Towards Binarized MobileNet via Structured Sparsity

Zhenmeng Zuo[1], Zhexin Li[1,2], Peisong Wang[2], Weihan Chen[1,2], and Jian Cheng[1,2(✉)]

[1] University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Automation, Chinese Academy of Sciences, NLPR, Beijing, China
jcheng@nlpr.ia.ac.cn

Abstract. The rising demand for deploying convolutional neural networks (CNNs) to mobile applications has promoted the booming of compact networks. Two parallel mainstream techniques include network compression and lightweight architecture design. Despite these two techniques can theoretically work together, the naive combination results in dramatic accuracy degradation. In this paper, we present Binarized MobileNet-Sp for mobile applications, by compression-architecture co-design. We first reveal the connection between MobileNets and low-rank decomposition, showing that decomposition-based architecture is not quantization friendly. Then, by adopting the view of sparsity, we propose the Binarized MobileNet-Sp, which significantly enhances the robustness to binarization. Experiments on ImageNet show that the proposed Binarized MobileNet-Sp achieves 61.2% top-1 accuracy, outperforming the naive binarization method by about 10% higher top-1 accuracy. Compared to the Bi-Real net which achieves 56.4% top-1 accuracy on the more heavy-weight and redundant ResNet-18 (which has comparable baseline accuracy with MobileNet in full-precision representation), the Binarized MobileNet-Sp achieves much higher accuracy with a significant reduction in computing complexity.

Keywords: Convolutional neural networks · MobileNet · Quantization

1 Introduction

Convolutional Neural Networks (CNNs) have been leading new state-of-the-arts in almost every computer vision tasks. One reason is the development of more advanced network architectures, like ResNet, DenseNet, etc. However, these networks are designed for higher accuracy, without optimizing the storage and computational complexity. In many real-world applications, storage consumption and latency are crucial, which on the other hand, pose great challenges to the deployment of these networks. Under this circumstance, reducing the complexity of CNNs becomes a hot topic in the computer vision field.

Z. Zuo and Z. Li—Equal contribution.

Y. Peng et al. (Eds.): ICIG 2021, LNCS 12888, pp. 688–699, 2021.
https://doi.org/10.1007/978-3-030-87355-4_57

To minimize the complexity of CNNs, two main directions are investigated by the community. The first straightforward way is to compress the learned models. Representative approaches include low-rank decomposition, sparsity, quantization, etc. Another parallel direction is to design efficient networks from scratch. Approaches like SqueezeNet, MobileNet, ShuffleNet fall into this direction. Theoretically, the two approaches mentioned above can work together to produce more efficient networks. However, lightweight networks tend to have limited redundancies, thus expressing more sensitivity to network compression. Previous compression methods are mainly evaluated on AlexNet, VGG and ResNet-18, which have the common characteristic of large bulk of convolutions. But the compressed models may still have higher complexity than the uncompressed efficient networks. For example, the deep compression reduced the size of AlexNet by 35× from 240 MB to 6.9 MB, which is still much larger than SqueezeNet of 4.8 MB. From this point of view, the compression makes more sense when combined with lightweight networks.

In this paper, we initiate the problem of compressing lightweight architectures for extremely efficient networks, and present Binarized MobileNet-Sp. We first reveal the connection between MobileNets and low-rank decomposition, showing that decomposition-based architecture is not quantization friendly. Then from the viewpoint of sparsity, the Binarized MobileNet-Sp is proposed, which significantly enhances the robustness to binarization. Experiments on ImageNet show that the proposed Binarized MobileNet-Sp achieves 61.2% top-1 accuracy, outperforming the naive binarization method by about 10% higher top-1 accuracy. Compared to the Bi-Real net which achieves 56.4% top-1 accuracy on the more heavy-weight and redundant ResNet-18 (which has comparable baseline accuracy with MobileNet in full-precision representation), the Binarized MobileNet-Sp achieves much higher accuracy with significantly reduced complexity. Our contributions are summarized as follows:

1. We initiate the problem of compressing lightweight architectures for extremely efficient networks.
2. We reveal the connection between MobileNets and low-rank decomposition, and propose a binarization robust module from the view of sparsity.
3. The proposed Binarized MobileNet-Sp dramatically outperforms traditional binarization method, achieving the new state-of-the-art on extremely efficient networks.

