip connection in the following block affects the batch normalization at the end of the current block. This has adverse effects on scale & shift of the quantization operator on the next block. Our new building block places a dedicated batch normalization in front of the quantization layer right after the starting point of the skip connection to stabilize activation distribution entering the quantization layer. In addition, since this study targets 2 to 4-bit low precision, the sign and RReLU functions are replaced by multi-bit quantization and ReLU functions, respectively.

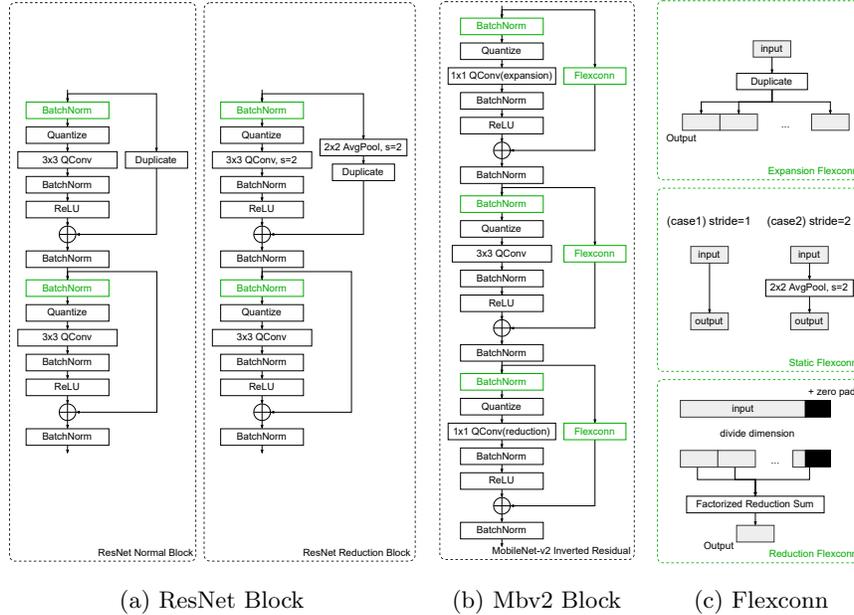


Fig. 3: (a), (b) New building block structures and (c) block skip connections (Flexconn) for low-bit quantization (differences from [29] & [47] are highlighted and Mbv2 represents MobileNet-v2.)

5.2 Flexconn: A Flexible Block Skip Connection for Fully Skip-Connected Layers

In order to bring stable effect to search algorithm with low-bit quantization through the new building block in Section 5.1, it is important to build fully skip-connected layers by connecting skip connections over all layers as in [29] and [47]. This fully skip-connected layer refers to a skip connection made of full precision without any quantized operation, unlike skip connection using 1x1 convolution in ResNets. However, it is non-trivial to build a fully skip-connected layer, especially if the number of output channels is not an integer multiple of that of input channels, e.g., a reduction layer in the inverted residual block of MobileNet-v2. Therefore, in this paper, we propose Flexconn, which enables the formation of a fully skip-connected layer, even in the dynamic change (expansion and reduction at any ratio) of channel dimension. Flexconn forms a fully skip-connected layer for the following three cases, as shown in Figure 3c.

– Expansion Flexconn on 1x1 channel expansion convolution

In channel expansion, Expansion Flexconn copies the input tensor up to the integer multiple of channel expansion and concatenates the copied tensors in the channel dimension. It can be used for the 1x1 channel expansion convolution of the inverted residual block in MobileNet-v2.

- **Static Flexconn on 3x3 same channel convolution**

In the case of the same channel size, Static Flexconn is used. It handles the skip connection in the same way as in [29]. If the stride is 1, we add the skip connection as is. If the stride is 2, we use 2x2 average pooling to adjust the resolution. Static Flexconn can be used for 3x3 convolution in the middle of the inverted residual block in MobileNet-v2.

- **Reduction Flexconn on 1x1 channel reduction convolution**

Reduction Flexconn is used to make a connection in case of channel reduction. It performs a factorized reduction sum operation after dividing the input channels by the integer multiple of the reduced channels. In MobileNet-v2, it is used for the last 1x1 channel reduction convolution of the inverted residual block. However, when the number of input channels is not an integer multiple of that of reduced channels, Flexconn uses zero tensors and concatenates them to the input channels on the last channel dimension to match the integer multiple reductions. Flexconn solves the corner case of non-integer multiple reductions in MobileNet-v2, thereby enabling the end-to-end identity connection.

