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

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

1 Clipping Threshold Behavior

1.1 Clipping Threshold on Training

Figure 1 illustrates how the clipping thresholds of BASQ change during the training of (block-1 of) 2-bit MobileNet-v2 [16]. They tend to become larger (smaller) as the associated L2 decay weights (λ) get smaller (larger) and stabilize early in the training.

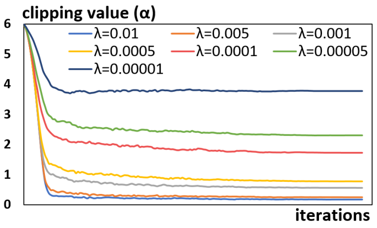
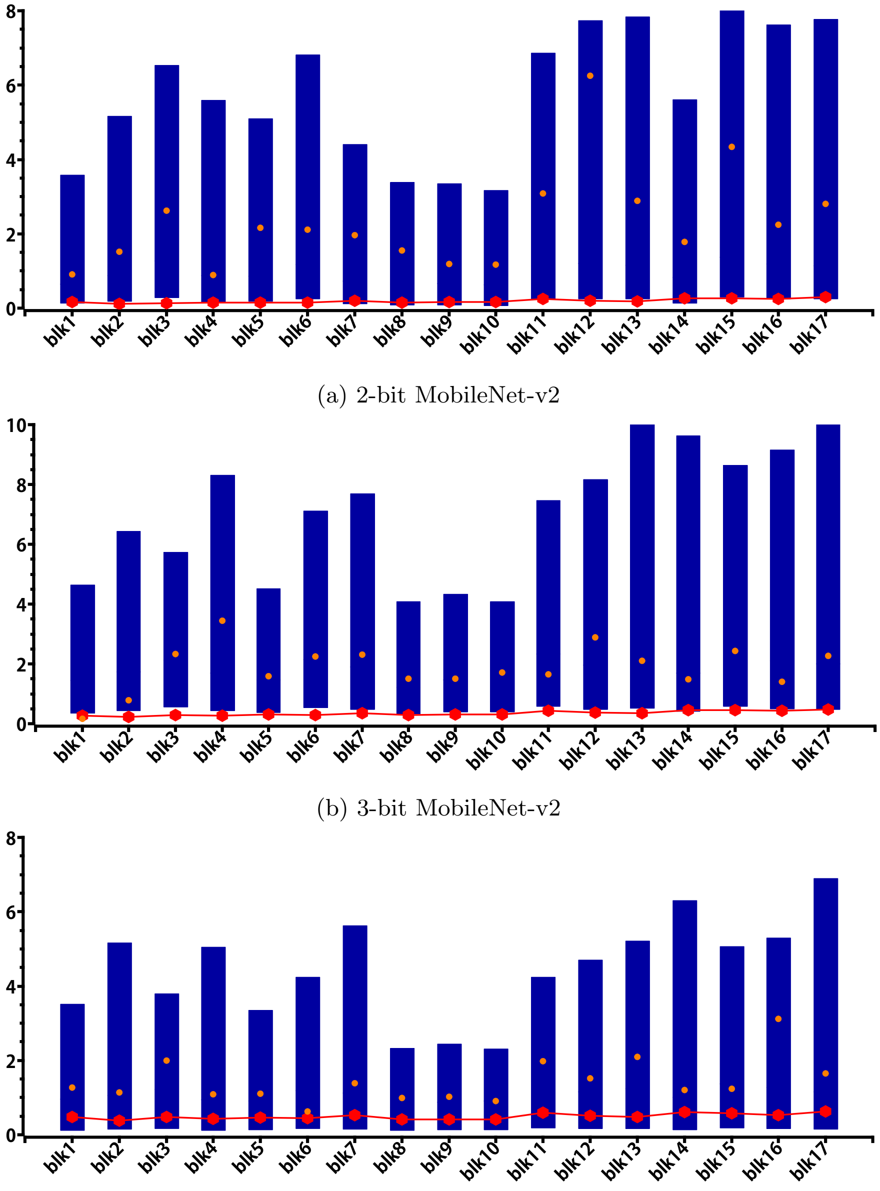


Fig. 1: Clipping thresholds in 2-bit MobileNet-v2 on training

1.2 Coverage of Clipping Threshold

Figure 2 illustrates the clipping thresholds of BASQ (orange dots) and LSQ [5] (red dots). The blue rectangle represents the range of minimum to maximum values of seven clipping thresholds since we used seven L2 decay weights as selection candidates in MobileNet-v2. The orange dots represent the average clipping thresholds of BASQ obtained in our 10-fold evaluation.

