CORING: Efficient tensor-based filter pruning

Van Tien PHAM [1] Yassine ZNIYED [1] Thanh Phuong NGUYEN [1]

[1]Université de Toulon, Aix Marseille Université, CNRS, LIS, UMR 7020, France

Overview

Keywords

- Network Compression
- Structured Pruning
- Tensor Decomposition

Motivations

- Reduced Memory Footprint
- Faster Inference
- Lower Energy Consumption
- Ease of Deployment on Cloud and Edge
- Interpretability and Understanding
- Privacy and Security

Hypothesis

- CNNs are over-parameterized
- Similar filters may generate duplicate features
- Redundancy can be compensated through fine-tuning

Research Gaps

❌ Flatten 3-D tensor to 1-D vector

🎲 Data-dependent

🥵 Computationally expensive

Our Contributions

🤗 Introducing tensor decompositions for filter pruning.

🥳 Novel method to compute filters' similarity.

✅ Filter selection algorithm.

🎯 Outstanding results.

References

[1] M. Alwani, V. Madhavan, and Y. Wang. Decore: Deep compression with reinforcement learning. *CVPR*, 2022.

[2] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao. Hrank: Filter pruning using high-rank feature map. *CVPR*, 2020.

[3] Y. Sui, M. Yin, Y. Xie, H. Phan, S. Zonouz, and B. Yuan. Chip: Channel independence-based pruning for compact neural networks. In *NeurIPS*, 2021.

[4] H. Wang and Y. Fu. Trainability preserving neural pruning. In *ICLR*, 2023.

CORING Framework

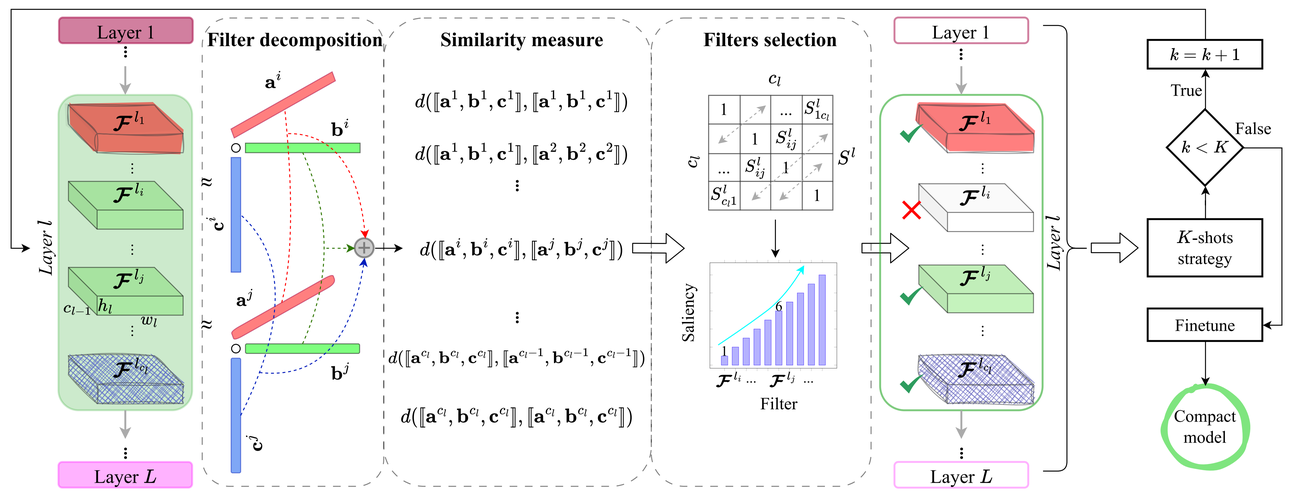


Figure 1. The CORING approach for filter pruning in one layer, summarized in three steps.

Filter decomposition

- The Tucker decomposition of $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \triangleq [\![\boldsymbol{\mathcal{S}}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]. \quad (1)$$

  $\{R_1, R_2, R_3\}$ forms the *multilinear rank* of $\boldsymbol{\mathcal{T}}$.

- Filter approximation:

$$\boldsymbol{\mathcal{F}} \approx s \times_1 \mathbf{a} \times_2 \mathbf{b} \times_3 \mathbf{c} = [\![s; \mathbf{a}, \mathbf{b}, \mathbf{c}]\!] \approx [\![\mathbf{a}, \mathbf{b}, \mathbf{c}]\!], \quad (2)$$

  where $\mathbf{a} \in \mathbb{R}^{c_{l-1}}$, $\mathbf{b} \in \mathbb{R}^{h_l}$, $\mathbf{c} \in \mathbb{R}^{w_l}$, and scalar $s \in \mathbb{R}^{1\times1\times1}$.

