Evolutionary Topology Search for Tensor Network Decomposition: an Extended Abstract

Chao Li[1], **Zhun Sun**[1], **Junhua Zeng**[1,2], **and Qibin Zhao**[1*]
[1]RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan
[2]Guangdong University of Technology
{chao.li, qibin.zhao}@riken.jp

Abstract

Tensor network (TN) decomposition is a promising framework to represent extremely high-dimensional problems with few parameters. However, it remains challenging to search the (near-)optimal topological structures for TN decomposition, since the number of candidate solutions exponentially grows with increasing the order of a tensor. In this work, we claim that the issue can be *practically* tackled by genetic algorithms (GAs) in an affordable manner, and the key is to encode the complex topological structures into fixed-length binary strings, *a.k.a.*, chromosomes in the context of GA. The experimental results on natural images demonstrate that, in the decomposition task, GA is able to discover more efficient topologys than the well-known TN models within a small number of generations.

1 Introduction

How to use fewer parameters to represent data and models is a crucial issue in both machine learning and scientific computing. To tackle this problem, tensor decomposition (TD), including CANDECOMP/PARAFAC decomposition [3], Tucker decomposition [30] and their variants [8], becomes a promising framework, which decomposes high-order tensors into low-dimensional forms [15]. In real-world applications, TD has been successfully applied to model compression [2], multi-modality and multi-task learning [18, 22, 36], latent variable model [1, 19], data restoration [13, 37] to name a few.

More recently, some studies brought the idea from physics to extend TD into a more sophisticated form, *i.e.*, tensor network (TN) [5, 6, 14], which derives various models such as tensor train (TT) [25], tensor ring (TR) [38], hierarchical Tucker (HT, or tensor tree) [9] and their variants [12, 27]. More interestingly, lots of studies in machine learning report overwhelming performance of the TN-based methods compared to the conventional TDs [4, 10, 17, 23, 29, 31, 32]. On the other hand, we notice that the existing TN-based methods are mainly developed by specifying the *topology* of the model, a graphical structure used to model TN, such as path, cyclic, tree and grid (*i.e.*, multi-cyclic) graphs. Such a fact raises the question *whether there exist "better or even optimal" topological structures of TN decomposition for a given tensor, and how to efficiently search them?* It is naturally expected that the optimal topology of TN would be out of the box of the manucrafted structures yet with more promising approximation capacity.

In this abstract, we investigate the topology search issue in a practical manner. The main idea is to formulate the problem as combinatorial optimization and to apply the well-developed genetic algorithms (GAs) to search the optimal structures. Specifically, we first reforge the graphical representation of TN [6, 34] to obtain a unique descriptor (*i.e.*, its adjacency matrix) for each

*Our code is available at https://github.com/minogame/icml2020-TNGA.

First Workshop on Quantum Tensor Networks in Machine Learning, 34th Conference on Neural Information Processing Systems (NeurIPS 2020).

topology, and then encode it into a fixed-length binary string. GAs are naturally applied by taking the string as chromosome. More details of this work can be found from our conference paper [20].

2 Problem Setting

We first reforge the existing definition of the tensor network's (TN's) graphical representation. The aim is to establish a bijective mapping from the TN structure to a simple graph. After that, we propose a formal description of the topology search issue of TN decomposition. Due to the page limit, we skip the background introduction of TN, and suggest to read the literature [6, 24] and the references therein for details of the TNs.

2.1 Graphical Representation of Tensor Network

Given an undirected graph $G = (V, E)$ we model a tensor network, by which the vertices V are labeled by different symbols and the edges E are assigned with non-negative integer weights to represent the index contraction [24] operations. Similar to [34], we assume that G is always a simple graph. It implies that there exists no self-loop or multiple edges. Furthermore, we impose additional rules onto the model to guarantee the bijection relationship:

Rule 1 *Weight-one edges are not allowed.*

Since dropping the weight-one edges does not change the expression of TN [34], Rule 1 eliminates the possibility that two different graphs correspond to the same TN structure. Hence, under Rule 1, the correspondence between the TN structure and the graph G becomes one-to-one, which we formalize as

Proposition 1 *Given the number of vertices that are labeled distinctly, there is a bijection from an arbitrary simple graph to the TN structure.*

Due to the fact that the connectivity of the graph is not restricted in the model, we should further define how to calculate TN from multiple isolated subgraphs.