The figure shows that BASQ tends to give larger clipping thresholds than LSQ, which demonstrates that the clipping of LSQ is sub-optimal (too aggressive in this experiment), possibly due to the fact that only a single clip value is optimized per block, thereby incurring too much truncation error and finally losing accuracy in low-bit precisions. On the contrary, our proposed BASQ can explore the design space of clipping thresholds over multiple branches and thus obtain better clipping solutions (larger clip values in this experiment) which offer better balance between truncation and rounding errors thereby improving accuracy.



(a) 2-bit MobileNet-v2

(b) 3-bit MobileNet-v2

(c) 4-bit MobileNet-v2

Fig. 2: Average clipping thresholds of BASQ obtained on evaluation (orange dots), LSQ clipping thresholds (red dots), and minimum to maximum clip value ranges of BASQ (blue rectangles)

Algorithm 1 Search strategy of BASQ

```
1. Supernet Training
Input: iteration $\mathcal{T}$, uniform sampling $\mathcal{N}$, $\lambda$-super-set $\Lambda$, $\alpha$-super-set $A$, supernet weight $W$, train dataset $D_{train}$, teacher network weight $W_T$, kl-divergence loss $\mathcal{L}$
Output: trained supernet
  for $i = 1 : \mathcal{T}$ do
    $\{(\lambda_{b1}, \alpha_{b1}), \cdots, (\lambda_{bn}, \alpha_{bn})\} = \mathcal{N}(\Lambda, A)$    ▷ sample selection from block-1 to n
    $out_s = Forward(W, D_{train}, \{(\lambda_{b1}, \alpha_{b1}), \cdots, (\lambda_{bn}, \alpha_{bn})\})$    ▷ student: supernet
    $out_t = Forward(W_T, D_{train})$    ▷ teacher: ResNet-50
    $Backward(\mathcal{L}(out_s, out_t))$
  end for
  Return $W$
2. Architecture Search
Input: fold number $K = 10$, iteration $\mathcal{T} = 20$, uniform sampling $\mathcal{N}$, $\lambda$-super-set $\Lambda$, $\alpha$-super-set $A$, supernet weight $W$, validation dataset $D_{val}$, population number $p = 45$, crossover number $c = 15$, mutation number $m = 15$, mutation probability $prob = 0.1$
Output: $K$ best clip value selections
  $\{D_{val,fold,1}, \cdots, D_{val,fold,K}\} = Divide\_fold(D_{val}, K)$    ▷ divide into $K$ subsets
  $Top1 = \emptyset$
  for $k = 1 : K$ do
    $Fold\_ACC, Fold\_Topn, P = \emptyset$
    $P_0 = Initialize\_population(p, \Lambda, A, D_{val,fold,k})$
    for $i = 1 : \mathcal{T}$ do
      $Fold\_ACC_{i-1} = Inference(W, D_{val,fold,k}, P_{i-1})$
      $Fold\_Topn = Update\_topn(Fold\_Topn, P_{i-1}, Fold\_ACC_{i-1})$
      $P_{crossover} = Crossover(Fold\_Topn, c)$
      $P_{mutation} = Mutation(Fold\_Topn, m, prob)$
      $P_{random} = \mathcal{N}(\Lambda, A, p - c - m)$
      $P_i = P_{crossover} \cup P_{mutation} \cup P_{random}$
    end for
    $Fold\_ACC_{\mathcal{T}} = Inference(W, D_{val,fold,k}, P_{\mathcal{T}})$
    $Fold\_Topn = Update\_topn(Fold\_Topn, P_{\mathcal{T}}, Fold\_ACC_{\mathcal{T}})$
    $Top1_k = Fold\_Topn_1$    ▷ select best architecture in $k$-th subset
  end for
  Return $Top1$
3. Finetuning
Input: fold number $K = 10$, iteration $\mathcal{T} = 3$, selected top1 architectures $Top1$, supernet weight $W$, train dataset $D_{train}$, validation dataset $D_{val}$
Output: accuracy of K-fold result from finetuned selections
  $\{D_{val,fold,1}, \cdots, D_{val,fold,K}\} = Divide\_fold(D_{val}, K)$    ▷ divide into $K$ subsets
  for $k = 1 : K$ do
    for $parameter \notin batch\_normalization$ or $clip\_value$ do
      $parameter.learning\_rate = 0$    ▷ freeze except batch norm and clip value
    end for
    for $i = 1 : \mathcal{T}$ do
      $Train(W, D_{train}, Top1_k)$
    end for
    $ACC_k = Inference(W, D_{val} - D_{val,fold,k}, Top1_k)$    ▷ avoid using same data
  end for
  $ACC = Average(ACC_1, \cdots, ACC_K)$
  Return $ACC$    ▷ final accuracy of BASQ
```

