

Presentation of paper: From probabilistic graphical models to generalized tensor networks for supervised learning (Glasser, I. Pancotti, N and Cirac, I)

Behnoush Khavari and Gustavo Patino

March 2020

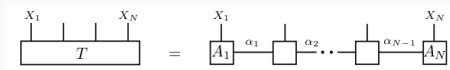
Plan of talk

- Review of Tensor Networks
- Probabilistic graphical models (PGMs) and the connection between TNs and PGMs
- TNs with copy tensors
- Copy operation and GTNs
- Supervised learning algorithms
- Learning feature vectors of data
- Numerical experiments
- Conclusion

Review of Tensor Networks

Tensor Train (MPS)

$$T_{X_1 \dots X_N} = \sum_{\alpha_i=1}^r A_{1, X_1}^{\alpha_1} A_{2, X_2}^{\alpha_1, \alpha_2} \dots A_{N-1, X_{N-1}}^{\alpha_{N-2}, \alpha_{N-1}} A_{N, X_N}^{\alpha_{N-1}}$$



Tensor network representation of MPS

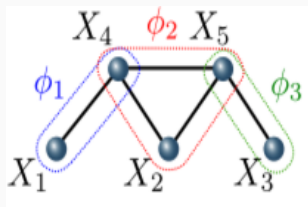
Graphical models and Tensor Networks

Graphical models

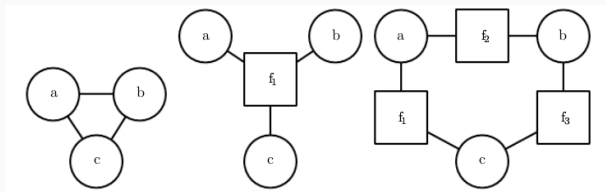
- probabilistic graphical model (PGM) : the factorization of joint probability distribution of random variables with a graph (directed or undirected)
- Here, we consider the undirected case
- Efficient tool for parameterizing probability models

Undirected PGM and factor graph

Any set of nodes that are all connected to each other in G is called a clique.

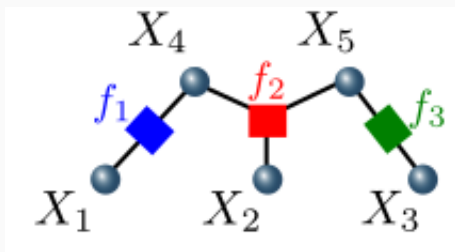


Ambiguity in factorization



Undirected PGM and factor graph

To resolve, we use factor graphs

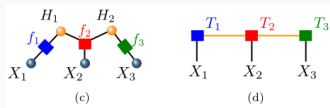
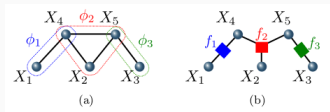


$$P(X = x) = \frac{1}{Z} \prod_c f_c(x_c)$$

$$P(X_1, X_2, X_3, X_4, X_5) = \frac{1}{Z}$$

$$f_1(X_1, X_4) f_2(X_4, X_2, X_5) f_3(X_5, X_3)$$

General PGMs in TN representation



$$P(X = x) = \frac{1}{Z} \prod_c f_c(x_c)$$

$$P(X_1, X_2, X_3, X_4, X_5) = \frac{1}{Z}$$

$$f_1(X_1, X_4) f_2(X_4, X_2, X_3) f_3(X_3, X_5)$$

$$P(X = x) = \frac{1}{Z} \sum_h \prod_c f_c(x_c, h_c)$$

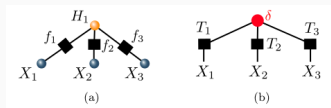
$$P(X_1, X_2, X_3) = \frac{1}{Z} \sum_{H_1, H_2}$$

$$f_1(X_1, H_1) f_2(H_1, X_2, H_3) f_3(H_3, X_3)$$

Hidden states play the role of **bond dimension** and the number of
Hidden states plays the role of **TT-rank**.

