Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration

He et al. CVPR 2019

Haein Choe

June 14, 2025

Pseudo Lab - On-Device AI: ON THE Air

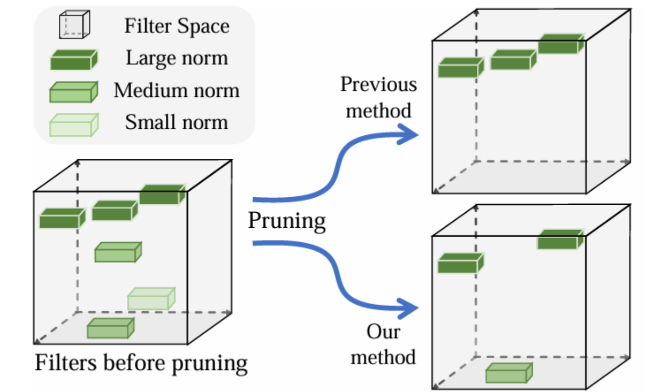
Table of contents

Introduction

Introduction

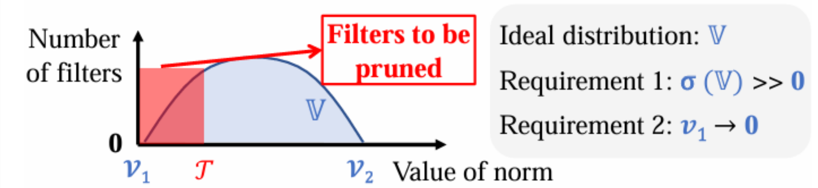
- **filter pruning** directly discards the whole selected filters and leaves a model with regular structures.
- **filter pruning** is more preferred for accelerating the networks and decreasing the model size.
- However, recent practice performs filter pruning by following the **“smaller-norm-less-important”** criterion which relies on two prerequisites that are not always true.
- To solve this problem, the authors propose a novel filter pruning method to compress the CNN model regardless of those two requirements namely **Filter Pruning via Geometric Median (FPGM)**.

Smaller-norm-less-important criterion



(a) Criterion for filter pruning

- in the **norm-based criterion**, only the filters with the largest norm are kept.
- the proposed method prunes the filters with **redundant information** in the network.



(b) Requirements for norm-based criterion

- Req 1 makes the searching space for $\mathcal{T}$ wide enough so that separating those filters needed to be pruned would be an easy task.
- Req 2 means that filters with smaller norms are expected to make **small contributions**.
- The **FPGM** chooses filters with **the most replaceable contribution** rather than **relatively less contribution**.

Methodology

Preliminaries

Notation

- N_i , N_{i+1} : the number of input channels and the output channels for the i_{th} convolution layer.
- $\mathcal{F}_{i,j} \in \mathbb{R}^{N_i \times K \times K}$: j_{th} filters of the i_{th} layer.
- K : kernel size of the network.
- L : the number of layers.
- $\mathbf{W}^{(i)} \in \mathbb{R}^{N_{i+1} \times N_i \times K \times K}, 1 \leq i \leq L$: model parameter of the i_{th} layer.

Filter Pruning via Geometric Median

Geometric Median

Given a set of n points $a^{(1)}, \cdots, a^{(n)}$ with each $a^{(i)} \in \mathbb{R}^d$, The geometric median is a point x^* such that minimizes the sum of Euclidean distances to them. i.e,

$$x^* = \underset{x \in \mathbb{R}^d}{\arg\min} f(x) \quad \text{where } f(x) \overset{\text{def}}{=} \sum_{i \in \{1, \cdots, n\}} \|x - a^{(i)}\|_2 \tag{1}$$

- by (1), We use the **geometric median** to get the **common information** of all the filters within the single i_{th} layer. The geometric median x^{GM} becomes

$$x^{GM} = \underset{x \in \mathbb{R}^{N_i \times K \times K}}{\arg\min} \sum_{j' \in \{1, \cdots, N_{i+1}\}} \|x - \mathcal{F}_{i,j'}\|_2 \tag{2}$$