2 Search Strategy

We explain the detailed search process of Section 4.2 in the main paper in Algorithm 1. First, the base network with various quantization operators is jointly trained as a supernet covering all possible configurations. Then, we explore the search space to make selections among candidates. Finally, we apply finetuning for stabilizing batch statistics.

3 Detailed Analysis

3.1 Effects of Components

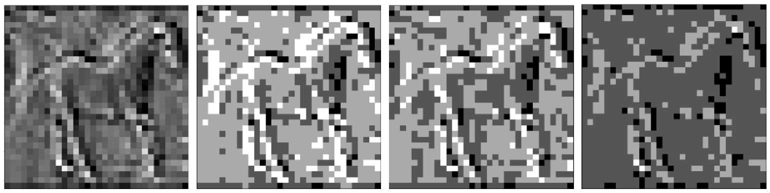
As a complement to Section 6.3 in the main paper, we evaluate 2-bit MobileNet-v2 model on ImageNet. Table 1 shows that BASQ proves better (57.48%) than LSQ (46.7%) for activation quantization on large scale dataset. It also demonstrates that the new components of our proposed method (new building block and flexconn) make a significant contribution (64.71%) to the accuracy improvement in the 2-bit MobileNet-v2 on ImageNet.

Table 1: Effects of BASQ and our proposed block structure in 2-bit MobileNet-v2 on ImageNet

Method	Configurations		Accuracy (%)
	BASQ	Our block structure (New building block & Flexconn)	
LSQ [5]			46.7
BASQ (without our block structure)	✓		57.48
BASQ (with our block structure)	✓	✓	64.71

3.2 Effects of BASQ

BASQ offers an automated per-layer optimization of clip value via controlling L2 decay weight. As shown in Figure 3, the resulting feature-map of activation highly relies on clip value (α) learned via L2 decay weight (λ). PACT [3] applies a single global L2 decay weight to the entire network and, thus, exhibits sub-optimal results on highly optimized networks, e.g. MobileNet-v2. It is mainly because activation distributions vary across layers, and thus a single global L2 decay weight fails to achieve per-layer optimization of clip value.

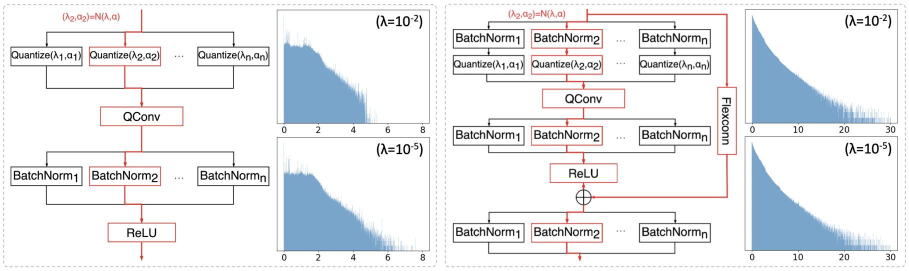


(a) (b) $\lambda = 0.01$ (c) $\lambda = 0.001$ (d) $\lambda = 0.00001$

Fig. 3: Effect of clipping threshold (α) learned via L2 decay weight (λ) on feature-map: (a) real valued input activation, (b-d) 2-bit quantized activations in BASQ branches

3.3 Effects of Our Block Structure

Our proposed block structure is adopted for bringing stabilization effect to activation distribution on training branch-wise searching structure with low-bit quantization. To demonstrate the effect of our block structure, we evaluate activation distribution behavior of BASQ without and with our block structure in 2nd and 3rd row of Table 1, respectively. Figure 4 (a) shows the original block structure and the distributions of output activation on two BASQ branches. Without the proposed block structure, the output distributions of the two BASQ branches having different L2 decay weights tend to have different value ranges. As Figure 4 (b) shows, the proposed block structure makes the distributions similar to each other, i.e., more consistent. The experiments demonstrate that our block structure gives stabilization effect on training branch-wise searching structure with low-bit precision. Also, having consistent output distributions across BASQ branches contributes to better convergence as shown in the Table 1.