2 Related Work

Convolutional neural networks often suffer from significant redundancy in parameter size and computation [3]. Consequently, a bulk of works have emerged, including but not limited to low-rank decomposition, sparsity, quantization and lightweight architecture design.

Low-Rank Decomposition: The motivation behind low-rank decomposition is to find an approximate tensor $\hat{W}$ that is close to W but facilitates more efficient

computation. [4] is one of the first methods to exploit low-rank decomposition of filters by applying truncated SVD along different dimensions. By decomposing the spatial dimension $w \times h$ into $w \times 1$ and $1 \times h$, [11] achieved 4.5× speedup. [24] proposed a non-linear response reconstruction based method and [13] adopted CP decomposition to decompose a layer into five layers with 4.5× speedup for the second layer of AlexNet. Tucker decomposition was also studied in [12].

Sparsity: Pruning can remove unimportant parameters to expand the sparsity of models significantly. [6] proposed to prune the deep CNNs in an unstructured way without drops in accuracy. [5] proposed a dynamic network surgery framework which can recover the incorrectly pruned connections. [17] proposed a filter-level sparsity method which utilizing the next layer's feature map to guide filter pruning in the current layer. By adding structured sparsity regularizer, [23] proposed to reduced trivial filters, channels or even layers.

Quantization: As full-precision parameters are not required to achieve high performance, low-bit quantization has recently received increasing interest. [25] proposed incremental quantization to reduced weight precision to 2–5 bits without accuracy loss. [2,14] constrained the weights to binary(e.g. -1 or $+1$) or ternary(e.g. -1, 0 or $+1$) values to obtain acceleration in inference. Recently, several works focused on quantizing both weights and activations while minimizing performance degradation. [1] introduced Binarized Neural Networks (BNNs) with binary weights and activations, and [20] improved BNN by introducing scale factors with accuracy improvement. Multi-bit networks [15,26] are also proposed to decompose a single convolution layer into multiple binary convolution operations to achieve higher accuracy.

Lightweight Architecture Design: Some works focus on building and training lightweight networks from scratch. ResNet [7] proposed the bottleneck structure and SqueezeNet [10] replacing 3×3 convolutions with 1×1 convolutions. Based on depthwise separable convolution and linear bottlenecks, MobileNet [9] and Mobilenet V2 [21] build a lightweight model with streamlined architecture. Besides, several lightweight network [8,22] have been proposed and obtain a new state-of-the-art trade-off between accuracy and efficiency.

3 Depthwise Separable Convolution and Its Binarization

A convolutional layer maps a three-dimensional tensor $X \in \mathbb{R}^{C_{in} \times H \times W}$ to $Y \in \mathbb{R}^{C_{out} \times H \times W}$ by a four-dimensional weight tensor $W \in \mathbb{R}^{C_{out} \times C_{in} \times K \times K}$, where C_{in} and C_{out} are the numbers of input and output channels, H and W represent the spatial height and width of the input as well as the output feature maps, K denotes the kernel height and width of the weight tensor. The computational cost of standard convolution is $C_{out} \times C_{in} \times K \times K \times H \times W$, which corresponds to the kernel size times by the spatial size of the input/output feature maps. To reduce the computations, efficient representations of kernel W are designed, among which network binarization, as well as the depthwise separable convolution of MobileNets are two representative directions.

3.1 Network Binarization

In binarized neural networks, all weights and activations are constrained to be either $+1$ or -1. Specifically, we get the binarized version of weights and activations through a sign function,

$$x^b = \text{Sign}(x^r) = \begin{cases} +1 & \text{, if } x^r \geq 0 \\ -1 & \text{, otherwise} \end{cases}$$

where x^r denotes the real weights or activations. Compared to the real-valued CNN model, binarized weights obtain up to 32× memory saving. Besides, most multiply-accumulate operations could be converted into 1-bit $popcnt - xnor$ operations in the inference stage with binarized activations, which reduces computation requirement significantly. However, previous binarization methods are mainly evaluated on the networks which possess many redundancies such as AlexNet, VGG and ResNet. To further improve model efficiency, it's necessary to combine binary techniques with lightweight networks.

