Searching for Accurate Binary Neural Architectures

Mingzhu Shen Kai Han Chunjing Xu Yunhe Wang
Huawei Noah's Ark Lab
{shenmingzhu,kai.han,xuchunjing,yunhe.wang}@huawei.com

Abstract

*Binary neural networks have attracted tremendous attention due to the efficiency for deploying them on mobile devices. Since the weak expression ability of binary weights and features, their accuracy is usually much lower than that of full-precision (*i.e. *32-bit) models. Here we present a new frame work for automatically searching for compact but accurate binary neural networks. In practice, number of channels in each layer will be encoded into the search space and optimized using the evolutionary algorithm. Experiments conducted on benchmark datasets and neural architectures demonstrate that our searched binary networks can achieve the performance of full-precision models with acceptable increments on model sizes and calculations.*

1. Introduction

Convolutional neural networks (CNNs) have been widely used in various computer vision tasks, such as image classification [9], object detection [19] and visual segmentation [15]. These neural networks are often of heavy design with massive parameters and computational costs, which cannot be directly deployed on portable devices without model compressing techniques, *e.g.* pruning [8], knowledge distillation [10], compact model design [11, 22], and quantization [18, 25].

Wherein, 1-bit quantization has been recently received a great attention, which represents the weights and activations in the network using only two values, *e.g.* -1 and $+1$. Thus, binarized networks could be efficiently applied in a series of real-world applications (*e.g.* camera and mobile phone). Nevertheless, the performance of binary neural networks (BNNs) are still far worse than that of their original models. Figure 1 summarizes the performance of state-of-the-art binarization methods [16, 13, 18, 25, 14, 5] on the ImageNet benchmark [3], including XNOR-Net [18], Bi-Real Net [14], PCNN [5], *etc*. Although they have made tremendous efforts for enhancing the performance of BNNs, the highest top-1 accuracy obtained by PCNN [5] is about 12.0% lower than that of the baseline ResNet-18 [9].

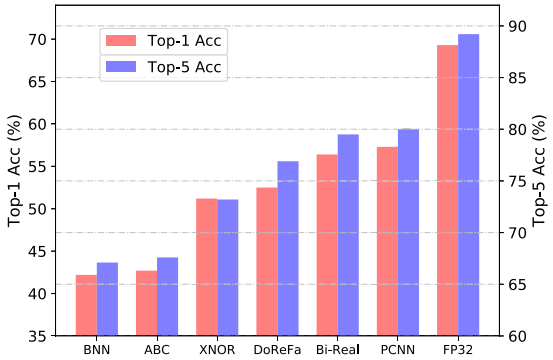

Figure 1. Performance of state-of-the-art methods for binarizing ResNet-18 on the ImageNet dataset.

The severe accuracy drop mentioned in Figure 1 greatly limits the practicality of BNNs, considering that there are a number of computer vision taks with very high precision requirements such as face recognition [21] and person re-identification [6]. The main reason could be derived from the fact that discrimination of binary features cannot match that of the full-precision features with the same dimensionality. Therefore, it is necessary to find a trade-off approach for establishing compact binary networks with acceptable model sizes by increasing the number of channels in each convolutional layer. Motivated by the recent neural architecture search (NAS [1, 4, 23]) hotspot, we present to appropriately modify channel numbers of binarized networks and search a new architecture with different channel numbers but high precision. In practice, expansion ratios of all layers in the desired binary network will be encoded to form the search space, and the evolutionary algorithm will be utilized for effectively find the lower bound of BNNs for achieving the same performance as that of their full-precision versions.

We conduct experiments on the CIFAR and ImageNet datasets using VGGNet [20] and ResNet [9] architectures. Results on these benchmarks show that the proposed approach is able to find excellent binary neural architectures for obtaining high precision with as few computation costs as possible.

2. Approach

Binarization Method. Following the widely-used DoReFa-Net [25], in the binary layer, the floating-point weights $\mathbf{w}$ is approximated by binary weights $\mathbf{w}_b$ and a floating-point scalar, while the floating-point activations $\mathbf{x}$ are represented by binary values $\mathbf{x}_b$. The feed-forward in DoReFa-Net is defined as:

$$\begin{aligned} \mathbf{w}_b &= \text{sign}(\mathbf{w}) \times E(|\mathbf{w}|), \\ \mathbf{x}_b &= \text{round}(\text{clip}(\mathbf{x}, 0, 1)), \end{aligned} \tag{1}$$

where $E(|\cdot|)$ calculates the mean of absolute value. In the back-propagation process, we adapt the "Straight-Through Estimator" method [2] to estimate the corresponding gradients. During the quantization process, we restrain the weights and activations of all convolution layers and fully-connected layers to only 1-bit except the first and last layer, following the existing works [25, 14].