6 Experiments

To evaluate the effectiveness of the proposed methods, we perform the BASQ experiment in ImageNet dataset [10]. The networks are trained along with the learning schedule of 2-phase training used in [32] for stable clipping parameter training. Therefore, training is performed with only activations quantized in phase-1 and both activations and weights quantized in phase-2. In MobileNet-v2, we use a batch size of 768 with an initial learning rate of 0.000375 for 466,000 iterations per phase. For selection candidates, seven L2 decay weights are used in MobileNet-v2. In MobileNet-v1 and ResNet-18, we use a batch size of 1024 with an initial learning rate of 0.0005 for 400,000 iterations per phase. For selection candidates, eight L2 decay weights are used in both networks. For all experiments, we use Adam optimizer [23] with cosine learning rate decay without restart. The weight decay is set to 5×10^{-6} . In addition, knowledge distillation [17] is applied with ResNet-50 as a teacher. The KL-divergence between teacher and student’s softmax output is used as loss [29].

Note that we apply LSQ as weight quantization in all experiments. Also, in MobileNet-v2, we apply BASQ only on activation quantization in front of the 3x3 convolution in the middle of the inverted residual block. In the evolutionary algorithm, we use 45 populations with 15 mutations and 15 crossovers in 20 iterations. We evaluate 2 to 4-bit quantizations for MobileNet-v2 and ResNet-18 and 4-bit for MobileNet-v1.

To evaluate BASQ, we apply the k -fold evaluation ($k = 10$) method in the ImageNet validation set by constructing a test set with exception to the dataset used in architecture selection.⁵ The method is as follows.

⁵ In [13], the accuracy is evaluated in a different way. [13] use about 16% of the validation set for architecture selection. The test set is constructed and evaluated

1. We divide the validation set into 10 subsets.
2. One subset is used to select the architecture while the other subsets are used for the test set to evaluate the accuracy of the selected architecture.
3. We perform step 2 for each of the remaining nine subsets that are not used for architecture selection. Each one is used for selecting the architecture and the rest are used as a test set. We average 10 evaluation results to calculate the final accuracy.

6.1 Evaluation with MobileNet-v2 and MobileNet-v1

Table 1: Top-1 ImageNet accuracy (%) of MobileNet-v2 and v1 models

Method	A2/W2	A3/W3	A4/W4
PACT [8]			61.4
DSQ [12]			64.8
QKD [22]	45.7	62.6	67.4
LSQ [11]	46.7	65.3	69.5
LSQ + BR [15]	50.6	67.4	70.4
LCQ [43]			70.8
PROFIT [33]	61.9	69.6	71.56
BASQ (Ours)	64.71	70.25	71.98

(a) MobileNet-v2

Method	A4/W4
PACT [8]	62.44
LSQ [11]	63.60
PROFIT [33]	69.06
BASQ (Ours)	72.05

(b) MobileNet-v1

Table 1a shows the accuracy of 2 to 4-bit quantized MobileNet-v2. Our 2-bit model (under ResNet-50 teacher) outperforms the state-of-the-art model, PROFIT [33] (with ResNet-101 teacher) by a large margin (64.71% vs 61.9%). Our 3-bit and 4-bit models also exhibit slightly better accuracy than PROFIT, while our 4-bit model gives 71.98% which is slightly better than the full-precision accuracy of 71.88%. Table 1b shows accuracy of MobileNet-v1. BASQ gives significantly better results on 4-bit (72.05%) compared to the state-of-the-art method, PROFIT (69.06%).

6.2 Evaluation with ResNet-18

Table 2 shows the accuracy of 2 to 4-bit quantized ResNet-18. Our models offer competitive results with the state-of-the-art one, LCQ [43]. Specifically, our 3-bit and 4-bit models give better results than LCQ by 0.8% and 1.1%, respectively, while our 2-bit model offers comparable (within 0.3%) results to LCQ. Note that LCQ is a non-uniform quantization method. Thus, it is meaningful that BASQ, as a uniform quantization method, shows comparable results to the state-of-the-art non-uniform method, LCQ.

using the entire validation set. In order to avoid such a duplicate use of the same data in architecture selection and evaluation, we adopt k -fold evaluation.