3.2 Standard Binarization of MobileNet

Table 1. Operations and parameters of depthwise and pointwise convolution.

	MAdds	Params
DW	17M	45K
DW percentage	3.1%	1.4%
PW	538M	3140K
PW percentage	96.4%	98.6%

Table 2. Baselines of MobileNet with its standard binarization accuracy.

Model	Top-1	Top-5
MobileNet	70.6%	–
Reproduced	70.1%	89.1%
Binary all layers	0.1%	0.5%
Binary 1 × 1	49.4%	73.3%

MobileNet is a class of streamlined and compact CNN models constructed through full utilization of depthwise separable convolutions. Briefly speaking, each depthwise separable convolution consists of two layers, i.e., depthwise convolution layer (DW) of which the number of convolution groups is equal to the number of input channels, and pointwise convolution layer (PW) with kernel size 1×1. The depthwise convolution and pointwise convolution realize intra-channel and inter-channel feature fusion, separately. Figure 1 illustrates the comparison between regular convolution and depthwise separable convolution.

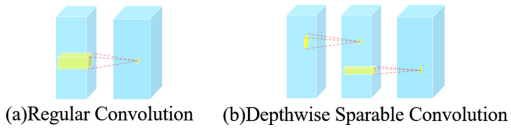

Fig. 1. Illustration of regular convolution (a) and depthwise separable convolution (b). The depthwise convolution realizes the spatial feature fusion, while pointwise convolution is responsible for the cross channel feature fusion.

Theoretically, the standard network binarization method could be naturally applied to MobileNet architecture. However, as shown in Table 2, the preliminary evaluation of direct binarization of MobileNets indicates that the network could not learn anything. We argue that this is caused by the weak representational power of binarized depthwise convolution. As illustrated in the previous section, all spatial information interactions are achieved by the depthwise convolutions, which only occupy a very small proportion of the overall computations and parameters. Table 1 gives the numbers as well as the percentages of multiply-addition operations and parameters for depthwise convolutions and pointwise convolutions, respectively. From Table 1, it is worth noting that the depthwise convolutions only take 3.1% of the computations and 1.4% of the parameters. This small proportion of resources must cover all the spatial information interaction, which will go underfitting when combined with binarization.

On the other hand, the small proportion of depthwise convolutions means that we could ignore these layers during the binarization, to achieve a better trade-off between the computation and storage gain and model accuracy degradation. Thus we propose to only binarize the input feature maps and weights of 1×1 convolutions. Table 2 shows the accuracy of full-precision MobileNet, as well as network binarization results. Only binarizing the 1×1 convolutions could achieve reasonable accuracy compared with binarizing all layers.

From the last row of Table 2 we can see that the direct binarization of MobileNet results in more than 20% top-1 accuracy loss, which means more advanced binarization technique for MobileNet-like networks are needed. In the next section, we propose a compression-architecture co-design method to improve the binarization performance of MobileNet.

4 Binarized MobileNet-Sp

In this section, we introduce our method for Binarized MobileNet-Sp in detail and step by step. We notice that the depthwise separable convolution can be considered as a kind of low-rank decomposition of the original 3×3 convolutional layer regardless of the intermediate batch norm and non-linear activation layers. Empirically, this kind of cascade decomposition method is not quantization friendly. Intuitively, it results from the information bottleneck effect along with the narrower single layer which makes it difficult to propagate gradient information, especially in binarization configuration. Inspired by this point of view, we consider altering to approximate the computation-heavy layer through sparsity connections. Going along with way, we propose our Binarized MobileNet-Sp which maintains the compact architecture of the original MobileNet while making it easier to binarize.