The extremely binary quantization brings enormous computation acceleration and memory reduction. However, most of the state-of-the-art binary networks cannot match the accuracy of the full-precision counterpart models. Recently, the uniform width expansion proposed by WRPN [17] expands all the layers with only one hyper-parameter for multi-bit quantization networks to pursue this goal.

Although widened binary networks can obtain acceptable performance, such a uniform expansion strategy will obviously increase the required memory and computational complexities, *e.g*. the binary network after expanding $4\times$ is $16\times$ larger than the original one. In fact, there is often strong redundancy in deep neural architectures, we do not need to expand all layers for achieving the desired performance. Thus, we propose to define a binary neural architecture search problem and utilize evolutionary algorithm to search the optimal architectures.

Search Space. For the search space, we only focus on the search for network width, *i.e*. the number of the channels of each layer. For a given network architecture which has n layers, we define $\mathbf{a} \in \mathbf{R}^n$ to encode the expansion ratio hyper-parameter of each layer. Our goal is to search $\mathbf{a}$ for higher accuracy with less FLOPs. All the other hyper-parameters and network settings like stride, kernel size, layer order, remain the same as the original full-precision models.

In the uniform width expansion experiments as shown in Table 2, we observe that by only expanding channels by 4 times, binary neural networks can obtain comparable performance to that of their full-precision model on the ImageNet classification task. Thus we assume that 4 is the empirical upper bound of expansion ratio to achieve full-precision accuracy. We set 4 as the largest expansion ratio, and use some smaller ratio to expand or even reduce channels. In practice, we have 6 expansion ratio candidates in $\mathbf{a}$ which is defined as follows:

$$\mathbf{a} = [a_1, ..., a_n], \quad \forall \ a_i \in \{0.25, 0.5, 1, 2, 3, 4\}. \tag{2}$$

Search Algorithm. As discussed above, we expect to search an optimal architecture with the expansion ratio set $\mathbf{a}^*$ for making the accuracy of the binarized neural networks similar to that of its full-precision models with as few parameters and floating-number opeartions (FLOPS) as possible. Therefore, the overall optimization can be described as:

$$\begin{aligned} \max_{\mathbf{a}} \quad & f(\mathbf{w}^*(\mathbf{a}), \mathbf{a}), \\ s.t. \quad & \mathbf{w}^* = arg\min_{\mathbf{w}} \mathcal{L}_{train}(\mathbf{w}, \mathbf{a}), \end{aligned} \tag{3}$$

where $f(\cdot)$ is the *fitness* function in evolutionary algorithm and $\mathcal{L}_{train}$ is loss on train set, $\mathbf{w}^*(\mathbf{a})$ is the corresponding trained weight with expansion ratio set $\mathbf{a}$. We first find an optimal $\mathbf{a}^*$ through evolutionary algorithm on a train subset. Then we train the corresponding binary network on full train set to obtain the final model.

Specifically, in every generation during evolution, we maintain a population of K individuals, *i.e*. $\{\mathbf{a}_1, ..., \mathbf{a}_K\}$, each of which denotes a bianry neural architecture according to a certain expansion ratio code satisfying Eq. 2. These individuals will be continuously updated with pre-designed operations (*e.g*. corssover and mutation) to have greater fitness. Here we have two objects: high performance on the specific task, *e.g*. classification accuracy, and low computation costs, *e.g*. FLOPs. Thus, the fitness $f(\mathbf{a}_k)$ of an individual $\mathbf{a}_k$ is defined as:

$$f(\mathbf{a}_k) = \max(\text{Acc} - \lambda \times \text{FLOPs}, 0) \tag{4}$$

where Acc and FLOPs are the Top-1 validation accuracy and FLOPs of the corresponding widened networks of the individual $\mathbf{a}_k$, λ is the trade-off parameter.

Compared with full-precision layers, the FLOPs of binary layers are divided by 64 as suggested in Bi-Real Net [14]. In the calculation of fitness in Eq. 4, we divide the FLOPs of the candidate models by the FLOPs of original binary network to get the same order of magnitude of accuracy. After defining the search space and fitness function, the evolutionary algorithm can effectively select excellent individuals with higher fitness during the evolution process until convergence.

