Supplementary Material: Linear Combination Approximation of Feature for Channel Pruning

Donggyu Joo Doyeon Kim Eojindl Yi Junmo Kim
Korea Advanced Institute of Science and Technology (KAIST)
{jdg105, doyeon_kim, djwld93, junmo.kim}@kaist.ac.kr

7. Appendix - Computation of LCAF Linear Combination Approximation (Eq. 3-6)

From Eq. 3, for any i and l,

$$\arg\min_{\lambda_l^{i,k}} ||\tilde{I}_l^i - \sum_{k\neq i} \lambda_l^{i,k} \tilde{I}_l^k||. \tag{14}$$

To simplify the problem, first we flatten every tensors I to a column vector $\tilde{I}$. The above minimization problem is re-arranged as following.

$$\begin{aligned}\arg\min_{\lambda_l^{i,k}} ||\tilde{I}_l^i - \sum_{k\neq i} \lambda_l^{i,k} \tilde{I}_l^k|| &= \arg\min_{\lambda_l^{i,k}} ||\tilde{I}_l^i - \sum_{k\neq i} \lambda_l^{i,k} \tilde{I}_l^k||^2 \\ &= \arg\min_{\lambda_l^{i,k}} (\tilde{I}_l^i - \sum_{k\neq i} \lambda_l^{i,k} \tilde{I}_l^k)^\top (\tilde{I}_l^i - \sum_{k\neq i} \lambda_l^{i,k} \tilde{I}_l^k) = \arg\min_{\lambda_l^{i,k}} f\end{aligned} \tag{15}$$

where we define f as above. By using partial derivative with respect to $\lambda_l^{i,1}$, we get

$$\frac{\partial f}{\partial \lambda_l^{i,1}} = -2\tilde{I}_l^{1\top} \tilde{I}_l^i + 2\tilde{I}_l^{1\top} (\sum_{k\neq i} \lambda_l^{i,k} \tilde{I}_l^k) = 0 \tag{16}$$

Then,

$$\sum_{k\neq i} \tilde{I}_l^{1\top} \tilde{I}_l^k \lambda_l^{i,k} = \tilde{I}_l^{1\top} \tilde{I}_l^i \tag{17}$$

By repeating this to all $\lambda_l^{i,k}$, we get following system of linear equations

$$\begin{cases} \sum_{k\neq i} \tilde{I}_l^{1\top} \tilde{I}_l^k \lambda_l^{i,k} = \tilde{I}_l^{1\top} \tilde{I}_l^i \\ \sum_{k\neq i} \tilde{I}_l^{2\top} \tilde{I}_l^k \lambda_l^{i,k} = \tilde{I}_l^{2\top} \tilde{I}_l^i \\ \vdots \\ \sum_{k\neq i} \tilde{I}_l^{i-1\top} \tilde{I}_l^k \lambda_l^{i,k} = \tilde{I}_l^{i-1\top} \tilde{I}_l^i \\ \sum_{k\neq i} \tilde{I}_l^{i+1\top} \tilde{I}_l^k \lambda_l^{i,k} = \tilde{I}_l^{i+1\top} \tilde{I}_l^i \\ \vdots \\ \sum_{k\neq i} \tilde{I}_l^{n_{l-1}\top} \tilde{I}_l^k \lambda_l^{i,k} = \tilde{I}_l^{n_{l-1}\top} \tilde{I}_l^i \end{cases} \tag{18}$$

This system of linear equations can be further simplified as following form.

$$\begin{bmatrix} \tilde{I}_l^{1\top} \\ \tilde{I}_l^{2\top} \\ \vdots \\ \tilde{I}_l^{i-1\top} \\ \tilde{I}_l^{i+1\top} \\ \vdots \\ \tilde{I}_l^{n_{l-1}\top} \end{bmatrix} \begin{bmatrix} \tilde{I}_l^1 & \tilde{I}_l^2 & \cdots & \tilde{I}_l^{i-1} & \tilde{I}_l^{i+1} & \cdots & \tilde{I}_l^{n_{l-1}} \end{bmatrix} \begin{bmatrix} \lambda_l^{i,1} \\ \lambda_l^{i,2} \\ \vdots \\ \lambda_l^{i,i-1} \\ \lambda_l^{i,i+1} \\ \vdots \\ \lambda_l^{i,n_{l-1}} \end{bmatrix} = \begin{bmatrix} \tilde{I}_l^{1\top} \\ \tilde{I}_l^{2\top} \\ \vdots \\ \tilde{I}_l^{i-1\top} \\ \tilde{I}_l^{i+1\top} \\ \vdots \\ \tilde{I}_l^{n_{l-1}\top} \end{bmatrix} \tilde{I}_l^i \tag{19}$$