4.1 Low-Rank Decomposition Perspective of Separable Convolution

To better understand the low-rank decomposition characteristic behind depthwise separable convolution module, we consider a standard convolution with

parameter $W \in \mathbb{R}^{C_{out} \times C_{in} \times K \times K}$. In other words, W has C_{out} 3D filters, each filter is consists of C_{in} 2D kernels. We reveal that all kernels correspond to the i-th input channel lies in a rank-1 subspace. More specifically, let $W_{dw} \in \mathbb{R}^{C_{in} \times K \times K}$ and $W_{pw} \in \mathbb{R}^{C_{out} \times C_{in} \times 1 \times 1}$ represent the parameter tensor for depthwise and pointwise convolutional layers. Considering the kernels correspond to the i-th input channel, we have

$$W(o, i, :, :) \approx W_{pw}(o, i) * W_{dw}(i, :, :), o \in [1, C_{out}], \tag{1}$$

which indicates that the C_{out} elements (2D kernels) corresponds to the i-th input channel lie in a rank-1 subspace with basis $W_{dw}(i, :, :)$. Figure 2 shows an example of the low-rank decomposition view of the depthwise separable convolution.

The low-rank architecture makes MobileNets quite efficient compared to other networks like VGG. However, when combined with network binarization techniques, the intrinsic low-rank characteristic of the depthwise separable block may cause an information bottleneck effect especially in the backpropagation phase, where gradient approximation is needed due to the binarization of weights and input features. More specifically, even without binarizing depthwise convolutions, the small portion of the full-precision depthwise convolutions could not recover the information lost during the feature binarization step of 1×1 convolutions. Consequently, directly binarized MobileNet would converge to a poor local minimum.

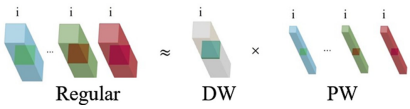

Fig. 2. Illustration of the low-rank perspective of depthwise separable convolution. All kernels of the regular filters (Regular) correspond to the i-th input channel lie in a rank-1 subspace, spaned by the i-th kernel of depthwise convolution (DW).

Fig. 3. Illustration of the proposed sparse connected convolutional filters. Each sparse convolution consists of a cross-spatial plane (yellow) and a cross-channel pillar (green) (Color figure online).

4.2 From Low-Rank to Sparse Connection

Through the above analysis, we know the cascade decomposition of standard convolution into depthwise separable convolution is not quantization friendly.

Instead, we consider altering to approximate the computation-heavy layer through sparse connections. Like in the low-rank approximation of the MobileNet building block, the sparse connection building block also needs to consider cross-spatial and cross-channel information fusion simultaneously. We take this property to the extreme, and propose the minimal sparse connected convolution, the process is shown in Fig. 3(a). Figure 3(b) lists some sparse convolutions, each of them has a cross-spatial plane and a cross-channel pillar, allowing both spatial and channel information fusion. Note that to make the sparse convolutional layer have a full perception of the input feature maps, the sparse convolutional filters have different sparse patterns, i.e., the cross-spatial planes are placed onto different positions of the cross-channel pillar.

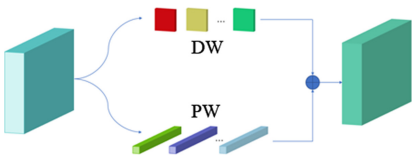

Fig. 4. Illustration of the building block for Binarized MobileNet-Sp.

Note that the sparse convolutional filter can be reformulated into the addition of two filters, i.e., the spatial filter (the plane) and the channel filter (the pillar). By collecting spatial parts and channel parts of all filters together, the sparse convolution can be reformulated into depthwise convolution and pointwise convolution, however, in a paralleled way instead of the cascade way of traditional MobileNet. Through this transformation, we get our ultimate building block for Binarized MobileNet as shown in Fig. 4. To distinguish from the traditional MobileNet, we denote our improved architecture as MobileNet-Sp, while the traditional counterpart as MobileNet-L, where "Sp" and "L" indicate the concept of sparse connection and low-rank decomposition respectively.