Rule 2 *The isolated subgraphs are "connected" by tensor outer product.*

Rule 2 implies that, if there are multiple isolated subgraphs in TN, we first calculate the sub-blocks defined by each isolated subgraph using index contraction, then merge all sub-blocks by trivial tensor outer product. Last, the work in [34] does not consider the existence of internal cores (*i.e.*, the core tensors whose all indices are connected to other cores), such as the ones in Tucker decomposition and multi-scale entanglement renormalization ansatz (MERA) [7]. Therefore, we further allow the existence of the internal cores (vertices):

Rule 3 *The vertices can be internal.*

2.2 TN Decomposition and Topology Search

Based on the above definitions on TN's graphical representation, below we give a general formulation of TN decomposition. Specifically, TN decomposition is defined as representing a tensor $\mathcal{X}$ by a series of index contraction operations on a collection of core tensors $\mathbb{V} := \{\mathcal{V}_i,\, i \in [N]\}$ under a structure *w.r.t.* graph $G = (V, E)$, and it can be therefore formulated as [21]

$$\mathcal{X} = TN(\mathbb{V}; \mathbf{A}), \tag{1}$$

where $TN(\,\cdot\,; \mathbf{A})$ denotes the index contraction operations over the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. Note that the general model (1) can be degenerated to the conventional decomposition like Tucker, TT, TR decomposition when we specify certain $\mathbf{A}$. Hence it is not difficult to see that there exist many TN decomposition models, which might efficiently represent a tensor yet do not belong to any known TN decomposition model. In this work, *we call the process of seeking the "unknown" TN models as the topology search of TN*, where the notion "topology" represents the graphical structure of a TN decomposition model. Below, we give a formal statement of topology search for TN decomposition.

Problem 1 *Given a tensor $\mathcal{X}$, the topology search of TN is to find the optimal adjacency matrix $\mathbf{A}_0$, such that there exist a set of core tensors $\widehat{\mathbb{V}}$ that satisfies Eq.* (1)*.*

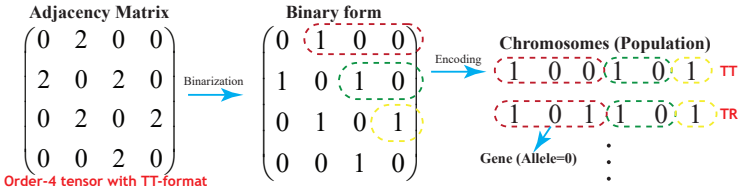

Figure 1: An example to illustrate how to encode an order-4 tensor with TT-format into a binary string *a.k.a* chromosome. First, the adjacency matrix is converted into a binary form; Subsequently, the upper triangle aspect of the matrix is encoded as a binary string. In the context of GA, we generally consider the binary strings as chromosomes or population and their entries as genes.

In different contexts, we can endow the notion "optimal" with different meanings, such as the sparsity of graph or the minimum of the largest degree of vertex. In this abstract, we temporally consider to seek a graphical structure of TN, which can *use the fewest parameters to represent a given tensor*.

3 GA-based Topology Search Method

From Eq. (1) and Problem 1 we see that the topology search of TN is equivalent to seeking the optimal adjacency matrix $\mathbf{A}$ under some criteria. Problem 1 can be therefore formulated as an optimization model:

$$\min_{\mathbf{A}\in\mathbb{A}} \frac{1}{\epsilon\left(\mathbf{A}\right)}, \quad s.t. \left\|\mathcal{X} - TN(\widehat{\mathbb{V}}; \mathbf{A})\right\|_F^2 \leq \delta, \text{ for some } \widehat{\mathbb{V}}, \tag{2}$$

where $\mathbb{A}$ denotes the set that contains all possible adjacency matrices of a given size, $\|\cdot\|_F$ is the Frobenius norm of a tensor, and $\epsilon(\mathbf{A})$ denotes the compression ratio of the TN decomposition under the given $\mathbf{A}$. It is defined as

$$\epsilon(\mathbf{A}) = \frac{\text{Uncompressed size of } \mathcal{X}}{\text{Parameter size of } \mathbb{V} \text{ under } \mathbf{A}}. \tag{3}$$