This equation is again arranged as following final form.

$$A^\top A\Lambda = A^\top \tilde{I}_l^i \tag{20}$$

where $A = \begin{bmatrix} \tilde{I}_l^1 & \tilde{I}_l^2 & \cdots & \tilde{I}_l^{i-1} & \tilde{I}_l^{i+1} & \cdots & \tilde{I}_l^{n_{l-1}} \end{bmatrix}$ and $\Lambda = \begin{bmatrix} \lambda_l^{i,1} & \lambda_l^{i,2} & \cdots & \lambda_l^{i,i-1} & \lambda_l^{i,i+1} & \cdots & \lambda_l^{i,n_{l-1}} \end{bmatrix}^\top$.

After we forward sample of training images, each feature map value I is obtained. For the linear combination approximation, the only thing we need is to solve Equation 20 which contains simple matrix operation. By solving above simplified problem, we find Λ without heavy computation. And then, each ϵ values can also be easily calculated with a simple operation.

8. Appendix - Computation of Global Criterion (Eq. 13)

For any feature map I , the loss difference $\Delta\mathcal{L}$ after pruning is a function $w.r.t.$ a feature map I. The tensor I is composed of $b \times h \times w$ elements. Then, the loss difference $\Delta\mathcal{L}$ is also a function of each element in I. We denote each element of I simply as $\tilde{I}_{\mathbf{b},\mathbf{h},\mathbf{w}}$ where $\mathbf{b} = \{1, \cdots, b\}, \mathbf{h} = \{1, \cdots, h\}$ and $\mathbf{w} = \{1, \cdots, w\}$ for the convenience. This notation is independent from the notation used in the paper. It is used for the detailed explanation in this appendix.

The first order Taylor polynomial of f at a is generally expressed as following:

$$f(x) = f(a) + f'(a)(x-a) + R_1(x) \approx f(a) + f'(a)(x-a) \tag{21}$$

In conventional pruning, for each element $I_{\mathbf{b},\mathbf{h},\mathbf{w}}$, when we prune this element ($I_{\mathbf{b},\mathbf{h},\mathbf{w}}$ becomes 0), the first order Taylor approximation of the loss $\mathcal{L}$ is expressed as following:

$$\mathcal{L}(0) \approx \mathcal{L}(I_{\mathbf{b},\mathbf{h},\mathbf{w}}) - \frac{\partial \mathcal{L}}{\partial I_{\mathbf{b},\mathbf{h},\mathbf{w}}} I_{\mathbf{b},\mathbf{h},\mathbf{w}} \tag{22}$$

Then,

$$\Delta\mathcal{L}(I_{\mathbf{b},\mathbf{h},\mathbf{w}}) \approx -\frac{\partial \mathcal{L}}{\partial I_{\mathbf{b},\mathbf{h},\mathbf{w}}} I_{\mathbf{b},\mathbf{h},\mathbf{w}} \tag{23}$$

This approximation can be applied to every element in I. Therefore, the pruning of entire feature map I can be simply expressed as following:

$$\begin{aligned} |\Delta\mathcal{L}(I)| &= |\sum_{\mathbf{b}}\sum_{\mathbf{h}}\sum_{\mathbf{w}} \Delta\mathcal{L}(I_{\mathbf{b},\mathbf{h},\mathbf{w}})| \\ &\approx |\sum_{\mathbf{b}}\sum_{\mathbf{h}}\sum_{\mathbf{w}} \frac{\partial \mathcal{L}}{\partial I_{\mathbf{b},\mathbf{h},\mathbf{w}}} I_{\mathbf{b},\mathbf{h},\mathbf{w}}| \\ &= |I \cdot \frac{\partial \mathcal{L}}{\partial I}| \end{aligned} \tag{24}$$

This is the derivation for the conventional pruning which converts feature map I to 0. Since the pruning in LCAF is equivalent to converting feature map I to ϵ, the final equation for the proposed LCAF is concluded as following.

$$|\Delta\mathcal{L}(I)| = |\epsilon \cdot \frac{\partial \mathcal{L}}{\partial I}| \tag{25}$$

where ϵ is the linear combination approximation error of the corresponding feature map I.

9. Results on CIFAR-100

Table 6. Comparison results of ResNet-56 on CIFAR-100 dataset.

