Tensor, Tensor Networks, Quantum Tensor Networks in Machine Learning: An Hourglass Architecture

Anonymous Author(s)
Affiliation
Address
email

Abstract

Quantum tensor networks in machine learning (QTNML) are envisioned to have great potential to advance AI technologies. Recent works show that (quantum) tensor networks provide powerful simulations of quantum machine learning algorithms on classical computers. We observe that tensor, tensor networks and quantum tensor networks in machine learning exhibit a layered architecture that resembles an hourglass. In this paper, we describe a seven-layer architecture to characterize the role of tensor, tensor networks and quantum tensor networks in machine learning, point out current challenges and discuss recent innovations. As a cornerstone data structure, tensor and tensor networks lie at the waist of the hourglass-shaped architecture, while the lower and upper layers tend to see frequent innovations. We expect tensor, tensor networks and quantum tensor networks continue to serve as an *amplifier* for computational intelligence, a *transformer* for machine learning innovations, and a *propeller* for AI industrialization.

1 Introduction

Why do conventional machine learning algorithms use vectors and matrices, while deep learning algorithms and neural networks mostly rely on tensors? A simple and direct answer is that deep learning usually involves hundreds, if not thousands, of features.

Quantum tensor networks [6] in machine learning (QTNML) are envisioned to have great potential to advance AI technologies. Quantum machine learning [7] promises quantum advantages (potentially exponential speedups in training, quadratic speedup in convergence, etc.) over classical machine learning, while (quantum) tensor networks provide powerful simulations of quantum machine learning algorithms on classical computers. QTNML is now a rapidly growing interdisciplinary area.

(Quantum) Tensor networks[1], a contracted network of factor tensors, have arisen independently in several areas of science and engineering. Such networks appear in the description of physical processes and an accompanying collection of numerical techniques have elevated the use of quantum tensor networks into a variational model of machine learning. Underlying these algorithms is the compression of high-dimensional data needed to represent quantum states of matter. These compression techniques have recently proven ripe to apply to many traditional problems faced in deep learning. (Quantum) Tensor networks have shown significant power in compactly representing deep neural networks [55], and efficient training and theoretical understanding of deep neural networks. More potential tensor network technologies are rapidly emerging, such as approximating probability

[1] A major difference of quantum tensor networks from tensor networks would be the contractions are required to be unitary transforms.

Submitted to First Workshop on Quantum Tensor Networks in Machine Learning, 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

functions and probabilistic graphical models [67, 22]. A merger of (quantum) tensor network algorithms with state-of-the-art approaches in deep learning is now taking place.

Quantum algorithms are typically described by quantum circuits (quantum computational networks). These networks are indeed a class of tensor networks, creating an evident interplay between classical tensor network contraction algorithms and executing tensor contractions on quantum processors. The modern field of quantum enhanced machine learning has started to utilize several tools from tensor network theory to create new quantum models of machine learning and to better understand existing ones.

The interplay between tensor networks, machine learning and quantum algorithms is rich. Indeed, this interplay is based not just on numerical methods but on the equivalence of tensor networks to various quantum circuits, rapidly developing algorithms from the mathematics and physics communities for optimizing and transforming tensor networks, and connections to low-rank methods for learning. A merger of tensor network algorithms with state-of-the-art approaches in deep learning is now taking place. A new community on quantum tensor networks in machine learning (QTNML) is forming, which this workshop aims to foster.

The three wagons (driving forces) for the success of machine learning are as follows

- Big data (represented as data structures of tensor and tensor networks): the past decade witnesses an exponential explosion of sensory data due to the great advances in sensor manufacturing, leading to the debate of "*More is less?* or *More is more!*" [5]. As a cornerstone data structure, tensor and tensor networks are powerful in representing unstructured, multi-modal data and are envisioned to have great potentials to promote the development and deployment of machine learning technologies.

- *Machine learning algorithms and models*: Treating datasets as past experiences, algorithms instruct machines on what they should do. Machine learning algorithms and models allow computers to learn on their own. On the other hand, deep reinforcement learning algorithms train themselves through interactions with an unknown environment.

