Supplementary Materials for: Discrete Model Compression with Resource Constraint for Deep Neural Networks

A. Derivation of Regularization Gradient

The detail derivation of regularization gradient is given below if $\hat{T} \neq pT$:

$$\begin{aligned}\frac{\partial \mathcal{R}_{\log}}{\partial \theta_{l,c}} &= \frac{1}{|\hat{T}-pT|+1} \cdot \frac{\hat{T}-pT}{|\hat{T}-pT|} \cdot \frac{\partial \sum_{l=1}^{L} (\widehat{\text{FLOPs}})_l}{\partial \theta_{l,c}} \\ &= \frac{1}{|\hat{T}-pT|+1} \cdot \frac{\hat{T}-pT}{|\hat{T}-pT|} \\ &\cdot k_l^2 \cdot \frac{\mathbf{1}^T \mathbf{g}_{l-1}}{\mathcal{G}_l} \cdot w_l \cdot h_l \cdot \frac{\partial \mathbf{1}^T \mathbf{g}_l}{\partial \theta_{l,c}} \\ &= \eta_l \cdot \frac{1}{|\hat{T}-pT|+1} \cdot \frac{\hat{T}-pT}{|\hat{T}-pT|} \cdot \frac{\partial g(\theta_{l,c})}{\partial \theta_{l,c}} \\ &= \eta_l \cdot \frac{1}{|\hat{T}-pT|+1} \cdot \frac{\hat{T}-pT}{|\hat{T}-pT|},\end{aligned}$$

where $\eta_l = k_l^2 \cdot \frac{\mathbf{1}^T \mathbf{g}_{l-1}}{\mathcal{G}_l} \cdot w_l \cdot h_l$ .The result of the second line is due to the definition of $(\widehat{\text{FLOPs}})_l$ and $(\widehat{\text{FLOPs}})_l = k_l \cdot k_l \cdot \frac{\mathbf{1}^T \mathbf{g}_{l-1}}{\mathcal{G}_l} \cdot \mathbf{1}^T \mathbf{g}_l \cdot w_l \cdot h_l$. The result of the fourth line is because of STE: $\frac{\partial g(\theta_{l,c})}{\theta_{l,c}} = 1$, if $\theta_{l,c} \in [0, 1]$. If $\hat{T} = pT$, then $\mathcal{R}_{\log} = 0$, and 0 can be used as the subgrident of this point. Thus, we have the sub-gradient given in the paper:

$$\frac{\partial \mathcal{R}_{\log}}{\partial \theta_{l,c}} = \begin{cases} \eta_l \cdot \frac{1}{|\hat{T}-pT|+1} \cdot \frac{\hat{T}-pT}{|\hat{T}-pT|}, & \text{if } \hat{T} \neq pT \\ 0, & \text{if } \hat{T} = pT \end{cases}$$

B. Detailed Choice of p

Architecture	ResNet-34	ResNet-50	ResNet-101	MobileNetV2
p	0.55	0.38	0.42	0.50

Table 1: Choice of p for ImageNet models. p is the **remained** FLOPs divided by the total FLOPs

In this section, we will give the detail number of p. In a CNN, we do not prune the first layer, the last layer and residual connections in ResNet. As a result, the actual remained FLOPs may not equal to p. We list the choice of p for ImageNet models in Tab. 1. For CIFAR-10 models, the unpruned FLOPs is quite small, thus, p is the same as the remained fraction of FLOPs.

C. Acceleration

The cpu run time of different models are shown in Tab. 2. The input is a mini-batch of 4 images.

Architecture	ResNet-34	ResNet-50	ResNet-101	MobileNetV2
Original Time (ms/batch)	113.5	195.7	331.5	106.8
Pruned Time (ms/batch)	81.4	126.2	205.4	67.5
Improvement (%)	28.3%	35.5%	38.0%	36.8%
Pruned FLOPs (%)	43.4%	55.0%	56.0%	46.0%

Table 2: CPU time for different ImageNet Models. The time is measured in millisecond.

D. Discussion of Difference between Proposed Method and Trainable Gate [2]

In trainable gate (TG) [2], they propose to turn non-differentiable gate to differentiable gate inspired by [1]. They add a perturbation to the gate function to make it differentiable. Our method, on the other hand, does not modify the gate function and use STE to handle gradient calculation. Thus, the approach of making gate differentiable is different. Moreover, in their framework, the gate calculation is deterministic. Consequently, they can not sample sub-networks as we do. Their work also applies a form of constraint to limit the resource of the pruned neural network.

References

[1] Sangchul Hahn and Heeyoul Choi. Gradient acceleration in activation functions. 2018.

[2] Jaedeok Kim, Chiyoun Park, Hyun-Joo Jung, and Yoonsuck Choe. Plug-in, trainable gate for streamlining arbitrary neural networks. *CoRR*, abs/1904.10921, 2019.