TN with copy tensor

Sometimes, Hidden states or Visible states or both are connected to several factors (like in restricted Boltzmann machine)

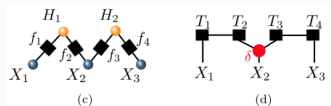


$$P(X_1, X_2, X_3) = \frac{1}{Z}$$

$$\sum_{H_1} f_1(X_1, H_1) f_2(X_2, H_1) f_3(X_3, H_1)$$

$$= \sum_{H_1, H_2, H_3} f_1(X_1, H_1) f_2(X_2, H_2) f_3(X_3, H_3)$$

$$\delta(H_1, H_2, H_3)$$



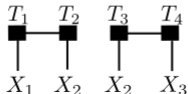
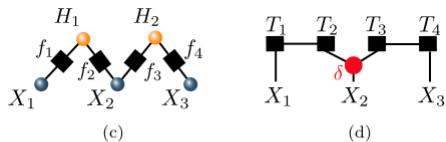
$$P(X_1, X_2, X_3) = \frac{1}{Z}$$

$$\sum_{H_1, H_2} f_1(X_1, H_1) f_2(H_1, X_2) f_3(X_2, H_2) f_4(H_2, X_3)$$

$$= \sum_{H_1, H_2, H_3, H_4} f_1(X_1, H_1) f_2(H_1, H_3) f_3(H_3, X_2, H_4)$$

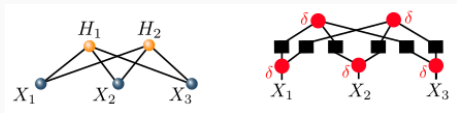
$$f_4(H_4, H_2) f_5(H_2, X_3)$$

Fixing the visible random variable

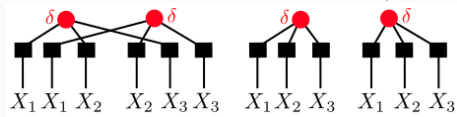


Restricted Boltzmann machine (RBM) and TN with copy tensor

RBM is an example of PGM with both hidden and visible variables connecting to several factors.



For fixed visible states TN is simplified



(binary) RBM probability function

Joint probability function :

$$P(X, H) = \frac{1}{Z} e^{\mathcal{H}(X, H)}$$

with

$$\mathcal{H}(X, H) = \sum_{i,j} \omega_{i,j} H_i X_j$$

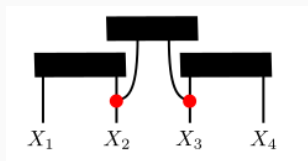
For binary RBM :

$$P(X) = \frac{1}{Z} \sum_H e^{\mathcal{H}(X, H)} = \frac{1}{Z} \prod_i (1 + e^{\sum_j \omega_{i,j} X_j})$$

index i goes over the number of hidden variables.

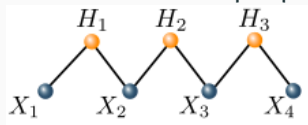
Physics inspired TNs as generalizations of RBMs

entangled plaquette states (EPS) : $T_{X_1, \dots, X_N} = \prod_{\rho=1}^P T_{\rho}^{X_{\rho}}$



$$P(X_1, X_2, X_3, X_4) = T_1(X_1, X_2)T_2(X_2, X_3)T_3(X_3, X_4)$$

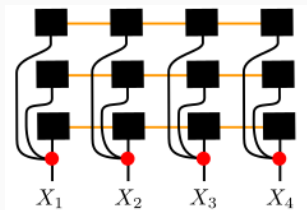
equivalent PGM : short-range RBM : X_i 's connected to the same H_j reside in the same **plaquette**



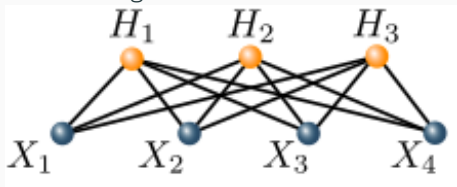
$$T_i^{X_i} = 1 + e^{\sum_j \omega_{i,j} X_j}$$

Physics inspired TNs as generalizations of RBMs

string-bond states(SBS) : $T_{x_1, \dots, x_N} = \prod_s \text{Tr}(\prod_{j \in s} A_{s,j}^{x_j})$



equivalent PGM : RBM : X_i 's connected to the same H_j reside on the same string



“No-Cloning” in Tensor Networks [Yoav Levine, Or Sharir, Nadav Cohen, and Amnon Shashua Phys. Rev.

