Efficient Layer-Adaptive Weight Pruning via Joint Optimization - Supplementary Material

Anonymous ICCV submission

Paper ID 7352

1. Implementation Details

Our implementation is built in Pytorch. Compared to baseline LAMP, we only changed the layerwise sparsity selection part to ours. For CIFAR-10 models, we chose SGD optimizer with learning rate of 0.1 and weight decay of 0.0001 for each round of finetuning. For ImageNet ones, we chose SGD optimizer with learning rate of 0.1 and weight decay of 0.0005 for each round of finetuning. other hyper-parameters settings the same (see LAMP [1] Appendix A).

For rate-distortion curve sampling, we sample 100 points for each curve. Following are some examples of empirically generated rate-distortion curves.

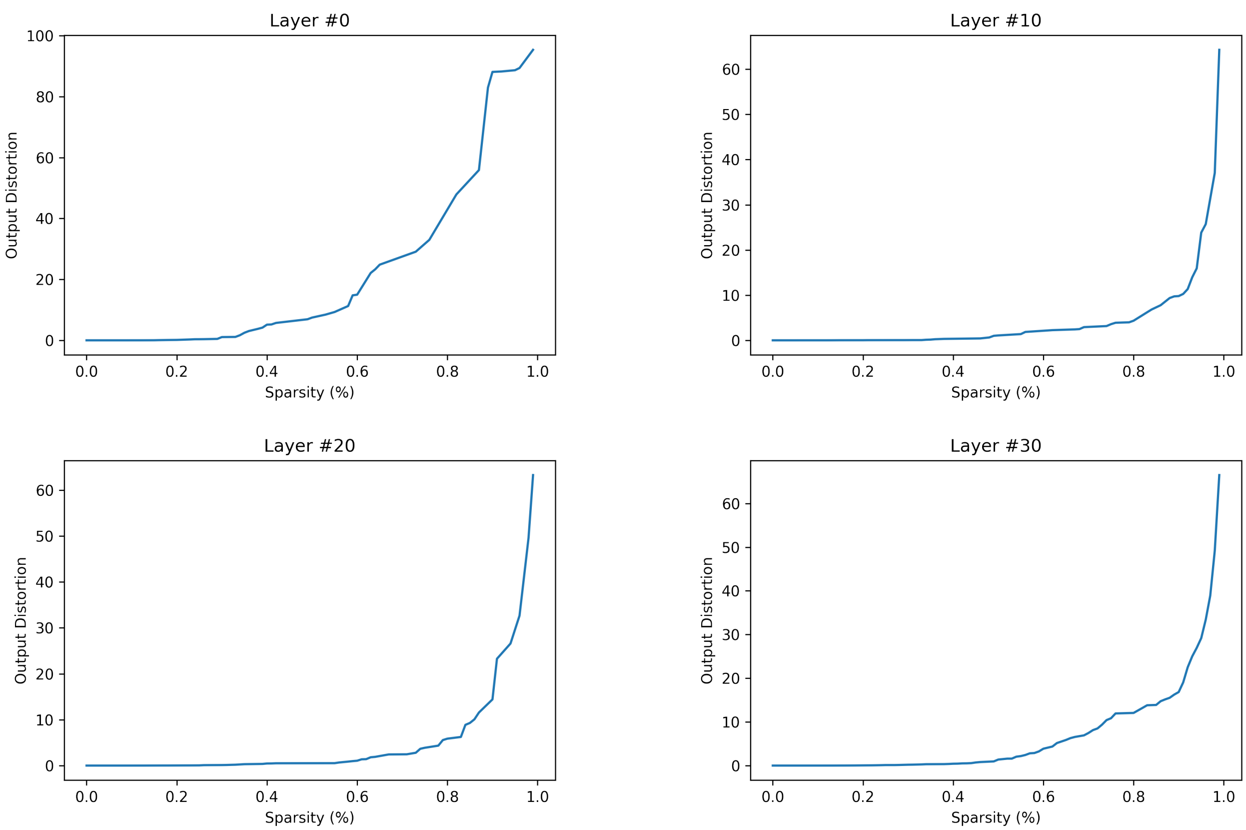
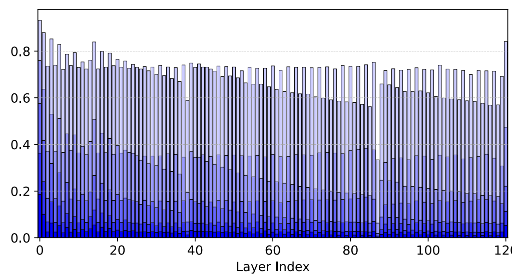


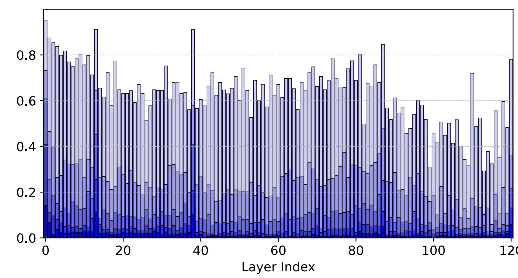
Figure 1: Example of rate-distortion curves of ResNet-32 on CIFAR-10.

2. More layer-wise distribution

As mentioned in the experiment section in the main text, we look into the layerwise statistics of DenseNet-121. As shown in Fig. 2, both methods demonstrate similar tendencies across different layers, where sparsity drops in deep layers. This implies that DenseNet has less redundant parameters in deep layers.



(a) LAMP.



(b) Ours.

Figure 2: Layer-wise sparsity of DenseNet-121 on CIFAR-10 of different methods during iterative pruning with $\{0.36, 0.74, 0.89, 0.96, 0.98\}$ model sparsities.

References

[1] Jaeho Lee, Sejun Park, Sangwoo Mo, Sungsoo Ahn, and Jinwoo Shin. Layer-adaptive sparsity for the magnitude-based pruning. In *International Conference on Learning Representations*, 2020.