Appendix for:
Are Straight-Through gradients and Soft-Thresholding all you need for Sparse Training?

Antoine Vanderschueren
UCLouvain, Belgium
antoine.vanderschueren@uclouvain.be

Christophe De Vleeschouwer
UCLouvain, Belgium
christophe.devleeschouwer@uclouvain.be

This appendix contains the following additional information:

1. An additional ablation study on the use of LAMP combined with ST-3
2. The full hyper-parameter list for our training
3. Our ST-3$^{(\sigma)}$ algorithm in pseudo-code format
4. The numerical values for the ImageNet results

1. Comparison to LAMP pruning

Figure 1 repeats the same experiences as Figure 5 in the main text, but with the addition of LAMP instead of l1-based pruning. The conclusions about the usefulness of our added rescaling and soft thresholding remain the same: rescaling is only useful when combined with soft-thresholding. LAMP doesn't show any additional performance increase aver traditional l1-based pruning while being more expensive FLOPS-wise. Although not shown on this graph, LAMP, combined with STE alone, has the advantage of not suffering from layer collapse due to its formal definition.

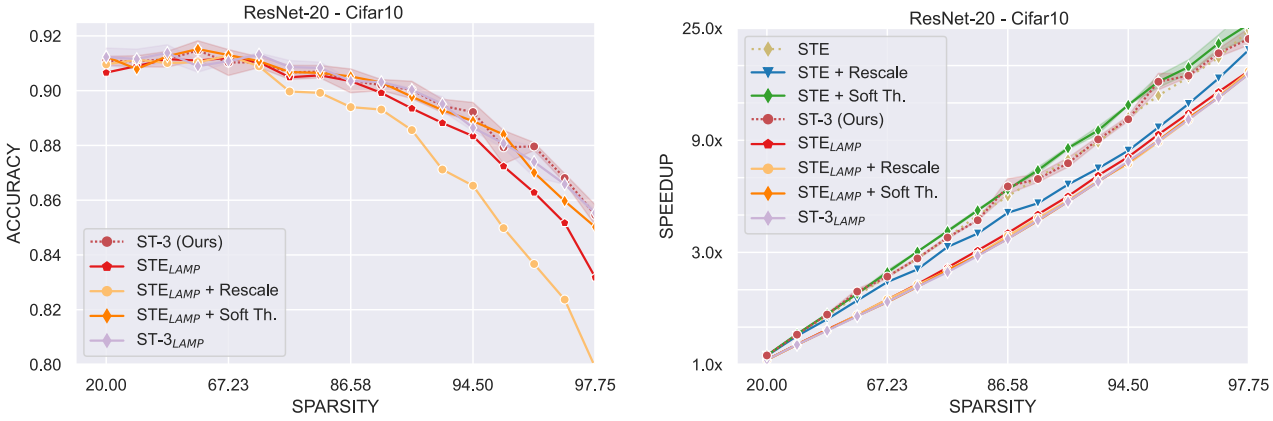

Figure 1. (left) A study dissecting the impact of straight-through estimation, soft-thresholding and weight-rescaling during sparse training with LAMP-based sparse-weight-selection. (right) FLOPS/Sparsity trade-off comparing l1 and LAMP sparse training. A log-scale is used for the x-axes.

2. Training parameters

All training configuration files and precise network structures are uploaded made open-source at `https://github.com/vanderschuea/stthree`. A quick rundown is provided below for easy comparison.

2.1. Hyper-parameters

These are the hyper-parameters used for our retraining of the following networks. All results on Cifar-10(0) are reported from a train/val/test split of sizes 45k/5k/10k. The results on ImageNet are reported on the validation set.

Model	ResNet-20/VGG-11/WideResNet-34x2	ResNet-50/MobileNetv1
epochs	160	100
optimizer	SGD	
lr	0.1	0.2
momentum	0.9	
batch-size	128	256
weight-decay	1e-4	see below
lr-decay	Step-lr@[80,120]	cosine-lr (+ warmup)
grad. clip.	3.0	

Due to the double regularization-effect of pruning and weight-decay, the weight-decay parameter is slowly decreased as follows at different sparsity ratios:

- ResNet50: 1e-4 (80-95), 1e-5 (96.5-98), 0.0 (99)
- MobileNetv1: 1e-4 (80), 1e-5 (89-95)

We do not apply decaying on the weight-decay parameter for the cifar-10(0) models as they don't benefit much from such optimizations.

2.2. Augmentations

Cifar-10(0) The standard 4x4 padding is applied before a 32x32 crop is taken. The data is normalized and also randomly horizontally flipped.

