Tangent-space gradient optimization: an efficient update scheme for tensor network machine learning and beyond

Zheng-Zhi Sun
School of Physical Sciences
University of Chinese Academy of Sciences
P. O. Box 4588, Beijing 100049, China
sunzhengzhi16@mails.ucas.edu.cn

Shi-Ju Ran*
Department of Physics
Capital Normal University
Beijing 100048, China
sjran@cnu.edu.cn

Gang Su†
Kavli Institute for Theoretical Sciences,
and CAS Center for Excellence in Topological Quantum Computation
University of Chinese Academy of Sciences
Beijing 100190, China
School of Physical Sciences
University of Chinese Academy of Sciences
P. O. Box 4588, Beijing 100049, China
gsu@ucas.ac.cn

Abstract

The gradient vanishing and exploding problems have seriously affected the effectiveness of gradient-based optimization method of deep machine learning models including tensor network (TN). In this extended abstract, we explain the tangent-space gradient optimization (TSGO) that can effectively avoid the gradient vanishing and exploding problems [Phys. Rev. E 102, 012152 (2020)]. The central idea is to keep the gradient vector on the tangent space of the variational parameter vector. Then the optimization process is performed by rotating the parameter vector angle θ. As θ is naturally restricted in $(0, \pi/2)$, a simple strategy where one gradually decreases the rotation angle can be adopted. TSGO has been compared with the off-the-shelf Adam and show better convergence on the generative TN models. We also compare TSGO and Adam combined with weight normalization method on Bayesian networks, showing the superiority of TSGO.

1 Introduction

The gradient based optimization of model involving deep network structure suffers from the well-known gradient vanishing and exploding problems [1]. The vanishing and exploding of the gradients make the optimization inefficient or unstable. To avoid such problems, many stochastic gradient-based optimization methods that adaptively determine the learning rate are proposed. The commonly used algorithms include stochastic gradient descent, root-mean-square propagation, adaptive learning rate method, and adaptive moment estimation (Adam) [2]. However the performance of these methods

*Corresponding Author
†Corresponding Author

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

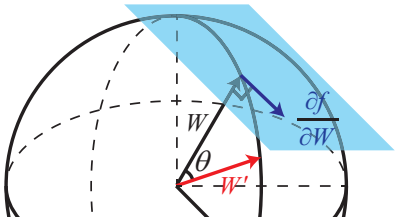

Figure 1: A sketch of updating (rotating) the parameter vector W to W' with an angle θ. The rotation direction $\frac{\partial f}{\partial W}$ is the gradient direction which is orthogonal to W. Its proof is given in text.

depends on the manual choices of the hyper parameters and the strategy of adaptively adjusting learning rates.

In this extended abstract, we explain the tangent-space gradient optimization (TSGO) proposed in Phys. Rev. E 102, 012152 (2020) [3], which is a gradient-based method to avoid the gradient vanishing and exploding problems. As the gradient is on the tangent space of the parameter vector, the optimization in TSGO is implemented as the rotation of the parameter vector. Such a process is illustrated in Fig. 1. Then the rotation angle θ plays the role of learning rate and can be well controlled in a simple way. For instance, θ reduces to its one third every time when the loss function begins to increase, which means the parameter vector is over rotated. The superiority of TSGO is demonstrated by comparing with Adam for optimizing generative tensor network (TN).

2 Tangent-space gradient optimization

TSGO requires that the parameter vector W is orthogonal to its gradient (i.e., on the tangent space) as

$$\langle W, \frac{\partial f}{\partial W} \rangle = 0, \tag{1}$$

where f is the loss function and $\langle *, * \rangle$ means the inner product. To apply TSGO in practice, one sufficient condition of Eq. (1) that is easy to satisfy is given as: the loss function does not change with rescaling the parameter vector. Mathematically, it is written as

$$f\left(X; \alpha W\right) = f\left(X; W\right), \forall \alpha \neq 0, \tag{2}$$

with X the samples and α a non-zero constant. The sufficiency can be easily proved by the two different definitions of the directional derivate. One definition of the directional derivate of f on the direction of W is

$$\partial_W f\left(X; W\right) = \lim_{h \to 0} \frac{f\left(X; W + hW\right) - f\left(X; W\right)}{h}. \tag{3}$$

It can be easily seen that $\partial_W f\left(X; W\right) = 0$ from Eq. (2). With another definition of directional derivate

$$\partial_W f\left(X; W\right) = \left\langle W, \frac{\partial f}{\partial W} \right\rangle. \tag{4}$$

It is proved that Eq. (2) is a sufficient condition of Eq. (1).