- *Intelligent computing*. Faster computers can process more data and play a critical role in future AI advancements. Deep learning [39] are computational models with multiple processing layers that learn representations of data with multiple levels of abstraction. In the post Moor's law era [69], the rise of deep learning [39] can be largely credited to a new paradigm *Intelligent computing for computational intelligence!* E.g., Google has designed and built TPU (Tensor Processing Units) specifically for machine learning and claims to be $15\times$ faster than a GPU (graphics processing units), while IBM and Google are both developing quantum computing systems.

We observe that tensor, tensor networks and quantum tensor networks in machine learning exhibit a layered architecture that resembles an hourglass. Such an observation is analogy to the hourglass structure [4] of the Internet protocol stack (known as TCP/IP) that successfully provides end-to-end data communication by specifying how data should be packetized, addressed, transmitted, routed, and received.

In this paper, we attempt to initiate a layered architecture for tensor and tensor networks, which will benefit the development of machine learning theory, AI chip manufacturing, and AI applications. This seven-layer architecture resembles an hourglass, namely, tensor, tensor networks and quantum tensor networks lie at the waist while the lower and upper layers tend to see frequent innovations. The bottom layer is the hardware, the highest layer is the AI applications and products. We point out current challenges and discuss recent innovations.

We advocate intelligent computing of quantum tensor networks to achieve computational intelligence. Quantum tensor networks provide a unified full-stack architecture to explore and fulfill the potential of quantum machine learning and general artificial intelligence (AI) on both classical computer and quantum computing platforms, which is envisioned to advance one-step ahead of the current deep learning driven AI. Actually, there is plenty of room at the bottom [19], in the middle, and at the top [40]. In the past, we have already saw $30\times \sim 100\times$ speedups in basic operations and $300\times \sim 1,000\times$ speedups in multiple AI tasks, such as image and video processing/recognition, autonomous driving, Internet of Things, gene data analysis, quantitative finance, etc.

Such an hourglass-shaped layer architecture enjoys disciplinary advantages, including layer-wise standardization, intra-layer modularity and inter-layer separability. The *layer-wise standardization* encourages an eco-system for machine learning research and industrialization. With the *intra-layer modularity*, one can update a functional module without interfering other modules. The *inter-layer separability* means that the lower layer is transparent to the upper layer that calls the APIs provided by the lower layer. We expect tensor and tensor networks continue to serve as an *amplifier* for computational intelligence, a *transformer* for machine learning innovations, and a *propeller* for AI industrialization.

We aim to promote discussions (by a series of workshops and academic events) among researchers investigating innovative tensor network technologies from perspectives of fundamental theory and algorithms, novel approaches in machine learning and deep neural networks, and various applications in computer vision, biomedical image processing, natural language processing, and many other related fields. In this survey, we also provide pointers to key references, hardware, software for beginners.

The remainder of this paper is organized as follows. Section 2 describes the proposed hourglass architecture. Section 3 discussed key challenges, recent innovations and broad impacts. We conclude this paper in Section 4.

2 The Proposed Hourglass Architecture

We propose a seven-layer architecture for tensor, tensor networks and quantum tensor networks, which resembles an hourglass. Quantum tensor networks provides a unified framework where both classical and quantum computing share the same theoretical and algorithmic developments, and the same model can be trained classically then transferred to a quantum processor. A take-home message would be: there is plenty of room at the bottom [19], in the middle, and at the top [40].

2.1 Layer 1: X Processing Unit

In the post Moor's law era [69], the rise of deep learning [39] can be largely credited to a new paradigm *Intelligent computing for computational intelligence!* The impetus to AI computation is made-for-AI processors, such as GPUs, FPGAs, ASICs (NPUs), quantum circuits/processors, etc. We call them *XPUs*.

