Global Sparse **Momentum** SGD for **Pruning** Very Deep Neural Networks

Xiaohan Ding (dxh17@mails.tsinghua.edu.cn) Guiguang Ding Xiangxin Zhou
Yuchen Guo Jungong Han Ji Liu
Tsinghua University, China The University of Warwick, UK Kwai AI platform

The take-home message

- Deep neural networks can still converge if you **only update very few params with their gradients**, and zero out most of them via ordinary **weight decay**. We achieve this by **directly transforming the gradient flow**, rather than changing the loss function to affect the gradients indirectly.
- Doing so makes most params infinitely close to zero.
- **Accelerated by momentum** SGD, such a process can be used for DNN pruning (unstructured or structured).
- Pruning in this way
  - requires the **final global compression ratio** as hyper-parameter
  - enables end-to-end training and lossless pruning
  - achieves high compression ratio
  - automatically discovers the **appropriate sparsity ratio for each layer**, given the final global compression ratio as requirement
  - finds more **powerful winning lottery tickets**.

Background

- Pruning methods seek to introduce sparsity into DNNs. Unstructured pruning (a.k.a. connection pruning) can achieve a high compression ratio (low percentage of non-zero params) but cannot directly reflect in acceleration on off-the-shelf platforms. Structured pruning (e.g., neuron-, kernel- or filter- level) is hardware-friendly but cannot achieve a high compression ratio. This paper focuses on connection pruning, but the method can be easily generalized to structured pruning.
- Momentum SGD: let k be the num of iterations, L be the loss function, w be a single parameter, α be the learning rate, β be the momentum coefficient, η be the ordinary weight decay coefficient (e.g., 0.0001 for ResNets), the update rule is

$$z^{(k+1)} \leftarrow \beta z^{(k)} + \eta w^{(k)} + \frac{\partial L}{\partial w^{(k)}} \,,$$

$$w^{(k+1)} \leftarrow w^{(k)} - \alpha z^{(k+1)} \,.$$

Global Sparse Momentum SGD (GSM)

- **In a nutshell**: important params are updated using gradients and weight decay, and unimportant params are reduced using weight decay only.
- The update rule:

$$\boldsymbol{Z}^{(k+1)} \leftarrow \beta \boldsymbol{Z}^{(k)} + \eta \boldsymbol{W}^{(k)} + \boldsymbol{B}^{(k)} \circ \frac{\partial L(x, y, \boldsymbol{\Theta})}{\partial \boldsymbol{W}^{(k)}} \,,$$

$$\boldsymbol{W}^{(k+1)} \leftarrow \boldsymbol{W}^{(k)} - \alpha \boldsymbol{Z}^{(k+1)} \,,$$

where **Θ** is the collection of all params which parameterizes the whole model, **W** is the kernel matrix of a fully-connected or conv layer and **Z** is its accumulated momentum, L(x, y, **Θ**) is the loss on the current inputs x and y. Same as ordinary momentum SGD, α is the learning rate, β is the momentum coefficient, and η is the weight decay coefficient

- **B** is the mask which decides which params are updated using the gradients derived from the loss, and which params are only reduced via weight decay. And ○ is the element-wise multiplication

$$B^{(k)}_{m,n} = \begin{cases} 1 & \text{if } T(x, y, W^{(k)}_{m,n}) \geq \text{the } Q\text{-th greatest value in } T(x, y, \boldsymbol{\Theta}^{(k)}) \,, \\ 0 & \text{otherwise} \,. \end{cases}$$

- Here Q is the desired number of non-zero params in the whole model (so we call it "global" sparse momentum SGD). And T is the criterion for the importance of parameters. We simply use first-order Taylor.

$$T(x, y, w) = \left| \frac{\partial L(x, y, \boldsymbol{\Theta})}{\partial w} w \right| \,.$$

Motivation

- No regularization terms! If we use some regularization terms to zero out some params, the strength of such regularization does not directly reflect the final sparsity, which is our primary concern. And the params cannot become infinitely close to zero, but can only be reduced to some extent.
- No non-differentiable optimization! We want end-to-end training. Some methods explicitly model the trade-off between final sparsity and accuracy as an optimization problem. As the sparsity (L0 norm) is not differentiable, the problem cannot be solved via end-to-end training.

