
Trained Rank Pruning for Efficient Deep Neural Networks

Yuhui Xu¹, Yuxi Li¹, Shuai Zhang², Wei Wen³, Botao Wang², Wenrui Dai¹, Yingyong Qi², Yiran Chen³, Weiyao Lin¹ and Hongkai Xiong¹

¹Shanghai Jiao Tong University, Email: {yuhuixu, lyxok1, daiwenrui, wylin, xionghongkai}@sjtu.edu.cn

²Qualcomm AI Research, Email: {shuazhan, botaow, yingyong}@qti.qualcomm.com

³Duke University, Email: {wei.wen, yiran.chen}@duke.edu

Abstract

To accelerate DNNs inference, low-rank approximation has been widely adopted because of its solid theoretical rationale and efficient implementations. Several previous works attempted to directly approximate a pre-trained model by low-rank decomposition; however, small approximation errors in parameters can ripple over a large prediction loss. Apparently, it is not optimal to separate low-rank approximation from training. Unlike previous works, this paper integrates low rank approximation and regularization into the training process. We propose Trained Rank Pruning (TRP), which alternates between low rank approximation and training. TRP maintains the capacity of the original network while imposing low-rank constraints during training. A nuclear regularization optimized by stochastic sub-gradient descent is utilized to further promote low rank in TRP. Networks trained with TRP has a low-rank structure in nature, and is approximated with negligible performance loss, thus eliminating fine-tuning after low rank approximation. The proposed method is comprehensively evaluated on CIFAR-10 and ImageNet, outperforming previous compression counterparts using low rank approximation. Our code is available at: <https://github.com/yuhuixu1993/Trained-Rank-Pruning>.

1 Introduction

Deep Neural Networks (DNNs) have shown remarkable success in many computer vision tasks such as image classification [8], object detection [15] and semantic segmentation [3]. Despite the high performance in large DNNs powered by cutting-edge parallel computing hardware, most of state-of-the-art network architectures are not suitable for resource restricted usage such as usages on always-on devices, battery-powered low-end devices, due to the limitations on computational capacity, memory and power.

To address this problem, low-rank decomposition methods [6, 10, 7, 17, 1] have been proposed to minimize the channel-wise and spatial redundancy by decomposing the original network into a compact one with low-rank layers. Different from precedent works, this paper proposes a novel approach to design low-rank networks.

Low-rank networks can be trained directly from scratch. However, it is difficult to obtain satisfactory results for several reasons. (1) *Low capacity*: Compared with the original full rank network, the capacity of a low-rank network is limited, which causes difficulties in optimizing its performances. (2) *Deep structure*: Low-rank decomposition typically doubles the number of layers in a network. The additional layers make numerical optimization much more challenging because of exploding and/or vanishing gradients. (3) *Rank selection*: The rank of decomposed network is often chosen as a

hyperparameter based on pre-trained networks; which may not be the optimal rank for the network trained from scratch.

Alternatively, several previous works [18, 7, 10] attempted to decompose pre-trained models in order to get initial low-rank networks. However, the heuristically imposed low-rank could incur huge accuracy loss and network retraining is required to recover the performance of the original network as much as possible. Some attempts were made to use sparsity regularization [17, 4] to constrain the network into a low-rank space. Though sparsity regularization reduces the error incurred by decomposition to some extent, performance still degrades rapidly when compression rate increases.

In this paper, we propose a new method, namely Trained Rank Pruning (TRP), for training low-rank networks. We embed the low-rank decomposition into the training process by gradually pushing the weight distribution of a well functioning network into a low-rank form, where all parameters of the original network are kept and optimized to maintain its capacity. We also propose a stochastic sub-gradient descent optimized nuclear regularization that further constrains the weights in a low-rank space to boost the TRP. The proposed solution is illustrated in Fig. 1.

Overall, our contributions are summarized below.