$(\lambda_2,\alpha_2)=N(\lambda,\alpha)$
Quantize(λ_1,α_1)
Quantize(λ_2,α_2)
...
Quantize(λ_n,α_n)
QConv
BatchNorm$_1$
BatchNorm$_2$
BatchNorm$_n$
ReLU
($\lambda=10^{-2}$)
($\lambda=10^{-5}$)
0 2 4 6 8
$(\lambda_2,\alpha_2)=N(\lambda,\alpha)$
BatchNorm$_1$
BatchNorm$_2$
BatchNorm$_n$
Quantize(λ_1,α_1)
Quantize(λ_2,α_2)
Quantize(λ_n,α_n)
QConv
Flexconn
ReLU
($\lambda=10^{-2}$)
($\lambda=10^{-5}$)
0 10 20 30

(a) without our block structure (b) with our block structure

Fig. 4: Block structures and output distributions of 2-bit MobileNet-v2 BASQ

4 Evaluation Methods

4.1 *k*-fold Evaluation

In BASQ, we evaluate our accuracy with k-fold (k=10) evaluation method on ImageNet validation dataset [4]. Since the fixed amount of dataset for architecture selection and accuracy evaluation could introduce the variance of results, we apply 10-fold evaluation and average 10 results for the final accuracy.

Table 2: ImageNet top-1 accuracy of MobileNet-v2 on 10-fold evaluation

Bit	set-1	set-2	set-3	set-4	set-5	set-6	set-7	set-8	set-9	set-10	Average
A2/W2	63.95	64.02	64.28	63.89	65.46	65.1	65.23	65.32	65.4	64.49	64.71
A3/W3	69.48	69.72	69.68	69.52	70.87	70.63	70.88	70.84	70.74	70.18	70.25
A4/W4	71.22	71.30	71.57	71.21	72.65	72.31	72.63	72.53	72.55	71.86	71.98

Table 2 shows the accuracy results on the 10-fold evaluation. Our 10-fold results demonstrate that the dataset dependency for architecture selection exists. Set-5 shows the highest accuracy in all available bit precisions while set-1 and set-4 shows the lowest accuracy among candidates. This indicates that 10-fold validation is beneficial for minimizing the statistical disturbance of the results induced by the dataset dependency.

4.2 Evaluation Method Adopted in SPOS

Table 3: ImageNet top-1 accuracy of MobileNet-v2 on 10-fold evaluation and SPOS evaluation

Method	A2/W2	A3/W3	A4/W4
PACT [3]			61.4
DSQ [6]			64.8
QKD [10]	45.7	62.6	67.4
LSQ [5]	46.7	65.3	69.5
LSQ + BR [8]	50.6	67.4	70.4
LCQ [19]			70.8
PROFIT [13]	61.9	69.6	71.56
BASQ (10-fold)	64.71	70.25	71.98
BASQ (SPOS [7])	64.85	70.24	71.95

As mentioned in the main paper, SPOS [7] use 16% of the validation set for architecture selection. The test set is constructed using the entire validation set. Our main results are obtained under 10-fold evaluation to avoid such a duplicated usage of the same data in both architecture selection and evaluation.

In order to investigate the difference between the two evaluation methods, we also evaluate our method following the evaluation method of SPOS. Table 3 shows that the duplicated usage (BASQ (SPOS) in the last row) gives slightly better accuracy in 2-bit precision, possibly due to the duplicated usage of a portion (16%) of validation data.

5 Binary (1-bit) BASQ

Our network structure is motivated by previous works related to the binary neural network (BNN). BNN is quite different from the other low-bit quantized networks in various aspects. BNN uses the sign function as a quantization function without allowing mapping value to zero. In addition, the representative study [11] uses the PReLU activation function with shift terms. Note that the low-bit networks (2-bit or more) adopt rounding and clipping for quantization and ReLU activation.

