Supplementary Materials of "Adaptive Tensor Networks Decomposition"

Chang Nie
changnie@njust.edu.cn
Huan Wang
wanghuanphd@njust.edu.cn
Le Tian
119106021993@njust.edu.cn

School of Computer Science and Engineering, Nanjing University of Science & Technology, Nanjing, China

1 ATN-based Models

We apply ATN to three typical high-dimensional optimization tasks: tensor completion, image denoising, and neural network compression. These optimization problems are mainly based on the low-rank assumption of the data. In other words, they aim to capture the intrinsic structure inside high-dimensional data and to eliminate redundancy through the low-rank representation.

1.1 Tensor completion

Tensor completion (TC) aims to obtain complete data by imposing low-rank constraints on the observed entries [3]. The general TC methods can be divided into two categories [8], rank-minimization-oriented and TD-oriented. Rank-minimization-oriented methods transfer tensor rank optimization into a matrix nuclear norm minimization by convex relaxation and matrixization. TD-oriented methods express data as the connection of a few factors in a multilinear space and maintain low-rank properties implicitly via the edge rank. Therefore, the TC problem can be expressed as:

$$\min_{\boldsymbol{\mathcal{X}}} \quad \boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{X}}) = R(\boldsymbol{\mathcal{X}}) + \frac{\lambda}{2}||P_{\Omega}(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}})||_F^2, \tag{1}$$

where the $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ are restored low-rank tensor and incomplete observing tensor, P_{Ω} is a projection operator that maps elements in the set $\Omega \subset [I_1] \times \cdots \times [I_N]$ (observed elements index) to itself and others to zero, and $R(\cdot)$ represents a certain low-rank restrain. We express $\boldsymbol{\mathcal{X}}$ in the form of ATN and remove all the rank-1edges since they have no substantial contribution to the contraction result. Then the following minimization problem can be deduced from (1) as follows:

$$\min_{\{\boldsymbol{\mathcal{Z}}^{(i)}\}_{i=1}^N} \boldsymbol{\mathcal{L}}(\{\boldsymbol{\mathcal{Z}}^{(i)}\}_{i=1}^N) = ||P_{\Omega}(TN(\boldsymbol{\mathcal{Z}}^{(1)}, \ldots, \boldsymbol{\mathcal{Z}}^{(N)}) - \boldsymbol{\mathcal{Y}}))||_F^2 \tag{2}$$

The optimization problem $\mathcal{L}$ is differentiable and the variational parameters can be updated iteratively based on the gradient descent algorithm, given by $\mathcal{Z}^{(i)} = \mathcal{Z}^{(i)} - \eta \frac{\partial \mathcal{L}}{\partial \mathcal{Z}^{(i)}}$. We can obtain the reconstructed tensor $\mathcal{X} = P_{\Omega}(\mathcal{Y}) + P_{\overline{\Omega}}(TN(\mathcal{Z}^{(1)}, \ldots, \mathcal{Z}^{(N)}))$ after $\mathcal{L}$ stops decreasing, where $\overline{\Omega}$ denote missing entries's indices set. The selection of hyperparameter κ has a significant impact on model performance. When the data missing ratio is relatively large, we suggest adopting a smaller κ to prevent overfitting.

1.2 Image Denoising

The intention of image denoising [7] is to recover the original feature from the image contaminated by various noise. We reshape the image into a high-order tensor and apply ATN decomposition to solve the denoising problem. For a given contaminated data tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ can be separated as [4] $\mathcal{X} = \mathcal{L}_0 + \mathcal{N}_0$, assuming that $\mathcal{L}_0$ is low-rank tensor and $\mathcal{N}_0$ is a small perturbation tensor. So we can depict the denoising optimization problem as:

$$\begin{aligned} \min_{\mathcal{L}_0, \{\mathcal{Z}^{(i)}\}_{i=1}^{N}} \quad & ||\mathcal{L}_0 - TN(\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}, \ldots, \mathcal{Z}^{(N)})||_F^2 \\ s.t. \quad & \mathcal{X} = \mathcal{L}_0 + \mathcal{N}_0. \end{aligned} \tag{3}$$

The (3) is solved by utilizing the gradient descent strategy, which is consistent with (2). Although TC and denoising are different types of tasks, they are based on the internal low-rank assumption of the original data, and derived optimization items are similar. Further, we can describe a class of low-rank optimization problems in a unified framework. To decompose Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ with ATN and obtain $TN(\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}, \ldots, \mathcal{Z}^{(N)})$, then the unified optimization form can be expressed as:

$$\min_{\{\mathcal{Z}^{(i)}\}_{i=1}^{N}} \boldsymbol{L}(\mathcal{X}, TN(\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}, \ldots, \mathcal{Z}^{(N)})) + \lambda \zeta(\kappa). \tag{4}$$

Where the $\boldsymbol{L}(\cdot)$ represents optimization objective likes structural loss and $\zeta(\cdot)$ indicates prior assumptions or complexity constraints of the model, λ is hyperparameter used to trade-off the two items and can be adjusted according to actual results. It explicates that we can employ ATN to more low-rank problems based on TD, such as neural network compression.

