Trained Rank Pruning For Efficient Deep Neural Networks

Yuhui Xu[1], Yuxi Li[1], Shuai Zhang[2], Wei Wen[3], Botao Wang[2], Wenrui Dai[1], Yingyong Qi[2], Yiran Chen[3], Weiyao Lin[1] and Hongkai Xiong[1]

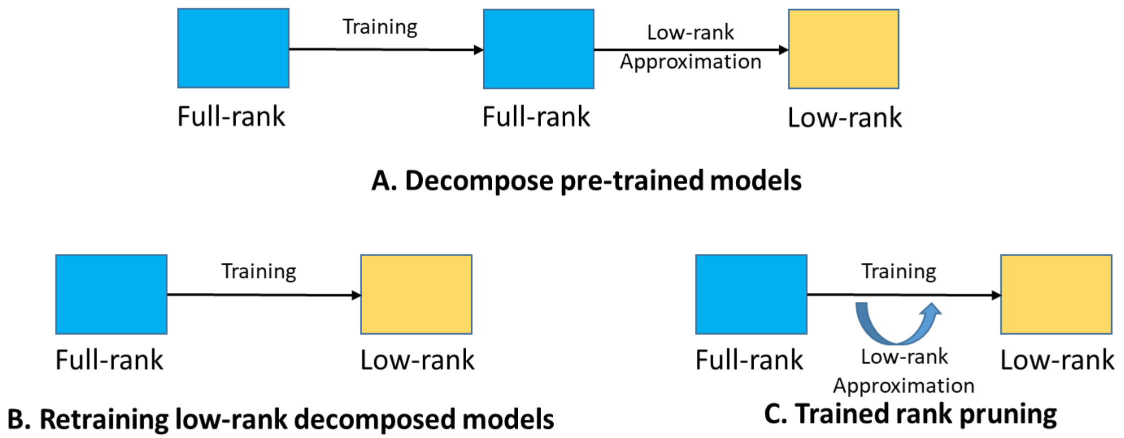
[1]Shanghai Jiao Tong University
[2]Qualcomm AI Research
[3]Duke University

Outline

- Low Rank (LR) Models
    - Methods on obtaining LR models
        - Decompose a pre-trained model
        - Retrain a LR decomposed model
    - Challenges on existing methods
- Trained Rank Pruning
    - Training LR model directly with 2 interleaved steps:
        - Step A: rank conditioning with nuclear norm constraint and sub-gradient
        - Step B: rank pruning with LR decomposition
- Experimental Results

LR Models

- Rank pruning with LR decomposition
- Decompose a pre-trained model
  - Small approximation errors can ripple a large prediction loss. Fine-tuning is required to recover some accuracy loss.
- Retrain low-rank decomposed model
  - Hard to select optimal rank for each layer to achieve good balance of model capacity and compression



A. Decompose pre-trained models

B. Retraining low-rank decomposed models

C. Trained rank pruning

Trained Rank Pruning

Our trained rank pruning method has 2 interleaved steps:

(A) Conventional SGD training with nuclear norm regularization and sub-gradient, conditioning the network to be LR compatible

- Nuclear norm constraint

$$min\left\{f(x;w) + \lambda \sum_{l=1}^{L} ||W||_*\right\}$$

- Sub-gradient descent[1]

$$g_{sub} = \Delta f + \lambda U_{tru} V_{tru}^T$$

where $W = U \sum V^T$ is the SVD decomposition and U_{tru}, V_{tru} are truncated U, V with $rank(W)$.

