Supplementary Materials for Network Pruning via Performance Maximization

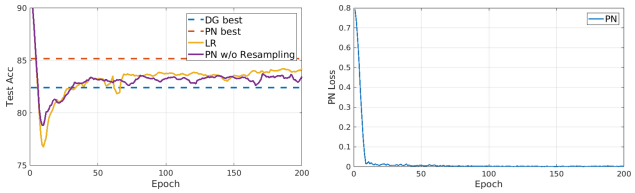

Figure 1: *Left*: More Pruning Settings. *Right*: PN Loss. Results are from CIFAR-10 with ResNet-56.

Dataset	Architecture	p
CIFAR-10	ResNet-56	0.50
	MobileNetV2	0.55
ImageNet	ResNet-34	0.50
	ResNet-50	0.38
	ResNet-101	0.42
	MobileNetV2	0.63
	ShuffleNetV2+	0.70

Table 1: Choice of p.

A. More Results

In Fig. 2, we plot the accuracy and classification loss on the test dataset given two settings: using performance maximization (PN) and differential gates (DG). From this figure, we can see that PN can outperform DG on test accuracy during pruning, but the difference of classification loss is much smaller. In summary, our method can achieve lower classification loss and higher accuracy. At some points, even the classification losses are close, the difference in accuracy can be larger than 1%, indicating that the classification loss is not always a good proxy for accuracy.

To verify the effectiveness of GRU, we add a comparison baseline for the linear regression (LR) model in Fig. 1-*Left*. Using PN is better than LR, we hypothesis that LR may omit hierarchical information of different layers while PN can capture it using GRU. The training loss of the PN is show in Fig. 1-*Right*, which is quite stable.

B. Choice of p

The choice of p can be analytically calculated. Let T_{all} be the overall FLOPs of a CNN, and T_{total} is the total prunable

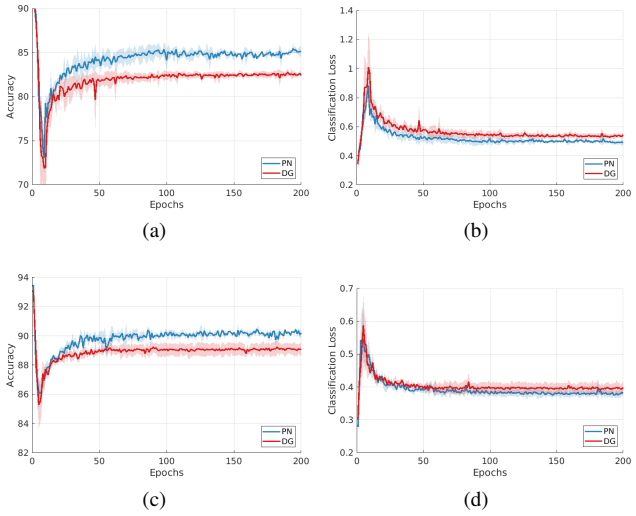

Figure 2: (a, b): Accuracy and classification loss given different settings with ResNet-56. (c, d): Accuracy and classification loss given different settings with MobileNetV2. Both experiments are done on CIFAR-10. Shaded areas represents variance from 5 runs.

Inputs a_i, i=1,$\cdots$, L
FC$_i$(C_i, 16), BN,ReLU GRU(16, 16),Avg
FC(16, 1), sigmoid

Table 2: The structure of PN used in our method.

FLOPs. Suppose we want to remove 50% of FLOPs, then $pT_{\text{total}} = 0.5T_{\text{all}}$, and $p = 0.5\frac{T_{\text{all}}}{T_{\text{total}}}$. The detailed p is listed in Tab. 1.

C. Orthogonal Projection of Gradients

Recall that $g_{\mathcal{L}}^i = \frac{\partial \mathcal{L}}{\partial \mathbf{w}_i}$ represents the gradient vector from the classification loss of ith layer, and let $g_{\text{P}}^i = \frac{\partial \log(\frac{1}{\text{PN}(\mathbf{a})})}{\partial \mathbf{w}_i}$ be the gradient vector from performance maximization. The

projection of g_{P}^i onto $g_{\mathcal{L}}^i$ is $\bar{g}_{\mathrm{P}}^i = \frac{(g_{\mathcal{L}}^i)^T(g_{\mathrm{P}}^i)}{\|g_{\mathcal{L}}^i\|^2} g_{\mathcal{L}}^i$, and

$$\hat{g}_{\mathrm{P}}^i = g_{\mathrm{P}}^i - \frac{(g_{\mathcal{L}}^i)^T(g_{\mathrm{P}}^i)}{\|g_{\mathcal{L}}^i\|^2} g_{\mathcal{L}}^i, \tag{1}$$

is orthogonal to $g_{\mathcal{L}}^i$. Thus, we have $g_{\mathrm{P}}^i = \bar{g}_{\mathrm{P}}^i + \hat{g}_{\mathrm{P}}^i$.

D. Structure of the Performance Prediction Network

The structure of the performance prediction network is shown in Tab. 2, where FC is a fully-connected layer, Avg average the outputs of all steps of GRU and $i = 1, \ldots, L$.