There is an emergence of dedicated AI accelerator using the ASIC (Application Specific Integrated Circuit) technology, called *NPU* (neural processing unit). Of particular interest are tensor-based NPUs, including Google TPU (tensor processing unit) [29], tensor cores in NVIDIA Volta/Turing Architecture, Intel Nervana neural network processors (NNP), Tensor Computing Processor BM1684, Alibaba Ali-NPU, Knupath Hermosa, Baidu XPU [57], the Huawei Ascend 910 using 32 DaVinci AI cores [44], etc.

The computing power of quantum circuits/processors lies in an optimistic believe of "quantum supremacy" [23], which is a key milestone when certain computational tasks might be executed exponentially faster on a quantum computer than on a classical computer. A recent breakthrough would be Google's Sycamore processor with 53 qubits. Interested readers may refer to [2] for principles of quantum computing and [1] for discussions of its limits.

Note that quantum circuits can be directly expressed as quantum tensor networks [53].

2.2 Layer 2: BLAS and Automatic Tensor Differentiation

To fully utilize the computing power of XPUs in Layer 1, *BLAS* (Basic Linear Algebra Subprograms, or Basic Tensor Algebra Subroutines *BTAS*) and *AutoDiff* (Automatic differentiation) [58] are "a knife and fork" to support effective implementation of machine learning models at the top.

Such BLAS standards are implemented and optimized in different programming languages. For example, numpy in Python, cuBLAS in CUDA.

- BLAS level 1 (1969): "vector-vector";
- BLAS level 2 (1972): "matrix-vector";

- BLAS level 3 (1980): " matrix-matrix";
- BLAS level 4 (Now?), "tensor-tensor": tensor operations include tensor (Kronecker) product, Khatri-Rao product, Hadamard product, tensor contraction, t-product [33] or $\mathcal{L}$-product [47], etc.

Automatic differentiation is a technique to numerically evaluate the derivative of a function, which is believed to be very powerful when combining the back-propagation algorithm. Interested readers may refer to AutoDiff [58] and DDSP (differentiable digital signal processing) [16], etc.

Of particular interest is the automatic differentiation for tensors [38] and (quantum) tensor networks [43]. Experimental results [38] show a speedup of up to two orders of magnitude over state-of-the-art frameworks when evaluating higher order derivatives on CPUs and a speedup of about three orders of magnitude on GPUs. H.-J. Liao *et al* [43] advocates that differential programming of tensor networks open the door to many innovations and applications.

2.3 Layer 3: Tensor Data Structure

Tensor is the most popular data structure in machine learning, especially in deep learning. For instance, a) input data: color image set, video sequence, MRI/fMRI, EEG, gene expression, traffic data, social network data, knowledge graph; b) High-order statistical information, high-order moment, covariance, cumulant, etc.; c) model parameters: fully connected layer, convolutional layer, multi-task weight parameters, multi-modal feature fusion, and etc.; and d) function: probability mass function of multiple discrete variables.

In the past, a unified notation set for tensors [34], tensor networks [10] and quantum tensor networks [7][6] successfully helps the adoption of tensor tools and the development of tensor network libraries in machine learning.

From a machine learning perspective, an N-th order[2] *tensor* is a container that can house N-dimensional data and associates with linear/multi-linear operations. A scalar is 0-dimensional, a vector has a single dimension (1D), a matrix has two dimensions (2D), and a higher-order tensor has more than two dimensions.

From a spectral (or transform) perspective, tubal-scalars [33][32][47] are vectors with the multiplication operation defined according to the convolution theorem. Considering a graph transform, one can have graph-tensors [52] or connected matrices [68], and graph tensor neural networks [49].

2.4 Layer 4: Tensor Decompositions, Tensor Networks and Quantum Tensor Networks

Many practically useful and efficient tensor models are built upon tensor decompositions, tensor networks and quantum tensor networks.

Tensor Decompositions: Canonical Polyadic (CP) tensor decomposition, Tucker tensor decomposition, TT [56] or TR [76] tensor decomposition, HT, tSVD, reshuffling TD. Sparse tensor decomposition and nonnegative tensor decompositions are also developed as extensions of CP, Tucker, TT, TR and HT.

The uniqueness of CP tensor decompositions [11] indicates that multilinear algebra may have theoretical advantages over bilinear and linear algebra.

