Learning Pruning-Friendly Networks via Frank-Wolfe: One-Shot, Any-Sparsity, And No Retraining

[ICLR 22] Miao Lu[1*], Xiaolong Luo[1*], Tianlong Chen[2], Wuyang Chen[2], Dong Liu[1], Zhangyang Wang[2]
[1]University of Science and Technology of China, [2]University of Texas at Austin,

Agenda

- Backgrounds
- Contributions
- Motivation & Methodology
- Experimental Results

Backgrounds

- **Pruning** is a commonly used way of DNN compression,
  - e.g., for deploying your model across platforms with different hardware performances.

- Usually, modern DNN pruning techniques require **retraining** or **fine-tuning** to obtain the compressed network.
  - huge computational cost
  - sensitive to retraining parameters

- Question：whether we can design an efficient pruning method that does not need retrain the neural network.

Our Contributions

- We propose **SFW(stochastic Frank-Wolfe)-pruning**, a **one-shot** unstructured pruning algorithm, which can guarantee consistent and competitive model performance under varying pruning ratios **without retraining**.

- We customize a meta-learning-based **initialization scheme** for SFW-based DNN training, leading to more consistent and competitive performance under varying pruning ratios.

- Empirical demonstrations.

Motivations & Methodology

- Idea: cast the DNN training as an explicit **pruning-aware** process, which actively enhances important weights and pushes less important weights smaller.

- To this end, we add an auxiliary **K-sparse polytope constraint** in training the training objective:

$$\min_{\boldsymbol{\theta} \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{\theta}; \mathbf{x}_i), y_i)$$

$$\mathcal{C}(K, \tau) = \mathbf{Span}_{[0,1]}(\{\boldsymbol{v} \in \mathbb{R}^p : \|\boldsymbol{v}\|_0 = K,\ (\boldsymbol{v})_i \in \{0, \tau\}\})$$

- solve the constrained OPT via Stochastic Frank-Wolfe (**SFW**).

Motivations & Methodology

- **Why K-sparse polytope constraint?** The optimization process of SFW for K-sparse polytope constraint is ideal for our goal!

- Update rule: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t(\boldsymbol{v}_t - \boldsymbol{\theta}_t) = \alpha_t \boldsymbol{v}_t + (1-\alpha_t)\boldsymbol{\theta}_t$. Here $\boldsymbol{v}_t$ solves a linear minimization oracle, $\arg\min_{\boldsymbol{v}\in\mathcal{C}} \langle \widehat{\nabla}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_t), \boldsymbol{v} \rangle$, which has closed form solution:

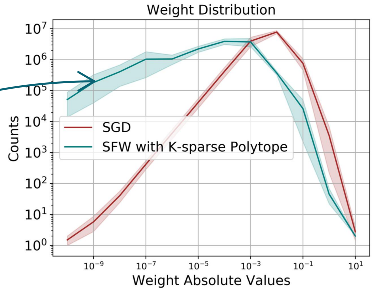
$$(\boldsymbol{v})_i = \begin{cases} -\tau \cdot \operatorname{sign}((\boldsymbol{m})_i) & \text{if } (\boldsymbol{m})_i \text{ is in the largest } K \text{ coordinates of } \boldsymbol{m}, \\ 0 & \text{otherwise,} \end{cases}$$

Motivations & Methodology

- Each step is equivalent to K “votes” on the weights to select important weights. Important weights are enhanced and less important weights are averaged with 0.

- Resulting in more smaller weights (but not exactly zero), and less large ones. This yields competitive test accuracies across the spectrum of pruning ratios, even without retraining.

- More smaller weights
- The amounts of weights, at different magnitude levels, change more “smoothly and “continually”, no “sudden jumps”



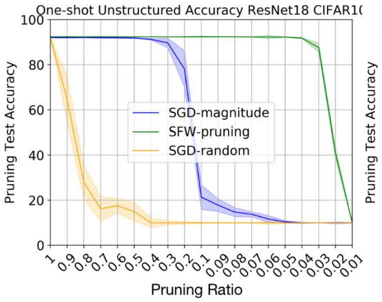
Motivations & Methodology

- Algorithm: **SFW-Pruning,**
  - **One-shot SFW-training** + **Magnitude Unstructured Pruning**
  - Achieving consistent and competitive model performance under varying pruning ratios **without retraining.**

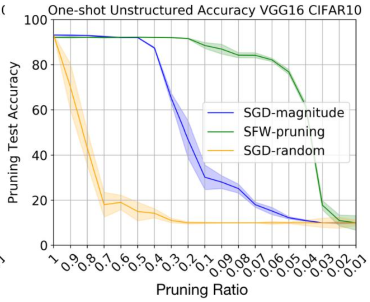
- Algorithm: **SFW-INIT,**
  - An **initialization scheme** tailored for SFW-training
  - Learning-based: learn the best initialization that allows the maximum loss reduction in the first SFW step.
  - Further boosting pruning test performance across different pruning ratios.

