Searching for Robust Binary Neural Networks via Bimodal Parameter Perturbation

Daehyun Ahn*
SqueezeBits Inc.
daehyun.ahn@squeezebits.com

Hyungjun Kim*
SqueezeBits Inc.
hyungjun.kim@squeezebits.com

Taesu Kim*
SqueezeBits Inc.
taesu.kim@squeezebits.com

Eunhyeok Park
Pohang University of Science and Technology
eh.park@postech.ac.kr

Jae-Joon Kim
Seoul National University
kimjaejoon@snu.ac.kr

Abstract

Binary neural networks (BNNs) are advantageous in performance and memory footprint but suffer from low accuracy due to their limited expression capability. Recent works have tried to enhance the accuracy of BNNs via a gradient-based search algorithm and showed promising results. However, the mixture of architecture search and binarization induce the instability of the search process, resulting in convergence to the suboptimal point. To address this issue, we propose a BNN architecture search framework with bimodal parameter perturbation. The bimodal parameter perturbation can improve the stability of gradient-based architecture search by reducing the sharpness of the loss surface along both weight and architecture parameter axes. In addition, we refine the inverted bottleneck convolution block for having robustness with BNNs. The synergy of the refined space and the stabilized search process allows us to find out the accurate BNNs with high computation efficiency. Experimental results show that our framework finds the best architecture on CIFAR-100 and ImageNet datasets in the existing search space for BNNs. We also tested our framework on another search space based on the inverted bottleneck convolution block, and the selected BNN models using our approach achieved the highest accuracy on both datasets with a much smaller number of equivalent operations than previous works.

1. Introduction

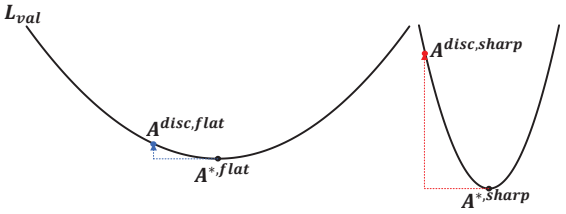
Deep neural networks (DNNs) have continuously improved with their outstanding performance on complex tasks, but the increasing amount of memory and operation cost has been consistently raised as an issue [3, 8, 11]. To reduce the overhead, model compression techniques, including quantization or pruning, have been widely studied. As an extreme case of quantization, binary neural networks (BNNs) where both input activations and weights are represented with 1-bit have advantages on saving a significant amount of memory and computational cost by replacing floating-point multiplications with simple XNOR operations with popcount [5, 20, 21, 22, 24].

However, one major drawback of BNNs is the limited-expression capability. Because the weights and activations should be restricted into the binary representation, output quality is degraded significantly compared to real-valued DNNs. Recently, several studies on employing neural architecture search (NAS) algorithm to mitigate the limitation of BNN have been reported [4, 17, 36, 38]. In particular, BNNs with differentiable architecture search (DARTS [19]) demonstrated modest gain in accuracy within the resource target, validating the potential of the architecture search for binarized networks.

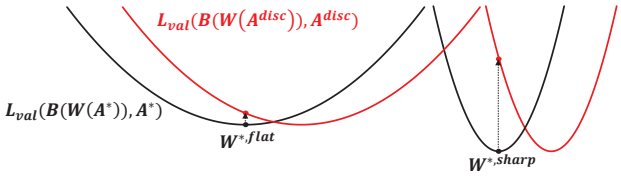
One of the important issues in DARTS is the robustness of the search process. Conventional DARTS algorithms have "collapsing cell" problem in which only parameter-free operations are selected. To mitigate this problem, several advanced techniques have recently been proposed [4, 6, 18, 17, 34, 36, 38]. Among them, we focus on the perturbation of architecture parameters. As shown in Fig. 1a, the relaxed representation of architecture A^* during searching and the discrete architecture A^{disc} for evaluating the searched model differs. Perturbing the architecture parameters leads the network to learn towards a flat minimum and helps to find a generalized model by narrowing the difference of loss values between A^* and A^{disc}.

In addition, previous studies [1, 10] showed that the flatness of loss surface regarding weight is also important for minimizing the accuracy degradation induced by the quantization operator. If we let the search process converge to

*Work done while at Pohang University of Science and Technology.



(a) Loss landscape on A axis



(b) Loss landscape on W axis

Figure 1: Illustration of loss landscape near flat and sharp minima on (a) A axis and (b) W axis.

the minimum point where the weight robustness is considered as shown in Fig. 1b, we may improve the network's output quality and transferability. However, the sharpness of the loss curve near minimum along W axis has not been considered in low-precision NAS fields yet.

Based on this intuition, we propose a BNN architecture search framework with perturbation in both architecture and weight parameters. In the proposed framework, weight parameters are trained using the loss value computed with random perturbation on architecture parameters, and architecture parameters are updated with the loss value computed by random noise-added weight parameters. Injection of the random noise to a parameter during the training phase of the other parameter can support a model to be trained to the flat minimum for both parameters.

On the other hand, we also conduct extensive studies for refining the search space of BNNs. Due to the instability of BNNs, overparameterized backbone networks are often utilized [20, 22, 17]. The inverted bottleneck block originated from MobileNet-v2 [25] are preferred in full-precison as an efficient backbone, but known to be vulnerable in low-precision [23]. We refine the inverted bottleneck judiciously by tuning the detailed configurations, i.e., groups size, expansion size, etc., for having robustness with BNNs. With bimodal perturbation, we could search for the most accurate network on the proposed search space having a much smaller number of operations than previous works.

