Supplementary Material for
Learning Architectures for Binary Networks

Dahyun Kim*[1][0000−0002−0820−4214], Kunal Pratap Singh*[2][0000−0003−3113−950X], and Jonghyun Choi[1][0000−0002−7934−8434]

[1] GIST (Gwangju Institute of Science and Technology), South Korea
[2] Indian Institute of Technology (IIT) Roorkee
killawhale@gm.gist.ac.kr ksingh@ee.iitr.ac.in jhc@gist.ac.kr

Note: We use blue color numbers to refer to figures, tables, section numbers and citations **in the main paper** (*e.g.*, [21]), and use green color numbers to refer to new hyper-linked references **here** (*e.g.*, [2]).

1 Qualitative Comparison of the Searched Cell

We qualitatively compare our searched cell with the XNOR-Net cell based on the ResNet18 architecture in Fig. 1. As shown in the figure, our searched cell has a contrasting structure to the handcrafted ResNet18 architecture. Both cells contain only two 3×3 binary convolution layer types, but the extra *Zeroise* layer types selected by our search algorithm help in reducing the quantization error. The topology in which the *Zeroise* layer types and convolution layer types are connected also contributes to improving the classification performance of our searched cell. In the following subsections, we show that our searched topology yields better binary networks that outperform the architectures used in state-of-the-art binary networks.

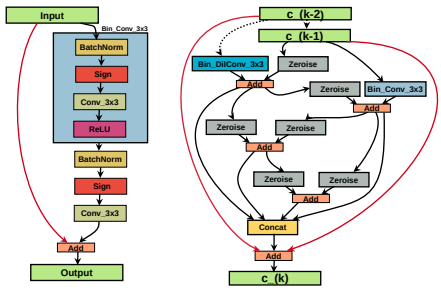
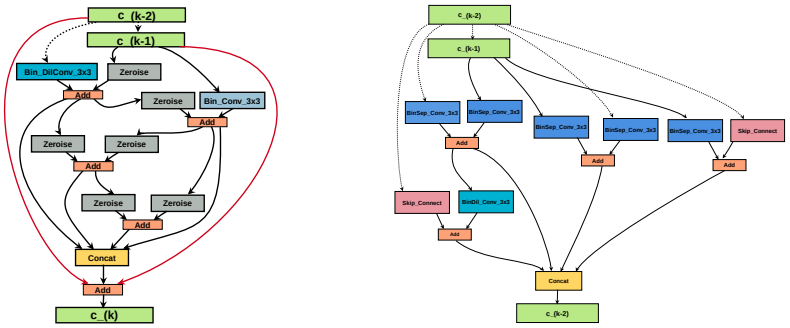


Fig. 1. Comparing BNAS cell (right) to XNOR-Net cell (left). *c_(k)* denotes the output of the kth cell. The dotted lines represents the connections from the second previous cell (*c_(k-2)*). The red lines correspond to skip connections

Below, we present more qualitative comparisons of both the normal cell and the reduction cell of our searched cell with the binarized DARTS cell [5].

Normal Cell. In Figure 2, we compare the normal cell of BNAS with the normal cell of DARTS [5]. Our cell has inter-cell skip connections which result in more stable gradients leading to better training, whereas the binarized DARTS cell does not train at all (achieving only 10.01% test accuracy on CIFAR10 in Figure 2 of the main paper). We hypothesize that the lack of inter-cell skip connections



(a) BNAS Normal Cell (b) Binarized DARTS Normal Cell

Fig. 2. Comparing the normal cell of BNAS (a) and the normal cell of binarized DARTS (b). *c_(k)* indicates the output of the k^{th} cell. The dotted lines represent the connections from the second previous cell (*c_(k-2)*). Red lines in (a) indicate the inter-cell skip connections. Note that the searched cell of binarized DARTS in (b) has only intra-cell skip connections (denoted by pink boxes) which have unstable gradients as compared to inter-cell skip connections in (a) (See discussions in Section 4.2 of the main paper)

in their cell template may also contribute to the failure of its architecture in the binary domain other than the excessive number of separable convolutions in the DARTS searched cell.