Lett. 122, 065301 – Published 12 February 2019]

- The required operation of duplicating a vector and sending it to be part of two different calculations, which is simply achieved in any practical setting, is actually impossible to represent in the framework of TNs [2]
- proof by contradiction :

$$\exists \phi, \forall v \in \mathbb{R}^P : \sum_{i=1}^P \phi_{ijk} v_i = v_j v_k$$

for basis vectors :

$$\begin{aligned} \sum_{i=1}^P \hat{e}_i^{(\alpha)} \phi_{ijk} &= \hat{e}_j^{(\alpha)} \hat{e}_k^{(\alpha)} \\ \rightarrow \phi_{\alpha jk} &= \delta_{\alpha jk} \end{aligned}$$

counterexample : $v = \vec{1}$

$$\sum_j \phi_{ijk} v_i = \sum_j \delta_{ijk} \mathbf{1} = \delta_{jk} \neq v_j v_k = 1$$

Generalized Tensor Networks (GTNs)

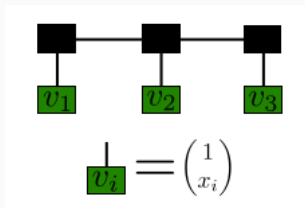
- a key factor that boosts the power of deep learning representations relative to common TNs :
inherent re-use of information in CNNs that cannot be naively represented in TN language
- To overcome, they introduce a new **copy operation** in tensor networks → **Generalized tensor network**

Input features

- We are concerned with data in the form of real numbers (non-discrete data)
- We make feature vectors as suggested in [4]

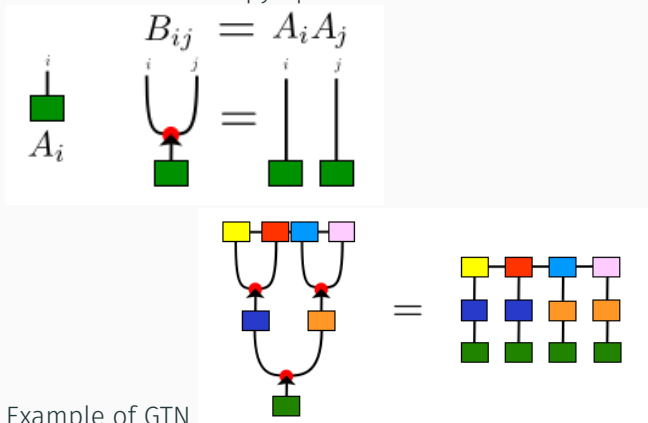
$$x_1, \dots, x_N \rightarrow v_1 \otimes v_2 \otimes \dots \otimes v_N, \quad v_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$$

The function of inputs is given by the contraction of weights in TN format with feature vectors



Copy operation with vector inputs

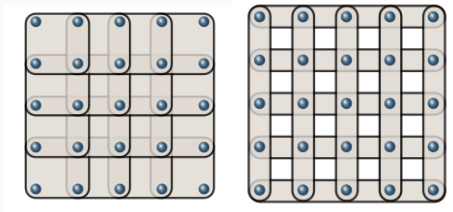
- Tensor Network + copy operation \rightarrow Generalized Tensor Network



- Example of GTN
- several copies of the inputs can be used
- GTNs use weight sharing between some tensors

GTNs used in the paper for 2-dim inputs : images

- EPS and SBS with copy tensors replaced by **copy operations**
- EPS with 2×2 overlapping plaquettes with weight sharing such that the tensor for each plaquette is the same



- SBS defined with horizontal and vertical strings covering the 2D lattice.