Overall, our contributions are summarized as follows.

• We proposed a new gradient-based BNN architecture search framework using the bimodal parameter perturbation, which drives the network to reach the flat minima in the loss curve along both W and A axes. In the DARTS-based BNN search space, our framework with noise in both parameters searched for a cell architecture which achieved the state-of-the-art performance on CIFAR-100 dataset.

• We designed a modified binary inverted mobilenet convolution (MBConv [25]) block, which is tolerant to binarization error. Using the modified binary MBConv block as a backbone architecture, we established a search space that consists of the number of channels, kernel size, and expansion, and then relaxed them as architecture parameters.

• Using the proposed NAS framework based on the modified binary MBConv blocks, we demonstrated the higher accuracy on CIFAR-100 and ImageNet dataset with smaller number operations compared to previous NAS-based BNNs.

2. Related works

2.1. Binary neural networks

BNNs were first proposed in [16] using the sign function and straight-through estimator (STE) for gradient computation, showing reasonable accuracy on MNIST and CIFAR-10 dataset. A scaling factor of binarized weights adopted in [24] allowed BNNs to obtain more representation capacity so that BNNs achieved noticeable improvement in accuracy on the large ImageNet dataset. Several studies presented the techniques to increase the representation ability with negligible increase in the size of parameters or number of operations. Liu et al. [21] increased the performance of BNNs by simply adding real-valued shortcut path to every convolution layer. Martinez et al. [22] further enhanced the representation ability of BNNs by adopting a data-driven channel scaling factor, knowledge distillation and 2-stage training schedule. Trainable parameters of activation function were also introduced as a method to increase the accuracy of BNN [20, 33].

2.2. Gradient-based neural architecture search

Gradient-based NAS [6, 13, 14, 18, 19, 28, 31, 34] searches for an optimal architecture with a differentiable method, and have the faster search time than evolutionary-based [7, 39] or reinforcement-based [28, 29, 30] NAS. One of the types of the gradient-based NAS is to search for the components of each convolution layer including kernel size or channel dimension [13, 14, 28, 31]. Components of each convolution layer to be searched for are parameterized, and the optimal architecture is found during the training of a super network and the component parameters in this method. DARTS [19] is another type of gradient-based NAS. In DARTS framework, the operation candidates in a super cell are expressed as the architecture parameters, and the objective to select optimal operations in the cell is relaxed by learning architecture parameters. DARTS suffers from the instability problem stated in Sec. 1, and to solve this issue, improved DARTS techniques have been actively studied [6, 18, 34]. Binary NAS based on DARTS framework recently has emerged to search for a binarization-optimized

architecture [4, 17, 36, 38], but the same instability problem may restrict the performance of the searching algorithm.

3. Preliminaries

3.1. Binary neural networks

Given a weight matrix $W \in \mathbb{R}^{C_{out} \times C_{in} \times k \times k}$, where C_{in}, C_{out} and k stand for the number of input channels, the number of output channels, and kernel size in a convolutional layer of a neural network, respectively, W can be binarized using the following equation,

$$B(W) = \alpha_W \cdot sign(W), \tag{1}$$

where $\alpha_W = E(|W|) = \frac{1}{n_W} \cdot \sum |W|$ means an L1 norm of W [24]. Similarly, an input matrix of the convolutional layer $X \in \mathbb{R}^{C_{in} \times w \times h}$, where w and h is the width and height of X, can be binarized as,

$$X_B = \begin{cases} 1, \text{ if } X \geq 0 \\ -1, \text{ otherwise} \end{cases} \tag{2}$$

[24]. In a binary neural network, a real-valued convolution operation $\otimes$ is estimated using the above binarized $B(W)$ and X_B as $W \otimes X \approx B(W) \oplus X_B$, where $\oplus$ means a binary XNOR convolution operation with popcount, which gives significant performance improvement compared to the full-precision operations in inference. During the training, we use straight-through estimation (STE) [16] for gradient computation to progressively update $B(W)$.

3.2. Differentiable architecture search (DARTS)

DARTS [19] is one of the state-of-the-art architecture search frameworks based on the continuous relaxation of discrete decision for the architecture parameters. The objective function of DARTS is formulated as the bi-level optimization problem of the training loss L_{train} and the validation loss L_{val} as following

$$\min_A L_{val}(W^*(A), A), s.t.\ W^* = \arg\min_W L_{train}(W, A). \tag{3}$$

With Eq. (3), weight W and the matrix of architecture parameters $A = [\alpha^o_{i,j}]$ are alternatively updated by gradient descent method.