BNN uses a back propagation gradient function with a limited value range. Specifically, it suffers from a zero gradient problem for the values outside of STE [1] (from -1 to 1). To resolve the zero gradient problem, IR-Net [14], for example, proposes adopting a large value range for gradients in the early stage of training while gradually shrinking its range in the later training steps. Our BASQ addresses the zero gradient problem by training with multiple clip values which offer various sizes of value ranges for gradient flow.

In order to investigate the difference between BNN and BASQ, we set the precision of BASQ as 1-bit and evaluate it on ImageNet [4]. We use the sign function as a weight quantization function. For activation quantization, we use the linear quantization function ranging from zero to clip value which is commonly used for the 2-bit to 4-bit models as in the main paper. As a result, our activation quantization of 1-bit gives only two values, zero and the clip value.

Table 4: Binary BASQ in ResNet-18 on ImageNet

Method	A1/W1 (binary)
Xnor-Net [15]	51.2
Bi-real-Net [12]	56.4
Xnor-Net++ [2]	57.1
IR-Net [14]	58.1
CI-Net [17]	59.9
ReActNet [11]	65.9
BASQ (Ours)	64.60

Table 4 compares our binary BASQ (under 10-fold evaluation) with the existing BNNs based on ResNet-18 [9]. Our binary BASQ offers 1.3% lower accuracy than ReActNet [11] while showing superior accuracy to the other binary networks. Considering that ReActNet benefits most of accuracy improvement from PReLU activation with shift terms, it is quite promising that our binary model shows 1.3% difference with ReLU activation and without sign activation function. We leave further investigation on the potential of BASQ on binary neural networks as our future work.

6 Comparison with NAS Method

Table 5 compares BASQ with the neural architecture search (NAS) methods for mixed-precision. We utilize BitOPs metric as in [7]. For example, in the case of the 2-bit model, we multiply the BitOPs from the network with binary precision by 2^2 (2-bit for activation and weight each). The table shows that our method outperforms the existing NAS methods with similar BitOps.

Table 5: Comparison of BASQ and NAS methods for mixed precision in ResNet-18 operation scale on ImageNet

Method	BitOPs	Accuracy(%)
SPOS [7]	6.21G	66.4
BASQ (2-bit)	6.70G	68.60
DNAS [18]	15.62G	68.7
SPOS [7]	13.49G	69.4
BASQ (3-bit)	15.08G	71.40
DNAS [18]	25.70G	70.6
SPOS [7]	24.31G	70.5
BASQ (4-bit)	26.82G	72.56

References

1. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
2. Bulat, A., Tzimiropoulos, G.: Xnor-net++: Improved binary neural networks. arXiv:1909.13863 (2019)
3. Choi, J., Wang, Z., Venkataramani, S., Chuang, P., Srinivasan, V., Gopalakrishnan, K.: Pact: Parameterized clipping activation for quantized neural networks. arXiv:1805.06085 (2018)
4. 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)
5. Esser, S., McKinstry, J., Bablani, D., Appuswamy, R., Modha, D.: Learned step size quantization. International Conference on Learning Representations (ICLR) (2020)
6. 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)
7. 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)
8. 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)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Computer Vision and Pattern Recognition (CVPR) (2016)
10. Kim, J., Bhalgat, Y., Lee, J., Patel, C., Kwak, N.: Qkd: Quantization-aware knowledge distillation. arXiv:1911.12491 (2019)
11. 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)
12. Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., Cheng, K.: Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. European Conference on Computer Vision (ECCV) (2018)
13. Park, E., Yoo, S.: Profit: A novel training method for sub-4-bit mobilenet models. European Conference on Computer Vision (ECCV) (2020)
14. Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., Song, J.: Forward and backward information retention for accurate binary neural networks. Computer Vision and Pattern Recognition (CVPR) (2020)
15. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. European Conference on Computer Vision (ECCV) (2016)
16. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. Computer Vision and Pattern Recognition (CVPR) (2018)
17. Wang, Z., Lu, J., Tao, C., Zhou, J., Tian, Q.: Learning channel-wise interactions for binary convolutional neural networks. Computer Vision and Pattern Recognition (CVPR) (2019)

18. 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)
19. Yamamoto, K.: Learnable companding quantization for accurate low-bit neural networks. Computer Vision and Pattern Recognition (CVPR) (2021)