Pruning experiments:

Table 1: Pruning results on MNIST.

Model	Result	Base Top1	Pruned Top1	Origin / Remain Params	Compress Ratio	Non-zero Ratio
LeNet-300	Han et al. [21]	98.36	98.41	267K / 22K	12.1×	8.23%
LeNet-300	L-OBS [13]	98.24	98.18	267K / 18.6K	14.2×	7%
LeNet-300	Zhang et al. [56]	98.4	98.4	267K / 11.6K	23.0×	4.34%
LeNet-300	DNS [18]	97.72	98.01	267K / 4.8K	55.6×	1.79%
LeNet-300	**GSM**	**98.19**	**98.18**	**267K / 4.4K**	**60.0×**	**1.66%**
LeNet-5	Han et al. [21]	99.20	99.23	431K / 36K	11.9×	8.35%
LeNet-5	L-OBS [13]	98.73	98.73	431K / 3.0K	14.1×	7%
LeNet-5	Srinivas et al. [47]	99.20	99.19	431K / 22K	19.5×	5.10%
LeNet-5	Zhang et al. [56]	99.2	99.2	431K / 6.05K	71.2×	1.40%
LeNet-5	DNS [18]	99.09	99.09	431K / 4.0K	107.7×	0.92%
LeNet-5	**GSM**	**99.21**	**99.22**	**431K / 3.4K**	**125.0×**	**0.80%**
LeNet-5	**GSM**	**99.21**	**99.06**	**431K / 1.4K**	**300.0×**	**0.33%**

Table 2: Pruning results on CIFAR-10.

Model	Result	Base Top1	Pruned Top1	Origin / Remain Params	Compress Ratio	Non-zero Ratio
ResNet-56	GSM	94.05	94.10	852K / 127K	6.6×	15.0%
ResNet-56	GSM	94.05	93.80	852K / 85K	10.0×	10.0%
DenseNet-40	GSM	93.86	94.07	1002K / 150K	6.6×	15.0%
DenseNet-40	GSM	93.86	94.02	1002K / 125K	8.0×	12.5%
DenseNet-40	GSM	93.86	93.90	1002K / 100K	10.0×	10.0%

Table 3: Pruning results on ImageNet.

Model	Result	Base Top1 / Top5	Pruned Top1 / Top5	Origin / Remain Params	Compress Ratio	Non-zero Ratio
ResNet-50	L-OBS[13]	- / ≈ 92	- / ≈ 92	25.5M / 16.5M	1.5×	65%
ResNet-50	L-OBS[13]	- / ≈ 92	- / ≈ 85	25.5M / 11.4M	2.2×	45%
ResNet-50	**GSM**	**75.72 / 92.75**	**75.33 / 92.47**	**25.5M / 6.3M**	**4.0×**	**25%**
ResNet-50	**GSM**	**75.72 / 92.75**	**74.30 / 91.98**	**25.5M / 5.1M**	**5.0×**	**20%**

Lottery ticket experiments:

Table 4: Eventual Top1 accuracy of the winning tickets training (step 5).

Model	Compression ratio	Magnitude tickets	GSM tickets
LeNet-300	60×	97.39	98.22
LeNet-5	125×	97.60	99.04
LeNet-5	300×	11.35	98.88

- In the lottery ticket paper [Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. ICLR 2019 best paper.], the authors found the winning tickets by simply pruning the params with smaller magnitude in the trained model.
- Our method GSM can find a better set of winner tickets, as training the GSM-discovered tickets yields higher eventual accuracy than those found by magnitude-based pruning.