DARTS's search space is motivated by the cell-based CNN design [39]; only the normal and reduction cell architectures are searched, then the whole neural network is organized as a stack of the searched cells. Each cell is composed of nodes and edges where each node x represents an intermediate activation value and each edge $e_{i,j}$ represents a set of operation candidates as a Direct Acyclic Graph (DAG). Given an operation o in a predefined search space O, a mixed operation $\hat{o}^{i,j}(x)$ is constructed as,

$$\hat{o}^{i,j}(x) = \sum_{o \in O} \frac{\exp(\alpha^o_{i,j})}{\sum_{o' \in O} \exp(\alpha^{o'}_{i,j})} \cdot o(x), \tag{4}$$

where $\alpha^o_{i,j}$ stands for an architecture parameter corresponding to the operation o at $e_{i,j}$. The cell architecture is determined by replacing the mixed operation with the operation that has the maximum $\alpha^o_{i,j}$ value per each $e_{i,j}$ after the searching process.

4. Binary neural architecture search with bimodal parameter perturbation

4.1. Searching BNN toward flat loss landscape via bimodal parameter perturbation

In case of differentiable NAS for binary neural networks [4, 17, 36, 38], Eq. (3) is redefined with $B(W)$ as follows:

$$\begin{aligned} &\min_A L_{val}(B(W^*(A)), A), \\ &s.t.\ W^* = \arg\min_W L_{train}((B(W), A), \end{aligned} \tag{5}$$

with the search space O_B, where convolution operations are replaced with binary convolution operations. This gradient-based binary NAS method, however, also experiences the same instability problem as the real-valued DARTS, which results in the cell collapsing. To stabilize the training of architecture parameters, previous binary NAS works adopted diverse methods, including softmax with temperature [4], gumbel softmax [36], penalties to parameter-free operation or entropy-based diverse regularizer [17, 36].

The instability problem of DARTS was already addressed in recent studies [6, 18, 34], commonly indicating that the phenomenon is highly related to the magnitude of $\nabla^2_A L_{val}$. The standard DARTS is often trained to converge to sharp minima (large $\nabla^2_A L_{val}$). Because the final discrete cell architecture (A^{disc}) to be deployed after the search may not be same as the relaxed architecture (A^*) found in the search step due to the discretization of architecture parameter selection, the difference in validation loss between two architectures can be large if DARTS is trained towards sharp minima. To alleviate this issue, early-stopping [18, 34] or perturbing architecture parameter [6] was proposed, which showed aforementioned techniques helped DARTS to be trained to decrease the eigen values of $\nabla^2_A L_{val}$ and to avoid to search for the collapsed cell.

The previous works, however, did not consider $\nabla^2_W L$ which determines the sharpness of minima of loss landscape along W axis. It was reported that injecting adversarial perturbation to model parameters leads the model to learn towards flat minima, thus a more generalized model can be obtained [12, 37]. In addition, a previous study [26] showed that low-precision models also achieved a high generalization capacity when converges to the flat minima. The main objective of BNN NAS is to find an architecture robust to binarization error (and expected to achieve high accuracy on validation/test set), so making models escape from sharp

minima regarding W by using perturbation on W could be another important objective for BNN search.

Motivated by the above relationship between generalization and parameter perturbation, we propose a BNN search framework where perturbation of both architecture and weight parameters is considered. The objective function for our framework is defined as follows:

$$\begin{aligned} &\min_{A} E_{\delta_W}(L_{val}(B(W^*(A)+\delta_W),A)), \\ &s.t.\ W^* = \arg\min_{W} E_{\delta_A}(L_{train}(B(W),A+\delta_A)). \end{aligned} \tag{6}$$

Perturbation on weight (δ_W) and architecture parameters (δ_A) are complementarily injected in Eq. (6). δ_W and δ_A follow random uniform distribution $U(-\epsilon_W\alpha_W, \epsilon_W\alpha_W)$ and $U(-\epsilon_A, \epsilon_A)$ with the scaling constants ϵ_W and ϵ_A, respectively.

Next, we explain how the injected noise δ_W and δ_A can regularize $\nabla^2 L$. By applying Taylor expansion to the upper objective function of Eq. (6) and approximating, the objective function can be approximated as follows

$$\begin{aligned} &E_{\delta_W} L(B(W+\delta_W),A) \approx E_{\delta'_W} L(B(W)+\delta'_W,A) \\ &\approx E_{\delta'_W}[L(B(W),A) + \delta'_W \nabla_W L(B(W),A) \\ &+ \frac{1}{2}\delta'^T_W \nabla^2_W L(B(W),A)\delta'_W] \\ &= L(B(W),A) + \frac{\epsilon^2_W \alpha^2_W}{6} Tr\{\nabla^2_W L(B(W),A)\}, \end{aligned} \tag{7}$$

where δ'_W follows $N(0, \epsilon_W\alpha_W/\sqrt{3})$. In Eq. (7), the term including $\nabla_W L$ was removed because $E_{\delta'_W}(\delta'_W) = 0$, and the term including $\nabla^2_W L$ acts as a regularization term in addition to the original loss term. Therefore, the training process to minimize the upper objective function of Eq. (6) can lead the model to converge to flat minima of loss surface along W axis by reducing the eigen values of $\nabla^2_W L$. Likewise, the lower objective function of Eq. (6) can be also similarly approximated applying the similar approach to the one used in Eq. (7)[1]

Therefore, the bimodal perturbation on architecture and weight parameters helps the model to converge to a flat minimum of loss landscape along axes of both parameters during search process. As a result, the gradient-based BNN NAS algorithm with perturbation can find out a robust BNN architecture among candidates. Note that both random noise and adversarial noise by projected gradient descent (PGD) can regularize Hessian. In this work, we only tested random noise because BNN search with PGD requires much longer search time than with random noise which increases search cost by 20%. Detailed analysis on search cost of our proposed framework is provided in the supplementary material.