- The filter(s) nearest to the geometric median x^{GM} $\mathcal{F}_{i,j^*}$ is

$$\mathcal{F}_{i,j^*} = \underset{\mathcal{F}_{i,j'}}{\arg\min} \|\mathcal{F}_{i,j'} - x^{GM}\|_2, \quad s.t \quad j' \in \{1, \cdots, N_{i+1}\} \tag{3}$$

- As geometric median is a non-trivial problem in computational geometry, the authors do not compute it directly, Instead, they find the filter which **minimizes the summation of the distance with other filters**:

$$\begin{aligned} \mathcal{F}_{i,x^*} = \arg\min_{x} \sum_{j' \in \{1,\cdots,N_{i+1}\}} \|x - \mathcal{F}_{i,j'}\|_2 \overset{\text{def}}{=} \arg\min_{x} g(x) \\ s.t \quad x \in \{\mathcal{F}_{i,1}, \cdots, \mathcal{F}_{i,N_{i+1}}\} \end{aligned} \tag{4}$$

- Note that even if $\mathcal{F}_{i,x^*}$ is not included in the calculation of the geometric median in (4), we obtain the same result.

$$\begin{aligned} g(x) &= \sum_{j' \in [1,N_{i+1}]} \|x - \mathcal{F}_{i,j'}\|_2 \quad s.t \quad x \in \{\mathcal{F}_{i,1}, \cdots, \mathcal{F}_{i,N_{i+1}}\} && (5) \\ &= \sum_{j' \in [1,N_{i+1}], \mathcal{F}_{i,j'} \neq x} \|x - \mathcal{F}_{i,j'}\|_2 + \left[\cancelto{0}{\|x - \mathcal{F}_{i,j'}\|_2} \right]_{\mathcal{F}_{i,j'} = x} && (6) \\ &= g'(x) && (7) \end{aligned}$$

FPGM Algorithm

Algorithm 1 Algorithm Description of FPGM

Input: training data: $\mathbf{X}$

1: **Given:** pruning rate P_i
2: **Initialize:** model parameter $\mathbf{W} = \{\mathbf{W}^{(i)}, 0 \leq i \leq L\}$
3: **for** $epoch = 1; epoch \leq epoch_{max}; epoch++$ **do**
4: Update the model parameter $\mathbf{W}$ based on $\mathbf{X}$
5: **for** $i = 1; i \leq L; i++$ **do**
6: Find $N_{i+1}P_i$ filters that satisfy Equation 4
7: Zeroize selected filters
8: **end for**
9: **end for**
10: Obtain the compact model $\mathbf{W}^*$ from $\mathbf{W}$

Output: The compact model and its parameters $\mathbf{W}^*$

Using FPGM algorithm, A compact model $\{\mathbf{W}^{*(i)} \in \mathbb{R}^{N_{i+1}(1-P_i)\times N_i(1-P_{i-1})\times K\times K}\}$ is obtained.

Experiments

Experimental Settings

- Dataset
  - CIFAR-10 : 50,000 training images, 10,000 testing images , 10 classes.
  - ILSVRC-2012 : 1.28M training images, 50k validation images, 1,000 classes.
- Models :
  - VGGNet(single-branch network)
  - ResNet(multiple-branch network)
- Pretrained model and scratch model are compared.
- The Pruning rate at each layer is the same. ($P_i = P \quad \forall i \in \{1, \cdots, L\}$).
- The authors use **a mixture of FPGM and previous norm-based method** to show that FPGM could serve as a supplement to pervious methods.

Single-Branch Network Pruning

Model \ Acc (%)	Baseline	Pruned w.o. FT	FT 40 epochs	FT 160 epochs
PFEC [21]	93.58 (±0.03)	77.45 (±0.03)	93.22 (±0.03)	93.28 (±0.07)
Ours	93.58 (±0.03)	**80.38** (±0.03)	**93.24** (±0.01)	**94.00** (±0.13)

Table 3. Pruning pre-trained VGGNet on CIFAR-10. "w.o." means "without" and "FT" means "fine-tuning" the pruned model.