Other important applications includes tensor completion [66, 45]. tensor time series [62, 50], spectral learning on matrix/tensor [26], and data privacy [35, 20, 18].

Tensor Networks and quantum tensor networks: TNs show advantages mostly in space complexity reduction and computation efficiency. Tensor Networks have been employed to a) large-scale optimization problems, large-scale eigenvalue problem, large-scale SVD, large-scale matrix pseudo-inverse; b) model compression in DNN, including fully connected layer and convolutional layer; c) expressive power analysis of DNN,

[2]In quantum physics, *rank* corresponds to *order*, while *bond dimension* corresponds to *rank*.

2.5 Layer 5: Tensor Libraries and Programming IDEs

Widely used tensor IDEs are TensorFlow [3], PyTorch [59], TensorRT [70], Theano, Keras, Apache MXNet, Caffe2, CNTK, PaddlePaddle, MindSpore, MegEngine, etc.

Other libraries include TensorLayer [15], TensorLy [37]; Tensor decomposition in TensorFlow [54], sparse tensor computing [60], and differentiating tensor networks library [43]

For quantum physics, iTensor (Intelligent Tensor) [3] provides a collection of optimized tensor network algorithms. TensorTrace [17] is an application designed to facilitate the implementation of tensor network algorithms and provides a drag-and-drop interface for building tensor networks.

Google's TensorNetwork [61] is a library for both physics and machine learning, which is wrapper for TensorFlow, JAX, PyTorch, and Numpy.

2.6 Layer 6: Machine Learning Algorithms and Models

There are active research on designing tensor-based machine learning models. We describe a few approaches in the following.

TensorFace [71][72] presents facial image ensembles, where the relevant factors include different faces, expressions, viewpoints, and illuminations. TensorMask [9] is proposed for dense object segmentation.

Tensor regression [36] extends the conventional regression models to tensor representation, while tensor mixture model [64] proposed a probabilistic graphic model in tensor form.

AutoEncoder can be extended to tensor form, such as tensor sparse coding [28].

The generative adversarial network framework is extended to tensor GAN [48] with application to real-time indoor localization for smartphones.

In the model-based direction, tensor neural networks are proposed by unfolding tensor algorithms into deep neural networks, e.g., [51][21] design fast decoders for snapshot compressive imaging cameras, [49] considered recovery of nodes' data matrices, and [75] investigated the video synthesis problem.

Many complicated TN models including DMRG, TTN, MERA, MERA 2D, PEPS, and etc, which have widely applied to machine learning but may have potential advantages in particular problems.

2.7 Layer 7: Applications and Products

Many products embracing AI is enjoying a booming market, penetrating our daily lives: from smartphones to self-driving cars and robotics, search engines, typing assistants (auto-completion), to healthcare services.

Compressing and optimizing neural networks for inference at mobile devices: (i) TVM (tensor virtual machine) [8]; (ii) the Tensor Algebra Compiler (taco) is a C++ library that computes tensor algebra expressions on sparse and dense tensors. It uses novel compiler techniques to get performance competitive with hand-optimized kernels in widely used libraries for both sparse tensor algebra and sparse linear algebra.

AutoML and neural architecture search (NAS) are promising, where the training and inference are performed at cloud servers. Many applications are now successfully deployed, including speech recognition, visual object recognition, object detection; others: drug discovery and genomics. Note that health-care is one of the hottest trends, while agriculture applications may have broad social impacts, including automatic quality check, mineral delivery optimization in hydroponics. Disaster recovery is also a critical application.

Big data analysis [65] for image, video; sensory data processing; EEG brain data; finance, genetics, etc.

AI is now being applied massively in entertainment industry, such as chess and poker, medias (e.g. Netflix), music industry (IBM Watson), and online games, etc.