[1]Approximation of the objective function with noise of architecture parameters was also explained in [6].

Table 1: CIFAR-100 test accuracy for the searched models on BNAS search space under various perturbation condition

		max ϵ_A			
		0.0	0.1	0.2	0.3
max ϵ_W	0.0	72.30	72.43	74.33	71.33
	0.125	74.30	73.98	74.47	71.44
	0.250	75.30	75.25	**75.38**	74.53
	0.375	75.10	75.04	75.04	72.57

4.2. Bimodal parameter perturbation on BNAS

The search space used in BNAS [17] includes 1) binary convolutions with 3×3 and 5×5 kernels, 2) binary dilated convolutions with 3×3 and 5×5 kernels, 3) max pooling and average pooling operations with 3×3 kernels, and 4) a 'Zero' operation. Compared to DARTS, 'Zero' operation will remain if selected and skip-connection (a real-valued convolution) is added to normal (reduction) cell. We first conducted a grid search to find the optimal magnitude of perturbations (ϵ_A, ϵ_W), and evaluated CIFAR-100 accuracy on various size of models with the tuned perturbations.

Experiment setup for architecture search Our experimental condition for architecture search is similar to the condition stated in [17]. We searched for the architecture by training a network (both weight and activation are 1-bit) with 8 cells and 16 initial channels for 50 epochs on CIFAR-10 dataset. 50% of the training CIFAR-10 dataset was used for training weights, and the other 50% was set to validation set which was used for training architecture parameters and evaluating the performance of relaxed architecture during search. ϵ_A and ϵ_W were linearly increased from $\times 0.1$ to $\times 1$ of the maximum ϵ_A and ϵ_W value.

Experiment details of the evaluation of the searched model The searched models were trained on CIFAR-100[2] with 2-stage. At stage 1, the models with real-valued weight (b_w=32) and binary activation (b_a=1) were trained for 400 epochs. Adam optimizer with learning rate 0.002, weight decay 1e-5, and cosine annealing learning rate schedule were used. The trained model at stage 1 was retrained under (b_w, b_a)=(1,1) condition for 400 epochs with the same training condition as the stage 1 except the weight decay 0. In both stages, mixup (α=0.2) [35] and cutout [9] were applied in addition to the basic data augmentation, and the real-valued ResNet-34 participated as a teacher model.

Searching for the optimal noise scaling To search for an optimal magnitude of (ϵ_A, ϵ_W), we tested our framework in BNAS search space under various scales of perturbation on architecture/weight parameters, and the results are shown in Tab. 1. Adding random noise to either architecture parameters or weight parameters helps finding the

[2]Training dataset of CIFAR-100 is split into 90% of training set and 10% of validation set.

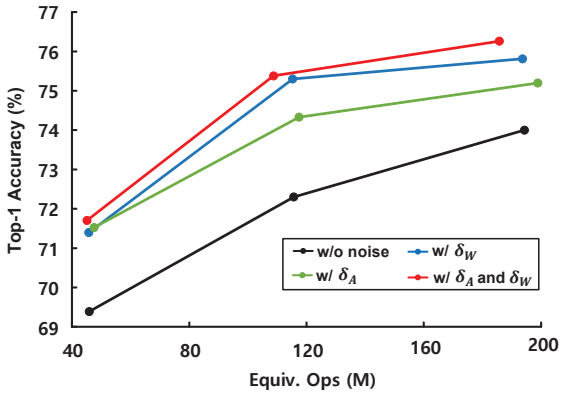


Figure 2: Test accuracy vs. the number of equivalent operations on CIFAR-100 dataset under different perturbation conditions (ϵ_A=0.2 and ϵ_W=0.25). The number of equivalent operations is computed as FLOPs + BinaryOPs/64 [36].

robust BNN architecture which achieves higher accuracy on CIFAR-100 than searching without perturbation. Perturbation of architecture parameters prevents a selected model from collapsing, while perturbation of weight parameters aids the search algorithm to find a cell more robust to binarization; both results in search models with higher generalization capacity. The combined effect of both perturbations is more effective, allowing the searching algorithm to discover the models even more robust to binarization. The optimal point with ϵ_A=0.2 and ϵ_W=0.25 shows the best accuracy. Note that too large perturbation interferes with learning, failing to find a better model. The architectures of searched cells are illustrated in the supplementary material.

Evaluation of searched BNAS models Based on the searched cell at the optimal point (ϵ_A=0.2 and ϵ_W=0.25), we further evaluated the performance of searched models with various sizes under different perturbation conditions (without noise, with noise to either weight/architecture parameters, and with noise to both parameters). As shown in Fig. 2, the searched model with both types of noise shows the highest accuracy in all aspects on the CIFAR-100. As a result, our bimodal perturbation-based NAS framework discovered the best cell architecture on the BNAS search space.