As shown in Eq. (2), we maximize the compression ratio, under which the tensor $\mathcal{X}$ can be decomposed into a collection of cores $\mathcal{V}_i$, $\forall i$. Compared to Problem 1, we further impose a tuning parameter δ to provide a tolerance of the noise. It is because in machine learning problems, we often demand a good TN approximation rather than the exact decomposition, and the latter can be accomplished by setting $\delta = 0$.

We can also see that Eq. (2) is a combinatorial optimization model when the weights of the edges are bounded. Since the exhaustive search for such problem is unacceptable even for a small-scale problem, we concern a practical alternative, where we are looking for near-optimal solutions that practically well-perform, instead of considering the worst-case behavior of the problem.

To do so, we first encode TN's topology into a fixed-length binary string, and then apply GAs to searching the solution. For simplicity, in the rest, we assume that the number of vertices is fixed, and the weights on edges are known and identical to each other. Then, we can encode the TN topology by the upper triangle part (the diagonal entries are omitted) of adjacency matrices due to its symmetry property. Figure 1 shows an example to encode an order-4 TT into a binary string. Taking the strings as chromosomes, any genetic operators can be applied in the task.

4 Experimental Results on Natural Images

Below, we exploit the real-world data (natural images) to evaluate the approximation capacity of TN decomposition with the discovered topology by GA.

Setup. We randomly select 10 natural images from the LIVE dataset [28]. The images have original sizes of 256×256 and are tensorized to order-8 of the size 4^8. Also we implement TT-SVD [25] and TR-SVD [38] in the experiment for comparison. In their methods, we manually adjust the tolerance value to meet RSE obtained by our methods. In ours, we change the weights to be $\{6, 7\}$ after the topology search to simulate the cases with different approximation errors.

Results. The results are given in Table 1. As shown, our methods result in less number of parameters (higher compression ratio) compared to TT-SVD and TR-SVD in over all data and settings. It implies

Table 1: Experimental results of approximation of image data. We compare the log compression ratio of GA (ours), TT-SVD, TR-SVD under close RSE value. In GA, the RSE between the original and reconstructed images are obtained from the optimal individuals with their weights setting to 6 and 7. A larger log compression ratio indicates fewer parameters of the model while a smaller RSE indicates better approximation quality.

Data	**Log compression ratio (CR)↑ + RSE↓ – *CR(RSE)***					
	GA(weights=6)	**TT**	**TR**	**GA(weights=7)**	**TT**	**TR**
0	0.901(*0.137*)	0.582(*0.142*)	0.469(*0.141*)	0.660(*0.115*)	0.325(*0.115*)	0.457(*0.127*)
1	1.352(*0.158*)	1.210(*0.170*)	1.216(*0.187*)	1.159(*0.155*)	1.137(*0.166*)	0.824(*0.155*)
2	1.452(*0.176*)	1.148(*0.187*)	1.231(*0.206*)	1.268(*0.171*)	0.898(*0.179*)	1.022(*0.182*)
3	1.649(*0.193*)	1.140(*0.191*)	1.416(*0.211*)	1.476(*0.189*)	1.265(*0.206*)	1.074(*0.191*)
4	0.859(*0.152*)	0.527(*0.156*)	0.403(*0.153*)	0.621(*0.121*)	0.408(*0.143*)	0.372(*0.141*)
5	1.726(*0.087*)	1.471(*0.087*)	1.471(*0.088*)	1.548(*0.083*)	1.531(*0.083*)	1.388(*0.085*)
6	1.332(*0.110*)	1.471(*0.113*)	1.212(*0.124*)	1.141(*0.104*)	1.088(*0.101*)	1.052(*0.102*)
7	1.573(*0.126*)	1.030(*0.139*)	1.112(*0.145*)	1.406(*0.120*)	1.179(*0.142*)	0.970(*0.125*)
8	1.679(*0.085*)	1.493(*0.082*)	1.387(*0.085*)	1.505(*0.081*)	1.493(*0.082*)	1.357(*0.084*)
9	1.164(*0.194*)	0.994(*0.227*)	0.836(*0.200*)	0.966(*0.185*)	0.774(*0.190*)	0.916(*0.226*)