3. Experiments

In this section, we conduct experiments to explore the empirical width lower bound of each layer in binary neural networks on several benchmark datasets, *i.e*. CIFAR-10 [12], and ImageNet [3]. We use two widely used network structures as baselines, VGG-small [24] and ResNet-18 [9].

3.1. Experimental Settings

For the evolution search process, we search for 50 generations with 32 individuals in each generation. We train each candidate model for 10 epochs on the trainset and obtain the accuracy on validation set as the accuracy used in Eq. 4. For the trade-off parameter λ, we set it to 4 to keep the value of accuracy and FLOPs comparable.

CIFAR-10 In CIFAR-10 dataset, it takes about 12 hours on 8 V100 GPUs. Then we train 200 epochs for full CIFAR-10 training. The learning rate starts as 0.1 and multiply by 0.1 in the epochs of 60, 120 and 180. We simply follow the same hyper-parameter setup as that in [24].

ImageNet As the ImageNet ILSVRC2012 dataset is very large, we do not use the whole train dataset in evolution process. We randomly sample a subset of 50,000 images from the original full trainset which belongs to 1000 classes with 50 images for each class in the evolution process and it takes about 180 hours on 8 V100 GPUs. Then we train 150 epochs to check if searched models reaches full-precision accuracy. The learning rate starts from 0.1 and decays by 0.1 in the epochs of 50, 100 and 135. We simply follow the same hyper-parameter setup as that in [9].

Initialization When evaluating each candidate, we train 10 epochs on a small subset in ImageNet dataset, the accuracy of candidate models is especially low and makes it difficult to distinguish the better models from the worse ones. Therefore, we train the model uniformly widened by $4\times$ on the subset with 150 epochs and use it to initialize all the candidate models which we simply intercept first corresponding channels values.

Table 1. Comparison of widened binary networks of VGG-small architecture on CIFAR-10.

Models	FLOPs	Speedup	Memory	Top-1(%)
Full-Precision	608M	-	149M	**93.48**
Uniform-1×	13.2M	46.1×	7.3M	90.24
Uniform-2×	45.3M	13.4×	23.7M	91.65
Uniform-3×	96.2M	6.3×	49.3M	91.87
Uniform-4×	166M	3.7×	84.1M	92.56
VGG-Auto-A	11.3M	53.6×	5.1M	92.17
VGG-Auto-B	59.3M	10.3×	23.4M	**93.06**

3.2. Results and Analysis

VGG-small on CIFAR-10 VGG-small [24] is a variant network of the original VGG-Net [20] designed for CIFAR-10. We compare the searched models, *i.e*. Automatic-A, B, with uniformly widened models in Table 1. The standard binarized VGG-Small decreases accuracy only by about 3%. As we uniformly increase the width, the accuracy increases subsequently. However with $4\times$ widened, the accuracy of binarized network still does not achieve that of full-precision network. Our Automatic-B model achieves higher accuracy than the Uniform-$4\times$ with about 1/4 FLOPs and memory. It has the smallest accuracy gap with the full-precision model. Although our Automatic-A model even has less channels than the original Uniform-$1\times$ model, it achieves higher accuracy with about 2% improvement. This phenomenon confirms our original intention in designing the search space, that some layers need to be expanded and some layers need to be narrowed.

Table 2. Comparison of widened binary networks and other binarization methods of ResNet-18 architecture on ImageNet dataset.

Models	FLOPs	Speedup	Top-1(%)	Top-5(%)
Full-Precision	1820M	-	**69.6**	89.2
PCNN	169M	10.8×	57.3	80.0
ABC$\{5/3\}$	520M	3.5×	62.5	84.2
ABC$\{5/3\}$	785M	2.3×	65.0	85.9
Uniform-1×	149M	12.2×	52.77	76.85
Uniform-2×	352M	5.2×	64.0	85.45
Uniform-3×	607M	3.0×	68.51	88.25
Uniform-4×	915M	2.0×	70.35	89.27
Res18-Auto-A	495M	3.7×	68.64	88.46
Res18-Auto-B	660M	2.8×	**69.65**	89.08

ResNet-18 on ImageNet We also conduct experiments on the large-scale ImageNet dataset. In the uniform expansion experiments, as the width increases, the top-1 accuracy can gradually approach that of the original full-precision model. From the results in Table 2, our Automatic-B binarized model can obtain the the same performance with the full-precision model with less than 1/3 computational cost. With similar FLOPs, Automatic-B outperform Uniform-$3\times$ by 1.1% in terms of Top-1 accuracy and 0.8% Top-5 accuracy. Our evolutionary search finds a more accurate widened models with as less FLOPs as possible.

