BNAS v2: A Summary with Empirical Improvements

Dahyun Kim[1] Kunal Pratap Singh[2] Jonghyun Choi[1]
GIST, South Korea[1] Allen Institute for AI[2]
killawhale@gm.gist.ac.kr, kunals@allenai.org, jhc@gist.ac.kr

1. Introduction

Backbone architectures of most binary networks are well-known floating point (FP) architectures such as the ResNet family. Questioning that the architectures designed for FP networks might not be the best for binary networks, we propose to search architectures for binary networks (BNAS). Specifically, based on the cell based search method, we define the new search space of binary layer types, design a new cell template, and rediscover the utility of and propose to use the *Zeroise* layer instead of using it as a placeholder. The novel search objective *diversifies early search* to learn better performing binary architectures. We also discuss simple empirical improvements that boost the performance of the searched architectures by over 6% in top-1 accuracy on the ImageNet classification task.

2. Approach

To search binary networks, we formulate the problem of cell-based architecture search as:

$$\alpha^* = \underset{\alpha \in A(\mathcal{S},T)}{\operatorname{argmin}} \mathcal{L}_S(D;\theta_\alpha), \tag{1}$$

where A is the feasible set of architectures, $\mathcal{S}$ is the search space, T is the cell template which is used to create valid networks, L_S is the search objective, D is the dataset, θ_α is the parameters of the searched architecture α which contain both architecture parameters (used in the continuous relaxation [11], Eq. 2) and the network weights (the learnable parameters of the layer types, Eq. 2), and α^* is the searched final architecture. Following [11], we solve the minimization problem using stochastic gradient descent (SGD).

In the above formulation, we propose a new search space ($\mathcal{S}_B$), cell template (T_B) and a new search objective $\widetilde{\mathcal{L}}_S$ for binary networks and outline the core idea of the proposed components in the following. Please refer to [6] for additional details.

2.1. Search Space for Binary Networks ($\mathcal{S}_B$)

The *Zeroise* layer outputs all zeros irrespective of the input [11]. It was originally proposed to model the lack of connections but they are simply used as a placeholder layer type in [11].

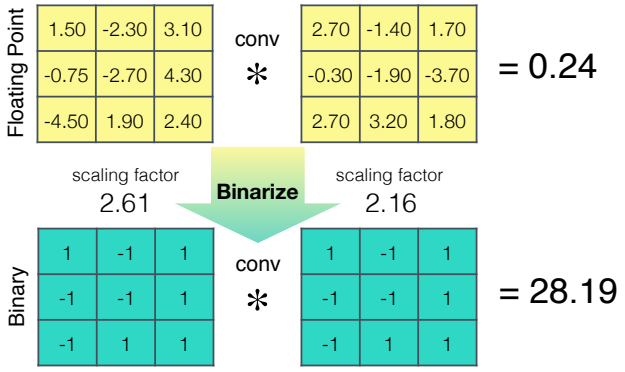


Figure 1. An example when the *Zeroise* layer is beneficial for binary networks. Since the floating point convolution is close to zero but the binarized convolution is far greater than 0, if the search selects the *Zeroise* layer instead of the convolution layer, the quantization error reduces significantly

Layer Type	Bin Conv.		Bin Dil. Conv.		MaxPool	AvgPool	Zeroise
Kernel Size	3×3	5×5	3×3	5×5	3×3	3×3	N/A

Table 1. Proposed search space for BNAS. *Bin Conv*, *Bin Dil. Conv*, *MaxPool* and *AvgPool* refer to the binary convolution, binary dilated convolution, max pooling and average pooling layers, respectively

However, we argue that the *Zeroise* layer can also reduce quantization error in binary networks such as the example shown in Fig. 1. Including the *Zeroise* layer in the final architecture is particularly beneficial when the situation similar to Fig. 1 happens frequently as the quantization error reduction is significant. Hence, we use the *Zeroise* layer for *reducing the quantization error* and first propose to keep it in the *final* architectures instead of using it as a placeholder.

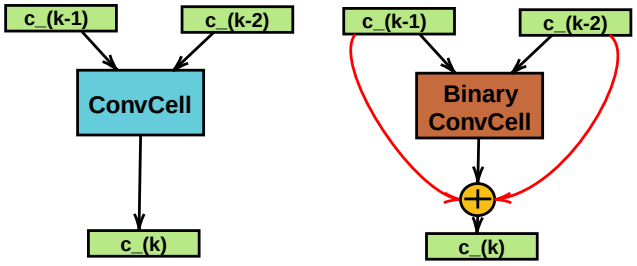
Along with the addition of the Zeroise layer, we summarize the defined search space for BNAS ($\mathcal{S}_B$) in Table 1.