Table 2: Top-1 ImageNet accuracy (%) of ResNet-18 models

Method	A2/W2	A3/W3	A4/W4
Dorefa [48]	62.6	67.5	68.1
PACT [8]	64.4	68.1	69.2
DSQ [12]	65.17	68.66	69.56
QIL [21]	65.7	69.2	70.1
PACT + SAWB + fpsc [7]	67.0		
APOT [26]	67.3	69.9	70.7
QKD [22]	67.4	70.2	71.4
LSQ [11]	67.6	70.2	71.1
LCQ [43]	68.9	70.6	71.5
BASQ (Ours)	68.60	71.40	72.56

Table 1 and 2 demonstrate that MobileNets are more difficult to quantize than ResNet-18. Thus, the existing methods, e.g., [8,12,22,11], which perform well on ResNet-18 in Table 2 give poor results on MobileNets in Table 1. It is also challenging to offer constant competitiveness across low-bit precisions. Some existing works, e.g., [22,11] show comparable results on 4-bit models while giving poor results in 2-bit cases. Our proposed BASQ is unique in that it constantly offers competitive results on both ResNet-18 and MobileNets across bit precisions from 2 to 4-bits. As will be explained in the next section, such benefits result from our proposed joint training of continuous and discrete parameters and the proposed block structures.

6.3 Ablation Study

Effects of components We use 2-bit ResNet-20 and MobileNet-v2 models on CIFAR10 [24] to evaluate the effect of each component in our proposed method. Table 3 shows the effect of adopting BASQ in activation quantization in the 2-phase training. The table shows that BASQ, adopted for activation quantization, can contribute to accuracy improvement in both ResNet-20 and MobileNet-v2.

Table 3: BASQ results of 2-bit models on CIFAR10

Model	Activation quantization	Weight quantization	Accuracy (%)
ResNet-20	LSQ	LSQ	89.40
	BASQ	LSQ	90.21
MobileNet-v2	LSQ	LSQ	89.62
	BASQ	LSQ	90.47

We also evaluate the effect of new building block (in Section 5.1), shown in Table 4. Note that, in the case of MobileNet-v2, we apply the new building

block structure in Figure 3b (with new batch normalization layer and without layer skip connections as original MobileNet-v2 block). The results show that the new building block improves the accuracy in both ResNet-20 and MobileNet-v2. MobileNet-v2 shows lower advances than ResNet-20 due to the absence of layer skip connections that obstructs the intention of the new batch normalization layer. We also evaluate the block structure of [47] in both networks and obtain accuracy degradation. This shows that the dedicated batch normalization layer in the new building block has the potential of stabilizing the convergence of low precision networks.

Table 4: Effect of new building block in 2-bit models on CIFAR10

Model	Block	Accuracy (%)
ResNet-20	preactivation ResNet	89.40
	Fracbnn [47]	89.39
	new building block	89.72
MobileNet-v2	basic inverted residual block	89.62
	Fracbnn [47]	89.32
	new building block	89.74

Table 5 shows the effect of Flexconn (in Section 5.2). Basically, we use the residual block connection originally used in MobileNet-v2 while using Flexconns. The table shows that we obtain the highest accuracy when all the three cases of Flexconn are adopted in the inverted residual block. This shows that constructing a fully skip-connected layer, making all layers connected in high precision, is critical in low precision networks. (For detailed analysis of our proposed block structure with BASQ, see Section 3 of supplementary.)

Table 5: Effect of Flexconn in 2-bit MobileNet-v2 on CIFAR10

Methods on Inverted Residual Block				Accuracy (%)
New Building Block	Expansion Flexconn	Static Flexconn	Reduction Flexconn	
				89.62
✓				89.74
✓	✓			90.02
✓		✓		88.95
✓			✓	89.61
✓	✓	✓		90.35
✓	✓		✓	89.86
✓		✓	✓	90.08
✓	✓	✓	✓	90.94

Importance of searching L2 decay weight in discrete search space BASQ jointly optimizes the weight and clip value in continuous search space while searching L2 decay weight for updating clip value in discrete search space. To investigate the importance of searching for L2 decay weight in discrete search space, we obtain the accuracy of 100 architectures by randomly selecting each block’s clip value within the supernet. 2-bit ResNet-18 and MobileNet-v2 are evaluated on ImageNet. The results are shown in Figure 4.