1. A new training method called TRP is presented by explicitly embedding the low-rank decomposition into the network training;
2. A nuclear regularization is optimized by stochastic sub-gradient descent to boost the performance of the TRP;
3. Improving inference acceleration and reducing approximation accuracy loss in both channel-wise and spatial-wise decomposition methods.

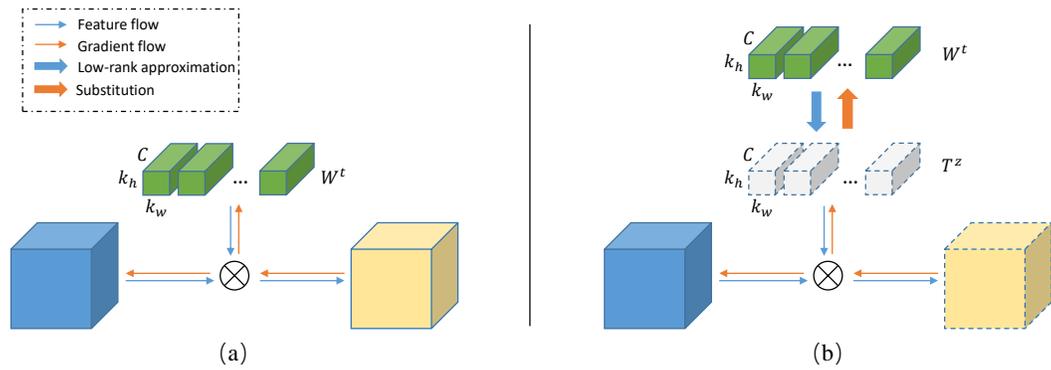


Figure 1: The training of TRP consists of two parts as illustrated in (a) and (b). (a) One normal iteration with forward-backward broadcast and weight update. (b) One training iteration inserted by rank pruning, where the low-rank approximation is first applied on current filters before convolution. During backward propagation, the gradients are directly added on low-rank filters and the original weights are substituted by updated low-rank filters. (b) is applied once every m iterations (*i.e.* when gradient update iteration $t = zm, z = 0, 1, 2, \dots$), otherwise (a) is applied.

2 Methodology

2.1 Preliminaries

Formally, the convolution filters in a layer can be denoted by a tensor $W \in \mathbb{R}^{n \times c \times k_w \times k_h}$, where n and c are the number of filters and input channels, k_h and k_w are the height and width of the filters. An input of the convolution layer $F_i \in \mathbb{R}^{c \times x \times y}$ generates an output as $F_o = W * F_i$. Channel-wise correlation [18] and spatial-wise correlation [10] are explored to approximate convolution filters in a low-rank space. In this paper, we focus on these two decomposition schemes. However, unlike the previous works, we propose a new training scheme TRP to obtain a low-rank network without re-training after decomposition.

2.2 Trained Rank Pruning

We propose a simple yet effective training scheme called Trained Rank Pruning (TRP) in a periodic fashion:

$$W^{t+1} = \begin{cases} W^t - \alpha \nabla f(W^t) & t \% m \neq 0 \\ T^z - \alpha \nabla f(T^z) & t \% m = 0 \end{cases} \quad (1)$$

$$T^z = \mathcal{D}(W^t), \quad z = t/m$$

where $\mathcal{D}(\cdot)$ is a low-rank tensor approximation operator, α is the learning rate, t indexes the iteration and z is the iteration of the operator \mathcal{D} , with m being the period for the low-rank approximation.

We apply low-rank approximation every m SGD iterations. This saves training time to a large extent. As illustrated in Fig. 1, for every m iterations, we perform low-rank approximation on the original filters, while gradients are updated on the resultant low-rank form. Otherwise, the network is updated via the normal SGD. Our training scheme could be combined with arbitrary low-rank operators. In the proposed work, we choose the low-rank techniques proposed in [10] and [18], both of which transform the 4-dimensional filters into 2D matrix and then apply the truncated singular value decomposition (TSVD). The SVD of matrix W^t can be written as:

$$W^t = \sum_{i=1}^{\text{rank}(W^t)} \sigma_i \cdot U_i \cdot (V_i)^T, \quad (2)$$