The sparse connection induced MobileNet-Sp and the traditional low-rank induced by MobileNet-L have several connections. (1) Both architectures use the depthwise convolution and pointwise convolution, however, the difference is that the MobileNet-L stacks these two layers while MobileNet-Sp utilizes a parallel pattern. Thus the computation and parameter size are almost the same for these two architectures. (2) When taking a global view of the whole network architectures, the MobileNet-Sp can be viewed as two twisted MobileNet-L models, with interactions across the intermediate layers.

From the above analysis, we find that like MobileNet-L, 1×1 convolution layers consume most computation and storage resources, thus we only binarize the pointwise convolution branch. More importantly, this form of network with the multi-branch structure that originates from sparsity connections could be more friendly to binarization. To be specific, the gradient backpropagation process benefits from the reserved full-precision computation and memory-efficient branch of depthwise convolution, which will be verified by detailed experiments in the following section.

5 Experiments

In this section, we thoroughly evaluate the performance of the proposed MobileNet binarization method on the ILSVRC12 ImageNet classification benchmark, as well as the properties of Binarized MobileNet-Sp through several ablation studies.

Table 3. Accuracy comparison between MobileNet-Sp and MobileNet-L.

Model	Top-1	Top-5
MobileNet (Reference)	70.6%	–
MobileNet-L	70.1%	89.1%
MobileNet-Sp	69.1%	88.6%
MobileNet-LS	70.4%	89.5%

Table 4. Accuracy comparison between binarized MobileNet-Sp and MobileNet-L.

Model	Top-1	Top-5
MobileNet (Reference)	70.6%	–
Binarized MobileNet-L	49.4%	73.3%
Binarized MobileNet-Sp	58.6%	80.9%

5.1 MobileNet-Sp vs MobileNet-L

This section compares the sparse connection based MobileNet-Sp and low-rank based MobileNet-L on the ImageNet classification task in detail. For the sake of fairness, all models for comparison are trained for 100 epochs with polynomial learning rate decay. Our results, as well as the reference MobileNet baseline, are shown in Table 3. From the results, it can be concluded that the sparse connection based MobileNet-Sp yields poorer performance than MobileNet-L, with about 1% top-1 accuracy gap. At the same time, to further compare the parallel module and the cascade module, we also report the accuracy when both the parallel and cascaded depthwise convolutions are incorporated, denoted by MobileNet-LS. The results indicate that the cascade depthwise convolution has a more powerful feature aggregation ability than parallel depthwise convolution used by MobileNet-Sp, under the circumstance of full-precision representations.

Next, we evaluate the binarization results of MobileNet-Sp and MobileNet-L. In both architectures, the depthwise convolutions, as well as the first convolution and the last fully-connected layer are not binarized. The results are shown in the last two rows of Table 4. Under the binarization setting, the behavior of these two architectures is quite different from the full-precision setting. The Binarized MobileNet-L model has a 20.7% accuracy drop than the full-precision. In contrast, the Binarized MobileNet-Sp only drops 11.5%, outperforming Binarized MobileNet-L by 9.2% top-1 accuracy, which proves our suppose.

Lightweight networks like MobileNet tend to need more training iterations to well converge. When coupled with binarization operations, it may need even more iterations. As shown in Table 5, When the training epochs doubled, i.e., from 100 epochs to 200 epochs, the top-1 accuracy improves 1.3%. There will be another 1.3% improvement when the training epochs reach 450. From the results, we can see that the binarized MobileNet-Sp can benefit from more training iterations.

Table 5. Accuracy results of Binarized MobileNet-Sp under different epochs.

Model	Epochs	Top-1	Top-5
Binarized MobileNet-Sp	100	58.6%	80.9%
Binarized MobileNet-Sp	200	59.9%	81.8%
Binarized MobileNet-Sp	450	61.2%	82.9%

Table 6. Accuracy results of different feature expanding options.