5. Differentiable binary NAS based on inverted bottleneck structure

5.1. Modified binary inverted bottleneck structure

To demonstrate the general applicability of the proposed method, we adopted the MBConv block, which is widely used for NAS [7, 15, 29, 30], as a backbone architecture. Networks based on MBConv, however, show large accuracy drop at low-precision, so we modified MBConv blocks to binary MBConv blocks as shown in Fig. 3. In the modified binary MBConv, ReAct Sign and ReAct PReLU with learnable threshold are used, which are known to increase the representation capacity of BNN [20]. In addition, shortcut path including average pooling (AvgPool) and a novel partial activation duplication is added to reduction block (Fig. 3b). It has been empirically accepted that connecting the identity connection in an end-to-end manner is effective to maintain the expression capability of BNNs [21]. However, the identity path is disconnected in the original inverted residual structure when the input/output channel sizes are different. In order to connect the identity path in a continuous manner without additional real-valued operations, we propose a partial activation duplication scheme. Given the number of input channels C_{in} and that of output channels C_{out} in a reduction binary MBConv block, the output of AvgPool X_{AP} is partially duplicated and concatenated as $[X_{AP}, X_{AP}[:C_{out} - C_{in}, :, :]]$, then added to the output of second 1×1 convolution. Without any shortcut path in the reduction block, BNN models with binary MBConv block failed to converge, but shortcut path with partial duplication allows the models to achieve reasonable accuracy. We tested on the various types of shortcut path, and adopted the type with partial duplication which shows the highest accuracy on average. Experiments with the modified shortcut candidates are described in the supplementary material. Finally, the depthwise convolution in MBConv block is replaced with group binary convolution in binary MBConv block because depthwise convolution is known

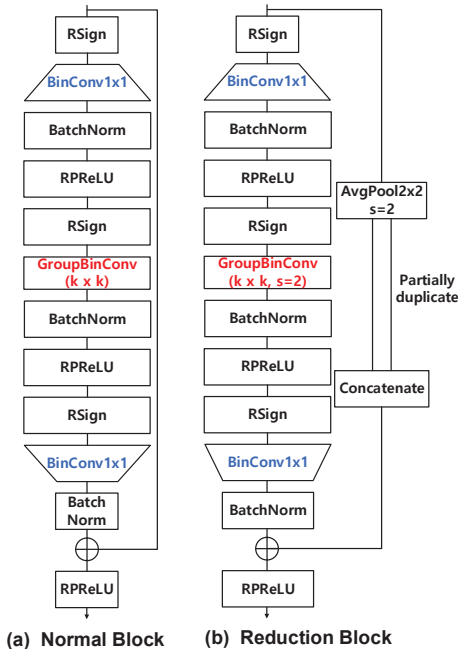


Figure 3: Structure of modified binary (a) normal and (b) reduction MBConv block. RSign and RPReLU stand for ReAct Sign and ReAct PReLU, respectively.

to be difficult to binarize due to the double approximation problem [4]. Based on the modified binary MBConv block, we search the various number of input/output channels (C), expansion e, and kernel size of group binary convolution (k), which are included in the search space.

5.2. Parameterization of search space based on binary MBConv

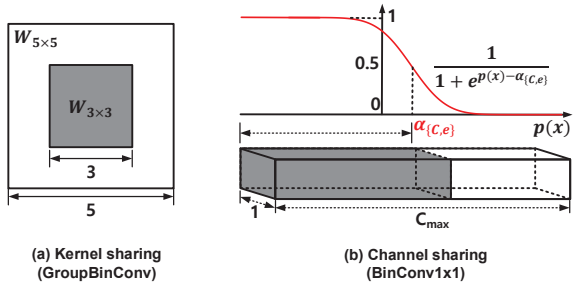


Figure 4: Parameterization methods to search for a optimal binary MBConv block architecture using (a) kernel sharing [27] and (b) channel (for input channel and expansion search) sharing.

For gradient-based BNN search, the search targets (C, e and k) need to be parameterized, and we chose the widely used weight sharing method [13, 14, 27, 31, 32]. In case of kernel size (k), we adopted the kernel sharing method introduced in the Single-Path NAS [27]. Using the kernel sharing method, $k \in \{3, 5\}$ of the group binary convolution can be parameterized with a super 5×5 kernel $W_{5\times5}$, 3×3 sub-kernel $W_{3\times3}$, and a kernel-gating parameter α_k as follows (Fig. 4a)

$$W_{5\times5} = W_{3\times3} + \frac{1}{1+\exp(-\alpha_k)} \cdot W_{5\times5-3\times3}. \quad (8)$$

The sign of the trained α_k during search determines the kernel size of group binary convolution to 3 or 5.

Next, we parameterize the number of channels C and expansion e as single gating parameters α_C and α_e. As shown in Fig. 4b and Eq. (9), the sigmoid value decided by the channel position x (parameterized to $p(x)$) and α_C or α_e is multiplied by the input of the first or second 1×1 convolution per channel, respectively.

$$W'_{[:,x,1,1]} = \frac{1}{1+\exp(p(x)-\alpha_{\{C,e\}})} \cdot W_{[:,x,1,1]} \quad (9)$$

After training α_C and α_e during search process, $p(\alpha_C)$ and $p(\alpha_e)$ determines the number of channels and expansion of each modified MBConv block.