Model	SA	Baseline	Pruned From Scratch	FLOPs↓(%)
PFEC [21]	Y	93.58 (±0.03)	93.31 (±0.02)	34.2
Ours	Y	93.58 (±0.03)	**93.54** (±0.08)	34.2
Ours	N	93.58 (±0.03)	93.23 (±0.13)	**35.9**

Table 4. Pruning scratch VGGNet on CIFAR-10. "SA" means "sensitivity analysis". Without sensitivity analysis, FPGM can still achieve comparable performances comparing to [21]; after introducing sensitivity analysis, FPGM can surpass [21].

Multiple-Branch Network Pruning - CIFAR-10

Depth	Method	Fine-tune?	Baseline acc. (%)	Accelerated acc. (%)	Acc. ↓ (%)	FLOPs	FLOPs ↓(%)
20	SFP [15]	✗	**92.20** (±0.18)	90.83 (±0.31)	1.37	2.43E7	42.2
	Ours (FPGM-only 30%)	✗	**92.20** (±0.18)	91.09 (±0.10)	1.11	2.43E7	42.2
	Ours (FPGM-only 40%)	✗	**92.20** (±0.18)	90.44 (±0.20)	1.76	**1.87E7**	**54.0**
	Ours (FPGM-mix 40%)	✗	**92.20** (±0.18)	**90.62** (±0.17)	**1.58**	**1.87E7**	**54.0**
32	MIL [5]	✗	92.33	90.74	1.59	4.70E7	31.2
	SFP [15]	✗	**92.63** (±0.70)	92.08 (±0.08)	0.55	4.03E7	41.5
	Ours (FPGM-only 30%)	✗	**92.63** (±0.70)	92.31 (±0.30)	0.32	4.03E7	41.5
	Ours (FPGM-only 40%)	✗	**92.63** (±0.70)	**91.93** (±0.03)	**0.70**	**3.23E7**	**53.2**
	Ours (FPGM-mix 40%)	✗	**92.63** (±0.70)	91.91 (±0.21)	0.72	**3.23E7**	**53.2**
56	PFEC [21]	✗	93.04	91.31	1.75	9.09E7	27.6
	CP [16]	✗	92.80	90.90	1.90	–	50.0
	SFP [15]	✗	**93.59** (±0.58)	92.26 (±0.31)	1.33	**5.94E7**	**52.6**
	Ours (FPGM-only 40%)	✗	**93.59** (±0.58)	**92.93** (±0.49)	**0.66**	**5.94E7**	**52.6**
	Ours (FPGM-mix 40%)	✗	**93.59** (±0.58)	92.89 (±0.32)	0.70	**5.94E7**	**52.6**
	PFEC [21]	✓	93.04	93.06	-0.02	9.09E7	27.6
	CP [16]	✓	92.80	91.80	1.00	–	50.0
	Ours (FPGM-only 40%)	✓	**93.59** (±0.58)	**93.49** (±0.13)	**0.10**	**5.94E7**	**52.6**
	Ours (FPGM-mix 40%)	✓	**93.59** (±0.58)	93.26 (±0.03)	0.33	**5.94E7**	**52.6**
110	MIL [5]	✗	93.63	93.44	0.19	-	34.2
	PFEC [21]	✗	93.53	92.94	0.61	1.55E8	38.6
	SFP [15]	✗	**93.68** (±0.32)	93.38 (±0.30)	0.30	1.50E8	40.8
	Ours (FPGM-only 40%)	✗	**93.68** (±0.32)	93.73 (±0.23)	-0.05	**1.21E8**	**52.3**
	Ours (FPGM-mix 40%)	✗	**93.68** (±0.32)	**93.85** (±0.11)	**-0.17**	**1.21E8**	**52.3**
	PFEC [21]	✓	93.53	93.30	0.20	1.55E8	38.6
	NISP [39]	✓	–	–	0.18	–	43.8
	Ours (FPGM-only 40%)	✓	**93.68** (±0.32)	**93.74** (±0.10)	**-0.16**	**1.21E8**	**52.3**