Since rescaling W does not change the loss function, W can be restricted on a unit sphere satisfying $\langle W, W \rangle = 1$. The name of TSGO comes from the fact that the gradient $\frac{\partial f}{\partial W}$ is on the tangent space of such a unit sphere. After normalizing the gradient $\frac{\partial f}{\partial W}$ to a unit vector, the parameter vector is optimized by

$$W' = W - \eta \frac{\partial f}{\partial W}. \tag{5}$$

The W' is then normalized to the unit sphere. It can be easily seen from the geometrical relation of Fig. 1 that the relation between the rotation angle and learning rate satisfies

$$\eta = \tan\theta. \tag{6}$$

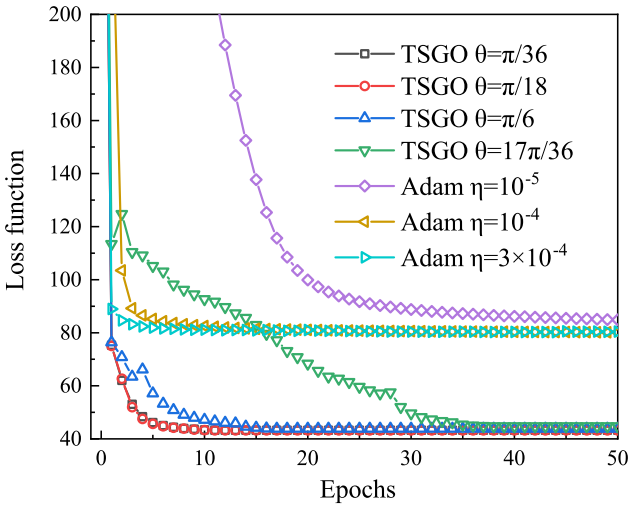

Figure 2: Loss function versus epoch by TSGO and Adam with different initial learning rates (or initial rotation angles). The computational times are around 2 and 70 s for TSGO and Adam, respectively, at each epoch. [Reused from Fig. 3 of Phys. Rev. E 102, 012152 (2020).]

3 Results and Discussions

TSGO is compared with the off-the-shelf Adam method to optimize a generative TN model [4, 5] on the MNIST dataset (a widely used dataset contains 70000 images of handwritten digits with a size of 28×28) [6]. The numerical tests are performed on NVIDIA Corporation GP102 [GeForce GTX 1080 Ti] and all codes are available at `https://github.com/crazybigcat/TSGO/tree/master`. The task is to optimize a TN generative model on 6000 images randomly selected from the MNIST dataset. The loss function takes the form of negative-log likelihood. The results with different hyper parameters (initial rotation angle θ and initial step η) are shown in Fig. 2.

From Fig. 2, the convergence values of TSGO are always less than 45, even at an obviously unreasonable $\theta = 17\pi/36$ [corresponding to $\eta \approx 11.4$ based on Eq. (6)]. As a comparison, even after 50 epochs the Adam shows to be slowly descending at a much higher (worse) value of loss function, which indicates the gradient vanishing problem. When $\eta = 3 \times 10^{-4}$, the optimization process becomes unstable [the loss function is NaN (not a number)] at the 98-th epoch, which indicates the gradient exploding problem. Larger η for Adam are tested and lead to faster instability. TSGO is also applied to the model optimized by Adam for different epochs.The loss function drops to lower than 45 quickly once TSGO is applied.

It is believed that the gradient vanishing and exploding problems occur when the computational graph becomes deep. This is verified by testing these two methods on the optimization of TN generative model with different depths. When the depth is lower than 64, Adam and TSGO have similar convergence loss functions, though Adam converges much slower. When the depth is over 144, the loss function by Adam converges to a much higher value, which indicates the depth of network structure still influences the effectiveness of Adam. As a comparison, TSGO always converges to a much lower value of loss function effectively even the depth is over 700. These results suggest that TSGO is much less influenced by the depth of network structure than Adam.

As TSGO has been applied to Bayesian networks [7] which is a probabilistic graphical model. Here we compare the performance of TSGO with the weight normalization (WN) [8]. WN is a commonly used optimization method sharing a similar idea as TSGO by normalizing the variational parameter vector. Fig. 3 shows the results on the Iris dataset [9], where the Bayesian network is optimized by Adam+WN or TSGO. When the hyper parameters are properly set, Adam combined with WN can effectively avoid the gradient vanishing problem as it converges to the position close to the convergence position of TSGO. TSGO converges much faster and more stable without tuning hyper parameters.