2.2. Cell Template for Binary Networks (T_B)

With the defined search space, we now search for a network architecture with the convolutional cell template pro-

FLOPs ($\times 10^8$)	Method (Backbone Arch.)	Binarization Scheme	Pretraining	Top-1 Acc. (%)	Top-5 Acc. (%)
~ 1.48	BinaryNet (ResNet18) [3]	Sign	✗	42.20	67.10
	ABC-Net (ResNet18) [9]	Clip + Sign	✗	42.70	67.60
	BNAS-D	Sign + Scale	✗	57.69	79.89
	BNAS-D-No-Reg	Sign + Scale	✗	61.60	82.91
	BNAS-D v2†	Sign + Scale	✗	63.82	84.25
	BNAS-D v2 Multi-Stage†	Sign + Scale	✓	66.03	85.42
	BATS† [1]	Sign + Scale	✓	66.10	87.00
~ 1.63	Bi-Real (Bi-Real Net18) [12]	Sign + Scale	✓	56.40	79.50
	XNOR-Net++ (ResNet18) [2]	Sign + Scale*	✗	57.10	79.90
	PCNN (ResNet18) [4]	Projection	✓	57.30	80.00
	BONN (Bi-Real Net18) [5]	Bayesian	✗	59.30	81.60
	BinaryDuo (ResNet18) [7]	Decoupled	✓	60.40	82.30
~ 1.78	ABC-Net (ResNet34) [9]	Clip + Scale	✗	52.40	76.50
~ 1.93	Bi-Real (Bi-Real Net34) [12]	Sign + Scale	✓	62.20	83.90
~ 6.56	CBCN (Bi-Real Net18) [10]	Sign + Scale	✓	61.40	82.80

Table 2. Comparison of ImageNet classification performance of various binary networks. Our architectures with various training schemes are indicated by gray cells. Models with the † symbol indicates using group convolutions following [1]. 'BNAS-D-No-Reg' refers to the BNAS-D model trained with minimal regularization. 'BNAS-D v2' and 'BNAS-D v2 Multi-Stage' refers to the BNAS-D with group convolutions trained with minimal regularization or with the process used in [1].



(a) CT in DARTS (b) CT in BNAS

Figure 2. Cell templates (CT) of (a) DARTS and (b) BNAS. Red lines in BNAS indicate inter-cell skip connections. ConvCell indicates the convolutional cell. *c_(k)* indicates the output of the kth cell

posed in [14]. However, the searched architecture still suffers from unstable gradients in the binary domain as shown in Fig. 4 of [6]. To mitigate this issue, we propose to add skip-connections between multiple cells in Fig. 2.

2.3. Search Objective for Binary Networks ($\widetilde{\mathcal{L}}_S$)

During the search, the layers with learnable parameters (*e.g.*, convolutional layers) are not selected as often early on as the layers requiring no learning, which is even more severe in the binary domain. To alleviate this, we propose to use an exponentially annealed entropy based regularizer in the search objective. Specifically, we subtract the entropy of the architecture parameter distribution from the search objective as:

$$\widetilde{\mathcal{L}}_S(D;\theta_{\alpha_B}) = \mathcal{L}_S(D;\theta,p) - \lambda H(p) e^{(-t/\tau)}, \quad (2)$$

where $\mathcal{L}_S(\cdot)$ is the cross-entropy, θ_{α_B} are the parameters of the sampled binary architecture, split into the architecture parameters p and the network weights θ, $H(\cdot)$ is the entropy, λ is a balancing hyper-parameter, t is the epoch, and τ is an annealing hyper-parameter. The changed search objective will encourage the search to explore diverse layer types.

2.4. Empirical Improvements

Following [1], we use group convolutions for the first convolution layer. Additionally, we conjecture that minimizing regularization during training by removing weight decay and color jittering is beneficial because the searched binary networks could be suffering from underfitting due to excess regularization. Hence, we remove the excess regularization during training and observe non-trivial gains (Table 2).

3. Experiments

We use CIFAR10 [8] (for the search process) and ImageNet (ILSVRC 2012) [13] datasets. Please refer to [6] for more experimental details. We summarize the ImageNet classification results in Tab. 2.

'BNAS-D v2' improves upon 'BNAS-D' by over 6% on the top-1 accuracy. Empirically, we also observe the training accuracy to be noticeably higher for 'BNAS-D v2', which demonstrates that the underfitting problem is fixed. We also train BNAS-D v2 using the training scheme used in [2] ('BNAS-D v2 Multi-Stage') and find the performance to be on-par with that of the network searched in [1].

References

[1] Adrian Bulat, Brais Martínez, and Georgios Tzimiropoulos. Bats: Binary architecture search. In *ECCV*, 2020. 2

[2] Adrian Bulat and Georgios Tzimiropoulos. Xnor-net++: Improved binary neural networks. In *BMVC*, 2019. 2

[3] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016. 2

[4] Jiaxin Gu, Ce Li, Baochang Zhang, Jungong Han, Xianbin Cao, Jianzhuang Liu, and David Doermann. Projection convolutional neural networks for 1-bit cnns via discrete back propagation. In *AAAI*, 2019. 2

[5] Jiaxin Gu, Junhe Zhao, Xiaolong Jiang, Baochang Zhang, Jianzhuang Liu, Guodong Guo, and Rongrong Ji. Bayesian optimized 1-bit cnns. In *CVPR*, 2019. 2

[6] Dahyun Kim, Kunal Pratap Singh, and Jonghyun Choi. Learning architectures for binary networks. In *ECCV*, 2020. 1, 2

[7] Hyungjun Kim, Kyungsu Kim, Jinseok Kim, and Jae-Joon Kim. Binaryduo: Reducing gradient mismatch in binary activation network by coupling binary activations. In *ICLR*, 2020. 2

[8] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009. 2

[9] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *NIPS*, 2017. 2

[10] Chunlei Liu, Yue Qi, Xue Xia, Baochang Zhang, Jiaxin Gu, Jianzhuang Liu, Rongrong Ji, and David S. Doermann. Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnns with circulant back propagation. In *CVPR*, 2019. 2

[11] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 1

[12] 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 *ECCV*, 2018. 2

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 2

[14] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018. 2