Multiple-Branch Network Pruning - ILSVRC-2012

Depth	Method	Fine-tune?	Baseline top-1 acc.(%)	Accelerated top-1 acc.(%)	Baseline top-5 acc.(%)	Accelerated top-5 acc.(%)	Top-1 acc. ↓(%)	Top-5 acc. ↓(%)	FLOPs↓(%)
18	MIL [5]	✗	69.98	66.33	89.24	86.94	3.65	2.30	34.6
	SFP [15]	✗	**70.28**	67.10	**89.63**	87.78	3.18	1.85	**41.8**
	Ours (FPGM-only 30%)	✗	**70.28**	67.78	**89.63**	88.01	2.50	1.62	**41.8**
	Ours (FPGM-mix 30%)	✗	**70.28**	**67.81**	**89.63**	**88.11**	**2.47**	**1.52**	**41.8**
	Ours (FPGM-only 30%)	✓	**70.28**	68.34	**89.63**	**88.53**	1.94	**1.10**	**41.8**
	Ours (FPGM-mix 30%)	✓	**70.28**	**68.41**	**89.63**	88.48	**1.87**	1.15	**41.8**
34	SFP [15]	✗	**73.92**	71.83	**91.62**	90.33	2.09	1.29	**41.1**
	Ours (FPGM-only 30%)	✗	**73.92**	71.79	**91.62**	**90.70**	2.13	**0.92**	**41.1**
	Ours (FPGM-mix 30%)	✗	**73.92**	**72.11**	**91.62**	90.69	**1.81**	0.93	**41.1**
	PFEC [21]	✓	73.23	72.17	–	–	**1.06**	–	24.2
	Ours (FPGM-only 30%)	✓	**73.92**	72.54	**91.62**	**91.13**	1.38	**0.49**	**41.1**
	Ours (FPGM-mix 30%)	✓	**73.92**	**72.63**	**91.62**	91.08	1.29	0.54	**41.1**
50	SFP [15]	✗	**76.15**	74.61	**92.87**	92.06	1.54	0.81	41.8
	Ours (FPGM-only 30%)	✗	**76.15**	**75.03**	**92.87**	**92.40**	**1.12**	**0.47**	42.2
	Ours (FPGM-mix 30%)	✗	**76.15**	74.94	**92.87**	92.39	1.21	0.48	42.2
	Ours (FPGM-only 40%)	✗	**76.15**	74.13	**92.87**	91.94	2.02	0.93	**53.5**
	ThiNet [25]	✓	72.88	72.04	91.14	90.67	0.84	0.47	36.7
	SFP [15]	✓	**76.15**	62.14	**92.87**	84.60	14.01	8.27	41.8
	NISP [39]	✓	–	–	–	–	0.89	–	44.0
	CP [16]	✓	–	–	92.20	90.80	–	1.40	50.0
	Ours (FPGM-only 30%)	✓	**76.15**	**75.59**	**92.87**	**92.63**	**0.56**	0.24	42.2
	Ours (FPGM-mix 30%)	✓	**76.15**	75.50	**92.87**	92.63	0.65	**0.21**	42.2
	Ours (FPGM-only 40%)	✓	**76.15**	74.83	**92.87**	92.32	1.32	0.55	**53.5**
101	Rethinking [38]	✓	**77.37**	75.27	–	–	2.10	–	**47.0**
	Ours (FPGM-only 30%)	✓	**77.37**	**77.32**	**93.56**	**93.56**	**0.05**	**0.00**	42.2

Conclusion

Conclusion

- this paper elaborates on the underlying requirements for norm-based criterion and points out their limitations.
- FPGM prunes the most replaceable filters containing redundant information, which can still achieve good performances when norm-based criterion fails;
- FPGM considers the mutual relations between filters. Thanks to this, FPGM achieves the SOTA performance in several benchmaerks.