Model	Expand	Top-1	Top-5
Binarized MobileNet-Sp	Copy	57.9%	80.1%
Binarized MobileNet-Sp	CReLU	58.5%	80.6%
Binarized MobileNet-Sp	2× DW	58.6%	80.9%

5.2 Feature Expanding Options

The MobileNet architecture increases channels by 2× at each time when feature maps are reduced, which can be easily accomplished by the 1 × 1 convolutions. However, in our sparse connection induced MobileNet-Sp, the 1 × 1 convolutions need to have the same number of channels as depthwise convolutions, also the same as the number of input channels. To deal with the channel expanding problem, we propose three expanding patterns. (1) **Copy:** means to duplicate the output feature maps of the depthwise convolutions. In this pattern, no extra information and computation are produced. (2) **CReLU:** means to utilize CReLU activation function to replace ReLU function for depthwise convolutions, producing in 2× number of channels. This pattern also introduces no extra computation and parameters, however, it can utilize the negative side of features which are ignored by ReLU function. (3) **2× DW:** means to concatenate the feature maps of two distinct depthwise convolutions. In this setting, the computation and parameters are also doubled.

Table 6 illustrates the comparison results for the above three feature expanding options. It can be concluded that the naive copy pattern achieves lower performance than the other settings. The reason is that the simple copy could not introduce extra information. In contrast, the CReLU pattern achieves much better results, outperforming a simple copy pattern by 0.6% top-1 accuracy. Moreover, the 2× DW pattern brings another 0.1% top-1 improvement than using CReLU. Considering that there is only several (5 for MobileNet) expanding layers, we choose the 2× DW pattern for the following experiments.

5.3 The Effect of Feature Width and Layer Depth

The choice of layer depth and the width for each layer is a trade-off between accuracy and computing performance. Generally speaking, increasing depth and width can boost the accuracy, at the cost of increased computation and parameters. This section evaluates the trade-off performance of layer depth and width about the proposed binarized MobileNet-Sp architecture. The results are shown in Table 7.

From Table 7 it can be concluded that increasing width and depth can dramatically improve the accuracy. Another finding is that the results of doubling width or doubling depth are similar, both reach 64.8% top-1 accuracy. However, using 2× depth, the multiply-addition operations are about half of 2× width. Thus increasing depth is more efficient than increasing width.

Table 7. Accuracy of Binarized MobileNet-Sp for different width (W) and depth (D).

Model	W/D	Top-1	Top-5
Binarized MobileNet-Sp	0.5/1	51.4%	74.6%
Binarized MobileNet-Sp	0.75/1	57.4%	79.7%
Binarized MobileNet-Sp	1.0/1	61.2%	82.9%
Binarized MobileNet-Sp	2.0/1	64.8%	85.4%
Binarized MobileNet-Sp	1.0/2	64.8%	85.6%
Binarized MobileNet-Sp	0.7/2	61.9%	83.6%

Table 8. Accuracy and FLOPs comparison with other state-of-the-art binary methods.

Networks	Top-1	Top-5	FLOPs
XNOR-AlexNet [20]	44.2%	69.2%	138M
XNOR-ResNet18 [20]	51.2%	73.2%	167M
Bi-Real Net18 [16]	56.4%	79.5%	163M
MoBiNet [18]	54.4%	77.5%	52M
Binary MobileNet [19]	60.9%	82.6%	154M
Our Method	**61.2%**	**82.9%**	**52M**

5.4 Comparison with State-of-the-Art Methods

In this section, to evaluate our method, we compare our Binarized MobileNet-Sp with several recent methods. Our baseline uses 2 × DW for feature expanding and is trained for 450 epochs. As shown in the Table 8, compared with Bi-RealNet18 [16], our method improves the accuracy by 4.8% with 3× lower FLOPs. MoBiNet [18] and [19] are recent methods for binary MobileNet. Our method outperforms the MoBiNet by 6.8% with comparable speedup ratio and is more efficient than [19] with 3× lower FLOPs.