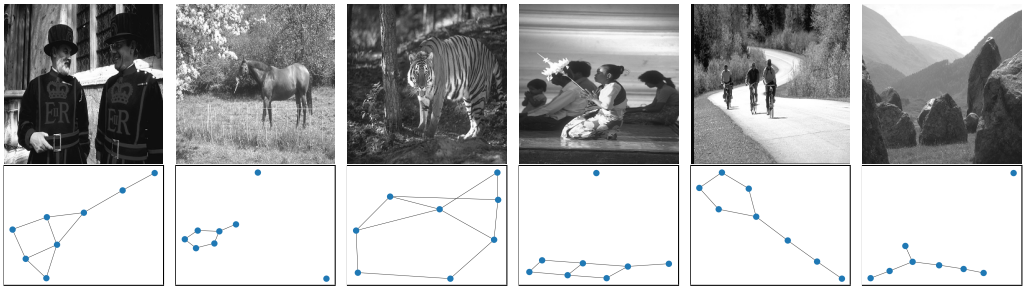

Figure 2: Example to illustrate the employed images and their corresponding TN topological structures obtained by GA.

that the discovered topological structures by GA has stronger express power on TN decomposition than the simple line (TT) and cycle (TR) structures. Figure 2 gives some examples of the topological structures obtained by GA. We can see that the obtained topology shows complex structures, which are generally a combination of lines, cycles and isolated points. Such results are expected because the natural images reflect complicated object relationship and abundant information.

5 Discussion and Concluding Remarks

Computational complexity. Although we have preferred applying the plain GA to the topology search and the convergence is quite fast, the computational requirement is still dramatically larger than TT-SVD and TR-SVD. The bottleneck mainly comes from the gradient-based decomposition algorithm and repeated calculation of the decomposition to obtain the fitness score.

Potential applications. TN has been widely used in the deep learning community [11, 13, 16, 26, 32, 33, 35], but the topological structures used in their methods are assumed to be known and fixed. Hence, we expect that searching more promising structures for TN may further improve the performance in those tasks.

Conclusion. One aim of this work is to highlight the importance of the topology search for TN decomposition and related tasks. We argue based on the empirical results that choosing a suitable topological structure could significantly boost the express power of TN decomposition, and near-optimal structures are sufficient to show their advantages compared to simple ones.

References

[1] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[2] Mohammad Taha Bahadori, Qi Rose Yu, and Yan Liu. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *Advances in Neural Information Processing Systems*, pages 3491–3499, 2014.

[3] Rasmus Bro. Parafac. tutorial and applications. *Chemometrics and Intelligent laboratory Systems*, 38(2):149–171, 1997.

[4] Song Cheng, Lei Wang, Tao Xiang, and Pan Zhang. Tree tensor networks for generative modeling. *arXiv preprint arXiv:1901.02217*, 2019.

[5] Andrzej Cichocki. Era of big data processing: A new approach via tensor networks and tensor decompositions. *arXiv preprint arXiv:1403.2048*, 2014.

[6] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[7] Lukasz Cincio, Jacek Dziarmaga, and Marek M Rams. Multiscale entanglement renormalization ansatz in two dimensions: quantum ising model. *Physical Review Letters*, 100(24):240603, 2008.

[8] Lieven De Lathauwer. Decompositions of a higher-order tensor in block terms—part I: Lemmas for partitioned matrices. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1022–1032, 2008.