SUPERVISED LEARNING ALGORITHM

Supervised Learning with Restricted Boltzmann Machines

Given the label training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$ A RBM can be used to approximate the joint probability distribution of the variables and labels as

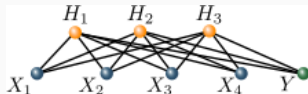
$$p(\mathbf{x}, y) = \frac{1}{Z} \sum_{\mathbf{h}} e^{\mathcal{H}(\mathbf{x}, \mathbf{h}, y)} ; \mathcal{H} = \sum_{i,j} w_{ij} h_i x_j \quad (1)$$

In supervised learning we are interested in calculating the conditional probability

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}, y)}{\sum_{y_j} p(\mathbf{x}, y_j)} \quad (2)$$

An optimal label predicted by the model is obtained minimizing the cost function

$$\mathcal{L} = - \sum_{i=1}^D \log p(y_i | \mathbf{x}_i) \quad (3)$$



RBM taken from [1]

Supervised Learning with Generalized Tensor Networks

The joint probability distribution of the variables and labels are approximated as a tensor network

$$p(\mathbf{x}, y) \sim \text{GTN}(\mathbf{x}, y) \quad (4)$$

GTN is the function resulting of the contraction of a generalized tensor network with the inputs features and with the discrete label. The contraction must be positive so

$$p(\mathbf{x}, y) \sim e^{\text{GTN}(\mathbf{x}, y)} \quad (5)$$

We then define, by analogy with the graphical model case

$$p(y_k | \mathbf{x}_i) = \frac{e^{\text{GTN}(\mathbf{x}_i, y_k)}}{\sum_{y_j} e^{\text{GTN}(\mathbf{x}_i, y_j)}} \quad (6)$$

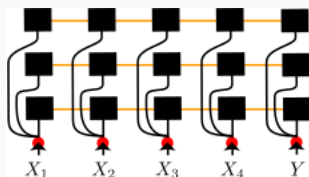
and the cost function

$$\mathcal{L} = - \sum_{i=1}^D \log p(y_i, \mathbf{x}_i) \quad (7)$$

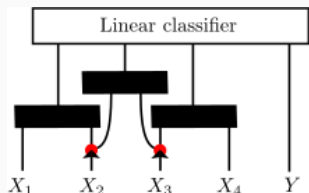
The gradient of the cost function can then be expressed as

$$\frac{\partial \log p(y_i, \mathbf{x}_i)}{\partial w} = \frac{\partial \text{GTN}(\mathbf{x}_i, y_i)}{\partial w} = \sum_{y_j} p(y_j, \mathbf{x}_i) \frac{\partial \text{GTN}(\mathbf{x}_i, y_j)}{\partial w} \quad (8)$$

Supervised Learning with Generalized Tensor Networks



SBS adds a node corresponding to the label, and corresponding tensors which connect it to the rest of the tensor network. Taken from [1]

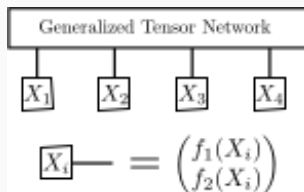


EPS output is a tensor that can be combined with a linear classifier. Taken from [1]

LEARNING FEATURE VECTORS OF DATA

Learning features vectors of data

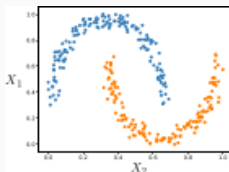
- Strategies that can be used to deal with data that is not discrete
- An approach is to map the real data to a higher dimensional feature space. Each variable is independently mapped to a vector of length (at least) two in order to be contracted with the open legs of the tensor network



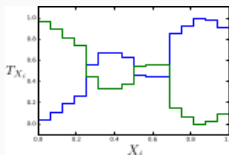
(a) Real inputs X_i are mapped to a feature vector. Taken from [1]

$$x \rightarrow \begin{pmatrix} \cos^2\left(\frac{\pi}{2}x\right) \\ \sin^2\left(\frac{\pi}{2}x\right) \end{pmatrix} \quad (9)$$