where σ_i is the singular value of W^t with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\text{rank}(W^t)}$, and U_i and V_i are the singular vectors. The parameterized TSVD ($W^t; e$) is to find the smallest integer k such that

$$\sum_{j=k+1}^{\text{rank}(W^t)} (\sigma_j)^2 \leq e \sum_{i=1}^{\text{rank}(W^t)} (\sigma_i)^2, \quad (3)$$

where $e \in (0, 1)$ is a pre-defined hyper-parameter of the energy-preserving ratio. After truncating the last $n - k$ singular values, we transform the low-rank 2D matrix back to 4D tensor.

2.3 Nuclear Norm Regularization

Nuclear norm is widely used in matrix completion problems. Recently, it is introduced to constrain the network into low-rank space during the training process [1].

$$\min \left\{ f(x; w) + \lambda \sum_{l=1}^L \|W_l\|_* \right\} \quad (4)$$

where $f(\cdot)$ is the objective loss function, nuclear norm $\|W_l\|_*$ is defined as $\|W_l\|_* = \sum_{i=1}^{\text{rank}(W_l)} \sigma_l^i$, with σ_l^i the singular values of W_l . λ is a hyper-parameter setting the influence of the nuclear norm. In this paper, we utilize stochastic sub-gradient descent [2] to optimize nuclear norm regularization in the training process. Let $W = U\Sigma V^T$ be the SVD of W and let U_{tru}, V_{tru} be U, V truncated to the first $\text{rank}(W)$ columns or rows, then $U_{tru}V_{tru}^T$ is the sub-gradient of $\|W\|_*$ [16]. Thus, the sub-gradient of Eq. (4) in a layer is

$$\nabla f + \lambda U_{tru}V_{tru}^T. \quad (5)$$

The nuclear norm and loss function are optimized simultaneously during the training of the networks and can further be combined with the proposed TRP.

3 Experiments

3.1 Implementation Details

We evaluate the performance of TRP scheme on two common datasets, CIFAR-10 [11] and ImageNet [5]. We implement our TRP scheme with NVIDIA 1080 Ti GPUs. For training on CIFAR-10, we

Model ("R-" indicates ResNet-.)	Top 1 (%)	Speed up
R-20 (baseline)	91.74	1.00×
R-20 (TRP1)	90.12	1.97×
R-20 (TRP1+Nu)	90.50	2.17 ×
R-20 ([18])	88.13	1.41×
R-20 (TRP2)	90.13	2.66×
R-20 (TRP2+Nu)	90.62	2.84 ×
R-20 ([10])	89.49	1.66×
R-56 (baseline)	93.14	1.00×
R-56 (TRP1)	92.77	2.31×
R-56 (TRP1+Nu)	91.85	4.48 ×
R-56 ([18])	91.56	2.10×
R-56 (TRP2)	92.63	2.43×
R-56 (TRP2+Nu)	91.62	4.51 ×
R-56 ([10])	91.59	2.10×
R-56 [9]	91.80	2.00×
R-56 [12]	91.60	2.00×

Table (2) Experiment results on CIFAR-10.

Method	Top1(%)	Speed up
Baseline	69.10	1.00×
TRP1	65.46	1.81×
TRP1+Nu	65.39	2.23 ×
[18]	63.1	1.41×
TRP2	65.51	2.60×
TRP2+Nu	65.34	3.18 ×
[10]	62.80	2.00×

Table (3) Results of ResNet-18 on ImageNet.

Method	Top1(%)	Speed up
Baseline	75.90	1.00×
TRP1+Nu	72.69	2.30 ×
TRP1+Nu	74.06	1.80×
[18]	71.80	1.50×
[13]	72.04	1.58
[14]	72.03	2.26

Table (4) Results of ResNet-50 on ImageNet.

start with base learning rate of 0.1 to train 164 epochs and degrade the value by a factor of 10 at the 82-th and 122-th epoch. For ImageNet, we directly finetune the model with TRP scheme from the pre-trained baseline with learning rate 0.0001 for 10 epochs. For both of the datasets, we adopt SGD solver to update weight and set the weight decay value as 10^{-4} and momentum value as 0.9.