The search space of BNN architecture based on modified binary MBConv is described in Tab. 2. Search space size is as large as $\approx 10^{45}$ for CIFAR-100 and $\approx 10^{40}$ for ImageNet dataset. To search for a network with the target number of operations t_{OPs}, we used the regularization term $\max(\log(OPs(\alpha_k, \alpha_C, \alpha_e)/t_{OPs}), 0)$, in which $OPs(\alpha_k, \alpha_C, \alpha_e)$ stands for the expected number of operations computed with α_k,α_C,and α_e.

Table 2: Network architecture and search space based on binary MBConv for (a) CIFAR-100 and (b) ImageNet. 'c', 'k', 'e', 's', and 'l' stand for the number of output channels, kernel size (3 or 5), the number of expanded channels, stride, and the number of layers, respectively. Two numbers in bracket means the minimum and maximum number of output/expanded channels whose step is 4

(a) Search space for CIFAR-100 dataset

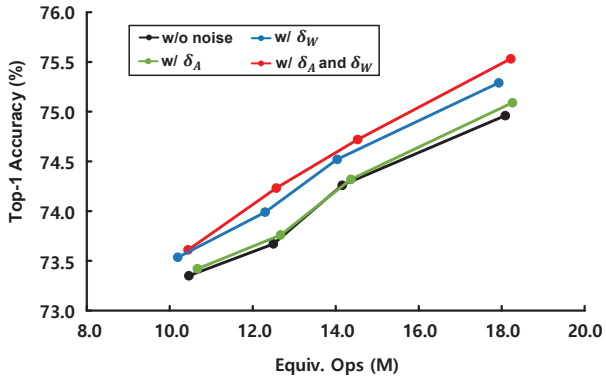
Type	c	k	e	s	l
Conv	[48, 96]	3	-	1	1
BinMBConv	[48, 96]	{3, 5}	[144,576]	1	4
BinMBConv	[96, 192]	{3, 5}	[144,576]	2	1
BinMBConv	[96, 192]	{3, 5}	[288,1152]	1	3
BinMBConv	[160, 384]	{3, 5}	[288,1152]	2	1
BinMBConv	[160, 384]	{3, 5}	[480,2304]	1	3
BinMBConv	[320, 512]	{3, 5}	[480,2304]	2	1
BinMBConv	[320, 512]	3	[960,3072]	1	3
AvgPool	[320, 512]	-	-	-	1
FC	100	-	-	-	1

(b) Search space for ImageNet dataset

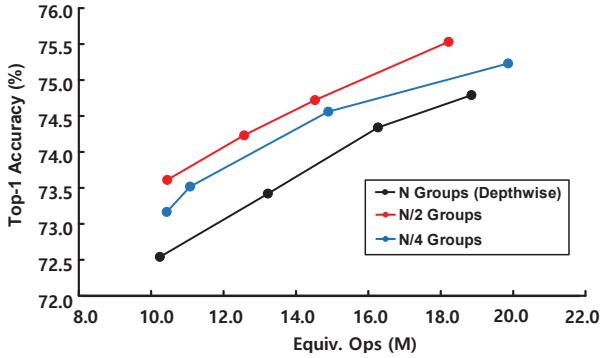
Type	c	k	e	s	l
Conv	32	3	-	2	1
BinConv	64	3	-	1	1
BinMBConv	[96, 144]	{3, 5}	[192,384]	2	1
BinMBConv	[96, 144]	{3, 5}	[288,864]	1	1
BinMBConv	[128, 288]	{3, 5}	[288,864]	2	1
BinMBConv	[128, 288]	{3, 5}	[384,1728]	1	1
BinMBConv	[256, 512]	{3, 5}	[384,1728]	2	1
BinMBConv	[256, 512]	{3, 5}	[768,3072]	1	5
BinMBConv	[512, 768]	{3, 5}	[768,3072]	2	1
BinMBConv	[512, 768]	{3, 5}	[1536,4608]	1	1
AvgPool	[512, 768]	-	-	-	1
FC	1000	-	-	-	1

5.3. Bimodal parameter perturbation on NAS with binary MBConv block

Experimental setup for CIFAR-100 The super network ($(b_w, b_a) = (1, 1)$) was trained for 80 epochs during searching. Architecture parameters were not updated during the first 40 epochs, but perturbation was added to architecture parameters to learn representation under diverse architecture conditions. Adam optimizer with learning rate 0.002 and weight decay 0 and the cosine annealing learning rate schedule was used for training weight parameters. A fixed learning rate 5e-4 and weight decay 0 were used for training



(a) Test accuracy of models searched under different perturbation conditions



(b) Test accuracy of models with various number of groups for group binary convolution

Figure 5: Test accuracy vs. the number of equivalent operations on CIFAR-100 dataset with searched (a) models under different perturbation condition ($N/2$ groups) and (b) models with various number of groups searched with bimodal parameter perturbation on binary MBConv search space.

architecture parameters. The searched models were trained on CIFAR-100 with the same training schedule, hyperparameters, and data augmentation stated in Sec. 4.2.