[9] Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[10] 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*, pages 5553–5563, 2019.

[11] Ming Hou, Jiajia Tang, Jianhai Zhang, Wanzeng Kong, and Qibin Zhao. Deep multimodal multilinear fusion with high-order polynomial pooling. In *Advances in Neural Information Processing Systems*, pages 12113–12122, 2019.

[12] Huyan Huang, Yipeng Liu, and Ce Zhu. Tensor grid decomposition with application to tensor completion. *arXiv preprint arXiv:1903.04735*, 2019.

[13] Zihao Huang, Chao Li, Feng Duan, and Qibin Zhao. H-owan: Multi-distorted image restoration with tensor 1x1 convolution. *arXiv preprint arXiv:2001.10853*, 2020.

[14] Boris N Khoromskij. *Tensor numerical methods in scientific computing*, volume 19. Walter de Gruyter GmbH & Co KG, 2018.

[15] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[16] Maxim Kuznetsov, Daniil Polykovskiy, Dmitry P Vetrov, and Alex Zhebrak. A prior of a googol gaussians: a tensor ring induced prior for generative models. In *Advances in Neural Information Processing Systems*, pages 4104–4114, 2019.

[17] Yoav Levine, David Yakira, Nadav Cohen, and Amnon Shashua. Deep learning and quantum entanglement: Fundamental connections with implications to network design. *arXiv preprint arXiv:1704.01552*, 2017.

[18] Binghua Li, Chao Li, Feng Duan, Ning Zheng, and Qibin Zhao. TPFN: Applying outer product along time to multimodal sentiment analysis fusion on incomplete data. In *Proceedings of the European Conference on Computer Vision*, 2020.

[19] Chao Li, Mohammad Emtiyaz Khan, Zhun Sun, Gang Niu, Bo Han, Shengli Xie, and Qibin Zhao. Beyond unfolding: Exact recovery of latent convex tensor decomposition under reshuffling. *arXiv preprint arXiv:1805.08465*, 2018.

[20] Chao Li and Sun Sun. Evolutionary topology search for tensor network decomposition. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[21] Chao Li, Zhun Sun, Jinshi Yu, Ming Hou, and Qibin Zhao. Low-rank embedding of kernels in convolutional neural networks under random shuffling. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3022–3026. IEEE, 2019.

[22] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2247–2256, 2018.

[23] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems*, pages 442–450, 2015.

[24] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[25] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[26] Yu Pan, Jing Xu, Maolin Wang, Jinmian Ye, Fei Wang, Kun Bai, and Zenglin Xu. Compressing recurrent neural networks with tensor ring for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4683–4690, 2019.

[27] Anh-Huy Phan, Andrzej Cichocki, Ivan Oseledets, Salman Ahmadi Asl, Giuseppe Calvi, and Danilo Mandic. Tensor networks for latent variable analysis: Higher order canonical polyadic decomposition. *arXiv preprint arXiv:1809.00535*, 2018.

[28] Hamid R Sheikh, Muhammad F Sabir, and Alan C Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451, 2006.

[29] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.

[30] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[31] Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5697–5705, 2017.

[32] Wenqi Wang, Yifan Sun, Brian Eriksson, Wenlin Wang, and Vaneet Aggarwal. Wide compression: Tensor ring nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9329–9338, 2018.

[33] Yinchong Yang, Denis Krompass, and Volker Tresp. Tensor-train recurrent neural networks for video classification. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3891–3900. JMLR. org, 2017.

[34] Ke Ye and Lek-Heng Lim. Tensor network ranks. *arXiv preprint arXiv:1801.02662*, 2019.

[35] Rose Yu, Stephan Zheng, Anima Anandkumar, and Yisong Yue. Long-term forecasting using higher order tensor rnns. *arXiv preprint arXiv:1711.00073*, 2017.

[36] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1114, 2017.

[37] Hongyang Zhang, Vatsal Sharan, Moses Charikar, and Yingyu Liang. Recovery guarantees for quadratic tensors with sparse observations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3322–3332, 2019.

[38] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.