Generative Tensor Network Classification for Supervised Learning

Ding Liu[1] , **Zheng-Zhi Sun**[2] , **Cheng Peng**[3] , **Gang Su**[2] , **Shi-Ju Ran**[4*]
[1]School of Computer Science and Technology, Tiangong University, Tianjin, 300387,China
[2]School of Physical Sciences, University of Chinese Academy of Sciences, P. O. Box 4588, Beijing 100049, China
[3]Stanford Institute for Materials and Energy Sciences, SLAC and Stanford University, Menlo Park, CA, USA
[4]Department of Physics, Capital Normal University, Beijing 100048, China
sjran@cnu.edu.cn

Abstract

Tensor network (TN) is developing rapidly into a powerful machine learning (ML) model that is built upon quantum theories and methods. Here, we introduce the generative TN classifier (GTNC), which is demonstrated to possess unique advantages over other relevant and well-established ML models such as support vector machines and naive Bayes classifiers. In specific, the GTNC is shown to rely much less on the hyper-parameters, and to be an adaptive model that avoids over-fitting by limiting the parameter complexity according to the entanglement. GTNC paves new paths to the quantum-inspired probabilistic ML models based on TN.

1 Introduction

Tensor network (TN)[Ran *et al.*, 2020] is a powerful tool that originates from quantum physics, and in recent years exhibits impressive performance and great potential in the field of machine learning (ML) [Stoudenmire and Schwab, 2016][Liu *et al.*, 2019][Han *et al.*, 2018][Sun *et al.*, 2020]. Compared with the conventional ML models such as neural network (NN), it is expected that the unique advantages of TN are from its foundations and successful applications in quantum physics. In specific, TN provides an efficient representation for quantum states, operators and circuit models; it is built upon and interpreted by the quantum many-body physics and quantum information theories.

Motivated by the inspiring progresses on TN ML, it becomes more and more urgent to demonstrated and understand the advantages of TN in practical ML problems. In this work, we introduce a generative-type of TN for classification tasks, known as generative TN classifier (GTNC) originally proposed in [Sun *et al.*, 2020]. In Sec. 2, we explain the basic ideas of GTNC with necessary formula. In Sec. 3, we demonstrate the main numerical results and discuss about the advantages of GTNC over other models. Our code of the implementation is available at [1]

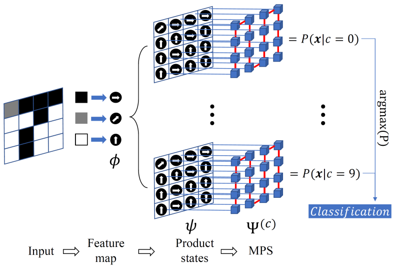
ϕ
$= P(\boldsymbol{x}|c=0)$
$= P(\boldsymbol{x}|c=9)$
argmax(P)
Classification
ψ
$\Psi^{(c)}$
Input ⇨ Feature map ⇨ Product states ⇨ MPS

Figure 1: (Color online) Illustration of the process in GTNC for classification. See the details in the text.

Table 1: Testing accuracy of GTN on MNIST and fashion-MNIST, compared with SVC, NB, a non-parametric baseline model and neural network.

model	GTNC	Baseline	SVC	NBC	CNN
MNIST	0.98	0.97	0.98	0.84	0.98
FMNIST	0.88	0.85	0.89	0.71	0.94

2 Generative Tensor Network Classifier

The central idea of GTNC is to train the generative model [Han *et al.*, 2018] using TN for each classes, and classify a given sample by comparing its probabilities from the generative models, i.e.,

$$\tilde{c} = \text{argmax}_c P(\mathbf{x}|c), \tag{1}$$

where c denotes the label corresponding to each class, $\tilde{c}$ denotes the prediction, and $\mathbf{x} = (x_1, x_2, \cdots, x_L)$ denotes the features of the sample to be classified (Fig. 1). The $P(\mathbf{x}|c)$ of each c is given by a generative TN [Han *et al.*, 2018].

GTNC is different from the discriminative TN classifiers (DTNC's), which model directly the mapping from the sample to the prediction of its classification as $\tilde{c} = \text{argmax}_c P(c|\mathbf{x})$. Two approaches are mathematically connected by Bayes' equation as $\tilde{c} = \text{argmax}_c P(c|\mathbf{x}) = \text{argmax}_c P(\mathbf{x}|c)P(c)$. Assuming the prior $P(c = 1) = P(c = 2) = \cdots = P(N_c)$, we have $\text{argmax}_c P(c|\mathbf{x}) = \text{argmax}_c P(\mathbf{x}|c)$.

[*] sjran@cnu.edu.cn

[1] The code of the implementation is available at https://github.com/crazybigcat/GTNC

Note that GTNC is not a "full" Bayesian treatment, as, for instance, that the causal relationship (represented by the conditional probabilities) among the features are difficult to infer from the TN representing $P(\mathbf{x}|c)$. To have a full Bayesian solution, one should carefully define the positivity and normalization conditions of the tensors in the TN, which leads to the Bayesian TN [Ran, 2019].