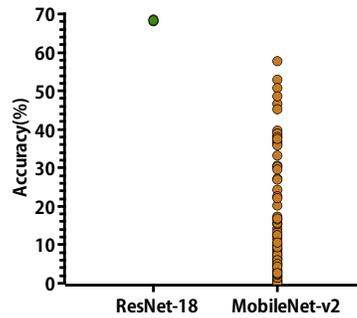


Fig. 4: Accuracies for 100 random architectures from 2-bit ResNet-18 and MobileNet-v2

In the case of ResNet-18, the overall accuracy deviation between architectures is as small as [67.98%, 68.53%]. However, in the case of MobileNet-v2, the difference of accuracy between architectures is quite large in [0.08%, 57.76%], much greater than that of ResNet-18. This demonstrates that a simple selection of L2 decay weight (or just one learning solution) adopted in the previous works [8,12,22,11] worked well for ResNets while such a strategy does not work in MobileNets as shown in our experiments, which advocates the necessity of selection methods for L2 decay weights like ours.

Loss landscape comparison Figure 5 illustrates the complexity of loss surface between LSQ and BASQ. We apply the loss-landscape [25] method to 2-bit ResNet-20 and MobileNet-v2 on CIFAR10 with LSQ and BASQ. We do not apply the new building block and Flexconn to evaluate pure quantization effects. As shown in Figure 5a, 5b of ResNet-20, BASQ shows improved loss surface with clear convexity while LSQ has several local minima. As shown in Figure 5c, 5d, in case of MobileNet-v2, both LSQ and BASQ do not show a clear loss surface, possibly due to the fact that MobileNet-v2 is an optimized network. Nevertheless, BASQ shows improved convexity while LSQ has a local minima near the center. The comparison shows that BASQ has the potential of better convergence in training low-bit networks, including optimized ones like MobileNet-v2.

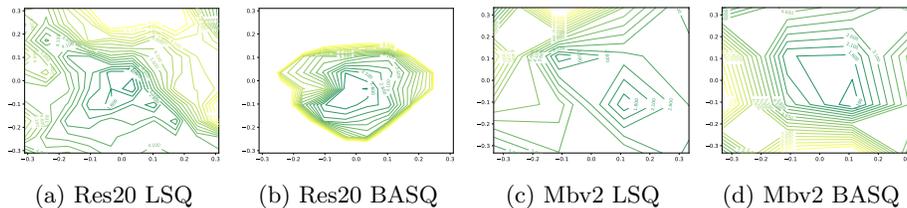


Fig. 5: Loss landscape of 2-bit quantized models on CIFAR10 (Res20 and Mbv2 represents ResNet-20 and MobileNet-v2, respectively.)

Table 6: Effect of training schedule in MobileNet-v2 on ImageNet

Method	A2/W2	A3/W3	A4/W4
LSQ [11]	50.0	66.5	70.2
BASQ	64.7	70.3	72.0

Table 7: BASQ results with shared and private convolution in search block

BASQ Method	Accuracy (%)
Shared conv.	90.21
Private conv.	89.90

Training schedule Quantization algorithms usually start with a full-precision pre-trained model and proceed with finetuning while quantizing activation and weight. However, BASQ trains itself without a full-precision pre-trained model. It starts training from scratch and proceeds with a 2-phase schedule from [32]. Progressive training is performed with only activation quantized (phase-1) and then with both activation and weight quantized (phase-2). This training schedule can affect MobileNet-v2 results. For example, [33] proposes good MobileNet results by special training schedule without changing the network. To evaluate the effect of the training schedule, we train LSQ [11] with the same training schedule as BASQ. The results are shown in Table 6. Although the accuracy of LSQ gets improved, there is still a large accuracy gap in 2-bit and 3-bit models.