Method	Top-1 (%)	FLOPs ↓ (%)	Params ↓ (%)
Baseline	71.41	-	-
MIL [1]	68.37	39.3	-
SFP [3]	68.79	52.6	-
FPGM [4]	69.66	52.6	-
LFPC [2]	70.83	51.6	-
LCAF	**70.91**	**53.1**	**41.8**

The CIFAR-100 dataset demonstrates the generalization ability of the proposed LCAF because it is a much finer dataset compared to the CIFAR-10. The experimental results are shown in Table 6. The ResNet-56 model was used for this comparison. Among the previous works, LCAF achieved the highest performance while reducing the most FLOPs. Compared with LFPC [2], we achieve a 0.08% higher Top-1 accuracy while reducing 1.5% more FLOPs. Compared to FPGM [4], it even shows 1.25% of performance improvement with similar FLOPs reduction.

10. Magnified View of Figure 4

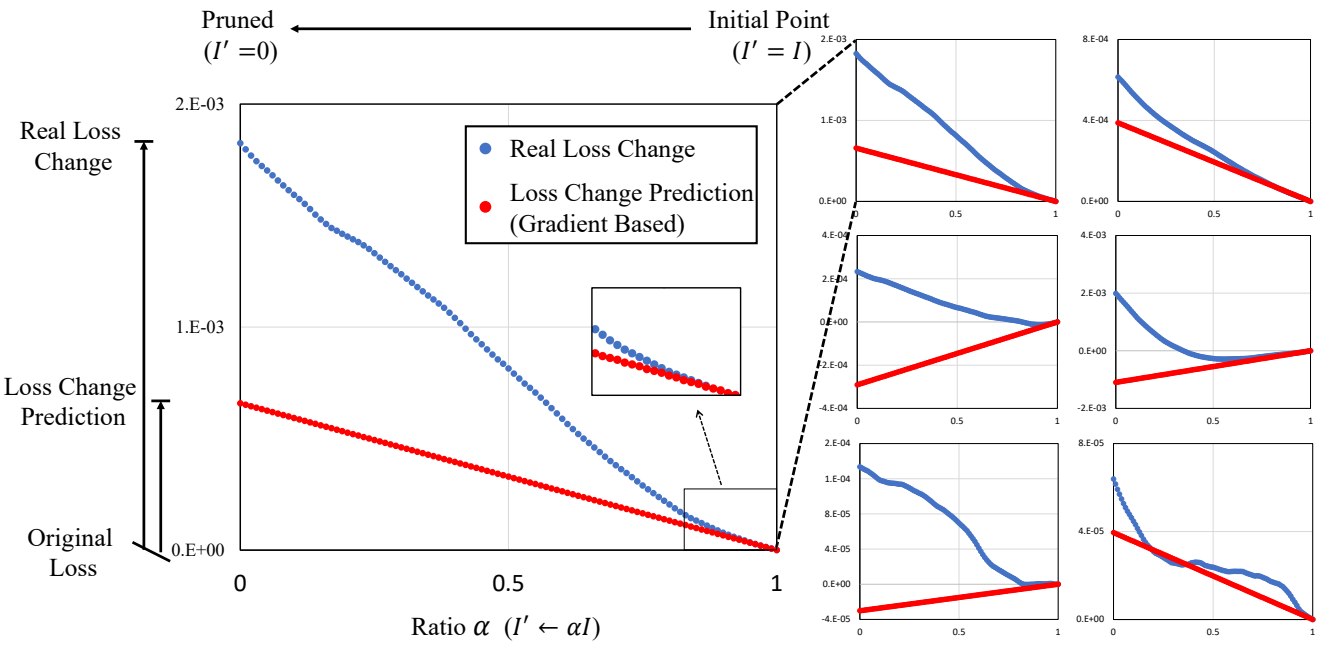

Figure 6. Magnified version of Figure 4 in manuscript for more clear view. Observations of loss change by gradually decreasing the value of each feature to zero. (Right) Each graph shows the pruning of six different features. (Left) Magnified view of one graph.

References

[1] Xuanyi Dong, Junshi Huang, Yi Yang, and Shuicheng Yan. More is less: A more complicated network with less inference complexity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5840–5848, 2017. 3

[2] Yang He, Yuhang Ding, Ping Liu, Linchao Zhu, Hanwang Zhang, and Yi Yang. Learning filter pruning criteria for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2009–2018, 2020. 3

[3] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018. 3

[4] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019. 3