[3] iTensor: `https://itensor.org/index.html`

Other AI products that benefits tensor network algorithms are listed as follows:

- reCAPTCHA is a CAPTCHA-like system designed to establish that a computer user is human.
- SIRI is one of many voice assitants available today.
- Gmail recently introduced autocomplet tools.
- Plagiarism checking by searching for matches in billions of documents.
- FaceID is a feature recently intoduced by Apple for authentication on iPhone.
- Recommendation systems in Amazon and Alibaba Taobao that suggests suers other products based on their preferences and click history.
- Facebook face detection and tagging is a services of Facebook which automatically detects faces in images and tags people from the user friendship set.

3 Challenges, Innovations and Broad Impacts

3.1 Challenges

The “4V+P" challenge of big data: IBM data scientists break big data into four dimensions [14]: volume for scale of data, variety for different forms of data, velocity for analysis streaming data, and veracity for uncertainty of data. We would like to advocate the privacy-preserving requirement as a plus aspect of tensor learning algorithms. Furthermore, the data acquisition process is expensive in terms of either time or budget.

The C^3-challenges of machine learning algorithms are the intertwined computing, caching and communication:

- *Computing*: Training a model requires substantial amount of time, which in turn slows down the development. How do we speed up machine learning by $100\times$? Real-time operations requires fast inference, e.g., cuTensor in NVIDIA CUDA.
- *Caching*: How to support Billion/Trillion-scale tensor computing? How to compress neural network for mobile platforms?
- *Communication*: the bandwidth between CPU and GPU, the link capacity of data centers, the communication between cloud and edge servers.

Quantitatively characterizing the efficiency-accuracy tradeoff: how to select tensor network models for different neural networks? How to tune the hyperparameters in the tensor network model?

Trustworthy AI: Explainability, interpretability, and understandable. Interpretability is about the extent to which a cause and effect can be observed within a system. Explainability (for decision making), meanwhile, is the extent to which the internal mechanics of a machine or deep learning system can be explained in human terms.

Understanding neural-intelligence: a two-layer feedforward network [27] is analyzed using CP tensor decomposition and such a network is believed to learn a mapping between data distribution priors and labels. On the other hand, an elementary function of neural net's intelligence is to recognize symmetry structures in the data [63]: *The glove for the left hand is able to fit the right hand if we turn it inside out like placing an imaginary mirror near the opening. Analogously, neural networks play a similar role as a glove when dealing with inputs of symmetry structures*. The classic Kruskal uniqueness theorem is exploited to provide a sufficient condition for the situations where such a generalization capability will hold.

Tensor networks provide a rigorous approach to investigate *Why deep is good?* Nadav [12] considered sum-product networks and CNN with ReLU activation functions [13]. Khrulkowv [31][30] took a similar approach to analyze RNNs.

Robustness of Machine Learning Models: deep adversarial learning; The notion of differential privacy is believed to be very power to construct ensemble methods that fuse sub-networks into a more robust one [42].

3.2 Innovations

One recent trend regarding both AI software and hardware is to consider inference and training as two separate different phases with different computational approaches. It is becoming standard to develop specific chips for training and specific chips for inference.

Cross-layer Codesign. High performance tensor learning operations by exploiting the massive parallelisms are important for both training and inference: 1). Tensor decompositions on GPUs/FPGA such as cuTensor library [74][46][24][25] and swTensor [77]; 2). Tensor completion [73].

Federated learning [35] or **Privacy-preserving tensor algorithms**; homomorphic encryption methods for tensor decompositions.

Quantum Machine Learning [7]: tensor networks provide powerful simulations of quantum machine learning algorithms on classical computers, which may promise quantum advantages, such as potentially exponential speedups in training, quadratic speedup in convergence, etc. [41]

Tensor network learning vs deep learning: TN has the power to express functions, will tensor network learning be used as a general machine learning model like deep learning?

4 Conclusion

Tensor, tensor networks and quantum tensor networks are envisioned to have great potentials to promote the development and deployment of machine learning technologies. In this paper, we have proposed a seven-layer architecture to characterize the role of tensor and tensor networks in machine learning, point out current challenges and discuss the development trends. Such a layered architecture resembles an hourglass. As a cornerstone data structure, tensor and tensor networks lie at the waist of the hourglass, while the lower and upper layers tend to see frequent innovations. We expect tensor and tensor networks continue to serve as a *transformer* for machine learning innovations, an *amplifier* for computational intelligence, and a *propeller* for AI industrialization.