1.3 Neural Network Compression

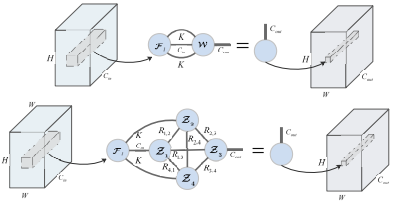

Figure 1: Illustration of standard convolution (top) and ATN-based convolution (bottom). The convolution process can be intuitively represented in the form of a tensor network.

Convolutional neural networks (CNNs) can be compressed through ATN decomposition to favor the development of mobile devices. For the input feature map $\mathcal{F}$ with size $W \times$

$H \times C_{in}$, where W, H, C_{in} denote the width, height, and the number of channels, respectively. The standard convolution utilizes a filter $\boldsymbol{\mathcal{W}}$ of the size $K \times K \times C_{in} \times C_{out}$ to act on $\boldsymbol{\mathcal{F}}$, where the K, C_{out} are the kernel size and the number of output channels [2]. We graphically illustrate the convolution process through TN as shown in Fig.1. The filter $\boldsymbol{\mathcal{W}}$ acts on the local receptive field tensor of the feature map $\boldsymbol{\mathcal{F}}_l$, and obtains a tubal of the output feature map by contracting three shared indexes. We decompose filter $\boldsymbol{\mathcal{W}}$ with ATN and obtain four cores factors $\{\boldsymbol{\mathcal{Z}}^{(1)}, \boldsymbol{\mathcal{Z}}^{(2)}, \boldsymbol{\mathcal{Z}}^{(3)}, \boldsymbol{\mathcal{Z}}^{(4)}\}$. The parameters and FLOPs of the ATN-based convolution are directly controlled by the edge rank R. The larger R is, the more computing resources are required.

We achieve an ATN-based convolution with the fully optimized Conv2d function in PyTorch [6]. The ATN rank was previously calculated through pre-trained model weights. First, the pointwise convolution (standard convolution with K=1) applies to $\boldsymbol{\mathcal{F}}$ and the number of output channels is $R_{12} \times R_{13} \times R_{14}$. Then the group convolution are performed (each group uses the same convolution kernel) with a group size R_{12} and a kernel of the size $1 \times K$, so the number of output channels is $R_{13} \times R_{14} \times R_{23} \times R_{24}$. After that, we use the channel shuffling [5, 9] strategy to change the order of contract indexes. Then we use another group convolution with group size $R_{14} \times R_{24}$ and kernel size $K \times 1$ and get the number of output channels $R_{13} \times R_{23} \times R_{43}$. Finally, another pointwise convolution is employed to obtain the output feature map. Reference [1] has enumerated the possible decomposition methods of the filters, but the size of the filter in each convolution layer is different and the edge rank is usually determined by handcraft. The layer-wise search decomposition is costly, and the implementation steps are inefficient.The ATN-based convolution filter decomposition can automatically select the edge rank according to the model capacity and has stronger robustness.

2 Numerical Experiments

Besides the above three tasks, the tensor decomposition task is also used to demonstrate that ATN can well capture the low-rank structure of data with less storage complexity. The test data is the same as the image denoising task, i.e. including 2 RGB color images Img1 and Img2 (both reshaped to the size of $8 \times 8 \times 8 \times 8 \times 3$) and 2 videos Vid1 and Vid2, each composed of 32 frames (reshaped to $8 \times 4 \times 8 \times 8 \times 8 \times 8 \times 3$). We carefully adjusted the rank of other TD models to make their compression ratio r as close as possible. Table 1 reports the evaluation indicators of four TD models under similar compression ratios. It can be seen that ATN obtains better results even with larger r on video1, where the RSE is 0.09 lower and the PSNR is 1.7 higher than TR. The TN topological structure of four tensors obtained by ATN is presented in Fig. 2. One vertice corresponding to the channel mode (upper right corner in Fig. 2) is not connected with others, which indicates the Img1 channel features are weakly correlated with spatial features. This experiment shows that ATN has better data approximation ability than other models.

3 Proof of Theorems

Theorem 1 *Let Nth-order tensor* $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *and* $I_1 = \cdots = I_N = I$. *The multilinear tensor rank is denoted by* $(rank(\boldsymbol{X}_{(1)}), \cdots, rank(\boldsymbol{X}_{(N)}))$, *where* $\boldsymbol{X}_{(k)}$ *is mode-k matircization*

Table 1: Comparison of RSE and PSNR values on four tensors after decomposition via different TD models.

