Sparse Model Soups: A Recipe for Improved Pruning via Model Averaging

Max Zimmer, Christoph Spiegel, Sebastian Pokutta

Cooperation: TU Berlin, ZIB

Funding: DFG Cluster of Excellence Math+, German Federal Ministry of Education and Research

Berlin Mathematics Research Center MATH+

ZIB ZUSE INSTITUTE BERLIN

Introduction: Model Soups, Pruning and IMP

Let θ denote the parameters of a Neural Network (NN).

- **Parameter Averaging or Model Soups:** Average the parameters of multiple models $\theta_i, 1 \leq i \leq m$, building a new model $\bar{\theta} = \sum_{1 \leq i \leq m} \lambda_i \theta_i$
  - Improves the generalization performance by combining multiple models.
  - Does not increase inference time: constant in the number of models m.
  - **Difficulty:** Models θ_i must reside in a linearly connected loss basin. Even averaging models trained with varying seeds but identical initialization degrades performance compared to individual models (Neyshabur et al., 2020).
- **Pruning:** Selectively removes parameters from NN θ by setting them to zero, inducing sparsity in the corresponding tensors.
  - Drastically reduces the parameter count, maintaining similar performance as the dense model.
  - Reduces memory requirements and computational complexity.
- **A classical algorithm:** Iterative Magnitude Pruning (IMP, Han et al., 2015)
  - Prunes weights based on their magnitude.
  - Retrains the model to restore performance after pruning.
  - Iterates these *prune-retrain cycles* until desired compression-performance tradeoff is reached.

Combining the benefits of both Model Averaging and Sparsity

Can we get the benefits of both averaging and sparsity? → Need to resolve two problems.
Problem 1: Averaging two sparse models may destroy the sparsity pattern (cf. Figure 1)!
Problem 2: It is unclear how we can obtain (sparse) models that are averageable in the first place!

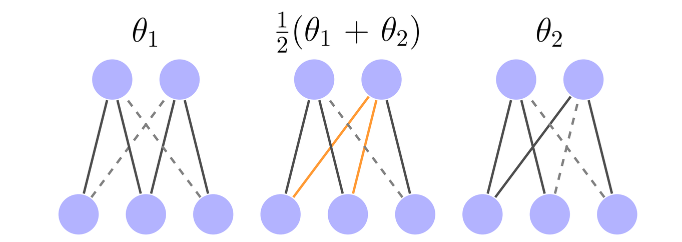

Figure 1: Creating the average (middle) of two networks with different sparsity patterns (left, right) may lower overall sparsity, changing pruned weights (dashed) to non-zero (solid), with reactivated weights highlighted in orange.

Observations

- Pruning a pretrained model and retraining multiple copies with varied hyperparameters (e.g., batch ordering, weight decay, retraining duration and length) yields models suitable for averaging.
- Averaged models exhibit superior generalization and out-of-distribution (OOD) performance compared to individual models.
- These models maintain the sparsity pattern of their pruned parent in their parameter average.

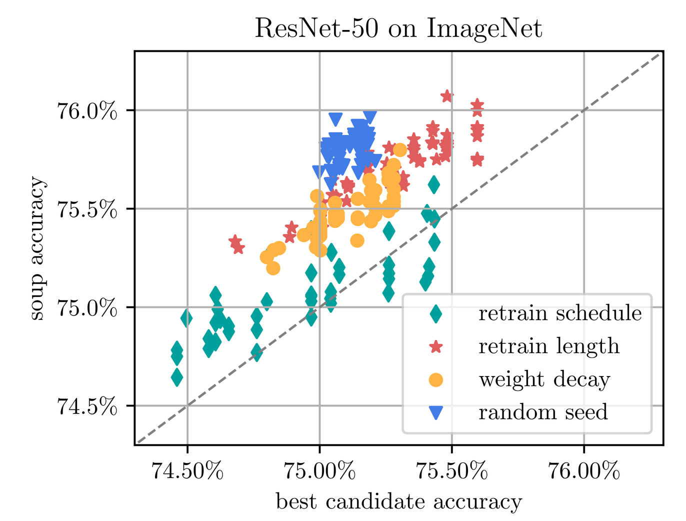

Figure 2: Accuracy of average of two models vs. the maximal individual accuracy. All models are pruned to 70% sparsity (One Shot) and retrained, varying the indicated hyperparameters.

