Efficient Neural Network Compression

Hyeji Kim, Muhammad Umar Karim Khan, Chong-Min Kyung

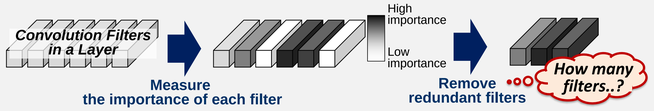
GitHub

SSAL Smart Sensor Architecture Lab

Background

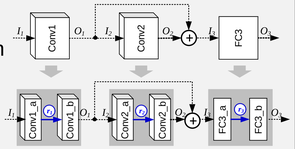
Concept of neural network compression

- Remove the redundant filters by the desired network complexity



Why we are using the kernel decomposition?

1. Decomposed filters are sorted by the importance (eigenvalues)
   → *We can focus on the choice of the optimal number of filters*
2. The in/out dimension of each original layer is not affected by the filter reduction
   → *Complex neural network can be easily compressed*



Network complexity, C

$$C(r_1, r_2, \dots, r_L) = (c_1 \times r_1) + (c_2 \times r_2) + \cdots + (c_L \times r_L)$$

- ✓ $(r_1, r_2, \dots, r_L)$: **configuration of the number of filters** for each layer
- ✓ c_l : complexity coefficient of l–*th* layer

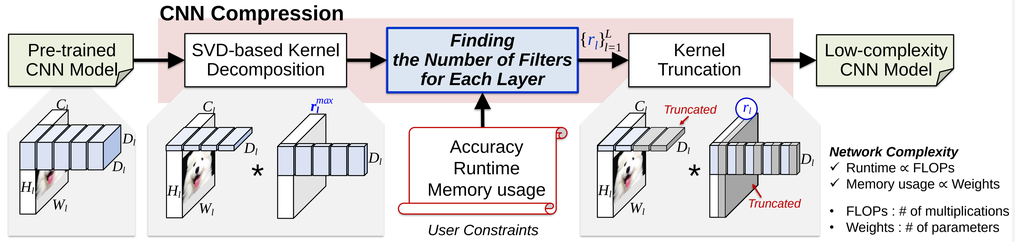
Motivation & Contribution

1. **Configuration of the number of filters strongly affects the accuracy of neural network**
2. **It is hard to find an optimal configuration of the number of filters in the overall search space**
   - Huge number of possible combinations & Significant evaluation time
3. **We need a simple and effective method to choose a configuration**

Idea : "How about forcing the equal accuracy loss for every layer during the compression?"

We propose the fast and high compression algorithm

Overall Framework



Approaches

Goal : Find an optimal configuration of the number of filters in holistic network

Constraint : *Network Complexity*

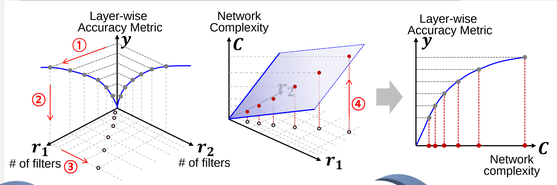
ENC-Map

- All layers have ***equal accuracy loss*** for the target complexity

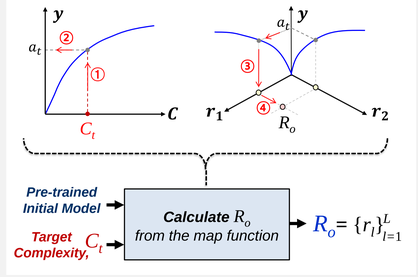
ENC-Model / ENC-Inf

- Search space is limited by **ENC-Map**
- Solution has maximum response of evaluation-metric

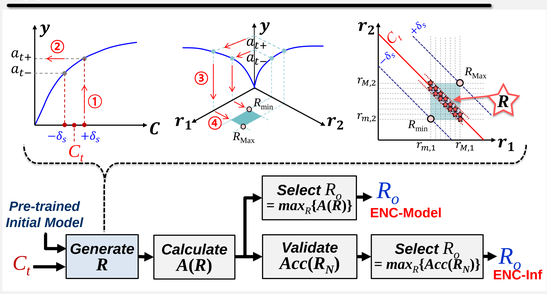
Mapping Function : Complexity & Accuracy Metric



***Approach #1 : ENC-Map*: Single-Shot Method**



***Approach #2 : ENC-Model/Inf*: Combinatorial Method**



R_o : final configuration of the number of filters

R :Candidate configurations of the number of filters

$A(R)$: whole-layer accuracy metric of R

R_N : Top-N configurations by $A(R)$

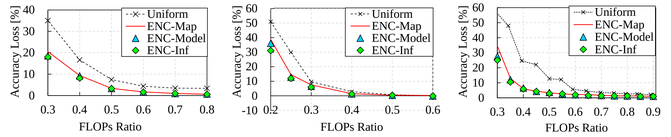
$Acc(R_N)$: validation accuracy of R_N

Experimental Results

Comparison of Proposed Methods

- **ENC-Map** ***is good at lower compression (higher FLOPs ratio)***
- **ENC-Model/Inf** ***is good at higher compression (lower FLOPs ratio)***

✓ Uniform : same compression ratio for all layers ✓ Smaller accuracy loss is better



Compression Time

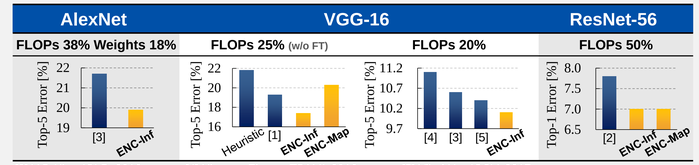
- ***Our ENC-Map is extremely fast !***

	Previous	Ours: ENC-Map	Ours: ENC-Model	Ours: ENC-Inf
AlexNet	-	~5s @CPU	~3m @CPU	~5m @4GPUs
VGG-16	4h@8GPUs [1]			
ResNet-56	1h@GPU [2]			

[1] Yihui He, *arXiv:1802.03494*, 2018 [2] Yihui He, "AMC", *ECCV*, 2018

Classification Error at Same Network Complexity

- ***Our ENC-Inf outperforms in all experiments (smaller error is better)***



[3] Y-D Kim, *ICLR*, 2016 [4] Xiangyu Zhang, *TPAMI*, 2016 [5] Yihui He, *ICCV*, 2017

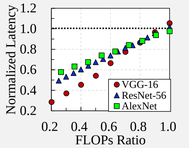
Inference on Embedded Board

Network Acceleration *without Accuracy Loss*

	Original	Compressed
ResNet-56	7.03 FPS	9.95 FPS (x 1.42)
AlexNet	1.21 FPS	2.00 FPS (x 1.65)
VGG-16	0.09 FPS	0.32 FPS (x 3.55)

✓ Compress only convolution layers

Test Board: ODROID-XU4



Acknowledgement This work was supported by MSIT as GFP / (CISS-2013M3A6A6073718). Thanks to Yeongmin Lee (KAIST, EE) for comments on the research.

Contact : hyejikim89@kaist.ac.kr, Hyeji Kim