Search block Unlike typical NAS methods, BASQ consists of search blocks where branches are made up of quantization operators so that the computation is shared across selections as shown in Figure 2b. To evaluate the effect of shared computation, we measure the accuracy of private computation, where each branch is equipped with private convolution as in typical NAS approaches. Table 7, on 2-bit ResNet-20 with CIFAR10, shows that the shared one offers better accuracy.

7 Conclusion

In this paper, we propose a novel quantization method BASQ and block structures (new building block and Flexconn). BASQ judiciously exploits NAS to search for hyper-parameters, namely, L2 decay weights of clipping threshold. Our proposed block structure helps form a fully skip-connected layer with a dedicated normalization layer, which contributes to the quantization for low-bit precisions. The experiments show that our proposed method offers constant competitiveness for both ResNet and MobileNets across low-bit precisions from 2 to 4-bits. Specifically, our 2-bit MobileNet-v2 model (though trained under a weaker teacher) outperforms the state-of-the-art 2-bit model by a large margin. In addition, our uniformly quantized 2-bit ResNet-18 model offers comparable accuracy to the non-uniformly quantized model of the state-of-the-art method.

Acknowledgment This work was supported by IITP and NRF grants funded by the Korea government (MSIT, 2021-0-00105, NRF-2021M3F3A2A02037893) and Samsung Electronics (Memory Division, SAIT, and SRFC-TC1603-04).

References

1. Bai, H., Cao, M., Huang, P., Shan, J.: Batchquant: Quantized-for-all architecture search with robust quantizer. *Advances in Neural Information Processing Systems* **34** (2021)
2. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013)
3. Bulat, A., Martinez, B., Tzimiropoulos, G.: Bats: Binary architecture search. *European Conference on Computer Vision (ECCV)* (2020)
4. Cai, H., Zhu, L., Han, S.: Proxylessnas: Direct neural architecture search on target task and hardware. *International Conference on Learning Representations (ICLR)* (2019)
5. Chen, P., Liu, J., Zhuang, B., Tan, M., Shen, C.: Towards accurate quantized object detection. *Computer Vision and Pattern Recognition (CVPR)* (2021)
6. Chen, X., Xie, L., Wu, J., Tian, Q.: Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. *International Conference on Computer Vision (ICCV)* (2019)
7. Choi, J., Venkataramani, S., Srinivasan, V., Gopalakrishnan, K., Wang, Z., Chuang, P.: Accurate and efficient 2-bit quantized neural networks. *Proceedings of Machine Learning and Systems* **1**, 348–359 (2019)
8. Choi, J., Wang, Z., Venkataramani, S., Chuang, P., Srinivasan, V., Gopalakrishnan, K.: Pact: Parameterized clipping activation for quantized neural networks. *arXiv:1805.06085* (2018)
9. Chu, X., Zhang, B., Xu, R.: Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. *International Conference on Computer Vision (ICCV)* (2021)
10. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A largescale hierarchical image database. *Computer Vision and Pattern Recognition (CVPR)* (2009)
11. Esser, S., McKinstry, J., Bablani, D., Appuswamy, R., Modha, D.: Learned step size quantization. *International Conference on Learning Representations (ICLR)* (2020)
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. *International Conference on Computer Vision (ICCV)* **1**, 348–359 (2019)
13. Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J.: Single path one-shot neural architecture search with uniform sampling. *European Conference on Computer Vision (ECCV)* (2020)
14. Habi, H., Jennings, R., Netzer, A.: Hmq: Hardware friendly mixed precision quantization block for cnns. *European Conference on Computer Vision (ECCV)* (2020)
15. Han, T., Li, D., Liu, J., Tian, L., Shan, Y.: Improving low-precision network quantization via bin regularization. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 5261–5270 (2021)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *Computer Vision and Pattern Recognition (CVPR)* (2016)
17. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. *arXiv:1503.02531* (2015)
18. Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q., Adam, H.: Searching for mobilenetv3. *International Conference on Computer Vision (ICCV)* (2019)

19. Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861 (2017)
20. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. *Computer Vision and Pattern Recognition (CVPR)* (2018)
21. Jung, S., Son, C., Lee, S., Son, J., Han, J., Kwak, Y., Hwang, S., Choi, C.: Learning to quantize deep networks by optimizing quantization intervals with task loss. *Computer Vision and Pattern Recognition (CVPR)* (2018)
22. Kim, J., Bhalgat, Y., Lee, J., Patel, C., Kwak, N.: Qkd: Quantization-aware knowledge distillation. arXiv:1911.12491 (2019)
23. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
24. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
25. Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the loss landscape of neural nets. *Advances in neural information processing systems (NIPS)* **31** (2018)
26. Li, Y., Dong, X., Wang, W.: Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. *International Conference on Learning Representations (ICLR)* (2020)
27. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. *International Conference on Learning Representations (ICLR)* (2019)
28. Liu, Z., Shen, Z., Li, S., Helwegen, K., Huang, D., Cheng, K.: How do adam and training strategies help bnns optimization. *International Conference on Machine Learning (ICML)* (2021)
29. Liu, Z., Shen, Z., Savvides, M., Cheng, K.: Reactnet: Towards precise binary neural network with generalized activation functions. *European Conference on Computer Vision (ECCV)* (2020)
30. Ma, N., Zhang, X., Zheng, H., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. *European Conference on Computer Vision (ECCV)* (2018)
31. Ma, Y., Jin, T., Zheng, X., Wang, Y., Li, H., Jiang, G., Zhang, W., Ji, R.: Ompq: Orthogonal mixed precision quantization. arXiv:2109.07865. (2021)
32. Martinez, B., Yang, J., Bulat, A., Tzimiropoulos, G.: Training binary neural networks with real-to-binary convolutions. *International Conference on Learning Representations (ICLR)* (2020)
33. Park, E., Yoo, S.: Profit: A novel training method for sub-4-bit mobilenet models. *European Conference on Computer Vision (ECCV)* (2020)
34. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. *European Conference on Computer Vision (ECCV)* (2016)
35. Real, E., Aggarwal, A., Huang, Y., Le, Q.: Regularized evolution for image classifier architecture search. In: *AAAI Conf. on Artificial Intelligence* **33**, 4780–4789 (2019)
36. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. *Computer Vision and Pattern Recognition (CVPR)* (2018)
37. Uhlich, S., Mauch, L., Cardinaux, F., Yoshiyama, K., Garcia, J., Tiedemann, S., Kemp, T., Nakamura, A.: Mixed precision dnns: All you need is a good parametrization. *International Conference on Learning Representations (ICLR)* (2020)

38. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision. *Computer Vision and Pattern Recognition (CVPR)* (2019)
39. Wang, T., Wang, K., Cai, H., Lin, J., Liu, Z., Wang, H., Lin, Y., Han, S.: Apq: Joint search for network architecture, pruning and quantization policy. *Computer Vision and Pattern Recognition (CVPR)* (2020)
40. Wu, B., Wang, Y., Zhang, P., Tian, Y., Vajda, P., Keutzer, K.: Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv:1812.00090* (2018)
41. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. *Computer Vision and Pattern Recognition (CVPR)* (2017)
42. Xie, S., Zheng, H., Liu, C., Lin, L.: Snas: stochastic neural architecture search. *International Conference on Learning Representations (ICLR)* (2018)
43. Yamamoto, K.: Learnable companding quantization for accurate low-bit neural networks. *Computer Vision and Pattern Recognition (CVPR)* (2021)
44. You, S., Huang, T., Yang, M., Wang, F., Qian, C., Zhang, C.: Greedynas: Towards fast one-shot nas with greedy supernet. *Computer Vision and Pattern Recognition (CVPR)* (2020)
45. Yu, H., Li, H., Shi, H., Huang, T., Hua, G.: Any-precision deep neural networks. *arXiv:1911.07346* (2019)
46. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. *Computer Vision and Pattern Recognition (CVPR)* (2018)
47. Zhang, Y., Pan, J., Liu, X., Chen, H., Chen, D., Zhang, Z.: Fracbnm: Accurate and fpga-efficient binary neural networks with fractional activations. *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* pp. 171–182 (2021)
48. Zhou, S., Ni, Z., Zhou, X., Wen, H., Wu, Y., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv:1606.06160* (2016)
49. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.: Learning transferable architectures for scalable image recognition. *Computer Vision and Pattern Recognition (CVPR)* (2018)