Learning features vectors of data

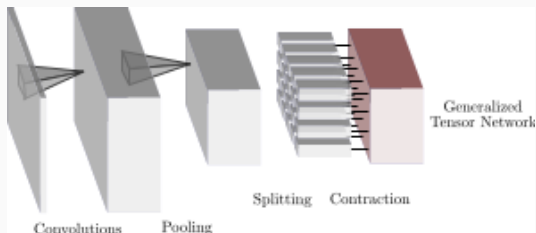


Dataset with two features (X_1, X_2) and two classes (colors) that cannot be learned by a MPS of bond dimension 2 using Eq.9) Taken from [1]



Two normalized features learned by a tensor classifying the previous data set with a MPS of bond dimension 2. Taken from [1]

Learning features vectors of data



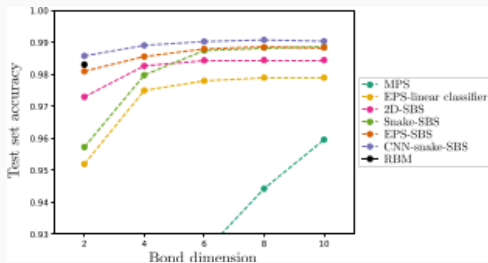
CNN used as feature vector extractors from real data. The output of the CNN is seen as an image with a third dimension collecting the different features. For each pixel of this image, the vector of features is contracted with the open legs of a tensor network. Taken from [1]

NUMERICAL EXPERIMENTS

Numerical experiments

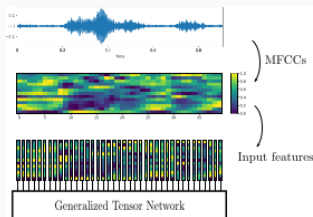


MNIST data set. Taken from [1]

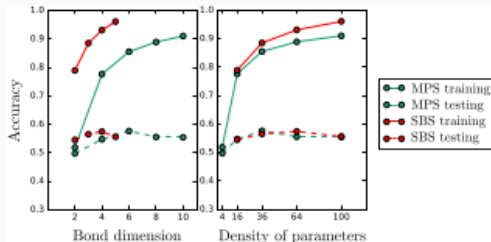


Test set accuracy of different generalized tensor networks on the MNIST data set. Taken from [1]

Numerical experiments



UrbanSound8K data set. Taken from [1]







Training and testing accuracy of a MPS and a SBS with 4 strings on the UrbanSound8K data set. The density of parameters is the total number of parameters divided by 174 (the length of the strings). Taken from [1]

Conclusions

Conclusions

- The authors show the relation between tensor network structures and graphical models such as RBM and SBS .
- One can generalize tensor networks to apply on data with vector features and strategies were discussed to use TN with real-valued data.
- Provide algorithms to train the models in supervised learning tasks, even when coupled with neural networks.
- The GTN show a better accuracy in multiclass classification tasks than regular TN and can be used as well in sound recognition.

-  Ivan Glasser, Nicola Pancotti, and J. Ignacio Cirac
From probabilistic graphical models to generalized tensor networks for supervised learning
arXiv:1806.05964
-  Y. Levine, O. Sharir, N. Cohen, and A. Shashua.
Quantum Entanglement in Deep Learning Architectures.
Phys. Rev. Lett. 122, 065301 (2019).
-  I. Glasser , R. Sweke , N. Pancotti , J. Eisert , J. Ignacio Cirac
Expressive power of tensor-network factorizations for probabilistic modeling
Advances in Neural Information Processing Systems 32, Proceedings of the NeurIPS 2019 Conference

-  E. Miles Stoudenmire, David J. Schwab Supervised Learning with Quantum-Inspired Tensor Networks
Advances in Neural Information Processing Systems 29, 4799
(2016)