Experimental setup for ImageNet We used ImageNet-100, which has $\times 0.1$ size of ImageNet with 100 labels for searching for a model, and used ImageNet for evaluating the searched models. 20% of validation set were randomly selected from training dataset of ImageNet-100 as following the widely used setup for NAS on ImageNet [31]. The super network ((b_w, b_a)=(32,1)) was first trained for 100 epochs without searching, then the trained super network at the previous step was retrained under (b_w, b_a)=(1,1) condition for 100 epochs. Architecture parameters were not updated during the first 50 epochs, but perturbation was added to architecture parameters. Adam optimizer with learning rate 5e-4, weight decay 0, and linear learning rate schedule was used for training weight parameters. The fixed learning rate 5e-4 and weight decay 0 were used for training architecture parameters. The searched model was trained under (b_w, b_a)=(32,1) condition for 128 epochs. Adam optimizer with learning rate 5e-4, weight decay 1e-5 and linear learning rate schedule were used. The trained model at stage 1 were retrained under (b_w, b_a)=(1,1) condition during 128 epochs with the same training condition at stage 1 except weight decay 0. In both stages, only the basic data augmentation was applied, and the real-valued ResNet-34 participated as a teacher model.

Experimental results on search space of binary MBConv block We first found the optimal perturbation scale point (ϵ_A, ϵ_W)=(0.3,0.5), where the searched model showed the best performance on CIFAR-100 (74.23%) with about 13M equivalent Ops. Fig. 5a describes CIFAR-100 test accuracy on searched models using $N/2$ groups in group binary convolution with N input channels under different perturbation conditions. Injecting noise to both weight/architecture parameters during search helps to find the architectures more optimized for binarization similar to the BNAS cases in Sec. 4.2. The searched models based on binary MBConv block achieve $> 75\%$ accuracy on CIFAR-100 with 83.2% reduced number of operations compared to models selected on BNAS. We further searched and evaluated the model by varying the number of groups of group binary convolution using our framework, and its result in Fig. 5b showed that $N/2$ is the optimal number of groups for the CIFAR dataset in which the highest accuracy is achieved with a similar number of operations. Similarly, we found that $N/4$ groups is optimal for ImageNet dataset.

Interestingly, we observed that the perturbation on architecture parameters in the binary MBConv-based search space was not as effective as the cases in the BNAS search space. The difference of the effect on generalization comes from the difference in the search space and the target to search. The objective of BNAS is to search for the optimal operations in a cell, whereas the objective of binary MBConv block-based NAS is to search for the optimal size of kernel or channels in a fixed operation. Because the architecture components targeted to find are relaxed in both search spaces, both BNAS and MBConv block-based NAS experience the difference in loss between the relaxed architecture representation during search and its discrete representation for evaluation ($|L(A^{disc}) - L(A^*)|$). But, the $|L(A^{disc}) - L(A^*)|$ is much more prominent on BNAS (discrepancy due to eliminating operations) than on binary MBConv block-based BNN search (discrepancy due to Sigmoid function), and hence perturbing architecture parameters in DARTS-based BNAS achieves better generalization effect.

Comparison with the previous works Tabs. 3 and 4 shows the comparison between the best model from our scheme and those from previous works. With the lower number of FLOPs and equivalent operations, our searched network achieves the higher accuracy than other BNN models that were handcrafted or searched with gradient-based

Table 3: Top-1 validation accuracy on ImageNet-1K for BNN models searched for with gradient-based NAS. 'Params' are computed by assuming that 32-bit is required for full-precision parameters and 1-bit for binarized parameters. Params of BARS [36] cannot be computed because channel dimensions of the searched models are not available. In case of BNAS [17], Ops and Params are calculated including real-valued convolutions at shortcut paths in reduction blocks

Model	BOPs (G)	FLOPs (M)	Equiv. Ops (M)	Params (MB)	Top-1 Acc. (%)
BNAS-D [17]	1.024	303.1	319.1	16.0	57.7
BNAS-E [17]	1.156	341.4	359.5	17.8	58.8
BNAS-F [17]	1.439	341.4	363.9	18.1	59.0
BNAS-G [17]	1.258	403.2	422.9	20.6	59.8
BNAS-H [17]	5.433	1190	1275	57.6	63.5
BARS-D [36]	1.645	129.0	154.7	N/A	53.2
BARS-E [36]	2.848	161.0	250.5	N/A	56.2
BARS-F [36]	5.188	254.0	335.1	N/A	60.3
BATS [4]	1.149	80.50	98.45	N/A	60.4
BATS (2×) [4]	2.157	121.0	154.7	N/A	66.1
Ours	**4.344**	**21.64**	**89.52**	5.78	**68.2**

Table 4: Top-1 validation accuracy on ImageNet-1K for manually designed BNN models

Model	BOPs (G)	FLOPs (M)	Equiv. Ops (M)	Params (MB)	Top-1 Acc. (%)
BNN [16]	1.695	131.4	157.9	4.18	42.2
XNOR-Net [24]	1.695	133.3	159.8	4.18	51.2
Bi-Real-Net [21]	1.676	154.4	180.6	4.18	56.4
Real-to-bin [22]	1.676	156.4	182.6	4.18	65.4
XNOR++ [5]	1.695	133.3	159.8	4.18	57.1
MeliusNet22 [2]	4.620	135.0	162.0	3.9	63.6
ReActNet-A [20]	4.820	12.0	87.31	7.89	69.4
Ours	4.344	21.64	89.52	5.78	68.2

method. Despite the outstanding performance of the inverted mobile structure in the full-precision domain, it is hard to utilize the benefit of the structure in the low-precision domain due to its vulnerability in quantization. The refined structure improves the robustness of the search space, and the proposed bimodal perturbation guides the search process toward a more robust space. The synergy of the two contributions produces the state-of-the-art BNN results for the ImageNet scale task. In addition, while previous BNN models heavily relied on floating point operations (>64% of equivalent operations) to improve the accuracy, only 24% of equivalent operations are floating point operations in our model.