In GTNC, $P(\mathbf{x}|c)$ is obtained in a quantum-inspired way. To this end, we introduce the quantum state $\mathbf{\Psi}^{(c)}$ (named as classifier states) to determine the joint probability distribution of the features in each class. For any sample $\mathbf{x}$, its probability can be given as

$$P(\mathbf{x}|c) = |\psi(\mathbf{x})^T \mathbf{\Psi}^{(c)}|^2, \tag{2}$$

where we have $\psi(\mathbf{x}) = \prod_{\otimes i=1}^{L} \phi(x_i)$ with

$$\phi(x_i) = (\cos x_i\pi/2, \sin x_i\pi/2)^T. \tag{3}$$

Eq. (3) maps a feature x_i to a normalized (column) vector $\phi(x_i)$ and is known as the feature map. The choice of the feature map is an assumption. $\psi(\mathbf{x})$ is a 2^L-component product vector, called a "vectorized" sample.

Each $\mathbf{\Psi}^{(c)}$ also has 2^L components. Unlike $\psi(\mathbf{x})$ that has a simple product form, $\mathbf{\Psi}^{(c)}$ possesses exponentially high complexity. Therefore, we use TN to represent $\mathbf{\Psi}^{(c)}$ to lower the complexity to be polynomial. In detail, we choose the matrix product state (MPS) that is given by the contraction of L tensors as

$$\Psi^{(c)}_{s_1 s_2 \cdots s_L} = \sum_{\alpha_1 \cdots \alpha_{L-1}} T^{[1,c]}_{s_1,\alpha_1} T^{[2,c]}_{s_2,\alpha_1\alpha_2} \cdots T^{[L,c]}_{s_L,\alpha_{L-1}}. \tag{4}$$

The indexes $\{s\}$ are called physical indexes, whose dimensions (denoted by d) equal to the dimension of the feature map [$d = 2$ for Eq. (3)]. $\{\alpha\}$ are called virtual indexes, whose dimensions are called virtual dimensions and (as hyper-parameters) control the parametric complexity of the MPS. The virtual dimensions are normally bounded by another hyper-parameter χ called dimension cut-off, i.e., $\dim(\alpha_l) \leq \chi$.

3 Results and discussions

We compare GTNC with naive Bayes classifiers (NBC's), support vector classifiers (SVC's) and neural network which possess several similarities and differences with GTNC (Tab. 1).

The central idea of the SVC is to classify the samples according to the distances in a higher-dimensional space. Similarly in GTNC, the samples are also mapped to the vector space of d^N dimensions (known as Hilbert space), and the conditional probabilities [Eq. 2] for classification are linear to the distances of the vectors in the Hilbert space. The "kernel" of the GTNC is determined by the feature map. Our results show that the GTNC achieves comparable or better accuracies without any prior knowledge of the hyper-parameters. In comparison, the accuracy of a SVC severely depends on the hyper-parameters. Ref. [Sun *et al.*, 2020] demonstrates that the better performance of GTNC is due to that the samples naturally cluster in the Hilbert space, making the classifications much easier.

We also compare GTNC with a baseline model. The idea is to compute the probability $\tilde{P}(\mathbf{x}|c) = \psi(\mathbf{x})^T \bar{\mathbf{\Psi}}^{(c)}$ with $\bar{\mathbf{\Psi}}^{(c)}$ obtained by the training samples in the c-th classes and $\psi(\mathbf{x})$ from a testing sample to be classified. Note that $\bar{\mathbf{\Psi}}^{(c)}$ is extremely difficult to compute from its definition but $\tilde{P}(\mathbf{x}|c)$ can be obtained easily as $\tilde{P}(\mathbf{x}|c) = \frac{1}{N} |\sum_{x' \in \mathcal{T}} \psi(\mathbf{x})^T \psi(\mathbf{x}')|^2$.

Moreover, for the MPS, the virtual dimensions are bounded by removing the small numbers in the entanglement spectra and the corresponding channels (basis). This suggests that the redundant information which might cause over-fitting are carried by such basis. Therefore, GTNC is an adaptive model that avoids over-fitting by using moderately large virtual dimensions.

Acknowledgments

S.J.R. is supported by Beijing Natural Science Foundation (Grants No. 1192005 and No. Z180013), Foundation of Beijing Education Committees (Grant No. KM202010028013), and by the Academy for Multidisciplinary Studies, Capital Normal University. Z.Z.S and G.S. are supported in part by the National Natural Science Foundation of China (Grants No. 11834014 and No. 11474279), the National Key R&D Program of China (Grant No. 2018YFA0305800), and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB28000000).

References

[Han *et al.*, 2018] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Phys. Rev. X*, 8:031012, Jul 2018.

[Liu *et al.*, 2019] Ding Liu, Shi-Ju Ran, Peter Wittek, Cheng Peng, Raul Blázquez García, Gang Su, and Maciej Lewenstein. Machine learning by unitary tensor network of hierarchical tree structure. *New Journal of Physics*, 21(7):073059, jul 2019.

[Ran *et al.*, 2020] Shi-Ju Ran, Emanuele Tirrito, Cheng Peng, Xi Chen, Luca Tagliacozzo, Gang Su, and Maciej Lewenstein. *Tensor Network Contractions*. Springer, Cham, Switzerland, 2020.

[Ran, 2019] Shi-Ju Ran. Bayesian tensor network with polynomial complexity for probabilistic machine learning. *arXiv:1912.12923*, 2019.

[Stoudenmire and Schwab, 2016] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4799–4807. Curran Associates, Inc., 2016.

[Sun *et al.*, 2020] Zheng-Zhi Sun, Cheng Peng, Ding Liu, Shi-Ju Ran, and Gang Su. Generative tensor network classification model for supervised machine learning. *Phys. Rev. B*, 101:075135, Feb 2020.