The interplay between tensor networks and machine learning algorithms is rich. Indeed, this interplay is based not just on numerical methods but on the equivalence of tensor networks to various arithmetic circuits, rapidly developing algorithms from the mathematics and physics communities for optimizing and transforming tensor networks, and connections to low-rank methods for learning. A merger of tensor network algorithms with state-of-the-art approaches in deep learning is now taking place. A new community is forming, which this workshop aims to foster.

References

[1] Scott Aaronson. The limits of quantum. *Scientific American*, 298(3):62–69, 2008.

[2] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In *32nd Computational Complexity Conference (CCC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[3] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 265–283, 2016.

[4] Saamer Akhshabi and Constantine Dovrolis. The evolution of layered protocol stacks leads to an hourglass-shaped architecture. In *Proceedings of the ACM SIGCOMM*, pages 206–217, 2011.

[5] Richard G Baraniuk. More is less: signal processing and the data deluge. *Science*, 331(6018):717–719, 2011.

[6] Jacob Biamonte. Lectures on quantum tensor networks. *arXiv preprint arXiv:1912.10049*, 2019.

[7] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[8] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. TVM: An automated end-to-end optimizing compiler for deep learning. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 578–594, 2018.

[9] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. TensorMask: A foundation for dense object segmentation. In *IEEE International Conference on Computer Vision*, pages 2061–2069, 2019.

[10] 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.

[11] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

[12] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016.

[13] Nadav Cohen and Amnon Shashua. Convolutional rectifier networks as generalized tensor decompositions. In *International Conference on Machine Learning*, pages 955–963, 2016.

[14] IBM Big Data and Analytics Hub. The four v's of big data. *IBM,[Online]. Available: http://www.ibmbigdatahub.com/infographic/four-vs-big-data*, 2013.

[15] Hao Dong, Akara Supratak, Luo Mai, Fangde Liu, Axel Oehmichen, Simiao Yu, and Yike Guo. TensorLayer: A versatile library for efficient deep learning development. *ACM Multimedia*.

[16] Jesse Engel, Lamtharn (Hanoi) Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. In *International Conference on Learning Representations*, 2020.

[17] Glen Evenbly. TensorTrace: an application to contract tensor networks. *arXiv preprint arXiv:1911.02558*, 2019.

[18] Jun Feng, Laurence Tianruo Yang, Ronghao Zhang, and Benard Safari Gavuna. Privacy preserving tucker train decomposition over blockchain-based encrypted industrial iot data. *IEEE Transactions on Industrial Informatics*, 2020.

[19] Richard P Feynman. There's plenty of room at the bottom. *California Institute of Technology, Engineering and Science magazine*, 1960.

[20] Cai Fu, Zhao Yang, Xiao-Yang Liu, Jia Yang, Anwar Walid, and Laurence T Yang. Secure tensor decomposition for heterogeneous multimedia data in cloud computing. *IEEE Transactions on Computational Social Systems*, 7(1):247–260, 2020.

[21] Xiaochen Han, Bo Wu, Zheng Shou, Xiao-Yang Liu, Yimeng Zhang Zhang, and Linghe Kong. Tensor fista-net for real-time snapshot compressive imaging. In *AAAI*, 2020.

[22] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Physical Review X*, 8(3):031012, 2018.

[23] Aram W Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203–209, 2017.

[24] Hao Hong, Tao Zhang, and Xiao-Yang Liu. cuTensor-TT/TR: High performance third-order tensor-train and -ring decompositions on gpus. In *IJCAI 2020 Workshop on Tensor Network Represenations in Machine Learning*, 2020.

[25] Hao Huang, Tao Zhang, and Xiao-Yang Liu. cuTensor-HT: High performance third-order hierarchical tucker tensor decomposition on gpus. In *IJCAI 2020 Workshop on Tensor Network Represenations in Machine Learning*, 2020.