3.2 Results on CIFAR-10

As shown in Table 2, for both spatial-wise (TRP1) and channel-wise (TRP2) decomposition, the proposed TRP outperforms basic methods [18, 10] on ResNet-20 and ResNet-56. Results become even better when nuclear regularization is used. For example, in the channel-wise decomposition (TRP2) of ResNet-56, results of TRP combined with nuclear regularization can even achieve $2\times$ speed up rate than [18] with same accuracy drop. Our method also outperforms filter pruning [12] and channel pruning [9]. For example, the channel decomposed TRP trained ResNet-56 can achieve 92.77% accuracy with $2.31\times$ acceleration, while [9] is 91.80% and [12] is 91.60%. With the help of nuclear regularization, our methods can obtain 2 times of the acceleration rate of [9] and [12] with higher accuracy.

3.3 Results on ImageNet

The results on ImageNet are shown in Table 3 and Table 4. For ResNet-18, our method outperforms the basic methods [18, 10]. For example, in the channel-wise decomposition, TRP obtains $1.81\times$ speed up rate with 86.48% Top5 accuracy on ImageNet which outperforms both the data-driven [18]¹ and data independent [18] methods by a large margin. Nuclear regularization can increase the speed up rates with the same accuracy.

For ResNet-50, to better validate the effectiveness of our method, we also compare the proposed TRP with [9] and [13]. With $1.80\times$ speed up, our decomposed ResNet-50 can obtain 73.97% Top1 and 91.98% Top5 accuracy which is much higher than [13]. The TRP achieves $2.23\times$ acceleration which is higher than [9] with the same Top5 degrade.

4 Conclusion

In this paper, we propose a new scheme Trained Rank Pruning (TRP) for training low-rank networks. It leverages capacity and structure of the original network by embedding the low-rank approximation in the training process. Furthermore, we propose stochastic sub-gradient descent optimized nuclear norm regularization to boost the TRP. The proposed TRP can be incorporated with any low-rank decomposition method. On CIFAR-10 and ImageNet datasets, we have shown that our methods can outperform basic methods both in channel-wise decomposition and spatial-wise decomposition.

¹the implementation of [7]

References

- [1] J. M. Alvarez and M. Salzmann. Compression-aware training of deep networks. In *NIPS*, 2017.
- [2] H. Avron, S. Kale, S. P. Kasiviswanathan, and V. Sindhvani. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In *ICML*, 2012.
- [3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40:834–848, 2018.
- [4] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. In *ICML*, 2015.
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009.
- [6] E. Denton, W. Zaremba, J. Bruna, Y. Lecun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, 2014.
- [7] J. Guo, Y. Li, W. Lin, Y. Chen, and J. Li. Network decoupling: From regular to depthwise separable convolutions. In *BMVC*, 2018.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016.
- [9] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *ICCV*, 2017.
- [10] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.
- [11] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Computer Science*, 2009.
- [12] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.
- [13] J.-H. Luo, J. Wu, and W. Lin. Thinet: A filter level pruning method for deep neural network compression. *ICCV*, 2017.
- [14] J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, and W. Lin. Thinet: pruning cnn filters for a thinner net. *TPAMI*, 2018.
- [15] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 39:1137–1149, 2015.
- [16] G. A. Watson. Characterization of the subdifferential of some matrix norms. *Linear algebra and its applications*, 170:33–45, 1992.
- [17] W. Wen, C. Xu, C. Wu, Y. Wang, Y. Chen, and H. Li. Coordinating filters for faster deep neural networks. In *ICCV*, 2017.
- [18] X. Zhang, J. Zou, K. He, and J. Sun. Accelerating very deep convolutional networks for classification and detection. *TPAMI*, 38(10):1943–1955, 2016.