6 Conclusion

In this paper, we present Binarized MobileNet-Sp for mobile applications, by compression-architecture co-design. We first reveal the connection between MobileNets and low-rank decomposition, showing that decomposition-based architecture is not quantization friendly. Then, by adopting the view of sparsity, we propose the Binarized MobileNet-Sp, which significantly enhances the robustness to binarization. Experiments on ImageNet show that the proposed Binarized MobileNet-Sp achieves 61.2% top-1 accuracy, outperforming the naive binarization method by about 10% higher top-1 accuracy. Compared to the Bi-Real

net which achieves 56.4% top-1 accuracy on the more heavy-weight and redundant ResNet-18 (which has comparable baseline accuracy with MobileNet in full-precision representation), the Binarized MobileNet-Sp achieves much higher accuracy with a significant reduction in computing complexity.

References

1. Courbariaux, M., Bengio, Y.: BinaryNet: training deep neural networks with weights and activations constrained to +1 or −1. CoRR arXiv:1602.02830 (2016)
2. Courbariaux, M., Bengio, Y., David, J.: BinaryConnect: training deep neural networks with binary weights during propagations. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, 7–12 December 2015, Montreal, Quebec, Canada, pp. 3123–3131 (2015)
3. Denil, M., Shakibi, B., Dinh, L., Ranzato, M., de Freitas, N.: Predicting parameters in deep learning. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States, pp. 2148–2156 (2013)
4. Denton, E.L., Zaremba, W., Bruna, J., LeCun, Y., Fergus, R.: Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, 8–13 December 2014, Montreal, Quebec, Canada, pp. 1269–1277 (2014)
5. Guo, Y., Yao, A., Chen, Y.: Dynamic network surgery for efficient DNNs. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, 5–10 December 2016, Barcelona, Spain, pp. 1379–1387 (2016)
6. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, 7–12 December, 2015, Montreal, Quebec, Canada, pp. 1135–1143 (2015)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778 (2016)
8. Howard, A., et al.: Searching for MobileNetv3. CoRR arXiv:1905.02244 (2019)
9. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861 (2017)
10. Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W.J., Keutzer, K.: SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <1 MB model size. CoRR arXiv:1602.07360 (2016)
11. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions. CoRR arXiv:1405.3866 (2014)
12. Kim, Y., Park, E., Yoo, S., Choi, T., Yang, L., Shin, D.: Compression of deep convolutional neural networks for fast and low power mobile applications. arXiv:1511.06530 (2015)
13. Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I.V., Lempitsky, V.S.: Speeding-up convolutional neural networks using fine-tuned CP-decomposition. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)

14. Li, F., Liu, B.: Ternary weight networks. CoRR arXiv:1605.04711 (2016)
15. Lin, X., Zhao, C., Pan, W.: Towards accurate binary convolutional neural network. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, pp. 344–352 (2017)
16. 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: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11219, pp. 747–763. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01267-0_44
17. Luo, J., Wu, J., Lin, W.: ThiNet: a filter level pruning method for deep neural network compression. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017, pp. 5068–5076 (2017)
18. Phan, H., He, Y., Savvides, M., Shen, Z., et al.: MobiNet: a mobile binary network for image classification. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3453–3462 (2020)
19. Phan, H., Liu, Z., Huynh, D., Savvides, M., Cheng, K.T., Shen, Z.: Binarizing mobileNet via evolution-based searching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13420–13429 (2020)
20. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-net: ImageNet classification using binary convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 525–542. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_32
21. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: MobileNetv 2: inverted residuals and linear bottlenecks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June, 2018, pp. 4510–4520 (2018)
22. Tan, M., Chen, B., Pang, R., Vasudevan, V., Le, Q.V.: MnasNet: platform-aware neural architecture search for mobile. CoRR arXiv:1807.11626 (2018)
23. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, 5–10 December, 2016, Barcelona, Spain, pp. 2074–2082 (2016)
24. Zhang, X., Zou, J., He, K., Sun, J.: Accelerating very deep convolutional networks for classification and detection. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 1943–1955 (2016)
25. Zhou, A., Yao, A., Guo, Y., Xu, L., Chen, Y.: Incremental network quantization: towards lossless CNNs with low-precision weights. In: International Conference on Learning Representations (ICLR). arXiv:1702.03044 (2017)
26. Zhu, S., Dong, X., Su, H.: Binary ensemble neural network: more bits per network or more networks per bit? In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019