[26] Majid Janzamin, Rong Ge, Jean Kossaifi, Anima Anandkumar, et al. Spectral learning on matrices and tensors. *Foundations and Trends® in Machine Learning*, 12(5-6):393–536, 2019.

[27] Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

[28] Fei Jiang, Xiao-Yang Liu, Hongtao Lu, and Ruimin Shen. Efficient multi-dimensional tensor sparse coding using t-linear combination. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[29] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12, 2017.

[30] Valentin Khrulkov, Oleksii Hrinchuk, and Ivan Oseledets. Generalized tensor models for recurrent neural networks. *ICLR*, 2019.

[31] Valentin Khrulkov, Alexander Novikov, and Ivan Oseledets. Expressive power of recurrent neural networks. *ICLR*, 2018.

[32] Misha E Kilmer, Karen Braman, Ning Hao, and Randy C Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[33] Misha E Kilmer and Carla D Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

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

[35] Linghe Kong, Xiao-Yang Liu, Hao Sheng, Peng Zeng, and Guihai Chen. Federated tensor mining for secure industrial internet of things. *IEEE Transactions on Industrial Informatics*, 2019.

[36] Jean Kossaifi, Zachary C Lipton, Aran Khanna, Tommaso Furlanello, and Anima Anandkumar. Tensor regression networks. *arXiv preprint arXiv:1707.08308*, 2017.

[37] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. TensorLy: Tensor learning in python. *The Journal of Machine Learning Research*, 20(1):925–930, 2019.

[38] Sören Laue, Matthias Mitterreiter, and Joachim Giesen. Computing higher order derivatives of matrix and tensor expressions. In *Advances in Neural Information Processing Systems*, pages 2750–2759, 2018.

[39] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[40] Charles E Leiserson, Neil C Thompson, Joel S Emer, Bradley C Kuszmaul, Butler W Lampson, Daniel Sanchez, and Tao B Schardl. There's plenty of room at the top: What will drive computer performance after moore's law? *Science*, 368(6495), 2020.

[41] Yoav Levine, David Yakira, Nadav Cohen, and Amnon Shashua. Deep learning and quantum entanglement: Fundamental connections with implications to network design. *ICLR*, 2018.

[42] Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, and Xiao-Yang Liu. Dp-lstm: A differential privacy framework for stock prediction based on financial news. *NeurIPS 2019 Workshop on Robust AI in Financial Services: Data, Fairness, Explainability, Trustworthiness, and Privacy (Robust AI in FS)*, 2019.

[43] Hai-Jun Liao, Jin-Guo Liu, Lei Wang, and Tao Xiang. Differentiable programming tensor networks. *Physical Review X*, 9(3):031041, 2019.

[44] Heng Liao, Jiajin Tu, Jing Xia, and Xiping Zhou. Davinci: A scalable architecture for neural network computing. In *IEEE Hot Chips 31 Symposium (HCS)*, pages 1–44. IEEE, 2019.

[45] Xiao-Yang Liu, Shuchin Aeron, Vaneet Aggarwal, and Xiaodong Wang. Low-tubal-rank tensor completion using alternating minimization. *IEEE Transactions on Information Theory*, 66(3):1714–1737, 2019.

[46] Xiao-Yang Liu, Han Lu, and Tao Zhang. cuTensor-CP: High performance third-order cp tensor decompositions on gpus. In *IJCAI 2020 Workshop on Tensor Network Represenations in Machine Learning*, 2020.

[47] Xiao-Yang Liu and Xiaodong Wang. Fourth-order tensors with multidimensional discrete transforms. *arXiv preprint arXiv:1705.01576*, 2017.

[48] Xiao-Yang Liu and Xiaodong Wang. Real-time indoor localization for smartphones using tensor-generative adversarial nets. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[49] Xiao-Yang Liu and Ming Zhu. Convolutional graph-tensor net for graph data completion. In *IJCAI 2020 Workshop on Tensor Network Represenations in Machine Learning*, 2020.