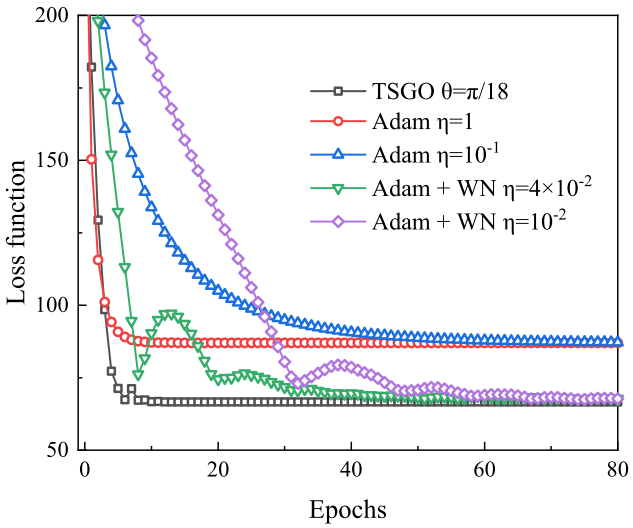

Figure 3: Loss function versus epoch by TSGO, Adam and Adam combined with WN on Iris dataset.

In summary, Adam is widely used in the optimization of deep machine learning task. It is an adaptive method that can adjust learning rate in the optimization process. Based on an empirical formula and adaptive estimates of lower-order moments, Adam takes optimization direction as a liner combination of gradients in current of previous iterations. The main idea of Adam is to increase the learning rate when the gradient is small and decreases the learning rate in the opposite case. While the parameter vector can exist in the whole Hilbert space and its scaling varies. Whether the gradient is large or small has no standard and can only rely on empirical formula. As a comparison, TSGO restricts the parameter vector on a unit sphere so that only the direction needs to be optimized. Since the angle is naturally restricted in $[0, 2\pi)$, it is easy to judge whether the rotation angle is large or small.

4 Summary

In this extended abstract, we introduce the tangent-space gradient optimization as a practical method to optimize tensor network machine learning model and it is free of gradient vanishing and exploding problems. The central idea of TSGO is to restrict the parameter vector on a unit sphere and only optimizes its direction. As the rotation angle is naturally selectable in a range of $(0, \pi/2)$, one can easily adjust the rotation angle to optimize the parameter vector. The gradient is on the tangent space of the parameter vector and only provides the direction of rotation. The scaling of gradient vector contributes nothing to the optimization and thus gradient vanishing and exploding problems are avoided.

The effectiveness of TSGO is verified on a generative TN model in comparison with Adam numerically. TSGO shows its superiority on both convergence and stability at different manual choices of hyper parameters. Similar methods involving normalization are used for avoiding the gradient vanishing and exploding problems in deep neural network such as layer normalization, batch normalization and weight normalization. It is believed that the normalization methods have implicit "early stopping" effect and help stabilize the optimization to convergence. Though TSGO has not been implemented to neural network due to the fact that nonlinearity in neural network makes it difficult to satisfy Eq. (2), it can be used for other probabilistic model in principle.

Acknowledgments and Disclosure of Funding

This work is supported in part by the National Natural Science Foundation of China (11834014), the National Key R&D Program of China (2018YFA0305800), and the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB28000000). S.J.R. is also supported by Beijing Natural Science Foundation (Grant No. 1192005 and No. Z180013) and by the Academy for Multidisciplinary Studies, Capital Normal University.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).

[2] D. P. Kingma and J. Ba, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (2015).

[3] Z.-Z. Sun, S.-J. Ran, and G. Su, Phys. Rev. E **102**, 012152 (2020).

[4] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, Phys. Rev. X **8**, 031012 (2018).

[5] Z.-Z. Sun, C. Peng, D. Liu, S.-J. Ran, and G. Su, Phys. Rev. B **101**, 075135 (2020).

[6] L. Deng, IEEE Signal Processing Magazine **29**, 141 (2012).

[7] F. V. Jensen et al., *An introduction to Bayesian networks*, vol. 210 (UCL press London, 1996).

[8] T. Salimans and D. P. Kingma, in *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016), pp. 901–909.

[9] D. Dua and C. Graff, *UCI machine learning repository* (2017).