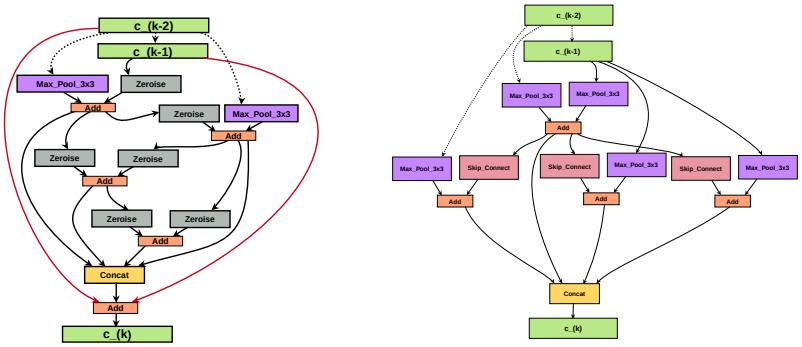
Reduction Cell. We also qualitatively compare the BNAS reduction cell to the binarized DARTS reduction cell in Figure 3. Note that the BNAS reduction cell has a lot of *Zeroise* layers which help reduce quantization error.

2 Additional Experimental Details.

Below we explain how we split our dataset for search and training and various configurations details on searching and training our architectures.

Dataset splitting. For searching binary networks, we use the CIFAR10 dataset. For training the final architectures from scratch, we use both CIFAR10 and ImageNet. During the search, we hold out half of the training data of CIFAR10 as the validation set to evaluate the quality of search. For final evaluation of the searched architecture, we train it from the scratch using the full training set and report Top-1 (and Top-5 for ImageNet) accuracy.

Details on Searching Architectures. We train a small network with 8 cells and 16 initial number of channels using SGD with the diversity regularizer (Section 4.3 of the main paper) for 50 epochs with batch size of 64. We use momentum 0.9 with initial learning rate of 0.025 using cosine annealing [7] and a weight decay of 3×10^{-4}. We use the same architecture hyper-parameters as [5] except for



(a) BNAS Reduction Cell (b) Binarized DARTS Reduction Cell

Fig. 3. Comparing the reduction cell of BNAS (a) and the reduction cell of binarized DARTS (b). *c_(k)* indicates the output of the k^{th} cell. The dotted lines represent the connections from the second previous cell (*c_(k-2)*). Red lines indicate the inter-cell skip connections. The intra-cell skip connections are denoted by the pink boxes. Interestingly, the BNAS reduction cell only uses the output from the second previous cell (*c_(k-2)*) as inputs to the max pool layers, utilizing the inter-cell skip connections more

the additional diversity regularizer where we use $\lambda = 1.0$ and $\tau = 7.7$. Our cell search takes approximately 10 hours on a single NVIDIA GeForce RTX 2080Ti GPU.

Details on Training the Searched Architectures. For CIFAR10, we train the final networks for 600 epochs with batch size 256. We use SGD with momentum 0.9 and weight decay of 3×10^{-6}. We use the one cycle learning rate scheduler [10] with the learning rate ranging from 5×10^{-2} to 4×10^{-4}. For ImageNet, we train the models for 250 epochs with batch size 512. We use SGD with momentum 0.9, with an initial learning rate of 0.1 and a weight decay of 3×10^{-5}. We use the cosine restart scheduler [7] with the minimum learning rate of 0 and the length of one cycle being 50 epochs.

Final Architecture Configurations. We vary the size of our BNAS to compare with the other binary networks with different FLOPs by stacking the searched cells and changing the output channels of the first convolutional layer and name them as BNAS-{Mini, A, B, C, D, E, F, G, H} as shown in Table 1.

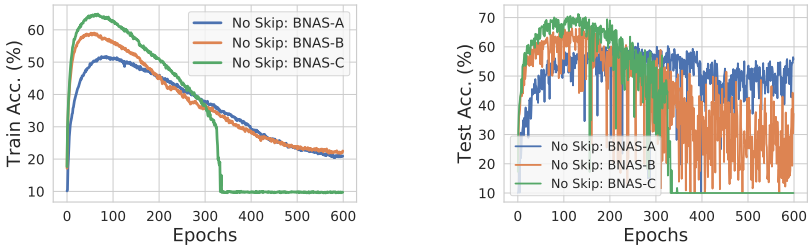
3 Additional Analyses on the Ablated Models

3.1 'No Skip' Setting

Besides the final classification accuracy presented in Table 7 of the main paper, here we additionally present the train and test accuracy curves for the *No Skip*

Table 1. Configuration details of BNAS variants. *# Cells*: the number of stacked cells. *# Chn.*: the number of output channels of the first convolution layer of the model. γ is the transferrability hyper-parameter in Eq.(5) of the main paper

BNAS-	Mini	A	B	C	D	E	F	G	H
# Cells	10	20	12	16	12	12	15	11	16
# Chn.	24	36	64	108	64	68	68	74	128
γ	1	1	1	1	3	3	3	3	3
Dataset	CIFAR10				ImageNet				



(a) Train Accuracy (%) on CIFAR10 (b) Test Accuracy (%) on CIFAR10

Fig. 4. Learning curve for the 'No Skip' ablation. The train (a) and test (b) accuracy of all three models collapse when trained for 600 epochs. Additionally, the test accuracy curves fluctuates heavily when compared to the train accuracy curve

ablation models of Table 7 of the main paper in Figure 4 for more detailed analysis. All three variants collapse to a very low training and test accuracy after a reasonable number of epochs (600).