	TUCKER			TT			TR			ATN		
	r	RSE	PSNR	r	RSE	PSNR	r	RSE	PSNR	r	RSE	PSNR
Img1	10.6	0.255	19.7	10.9	0.195	22.1	9.75	0.123	26.1	10.6	**0.101**	**27.8**
Img2	10.6	0.205	22.5	10.9	0.150	23.2	9.75	0.112	25.7	12.4	**0.102**	**26.6**
Vid1	10.6	0.183	20.1	9.68	0.074	27.8	10.7	0.051	31.1	13.4	**0.042**	**32.8**
Vid2	10.6	0.328	17.2	9.68	0.191	21.9	10.7	0.175	22.7	9.56	**0.114**	**26.4**

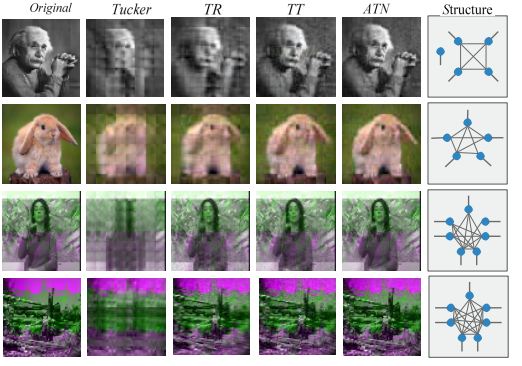

Figure 2: Illustration of four tensors decomposed by different TD models and the corresponding topological structures of ATN. The first two images in column one are Img1 and Img2, and the last two images are the 1-*st* frame of Vid1 and Vid2.

of $\mathcal{X}$*, then the following inequalities holds for* $i = 1, \cdots, N$*:*

$$\begin{aligned} &\min\{N-1, rank(\boldsymbol{X}_{(i)})\} \leq \sum_{j=1, j\neq i}^{N} rank(\mathcal{X}_{(i,j)}) \leq (N-1) rank(\boldsymbol{X}_{(i)}) \\ &\max\{rank(\mathcal{X}_{(i,j)})\}_{j=1,j\neq i}^{N} \leq rank(\boldsymbol{X}_{(i)}) \leq I(\frac{\sum_{j=1,j\neq i}^{N} rank(\mathcal{X}_{(i,j)})}{N-1})^{N-2}. \end{aligned} \tag{5}$$

Proof. The mode-i matircization of $\mathcal{X}$ is denoted by $\boldsymbol{X}_{(i)} \in \mathbb{R}^{I_i \times \prod_{j=1,j\neq i}^{N} I_j}$ and satisfy $rank(\boldsymbol{X}_{(i)}) = rank([X_{(i,j)}^{(1)}, \cdots, X_{(i,j)}^{(\prod_{k=1,k\neq i,j}^{N} I_k)}])$. According to the definitions of generalized tensor rank, we can obtain $rank(\mathcal{X}_{(i,j)}) \leq rank(\boldsymbol{X}_{(i)})$, for $j \neq i$, then the above inequalities can be easily obtained. In addition, for the case of $I_i \gg I_1 = \cdots I_{i-1} = I_{i+1} \cdots I_N = I$, the lower inequality still holds since $rank(\boldsymbol{X}_{(i)}) \leq \frac{I(\prod_{j=1,j\neq i}^{N} rank(\mathcal{X}_{(i,j)}))}{\max\{rank(\mathcal{X}_{(i,j)})\}_{j=1,j\neq i}^{N}} \leq I(\frac{\sum_{j=1,j\neq i}^{N} rank(\mathcal{X}_{(i,j)})}{N-1})^{N-2}$.

Theorem 2 *For Nth-order tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *and Mth-order tensor* $\mathcal{Y} \in \mathbb{R}^{I_{N+1} \times \cdots \times I_{N+M}}$*, let* $\mathcal{Z} = \mathcal{X} \circ \mathcal{Y}$*, The generalized tensor rank of* $\mathcal{Z}_{(n,m)}$ *is equal to 1 consistently for any* $1 \leq n \leq N, N+1 \leq m \leq N+M$*.*

Proof. The element-wise form of $\mathcal{Z}_{(n,m)} \in \mathbb{R}^{I_n \times I_m \times \prod_{j\neq m,n} I_j}$ is given by:

$$\mathcal{Z}_{(n,m)}(i_n, i_m, \overline{i_1 \cdots i_{n-1} i_{n+1} \cdots i_{m-1} i_{m+1} \cdots i_{N+M}}) = \mathcal{X}(i_1, \cdots, i_N) \mathcal{Y}(i_{N+1}, \cdots, i_{N+M}). \tag{6}$$