We also compare our models with some state-of-the-art binarization methods in Table 2. PCNN [5] does not quantize the downsample layer and adds additional shortcut connections which could inevitably increase end-to-end inference time. In the comparison of ABC-Net with multiple bases, which $5/3$ means 5 binary bases for weight and 3 bases for activations, Our Uniform and Automatic models consistently performs better than ABC-Net by a large margin.

Searched Architecture To further analyze the searched network architecture, we show the number of output channels in each layer of two binary networks with similar accuracy, *i.e*. Res18-Auto-A and Uniform-$3\times$ in Table 2. From Fig. 2, we observe that compared with Uniform-$3\times$, the

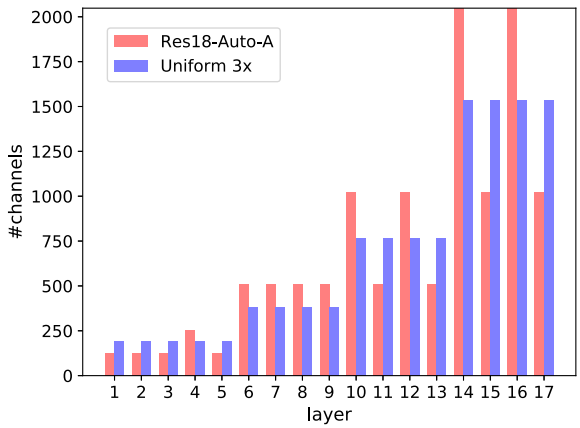

Figure 2. Number of channels in each layer of widened ResNet-18.

searched architecture Res18-Auto-A has fewer output channels in the 1st, 2nd and last stages. In addition, Res18-Auto-A needs more channels for the middle feature maps inside each block. These observations could inspire us to design blocks or architectures for more efficient convolutional neural networks.

4. Conclusion

To establish binary neural networks with higher precision and lower computational costs, this paper studies the binary neural architecture search problem. Based on the empirical study on uniform width expansion, we define a novel search space and utilize evolutionary algorithm to adjust the number of channels in each convolutional layer after binarizing. Experiments on benchmark datasets and neural architectures show that the proposed method can produce binary networks with acceptable parameters increment and the same performance as that of the full-precision original network.

References

[1] Z. Barret and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[2] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[4] R. Esteban, A. Alok, H. Yanping, and Q. V. Le. Regularized evolution for image classifier architecture search. *arXiv preprint arXiv:1802.01548*, 2018.

[5] J. Gu, C. Li, B. Zhang, J. Han, X. Cao, J. Liu, and D. Doermann. Projection convolutional neural networks for 1-bit cnns via discrete back propagation. In *AAAI*, 2019.

[6] K. Han, Y. Wang, H. Shu, C. Liu, C. Xu, and C. Xu. Attribute aware pooling for pedestrian attribute recognition. *arXiv preprint arXiv:1907.11837*, 2019.

[7] K. Han, Y. Wang, C. Zhang, C. Li, and C. Xu. Autoencoder inspired unsupervised feature selection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2941–2945. IEEE, 2018.

[8] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[10] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[12] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[13] X. Lin, C. Zhao, and W. Pan. Towards accurate binary convolutional neural network. In *NeurIPS*, pages 345–353, 2017.

[14] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*, 2018.

[15] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[16] C. Matthieu and B. Yoshua. Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. In *CoRR*, 2016.

[17] A. Mishra, E. Nurvitadhi, J. J. Cook, and D. Marr. Wrpn: wide reduced-precision networks. In *ICLR*, 2018.

[18] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, pages 525–542. Springer, 2016.

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. 2015.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[21] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[22] Y. Wang, C. Xu, X. Chunjing, C. Xu, and D. Tao. Learning versatile filters for efficient convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1608–1618, 2018.

[23] Y. Wang, C. Xu, J. Qiu, C. Xu, and D. Tao. Towards evolutional compression. *arXiv preprint arXiv:1707.08005*, 2017.

[24] C. Zhaowei, H. Xiaodong, S. Jian, and N. Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *CVPR*, pages 5918–5926, 2017.

[25] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.