In Figure 5, which shows the gradients of the ablated models at epoch 100 similar to Figure 4 of the main paper, we again observe that the ablated BNAS-{A,B,C} without the inter-cell skip connections have unstable (spiky) gradients. We additionally provide temporally animated plots of the gradients to demonstrate how they change at every 10 epochs starting from 100 epoch to 600 epoch in the accompanied animated gif file – 'comb_grads.gif'. Table 2 shows the details of the plots in the 'comb_grads.gif' file.

Table 2. Plot details in 'comb_grads.gif' file. We provide an animated plot for 'BNAS-A w/ SC' for comparison to those of the other ablated models without the skip connection (BNAS-A w/o SC, BNAS-B w/o SC, BNAS-C w/o SC). Other models (BNAS-B and C) with skip connections show similar trend with BNAS-A, and thus omitted for clear presentation

Plot Title	BNAS-A w/ SC	BNAS-A w/o SC	BNAS-B w/o SC	BNAS-C w/o SC
Model	BNAS-A	BNAS-A	BNAS-B	BNAS-C
Skip Connections	✓	✗	✗	✗

Note that the ablated models ('BNAS-A w/o SC', 'BNAS-B w/o SC' and 'BNAS-C w/o SC') have unstable (spiky) gradients in the early epochs while the full model ('BNAS-A w/ SC') shows relatively stable (less spiky) gradients in all epochs. All models eventually show small gradients, indicating the models have stopped learning. However, while the training curve of the full model (Figure 4) implies that it has converged to a reasonable local optima, the training curves of the ablated models (Figure 4) imply that they converged to a poor local optima instead.

3.2 'No Zeroise' Setting

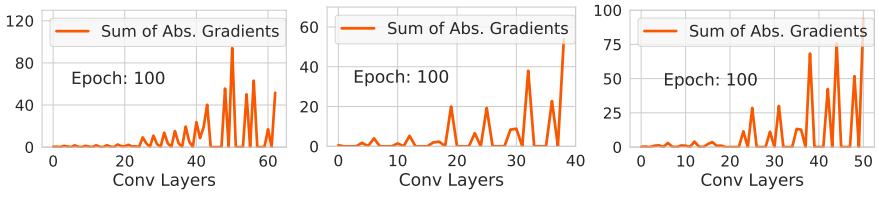
In addition to reducing the quantization error, the *Zeroise* layers has additional benefits of more memory savings, reduced FLOPs and more inference speed-up as it does not require any computation and has no learnable parameters.

We summarize the memory savings, FLOPs and inference speed-up of our BNAS-A model in Table 3 by comparing our BNAS-A model with and without *Zeroise* layers. With the *Zeroise* layers, not only does the accuracy increase, but we also observe significantly more memory savings and inference speed-up.

4 Additional Discussion on Memory Saving and Inference Speed-up of Our Method

Following that other binary networks compare memory savings and inference speed-up with respect to their floating point counterpart [6], we compute the memory savings and inference speed-up by comparing it to the floating point version of our searched binary networks and summarize the results in Table 4 for the models for experiments with ImageNet dataset.

Note that all our models achieve higher or comparable memory savings and inference speed-up for the respective FLOPs budgets compared to Bi-Real models [6].



(a) BNAS-A w/o SC (b) BNAS-B w/o SC (c) BNAS-C w/o SC

Fig. 5. Unstable gradients in the 'No Skip' ablation (Similar to 'w/o SC' in Figure 4-(b) of the main paper) of BNAS-{A,B,C} models. We show the sum of gradient magnitudes for convolution layers in all three models for the *No Skip* setting. All three models show spiky gradients without the proposed skip-connections

Table 3. Comparing our searched binary networks (BNAS-A) with and without the Zeroise layer on CIFAR10. *Test. Acc. (%)* indicates the test accuracy on CIFAR10. The two models compared have the exact same configuration except the usage of *Zeroise* layers in the search space. Note that the memory savings and inference speed-up were calculated with respect to the floating point version of BNAS-A without *Zeroise* since *Zeroise* layers are not used in floating point domain (see Section 4 in this supplement for related discussion)