(B) Training with LR decomposition, obtaining the LR network with rank pruning

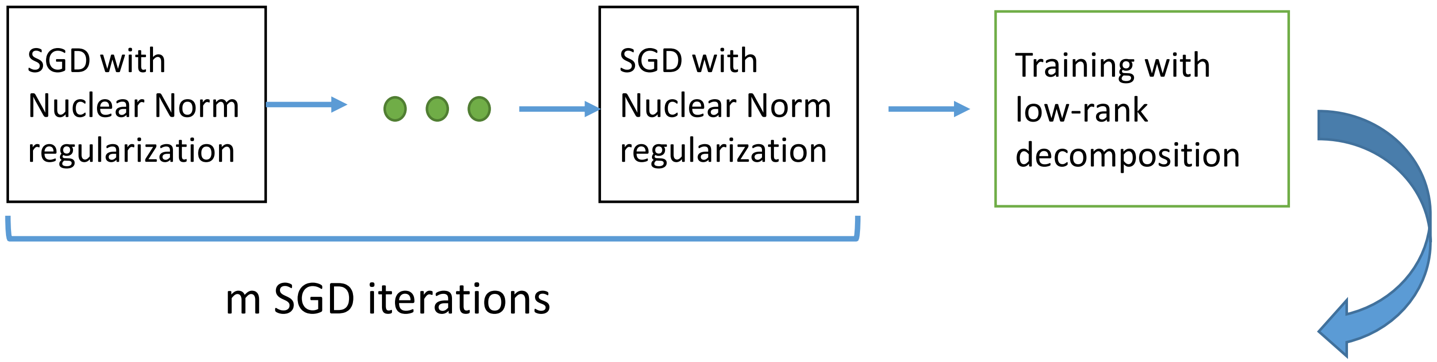
-- forward: decompose original filters T into LR filters T_low;

-- backward: update decomposed LR filters T_low with SGD and then substitute original filters.

[1] H. Avron, S. Kale, S. P. Kasiviswanathan, and V. Sindhwani. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In ICML, 2012.

Trained Rank Pruning

- Step B is inserted into training process after every *m* SGD iterations of step A.



- Capable of generating LR model parameters with diverse optimal ranks.
- Applicable to most existing decompositions, i.e. channel-wise and spatial-wise decompositions.

Experimental Results

All comparison decomposition and pruning results here are finetuned to improve accuracy, while our methods results are from direct decomposition after training.

- **TRP_spatial**: our trained rank pruning method with spatial-wise decomposition;
- **TRP_channel**: our trained rank pruning method with channel-wise decomposition;
- **Nu**: nuclear norm regularization in training;
- **Speedup**: the reduction ratio of model FLOPs

Model	Top 1 (%)	Speed up
ResNet-20 (baseline)	91.74	1.00×
ResNet-20 (TRP_spatial)	90.12	1.97×
ResNet-20 (TRP_spatial + Nu)	**90.50**	**2.17**×
ResNet-20 (Spatial-decomp)	88.13	1.41×
ResNet-20 (TRP_channel)	90.13	2.66×
ResNet-20 (TRP_channel + Nu)	**90.62**	**2.84**×
ResNet-20 (Channel-decomp)	89.49	1.66×

Table 1: Experiment results on CIFAR-10.

Method	Top1(%)	Speed up
Baseline	69.10	1.00×
TRP_spatial	**65.46**	1.81×
TRP_spatial + Nu	65.39	**2.23**×
Spatial-decomp	63.1	1.41×
TRP_channel	**65.51**	2.60×
TRP_channel + Nu	65.34	**3.18**×
Channel-decomp	62.80	2.00×

Table 2: Results of ResNet-18 on ImageNet.

Method	Top1(%)	Speed up
Baseline	75.90	1.00×
TRP_spatial + Nu	72.69	**2.30**×
TRP_spatial + Nu (diff hyper-param)	**74.06**	1.80×
Spatial-decomp	71.80	1.50×
Filter pruning-ICCV2017	72.04	1.58
Thinet-TPAMI2018	72.03	2.26

Table 3: Results of ResNet-50 on ImageNet.

On both CIFAR-10 and ImageNet datasets, it shows that our TRP methods can outperform other existing methods both in channel-wise decomposition and spatial-wise decomposition formats. It achieves better balance of accuracy and complexity.