Instability to randomness and recovering it

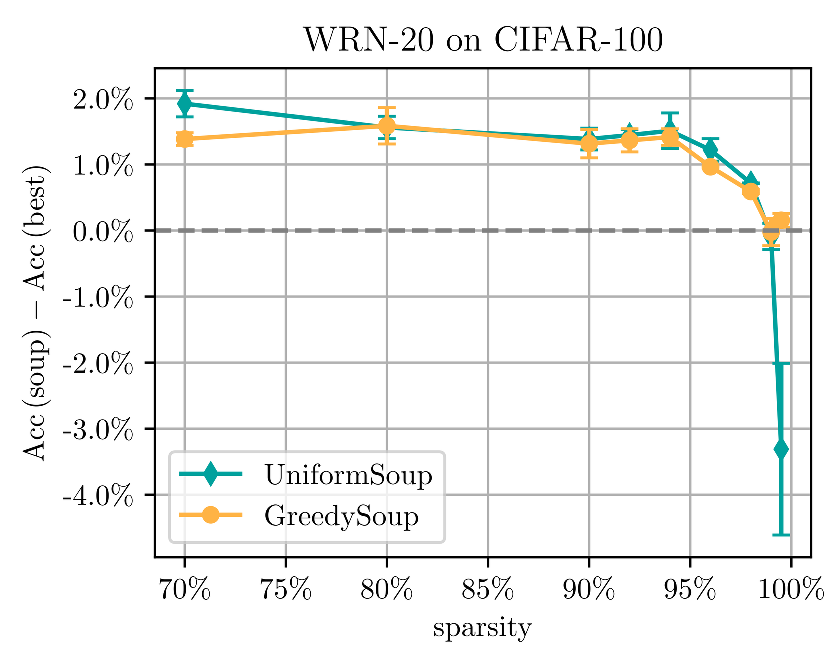

Figure 3: WideResNet-20 on CIFAR-100: Accuracy difference between the soup ($m = 5$) and best averaging candidate after One Shot pruning and retraining for varying sparsity levels.

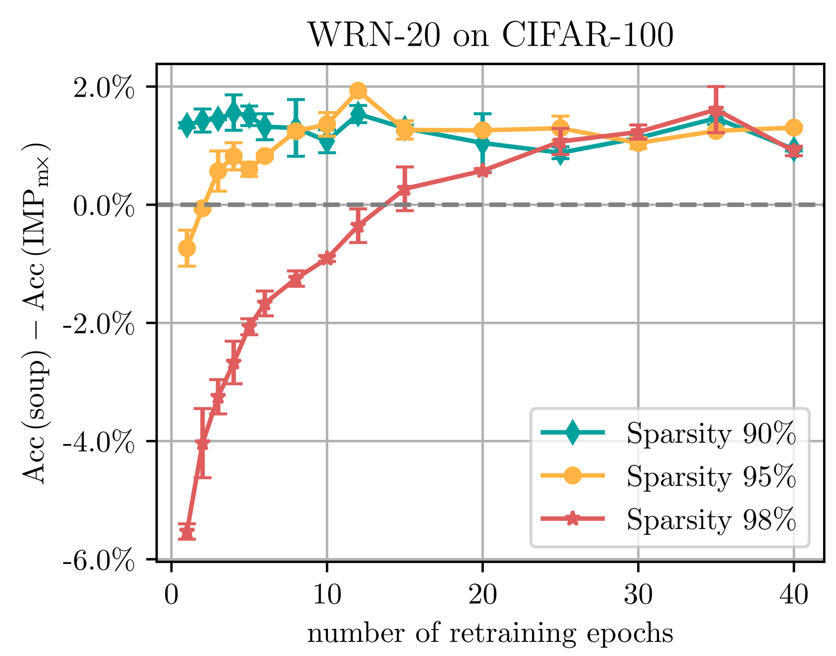

Figure 4: WideResNet-20 on CIFAR-100: Accuracy difference between the soup ($m = 3$) and IMP$_{3\times}$ retrained three times as long as indicated on the x-axis, using One Shot pruning.

The Recipe: Sparse Model Soups

A single phase of IMP yields models suitable for averaging without destroying the sparsity pattern.
→ **Another problem: We cannot guarantee identical sparse connectivity after multiple prune-retrain cycles.**
→ **Average models after each phase and begin next one with averaged model.**

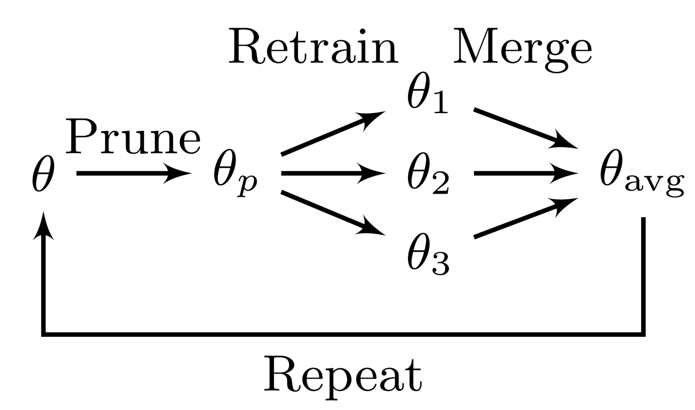

Figure 5: Sketch for a single phase, $m = 3$.