BNAS-A	w/o Zeroise	w/ Zeroise
# Cells/# Chn.	20 / 36	20 / 36
Memory Savings	31.79×	**91.06**×
FLOPS ($\times 10^8$)	0.36	**0.14**
Inference Speed-up	58.01×	**149.14**×
Test. Acc. (%)	89.47	**92.70**

Table 4. Memory savings and inference speed-up compared to floating point counter part of our searched binary network (models for ImageNet experiments). Note that the memory savings and inference speed-up differ for different networks, as described in [9]. The FLOPs, memory savings and inference speed-up for Bi-Real Net is from Table 3 in [6]. All our models achieve higher or comparable memory savings and inference speed-up compared to Bi-Real Net [6]

Model	BNAS-D	BNAS-E	BNAS-F	BNAS-G	BNAS-H	Bi-Real (Bi-Real Net18)	Bi-Real (Bi-Real Net34)
FLOPs ($\times 10^8$)	~ 1.48	~ 1.63	~ 1.78	~ 1.93	~ 6.56	~ 1.63	~ 1.93
Memory Savings	13.91×	14.51×	30.37×	14.85×	21.75×	11.14×	15.97×
Inference Speed-up	20.85×	21.15×	24.29×	19.34×	24.81×	11.06×	18.99×

5 Remarks on BATS [1]

BATS [1] is a concurrent work which will be presented in the same conference. It also aims to search binary networks. Note that there are numerous differences in how BATS searches their architectures compared to ours. However, training the searched architecture has little differences, similar to how the training of P-DARTS [3] is exactly same as that of DARTS [5]. Referring to the preprint version of BATS [1], we try to compare ours to it by training the reported searched BATS architecture. Unfortunately, we were not able to reproduce the reported accuracy, even when we followed all the configuration details presented in the preprint version. Interestingly, we were able to reproduce their reported accuracy almost exactly (less than 0.1 difference in top-1 accuracy on ImageNet) by not binarizing certain layers that downsample channels or spatial resolution at the expense of using roughly 4 times more FLOPs than what was reported in BATS [1]. We reached out to the authors regarding this issue but have yet to clarify it at the time of the camera ready version deadline. When the authors release the code, we expect the issue to be resolved soon.

6 Additional Remarks on Quantized ('Non 1-bit') or not fully binary CNNs

In the introduction of the main paper, we mention that binary networks or 1-bit CNNs are distinguished from quantized networks (using more than 1 bit) and not fully binary networks (networks only with binary weights but floating point activations) due to the extreme memory savings and inference speed-up they bring. Quantized or not fully binarized networks that incorporate search are a type of efficient networks that are not comparable to 1-bit CNNs because they cannot utilize XNOR and bit counting operations in the inference which significantly brings down their memory savings and inference speed up gains.

It is, however, interesting to note that there are a line of work for efficient networks with more resource consumption, especially the recent ones. Notably, [4, 8, 11, 12] search for multi-bit quantization policies only and solely [4] search for network architectures as well. [2] also search for network architectures for binary weight (not fully binarized) CNNs. Their networks are not fully binarized (networks only with binary weights) which makes them incomparable to other binary networks. Moreover, [8, 11, 12] all search for quantization policies, not network architectures, further differentiating it from our method.

References

1. Bulat, A., Martínez, B., Tzimiropoulos, G.: Bats: Binary architecture search. ArXiv preprint arXiv:2003.01711 **abs/2003.01711** (2020) 6
2. Chen, H., Zhuo, L., Zhang, B., Zheng, X., Liu, J., Doermann, D.S., Ji, R.: Binarized neural architecture search. ArXiv **abs/1911.10862** (2019) 1, 7
3. Chen, X., Xie, L., Wu, J., Tian, Q.: Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In: ICCV. pp. 1294–1303 (2019) 6
4. Chen, Y., Meng, G., Zhang, Q., Zhang, X., Song, L., Xiang, S., Pan, C.: Joint neural architecture search and quantization. ArXiv **abs/1811.09426** (2018) 7
5. Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable architecture search. In: ICLR (2019), https://openreview.net/forum?id=S1eYHoC5FX 1, 2, 6
6. Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., Cheng, K.T.: Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In: ECCV (2018) 5, 6
7. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 2, 3
8. Lou, Q., Liu, L., Kim, M., Jiang, L.: Autoqb: Automl for network quantization and binarization on mobile devices. ArXiv **abs/1902.05690** (2019) 7
9. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: ECCV (2016) 6
10. Smith, L.N.: A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820 (2018) 3
11. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision. In: CVPR (2019) 7

12. Wu, B., Wang, Y., Zhang, P., Tian, Y., Vajda, P., Keutzer, K.: Mixed precision quantization of convnets via differentiable neural architecture search. ArXiv **abs/1812.00090** (2018) 7