Experimental Results

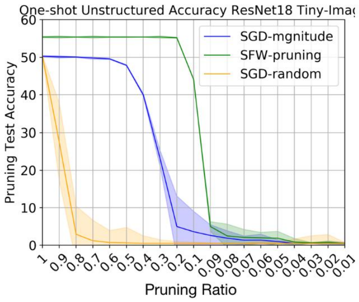
- Unstructured pruning NN with different sparsity ratios without retraining
  - SFW-pruning significantly outperforms magnitude-based and random pruning by SGD, across different datasets and architectures.
  - Over a wide range of sparsity ratios, SFW can keep pruning while maintaining a highly competitive performance.



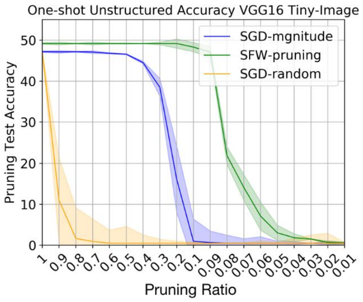
(a) CIFAR-10 ResNet-18



(b) CIFAR-10 VGG-16



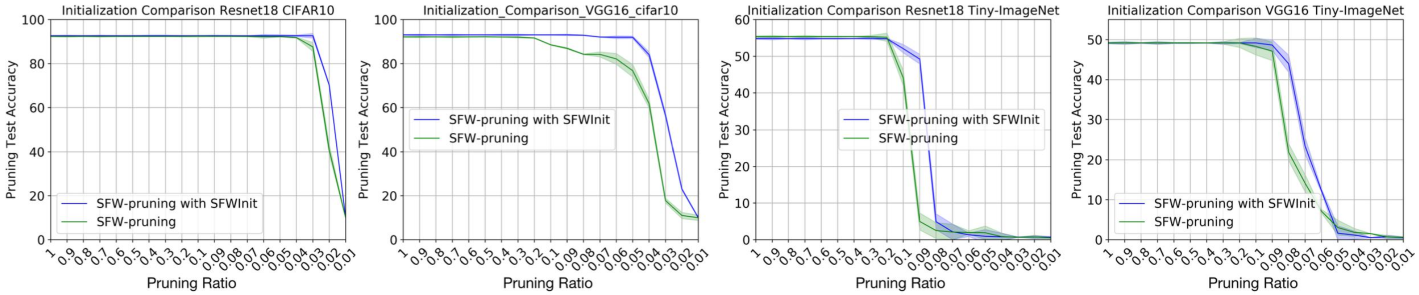
(c) Tiny-Image ResNet-18



(d) Tiny-Image VGG-16

Experimental Results

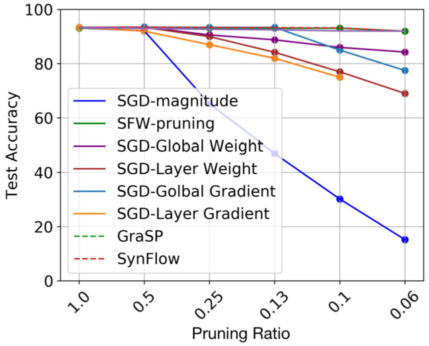
- SFW-pruning with and without SFWInit
  - SFW + SFWInit consistently achieves higher accuracies compared with SFW across different datasets and architectures.



(a) CIFAR-10 ResNet-18 (b) CIFAR-10 VGG-16 (c) Tiny-Image ResNet-18 (d) Tiny-Image VGG-16

Experimental Results

- Comparison to SOTA methods:
  - One-shot no retraining methods
  - pruning at initialization methods
  - pruning-during-training methods
  - iterative pruning methods
  - group sparsity methods



Pruning Ratios	**50%**	**70%**	**80%**	**90%**	**95%**
SFW-Pruning (ours)	93.10	93.10	93.10	93.10	92.00
One-Cycle Pruning (Hubens et al., 2021)	-	-	90.87	90.72	90.67
Early Bird (You et al., 2019)	93.2	92.8	-	-	-
OTO (Chen et al., 2021)	90.35	90.35	90.35	90.35	90.35
DPF (Lin et al., 2020)	-	-	-	-	93.87
Group MDP (Deleu & Bengio, 2021)	-	-	-	89.38	-

Q&A