Supplementary Material: Dynamic Group Convolution for Accelerating Convolutional Neural Networks

Zhuo Su[1,⋆], Linpu Fang[2,⋆], Wenxiong Kang[2],
Dewen Hu[3], Matti Pietikäinen[1], and Li Liu[3,1,†]

[1] Center for Machine Vision and Signal Analysis, University of Oulu, Finland
[2] South China University of Technology, China
[3] National University of Defense Technology, China

1 Global Threshold and its Derivations

This section is, for anyone's attention, a more detailed illustration about the global threshold mentioned in Section 3.2 and Section 4.3 of the original paper.

1.1 Detailed derivations and Updating strategy

Let $\boldsymbol{x} \in \mathbb{R}^{C\times H\times W}$, $\boldsymbol{x'} \in \mathbb{R}^{C'\times H'\times W'}$ be the input and output of a particular DGC layer and the pruning rate is denoted as ξ.

For a dynamic group convolution (DGC) network, the head-wise threshold makes sure each head in the network exactly selects a certain number of channels according to the target pruning rate ξ after training, *i.e.,* $(1-\xi)C$ channels are selected from the input volume $\boldsymbol{x}$ (see Eq. 6 in the paper). While the global threshold $\mathcal{T}$ makes DGC structures more flexible allowing an uneven channel selection among heads within any DGC layer, while at the same time keeping the average pruning rate of the whole structure meeting the target ξ with tiny deviation.

To obtain $\mathcal{T}$, firstly, all saliency vectors throughout the network are collected and concatenated as a single saliency vector $\boldsymbol{G}$:

$$\boldsymbol{G} = [\boldsymbol{g}^{1,1}, \boldsymbol{g}^{1,2}, ..., \boldsymbol{g}^{1,\mathcal{H}}, \boldsymbol{g}^{2,1}, ..., \boldsymbol{g}^{\mathcal{L},\mathcal{H}}], \tag{1}$$

where, $\mathcal{L}$ and $\mathcal{H}$ represents the number of DGC layers and number of heads in each DGC layer of the network respectively, $\boldsymbol{g}^{i,j}$ is the saliency vector from the jth head in the ith DGC layer which is derived from Eq. 5 in the original paper but remove the ReLU activation to keep the negative saliencies for a further exploration. After that, similar to Eq. 6 in the paper, we find the global threshold by meeting:

$$\xi = \frac{|\{g \mid \text{abs}(g) < \mathcal{T}, g \in \boldsymbol{G}\}|}{|\{g \mid g \in \boldsymbol{G}\}|}, \tag{2}$$

where $|\mathcal{S}|$ is the length of set $\mathcal{S}$, abs(z) returns the absolute value of z.

⋆ Equal contributions. † Corresponding author: li.liu@oulu.fi

Table 1. Comparison of Top-1 classification error (%) with state-of-the-art filter-level weight pruning methods.

Model	MACs	CIFAR-10	CIFAR-100
VGG-16-pruned [4]	206M	6.60	25.28
VGG-19-pruned [6]	195M	6.20	-
VGG-19-pruned [6]	250M	-	26.52
ResNet-56-pruned [2]	62M	8.20	-
ResNet-56-pruned [4]	90M	6.94	-
ResNet-110-pruned [4]	213M	6.45	-
ResNet-110-pruned [1]	121M	6.15	-
ResNet-164-B-pruned [6]	124M	5.27	23.91
DenseNet-40-pruned [6]	190M	5.19	25.28
DenseNet-40-pruned [5]	183M	5.39	-
DenseNet-40-pruned [5]	81M	6.77	-
CondenseNet-86 [3]	65M	5.00	23.64
CondenseNet-86-DGC	71M	**4.77**	**23.41**
CondenseNet-86-DGC-G	71M	**4.42**	**23.36**

Since saliency vectors are dynamically changing during the training process, we update the global vector every three epochs based on the last N iterations at the third epoch. Therefore, assuming the batch size for each iteration is B, NB different $\boldsymbol{G}$s are obtained, which is regarded as the "saliency library" by further concatenating these $\boldsymbol{G}$s as a new $\boldsymbol{G}$ and put it to Eq. 2 to get $\mathcal{T}$. In our experiments for ImageNet, N and B is set as 5 and 256 respectively. This naive strategy works since the training set is randomly shuffled for each epoch, while other methods can also be tried such as introducing a running mean for $\mathcal{T}$ like the parameter updating process in batch normalization (BN) layers. Finally, like the BN layer, we adopt the finally updated $\mathcal{T}$ for inference. The experiments show that the actual pruning rate during testing is almost the same as ξ (see Table 1 and 2 in the paper).