[50] Weijun Lu, Xiao-Yang Liu, Qingwei Wu, Yue Sun, and Anwar Walid. Transform-based multilinear dynamical system for tensor time series analysis. *NeurIPS Workshop on Modeling and Decision-Making in the Spatiotemporal Domain*, 2018.

[51] Jiawei Ma, Xiao-Yang Liu, Zheng Shou, and Xin Yuan. Deep tensor admm-net for snapshot compressive imaging. In *IEEE ICCV*, pages 10223–10232, 2019.

[52] Osman Asif Malik, Shashanka Ubaru, Lior Horesh, Misha E Kilmer, and Haim Avron. Tensor graph neural networks for learning on time varying graphs. *NeurIPS Workshop on Graph Representation Learning*, 2019.

[53] Igor L Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3):963–981, 2008.

[54] Alexander Novikov, Pavel Izmailov, Valentin Khrulkov, Michael Figurnov, and Ivan Oseledets. Tensor train decomposition on TensorFlow (t3f). *Journal of Machine Learning Research*, 21(30):1–7, 2020.

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

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

[57] Jian Ouyang. XPU: A programmable fpga accelerator for diverse workloads. In *IEEE Hot Chips 29 Symposium*, 2017.

[58] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. *NeurIPS Autodiff Workshop*.

[59] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[60] Eric T Phipps and Tamara G Kolda. Software for sparse tensor decomposition on emerging computing architectures. *SIAM Journal on Scientific Computing*, 41(3):C269–C290, 2019.

[61] Chase Roberts, Ashley Milsted, Martin Ganahl, Adam Zalcman, Bruce Fontaine, Yijian Zou, Jack Hidary, Guifre Vidal, and Stefan Leichenauer. Tensornetwork: A library for physics and machine learning. *arXiv preprint arXiv:1905.01330*, 2019.

[62] Mark Rogers, Lei Li, and Stuart J Russell. Multilinear dynamical systems for tensor time series. In *Advances in Neural Information Processing Systems*, pages 2634–2642, 2013.

[63] Chen Shang and Xiao-Yang Liu. Neural networks' capability of recognizing symmetry structures: A tensor perspective. *IEEE MIT Undergraduate Research Technology Conference*, 2019.

[64] Or Sharir, Ronen Tamari, Nadav Cohen, and Amnon Shashua. Tensorial mixture models. *arXiv preprint arXiv:1610.04167*, 2016.

[65] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

[66] Qingquan Song, Hancheng Ge, James Caverlee, and Xia Hu. Tensor completion algorithms in big data analytics. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(1):1–48, 2019.

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

[68] Qingyun Sun, Mengyuan Yan, David Donoho, et al. Convolutional imputation of matrix networks. In *International Conference on Machine Learning*, pages 4818–4827, 2018.

[69] Thomas N Theis and H-S Philip Wong. The end of Moore's law: A new beginning for information technology. *Computing in Science & Engineering*, 19(2):41–50, 2017.

[70] Han Vanholder. Efficient inference with TensorRT, 2016.

[71] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision*, pages 447–460. Springer, 2002.

[72] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–93. IEEE, 2003.

[73] Tao Zhang, Xiao-Yang Liu, and Xiaodong Wang. High performance GPU tensor completion with tubal-sampling pattern. *IEEE Transactions on Parallel and Distributed Systems*, 2020.

[74] Tao Zhang, Xiao-Yang Liu, Xiaodong Wang, and Anwar Walid. cuTensor-tubal: Efficient primitives for tubal-rank tensor learning operations on GPUs. *IEEE Transactions on Parallel and Distributed Systems*, 2019.

[75] Yimeng Zhang, Xiao-Yang Liu, Bo Wu, and Anwar Walid. Video synthesis via transform-based tensor neural network. *ACM Multimedia*, 2020.

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

[77] Xiaogang Zhong, Hailong Yang, Zhongzhi Luan, Lin Gan, Guangwen Yang, and Depei Qian. swtensor: accelerating tensor decomposition on sunway architecture. *CCF Transactions on High Performance Computing*, 1(3-4):161–176, 2019.