6. Supplementary material

In supplementary material we show additional experimental results on ResNet20 with CIFAR10, ResNet101 on ImageNet. Additionally, we evaluate inference speed of pruned ResNet101 models.

6.1. ResNet20 on CIFAR10

We experiment on ResNet20 trained on CIFAR10 in order to compare with the work of [32] (referred to as BN-ISTA) and to evaluate the effect of the "pruning paradox" reported in [1]. Our setup is as follows: initial model was trained for 200 epochs with learning rate 0.1 and decay by 10 after 80 epochs. Final model obtained 92% on the test split and we picked it as an initial model for pruning. Pruning and fine-tuning setup is: initial learning rate of 0.1, decayed by 10 every 20 epochs for a total number of 70 epochs. While pruning, we remove 10 neurons every 30 mini-batches until the predefined number of pruned neurons is reached.

Results of pruning and training from scratch are summarized in the Table 4. We observe that pruning with Random or magnitude based criteria results in the worst performance, primary because they introduce uncorrelated bias to the estimate, we also observe that these 2 methods can be affected by "pruning paradox" as their difference is within a standard deviation of experiments. Our proposed method that relies on estimating feature importance with Taylor expansion of first and second orders outperform BT-ISTA[32]. The difference between Optimal Brain Damage and Our methods is not large and within a single standard deviation. We conclude that first order Taylor expansion applied to the gates after BN is a reasonable choice for residual network. It is not affected by "prunung paradox" discovered in [1].

6.2. Additional details on ResNets pruning

Our method can be applied with various pruning scheduling. The scheduling we apply in the paper, named here as *iterative*, removes 100 neurons per every 30 mini-batch updates until we reach predefined number of neurons. Also, all neurons can be removed at once, named as *pruning with a single step*. One more setting, named as *continuous*, prunes 100 neurons every 30 mini-batches only if the training loss is above the predefined threshold (we set it to 1.04).

Progress of ResNet-101 pruning on ImageNet with 3 different pruning scheduling settings is illustrated in Fig. 6. All settings had the maximum number of neurons to be pruned as 10000 out of 20096, and the *Iterative* corresponds to *TaylorFO-BN-50%* in the main paper. Iterative pruning clearly outperforms other settings over all epochs.

Finetuning details on ImageNet dataset. When a small number of neurons are removed we found that starting with the smaller learning rate works better. Therefore we use

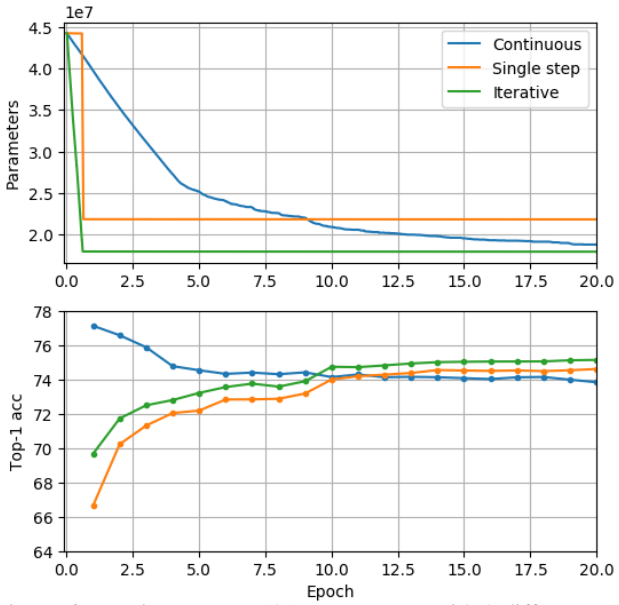

Figure 6: Pruning ResNet-101 on Imagenet with 3 different settings.

starting learning of 0.001 for the following pruning models: ResNet-101 (Taylor-FO-BN-40%, Taylor-FO-BN-55% and Taylor-FO-BN-50% and Taylor-FO-BN-22%), ResNet-50 (Taylor-FO-BN-56%). All other networks were finetuned with initial learning of 0.01. Weight decay is set to 0.0 during finetuning.

Inference speed. The main reason behind filter level pruning is computation cost reduction. We evaluate inference time of pruned models in the Table 5. Pruning results in inference speed reduction, especially for the larger batch size. Pruning skip connections results in higher time reduction compared to pruning all layers. For example, only by removing 33% of FLOPs result in 1.59× speed up of *Taylor-FO-BN-22%*, while by removing 68% of FLOPs results only in 1.51× speed up of *Taylor-FO-BN-50%*.

6.3. Oracle computation details

Oracle for Table 2 is computed from the training set (as [27]) with Eq. (3). To check if correlation study is representative we compute the Oracle from the test set. Correlation between Oracles computed on training and testing sets, respectively, is 95.67%. After recomputing Table 2 using the test set, we observed little change (avg. deviation of 0.04 between raw table entries) and no reordering of the methods.

Strategy	Neurons	BN-ISTA [32]	Random	Oracle	Weight magnitude	OBD [22]	Taylor FO (Ours)	Taylor SO (Ours)
Prune A	223($\approx 70\%$)	90.9%	88.22($\pm$0.51)	91.61($\pm$0.10)	86.93($\pm$0.25)	91.57 ($\pm$0.15)	91.52 ($\pm$0.11)	91.56 ($\pm$0.14)
Prune A - train from scratch	223($\approx 70\%$)		86.28($\pm$3.59)	89.55($\pm$0.22)	80.97($\pm$4.07)	89.62($\pm$0.24)	89.56($\pm$0.19)	89.63($\pm$0.20)
Prune B	119($\approx 35\%$)	88.8%	71.49($\pm$2.35)	89.72($\pm$0.10)	62.03($\pm$1.36)	89.78 ($\pm$0.16)	89.78 ($\pm$0.18)	89.76 ($\pm$0.17)
Prune B - train from scratch	119($\approx 35\%$)		77.90($\pm$7.01)	88.25($\pm$0.28)	62.08($\pm$1.08)	88.14($\pm$0.22)	88.17($\pm$0.22)	88.29($\pm$0.19)

Table 4: Pruning results on ResNet20 for CIFAR10. Only the first layer in every residual block is pruned. Results are averaged over 10 seeds.

Pruning Method	GFLOPs	Params(10^7)	$\downarrow$ Error, %	Time, B16	Time, B256
No pruning	7.80	4.47	22.63	29.0	379.8
Taylor-FO-BN-75%	4.70	3.12	22.65	24.1	313.5
Taylor-FO-BN-55%	2.85	2.07	24.05	21.6	261.7
Taylor-FO-BN-50%	2.47	1.78	24.62	20.9	251.4
Taylor-FO-BN-40%	1.76	1.36	25.84	21.0	223.4
pruning only skip connections					
Taylor-FO-BN-52%	6.57	3.60	22.94	25.3	326.7
Taylor-FO-BN-22%	5.19	2.86	24.77	19.5	239.3

Table 5: Batch inference time of models obtained by pruning ResNet-101, time is measured on NVIDIA Tesla V100 in ms with different batch sizes.