ImageNet Images are randomly cropped and resized to 224x224. The data is normalized and also randomly horizontally flipped. (Test images are resized to 256x256 and a centercrop of size 224x224 is taken for evaluation)

3. Pseudo-code

Algorithm 1 PyTorch-like Pseudo-code for ST-3 $^{\sigma}$ without and with Sparsity Distribution Suggestion

```
for i = 1 → n_epochs do
    for batch = 1 → len(train_loader) do
        sp_ratio = get_sp_ratio(i, batch)                          ▷ update sparsity ratio
        weights = torch.tensor([])                                 ▷ aggregate all weights
        for module in model.modules do
            w = module.weight
            weights.append(w.flatten() * √w.shape.sum() )
        th = weights.abs().quantile(sp_ratio)
        W_sparse = {}                                              ▷ aggregate sparsified weights
        for name, module in model.named_modules do
            mth = th / √w.shape.sum()
            with STE():                                            ▷ straight-through gradients
                w = module.weight                                  ▷ dense weights
                w_sparse = sign(w) * (||w||-mth).clip(0)           ▷ soft-th
                w_sparse *= get_scale(w, mth)                      ▷ apply rescaling
                W_sparse[name] = w_sparse
        inputs = trainloader[batch]
        loss = model(W_sparse, inputs)
        loss.backward()                                            ▷ + update all dense model weights
```

4. Table of results

Below are the exact numerical values used for the graphs presenting the results on ImageNet. The values for the other methods, are taken from their respective papers which use the same training times and data-augmentations. No 'best' values are put in bold as it would depend on whether comparison is made on Accuracy or GFLOPS as the common denominator. For a better understanding, we refer to the figures presented in the paper's main body.

Method	Accuracy [%]	Sparsity [%]	GFLOPS
GMP	67.7	74.11	163
STR	68.35	75.28	101
STR	66.52	79.07	81
RigL+ERK	68.39	75	296
ST-3$^{\sigma}$ (Ours)	68.67	80	135
ST-3 (Ours)	69.1	80	181
STR	64.83	85.8	55
GMP	61.8	89.03	82
STR	62.1	89.01	42
STR	61.51	89.62	40
RigL+ERK	62.33	90	154
ProbMask	65.19	89	63
ST-3$^{\sigma}$ (Ours)	66.15	89	86
ST-3 (Ours)	66.67	89	119
RigL+ERK	56.27	95	103
ProbMask	60.1	94.1	34
ST-3$^{\sigma}$ (Ours)	60.94	94.1	52
ST-3 (Ours)	61.19	94.1	74

Table 1. Numerical Values for the figures with results on ImageNet w/ MobileNetv1

Method	Accuracy [%]	Sparsity [%]	GFLOPS
RigL	74.6	80	940
RigL+ERK	75.1	80	1717
DNW	76	80	818
GMP	75.6	80	818
STR	76.19	79.55	766
STR	76.12	81.27	705
GraNet	76	80	1431
ST-3$^{\sigma}$ (Ours)	76.44	80	739
ST-3 (Ours)	76.95	80	1215
RigL	72.0	90	531
RigL+ERK	73.0	90	981
DNW	74	90	409
GMP	73.91	90	409
STR	74.73	87.7	402
STR	74.31	90.23	343
STR	74.01	90.55	341
GraNet	74.5	90	818
ProbMask	74.68	90	381
ST-3$^{\sigma}$ (Ours)	75.28	90	397
ST-3 (Ours)	76.03	90	764
RigL	67.5	95	327
RigL+ERK	70	95	491
DNW	68.3	95	204
GMP	70.59	95	204
STR	70.97	94.8	182
STR	70.4	95.03	159
STR	70.21	95.15	162
ProbMask	71.5	95	205
ST-3$^{\sigma}$ (Ours)	73.69	95	219
ST-3 (Ours)	74.46	95	436
RigL	64.5	96.5	164
RigL+ERK	67.2	96.5	286
STR	67.78	96.11	127
STR	67.22	96.53	117
ST-3$^{\sigma}$ (Ours)	72.62	96.5	167
ST-3 (Ours)	73.31	96.5	351
DNW	58.2	98	82
GMP	57.9	98	82
STR	62.84	97.78	80
STR	61.46	98.05	73
STR	59.76	98.22	64
ProbMask	66.83	98	105
ST-3$^{\sigma}$ (Ours)	69.75	98	116
ST-3 (Ours)	70.46	98	220
GMP	44.78	99	41
STR	54.79	98.79	54
STR	51.82	98.98	47
STR	50.35	99.1	44
ProbMask	61.07	99	55
ST-3$^{\sigma}$ (Ours)	63.25	99	69
ST-3 (Ours)	63.88	99	120

Table 2. Numerical Values for the figures with results on ImageNet w/ ResNet-50