Similarity measure

The distance between $\boldsymbol{\mathcal{F}}^i$ and $\boldsymbol{\mathcal{F}}^j$:

$$d(\boldsymbol{\mathcal{F}}^i, \boldsymbol{\mathcal{F}}^j) = d\big([\![\mathbf{a}^i, \mathbf{b}^i, \mathbf{c}^i]\!], [\![\mathbf{a}^j, \mathbf{b}^j, \mathbf{c}^j]\!]\big) = \frac{d(\mathbf{a}^i, \mathbf{a}^j) + d(\mathbf{b}^i, \mathbf{b}^j) + d(\mathbf{c}^i, \mathbf{c}^j)}{3}. \quad (3)$$

3 similarity metrics d:

- Euclidean
- Cosine Similarity
- Variance-Based Distance:

$$d_{VBD}(\boldsymbol{\mathcal{F}}^i, \boldsymbol{\mathcal{F}}^j) = \frac{\mathrm{Var}(\boldsymbol{\mathcal{F}}^i - \boldsymbol{\mathcal{F}}^j)}{\mathrm{Var}(\boldsymbol{\mathcal{F}}^i) + \mathrm{Var}(\boldsymbol{\mathcal{F}}^j)} \quad (4)$$

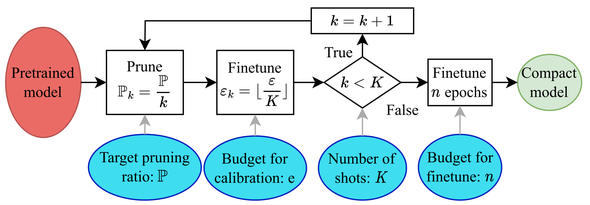
For a distance metric $d(\cdot,\cdot)$, a similarity matrix $\mathbf{S}$ of size $c \times c$ can be constructed such that $S_{ij} = d(\boldsymbol{\mathcal{F}}^i, \boldsymbol{\mathcal{F}}^j)$.

Filters selection algorithm

Require: Similarity matrix $\boldsymbol{S} \in \mathbb{R}^{c\times c}$, filters $\boldsymbol{\mathcal{F}}^1, \boldsymbol{\mathcal{F}}^2, \ldots, \boldsymbol{\mathcal{F}}^c$, sparsity κ.
Ensure: Selected filters $\boldsymbol{\mathcal{F}}^{p_1}, \boldsymbol{\mathcal{F}}^{p_2}, \ldots, \boldsymbol{\mathcal{F}}^{p_\kappa}$.

```
for t = 1 to c − κ do
    Find the highest similarity: (i, j) = argmax_{(x,y), x≠y} S_{x,y}
    if Σ_{k=1}^{c} S_{i,k} ≥ Σ_{k=1}^{c} S_{j,k} then
        Delete F^i.
    else
        Delete F^j.
    end if
    Remove the row, column of the deleted filter from S.
end for
```

K-shots pruning strategy



Experiments

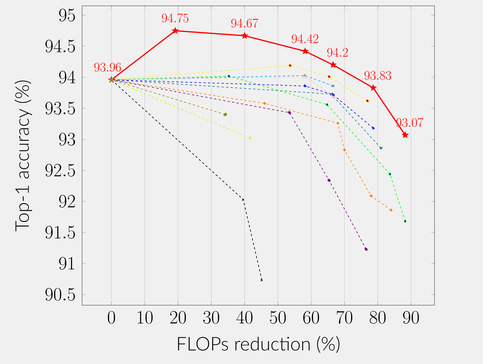


Figure 2. **Comparison** of pruning methods for VGG-16 on CIFAR-10.

Table 1. Pruning results of **ResNet-50 on ImageNet**

Model	Top-1	Top-5	Params (↓%)	FLOPs (↓%)
ResNet-50	76.15	92.87	25.50M(00)	4.09B(00)
DECORE-8 [1]	76.31	93.02	22.69M(11)	3.54B(13)
CHIP [3]	76.30	93.02	15.10M(41)	2.26B(45)
TPP [4]	76.44	N/A	N/A	2.74B(33)
CORING-V (Ours)	**76.78**	**93.23**	**15.10M(41)**	**2.26B(45)**
HRank-1 [2]	74.98	92.33	16.15M(37)	2.30B(44)
DECORE-6 [1]	74.58	92.18	14.10M(45)	2.36B(42)
CHIP [3]	76.15	92.91	14.23M(44)	2.10B(49)
CORING-C (Ours)	**76.34**	**93.06**	**14.23M(44)**	**2.10B(49)**
HRank-2 [2]	71.98	91.01	13.77M(46)	1.55B(62)
CHIP [3]	75.26	92.53	11.04M(57)	1.52B(63)
CORING-V (Ours)	**75.55**	**92.61**	**11.04M(57)**	**1.52B(63)**
HRank-3 [2]	69.10	89.58	8.27M(67)	0.98B(76)
DECORE-5 [1]	72.06	90.82	8.87M(65)	1.60B(61)
CHIP [3]	72.30	90.74	8.01M(69)	0.95B(77)
CORING-V (Ours)	**73.99**	**91.71**	**8.01M(69)**	**0.95B(77)**

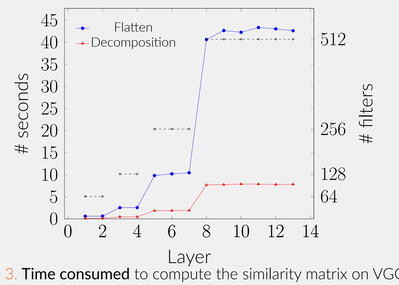


Figure 3. **Time consumed** to compute the similarity matrix on VGG-16.