1.2 Training with Angle Enlargements

We further reduce the inner product among saliency vectors from different heads within a DGC layer by introducing a third loss, in order to encourage learning saliency vectors in orthogonal directions to forcefully diversify feature representations. Our experiments show that it is automatically achieved during training if head-wise threshold is adopted, adding such loss hardly gives any improvement on the performance. However, this is slightly not the case if global threshold is applied (25.2% *vs.* 25.8% of Top-1 error of the CondenseNet-DGC structure used in Section 4.3 on ImageNet dataset with and without this angle enlargements). Specifically, the angle enlargement loss is defined as:

$$L_a = \lambda \frac{2}{\mathcal{L}\mathcal{H}(\mathcal{H}-1)} \sum_{l=1}^{\mathcal{L}} \sum_{i=1}^{\mathcal{H}} \sum_{j=i+1}^{\mathcal{H}} \text{abs}\left(\frac{\boldsymbol{g}^{l,i}}{\|\boldsymbol{g}^{l,i}\|_2} \odot \frac{\boldsymbol{g}^{l,j}}{\|\boldsymbol{g}^{l,j}\|_2} \right), \tag{3}$$

where $\odot$ represents the inner product between two vectors, $\|\boldsymbol{z}\|_2$ is the ℓ^2 norm of vector $\boldsymbol{z}$. We set λ to 10^{-4} in our experiments when using global threshold.

2 More results on CIFAR datasets

We further evaluate our method on the CondenseNet-86 structure used in [3] by replacing the learnt group convolution (LGC) of CondenseNet with DGC, and compare it with the original CondenseNet and other state-of-the-art filter-level pruning methods. The parameter settings are the same as the CondenseNet. Results are shown in Table 1. In this table, the model with a suffix "G" means we adopt the global threshold.

3 Further Visualization

Please see Fig. 1 and Fig. 2.

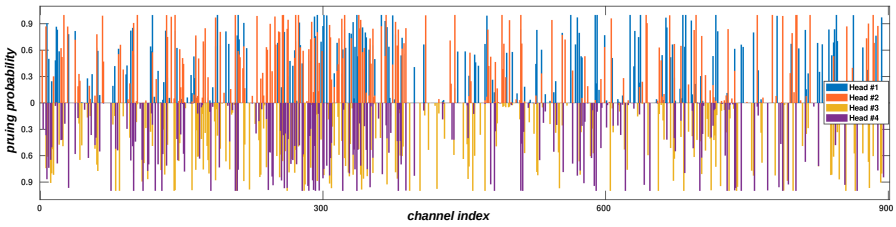

Fig. 1. Extension for Fig. 8 in the original paper, with the other two heads visualized.

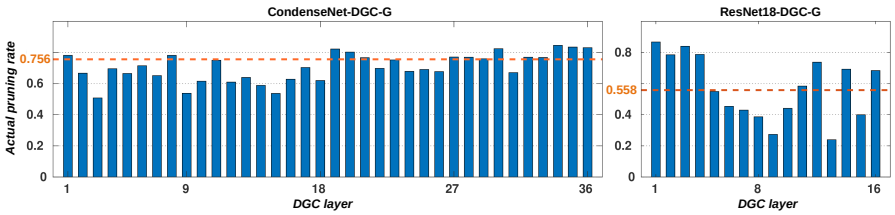

Fig. 2. Actual pruning rate of each DGC layer for CondenseNet (left) and ResNet (right) structure on the validation set of ImageNet dataset using the global threshold. *CondenseNet-DGC* corresponds to the model in Table 2 from the original paper with the same name. *ResNet18-DGC-G* corresponds to the model *DGC-G* in Table 1 from the original paper. The red line represents the overall pruning rate for the model. It can be seen that by using global threshold, the network is given more flexibility that allows each layer adapting to a particular pruning rate, leading to a slightly better performance than the one with head-wise thresholds, but at the same time bringing extra irregularity to the model structure (*e.g.*, even within a single layer, different input smaples may also lead to different numbers of channels selected). A balance between such irregularity and performance need to be considered during network design.

References

1. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4340–4349 (2019) 2
2. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1389–1397 (2017) 2
3. Huang, G., Liu, S., Van der Maaten, L., Weinberger, K.Q.: Condensenet: An efficient densenet using learned group convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2752–2761 (2018) 2, 3
4. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016) 2
5. Lin, S., Ji, R., Yan, C., Zhang, B., Cao, L., Ye, Q., Huang, F., Doermann, D.: Towards optimal structured cnn pruning via generative adversarial learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2790–2799 (2019) 2
6. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2736–2744 (2017) 2