6. Conclusion

We proposed a gradient-based binary architecture search framework based on bimodal parameter perturbation to alleviate the instability problem of gradient-based NAS methods. We mathematically derived that random noise injected into architecture/weight parameters could drive a model to a stable flat minima in the loss landscape. Experimental results on DARTS-based BNN search showed that our proposed method allowed the search algorithm to find more robust BNN architecture. To search for more operation-efficient BNNs using our framework, we modified the MBConv block to be optimized to binarization, and used it as a backbone layer for search. In the search space for the ImageNet datasets based on our modified binary MBConv block, our proposed framework found an architecture that achieved higher performance with the smaller number of operations than previous BNN NAS methods.

7. Acknowledgement

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00105, Development of model compression framework for scalable on-device AI computing on Edge applications (80%), No. 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University) (10%), and No.2021-0-02068, Artificial Intelligence Innovation Hub (10%)).

References

[1] Milad Alizadeh, Arash Behboodi, Mart van Baalen, Christos Louizos, Tijmen Blankevoort, and Max Welling. Gradient ℓ_1 regularization for quantization robustness. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[2] Joseph Bethge, Christian Bartz, Haojin Yang, Ying Chen, and Christoph Meinel. Meliusnet: An improved network architecture for binary neural networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1439–1448, 2021.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[4] Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. Bats: Binary architecture search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 309–325. Springer, 2020.

[5] Adrian Bulat and Georgios Tzimiropoulos. Xnor-net++: Improved binary neural networks, 2019.

[6] Xiangning Chen and Cho-Jui Hsieh. Stabilizing differentiable architecture search via perturbation-based regularization. In *International Conference on Machine Learning*, pages 1554–1565. PMLR, 2020.

[7] Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, et al. Fbnetv3: Joint architecture-recipe search using predictor pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16276–16285, 2021.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[9] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017.

[10] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: hessian aware quantization of neural networks with mixed-precision. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 293–302. IEEE, 2019.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[12] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

[13] Yushuo Guan, Ning Liu, Pengyu Zhao, Zhengping Che, Kaigui Bian, Yanzhi Wang, and Jian Tang. DAIS: automatic channel pruning via differentiable annealing indicator search. *CoRR*, abs/2011.02166, 2020.

[14] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision*, pages 544–560. Springer, 2020.

[15] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[16] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in neural information processing systems*, pages 4107–4115, 2016.

[17] Dahyun Kim, Kunal Pratap Singh, and Jonghyun Choi. Bnas v2: Learning architectures for binary networks with empirical improvements. *CoRR*, abs/2110.08562, 2021.

[18] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. Darts+: Improved differentiable architecture search with early stopping, 2020.

[19] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018.

[20] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In *European Conference on Computer Vision*, pages 143–159. Springer, 2020.

[21] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*, pages 722–737, 2018.

[22] Brais Martinez, Jing Yang, Adrian Bulat, and Georgios Tzimiropoulos. Training binary neural networks with real-to-binary convolutions. *arXiv preprint arXiv:2003.11535*, 2020.

[23] Eunhyeok Park and Sungjoo Yoo. PROFIT: A novel training method for sub-4-bit mobilenet models. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, volume 12351 of *Lecture Notes in Computer Science*, pages 430–446. Springer, 2020.

[24] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. *CoRR*, abs/1603.05279, 2016.

[25] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4510–4520. Computer Vision Foundation / IEEE Computer Society, 2018.

[26] Sungho Shin, Jinhwan Park, Yoonho Boo, and Wonyong Sung. Hlhlp: Quantized neural networks training for reaching flat minima in loss surface. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5784–5791, 2020.

[27] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-path NAS: designing hardware-efficient convnets in less than 4 hours. *CoRR*, abs/1904.02877, 2019.

[28] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[29] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[30] Mingxing Tan and Quoc V Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021.

[31] Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12965–12974, 2020.

[32] Jiaxing Wang, Haoli Bai, Jiaxiang Wu, Xupeng Shi, Junzhou Huang, Irwin King, Michael Lyu, and Jian Cheng. Revisiting parameter sharing for automatic neural channel number search. *Advances in Neural Information Processing Systems*, 33, 2020.

[33] Peisong Wang, Xiangyu He, Gang Li, Tianli Zhao, and Jian Cheng. Sparsity-inducing binarized neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12192–12199, 2020.

[34] Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. Understanding and robustifying differentiable architecture search. *CoRR*, abs/1909.09656, 2020.

[35] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[36] Tianchen Zhao, Xuefei Ning, Xiangsheng Shi, Songyi Yang, Shuang Liang, Peng Lei, Jianfei Chen, Huazhong Yang, and Yu Wang. Bars: Joint search of cell topology and layout for accurate and efficient binary architectures, 2021.

[37] Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8156–8165, 2021.

[38] Baozhou Zhu, Zaid Al-Ars, and H. Peter Hofstee. Nasb: Neural architecture search for binary convolutional neural networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[39] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8697–8710. Computer Vision Foundation / IEEE Computer Society, 2018.