Algorithm 1 Sparse Model Soups
Require: Pretrained model θ
Ensure: Sparse model soup
1: **for** each prune-retrain cycle **do**
2:   Prune θ
3:   **for** $i \leftarrow 1$ to m **do** ▷ Fully parallelizable
4:     $\theta_i \leftarrow \theta$
5:     Retrain θ_i with specific hyperparameters
6:   **end for**
7:   $\theta \leftarrow \text{Merge}(\theta_1, \ldots, \theta_m)$
8: **end for**
9: **return** θ

Comparing SMS against suitable baselines

In each phase, SMS trains m models in parallel for k epochs each.

Suitable baselines:

- IMP: Regular IMP without averaging, i.e., $m = 1$.
- IMP$_{m\times}$: Extended IMP, where the IMP retraining duration is extended by a factor of m, resulting in $k \cdot m$ retraining epochs per prune-retrain cycle as as many overall epochs as SMS.
- IMP-RePrune: Regular IMP executed m times, averaging performed after the final phase, followed by repruning to address sparsity reduction after averaging.
- Best candidate: Best accuracy among all averaging candidates.
- Mean candidate: Mean accuracy of the averaging candidates.

Table 1: WideResNet-20 on CIFAR-100 and ResNet-50 on ImageNet: Test accuracy comparison for target sparsities 98% (top) and 90% (bottom) given three prune-retrain cycles. The best value is highlighted in bold.

CIFAR-100 (98%)

	Sparsity 72.8% (Phase 1)			Sparsity 92.6% (Phase 2)			Sparsity 98.0% (Phase 3)		
Accuracy of	$m=3$	$m=5$	$m=10$	$m=3$	$m=5$	$m=10$	$m=3$	$m=5$	$m=10$
SMS	**76.50** ±0.16	**76.59** ±0.13	**76.75** ±0.28	**75.55** ±0.60	**76.19** ±0.37	**76.21** ±0.43	**72.67** ±0.29	**72.90** ±0.64	**73.05** ±0.45
best candidate	75.58 ±0.19	75.71 ±0.08	75.96 ±0.13	74.51 ±0.47	75.01 ±0.74	75.00 ±0.34	71.77 ±0.04	71.77 ±0.37	72.21 ±0.02
mean candidate	75.37 ±0.12	75.58 ±0.03	75.55 ±0.26	74.32 ±0.40	74.71 ±0.48	74.70 ±0.42	71.41 ±0.09	71.61 ±0.40	71.66 ±0.19
IMP$_{m\times}$	75.85 ±0.26	76.05 ±0.00	75.76 ±0.24	74.09 ±0.24	74.19 ±0.44	74.74 ±0.06	70.92 ±0.07	70.31 ±0.52	71.85 ±0.15
IMP-RePrune		— N/A —			— N/A —		68.19 ±0.44	65.53 ±0.06	63.62 ±0.90
IMP		— 75.54 ±0.41 —			— 74.09 ±0.13 —			— 70.74 ±0.08 —	

ImageNet (90%)

	Sparsity 53.6% (Phase 1)			Sparsity 78.5% (Phase 2)			Sparsity 90.0% (Phase 3)		
Accuracy of	$m=3$	$m=5$	$m=10$	$m=3$	$m=5$	$m=10$	$m=3$	$m=5$	$m=10$
SMS	**76.74** ±0.20	**76.89** ±0.18	**77.01** ±0.05	**76.04** ±0.21	**76.30** ±0.13	**76.49** ±0.12	**74.53** ±0.04	**74.82** ±0.08	**74.96** ±0.16
best candidate	76.07 ±0.01	76.07 ±0.21	76.14 ±0.18	75.48 ±0.16	75.46 ±0.11	75.70 ±0.03	74.00 ±0.03	74.19 ±0.08	74.25 ±0.13
mean candidate	75.99 ±0.04	75.95 ±0.14	75.96 ±0.08	75.40 ±0.11	75.42 ±0.10	75.55 ±0.05	73.94 ±0.03	74.11 ±0.11	74.13 ±0.12
IMP$_{m\times}$	76.25 ±0.08	76.21 ±0.14	76.46 ±0.04	75.74 ±0.03	75.87 ±0.11	75.93 ±0.03	74.34 ±0.09	74.56 ±0.24	74.50 ±0.09
IMP-RePrune		— N/A —			— N/A —		72.97 ±0.25	72.58 ±0.01	72.08 ±0.12
IMP		— 75.97 ±0.16 —			— 75.19 ±0.14 —			— 73.59 ±0.04 —	