The k-th frontal slices of $\boldsymbol{\mathcal{Z}}_{(n,m)}$, for $1 \leq k \leq \prod_{i \neq n,m} I_i$, denote as $\boldsymbol{\mathcal{Z}}_{(n,m)}^{(k)} \in \mathbb{R}^{I_n \times I_m}$, then we have

$$\boldsymbol{\mathcal{Z}}_{(n,m)}^{(k)} = \boldsymbol{\mathcal{X}}(i_1, \cdots, i_{n-1}, :, i_{n+1}, \cdots i_N) \circ \boldsymbol{\mathcal{Y}}(i_{N+1}, \cdots, i_{m-1}, :, i_{m+1}, \cdots i_{N+M}), \tag{7}$$

where the $k = 1 + \sum_{l=1, l \neq n,m}^{N+M} (i_l - 1) \prod_{t=1, t \neq n,m}^{l-1} I_t$. That indicate each frontal slices of $\boldsymbol{\mathcal{Z}}_{(n,m)}$ can be expressed as outer product of a mode-n fibers in $\boldsymbol{\mathcal{X}}$ and a mode-m fibers in $\boldsymbol{\mathcal{Y}}$, so $rank(\boldsymbol{\mathcal{Z}}_{(n,m)}) = 1$ is always established.

Theorem 3 *Let Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $1 < I_1 \leq \cdots \leq I_N$. Under a certain κ, the number of parameter' upper bound and computational complexity of ATN decomposition is $\sum_{i=1}^{N} (\prod_{j=1}^{i} I_j) I_i^{N-i} \kappa^{N-1}$ and $\mathcal{O}(\sum_{j=2}^{N} (\kappa^{j(N-j)+j-1} \prod_{i=1}^{j} I_i^{N-j+1})/I_j)$.*

Proof. Assuming that $\boldsymbol{\mathcal{X}}$ can be decomposed by ATN and obtain $\boldsymbol{\mathcal{Z}}^{(K)} \in \mathbb{R}^{R_{1,k} \times \cdots \times R_{k-1,k} \times I_k \times \cdots \times R_{N,k}}$ for $k \in [N]$. According to the definition of $R_{(i,j)}$, we have $R_{i,j} \leq \kappa I_i$ for $1 \leq i < j \leq N$. Then the upper bound of the number of parameters required is $\sum_{i=1}^{N} (\prod_{j=1}^{i} I_j) I_i^{N-i} \kappa^{N-1}$. The computational complexity of ATN is mainly generated by $N-1$ tensor contraction, and the k-th contraction can represent with the multilinear operation form as:

$$\boldsymbol{\mathcal{X}}^k = \boldsymbol{\mathcal{X}}^{k-1} \times_{R_{1,k+1}, \cdots, R_{k,k+1}} \boldsymbol{\mathcal{Z}}^{(K+1)}, \tag{8}$$

where $\boldsymbol{\mathcal{X}}^{k-1}$ is size of $I_1 \cdots I_k R_{1,k+1} \cdots R_{1,N} \cdots R_{k,k+1} \cdots R_{k,N}$ and $\times_{R_{1,k+1}, \cdots, R_{k,k+1}}$ denote summation in the indexes represented by R. The computational complexity of (8) is

$$\begin{aligned} \mathcal{O}((\prod_{m=1}^{k} R_{m,k+1})(\prod_{i=1}^{k+1} I_i (\prod_{j=k+2}^{N} R_{i,j}))) &\leq \mathcal{O}((\prod_{m=1}^{k} \kappa I_m)(\prod_{i=1}^{k+1} I_i (\prod_{j=k+2}^{N} \kappa I_i))) \\ &\leq \kappa^{(k+1)(N-k-1)+k} \prod_{i=1}^{k+1} I_i^{N-k}) / I_{k+1} \end{aligned} \tag{9}$$

Therefore, we can obtain the computational complexity of ATN is $\mathcal{O}(\sum_{j=2}^{N} (\kappa^{j(N-j)+j-1} \prod_{i=1}^{j} I_i^{N-j+1})/I_j)$.

References

[1] Kohei Hayashi, Taiki Yamaguchi, Yohei Sugawara, and Shin ichi Maeda. Exploring unexplored tensor network decompositions for convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 32, pages 5552–5562, 2019.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[3] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 35, pages 208–220, 2009.

[4] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):925–938, 2020.

[5] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 122–138, 2018.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037, 2019.

[7] Yue Wu, Leyuan Fang, and Shutao Li. Weighted tensor rank-1 decomposition for nonlocal image denoising. *IEEE Transactions on Image Processing*, 28(6):2719–2730, 2019.

[8] Longhao Yuan, Chao Li, Danilo P. Mandic, Jianting Cao, and Qibin Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9151–9158, 2019.

[9] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.