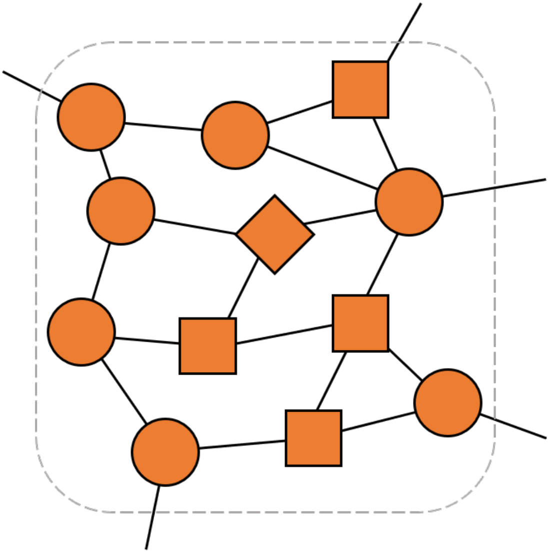
Tensor Networks in Machine Learning

Luyan Yu

University of Texas at Austin

May 13, 2019

Tensor Networks



Outline

Tensors

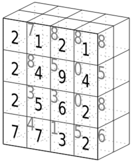
Tensors (multi-linear maps) can be simply viewed as multi-dimensional arrays. The dimension of such array is called the order of a tensor:

- Order-0 tensor: number x
- Order-1 tensor: vector v_i
- Order-2 tensor: matrix A_{ij}
- Order-n tensor: $T_{i_1 i_2 \cdots i_n}$

Important: not to confuse the order with the dimension of each index!

5
7
3
1
4
2

3	1	4	1
5	9	2	6
5	3	5	8
9	7	9	3
2	3	8	4
6	2	6	4



Tensor product

The operation to form high order tensor from low order tensors.

$$T_{i_1 \cdots i_n \, j_1 \cdots j_m \, k_1 \cdots k_p \, \cdots\cdots} = Q_{i_1 \cdots i_n} R_{j_1 \cdots j_m} S_{k_1 \cdots k_p} \cdots\cdots$$

Example: Tensor product of two vectors (order-1 tensor) is a matrix (order-2 tensor). It's also called outer product.

$$A_{ij} = v_i u_j$$

The order of the result tensor is sum of the orders of component tensors.

Tensor contraction

The operation of summing a pair of indices of a tensor.

$$\tilde{T}_{i_1 \cdots i_p i_{p+2} \cdots i_q i_{q+2} \cdots i_m} = \sum_k T_{i_1 \cdots i_p \, k \, i_{p+2} \cdots i_q \, k \, i_{q+2} \cdots i_m}$$

Examples:

- Matrix trace: $\mathrm{Tr} A = \sum_i A_{ii}$
- Dot product: $\boldsymbol{v} \cdot \boldsymbol{u} = \sum_i v_i u_i$
- Matrix-vector product: $A\boldsymbol{v} = \sum_j A_{ij} v_j$
- Matrix-matrix product: $AB = \sum_k A_{ik} B_{kj}$

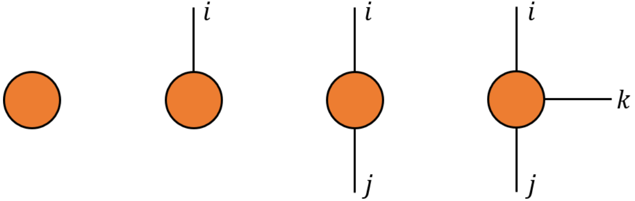
Contractions can be nested:

$$A_{ij} B_{kl} \longrightarrow \sum_j A_{ij} B_{jl} \longrightarrow \sum_i \left(\sum_j A_{ij} B_{ji} \right) \longrightarrow \mathrm{Tr}(AB)$$

Each contraction reduces the order of tensor by 2.

Graphical representation of tensors

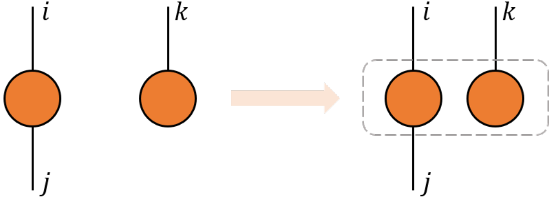
Represent tensor by a shape with arms. (not necessarily by a circle)



Number of arms = order of the tensor

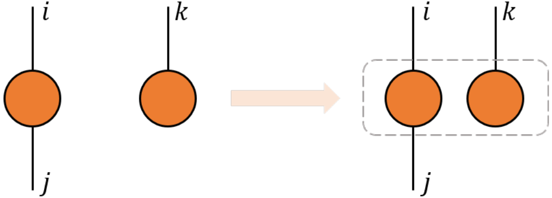
Graphical representation of tensors

Tensor product: put tensors in the juxtaposition.

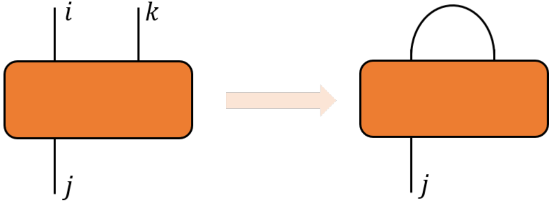


Graphical representation of tensors

Tensor product: put tensors in the juxtaposition.

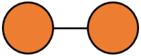


Tensor contraction: connect arms of tensor(s).

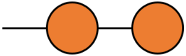


Examples

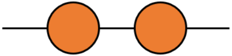
Dot product



Matrix vector product

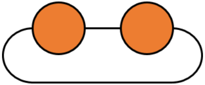


Matrix matrix product

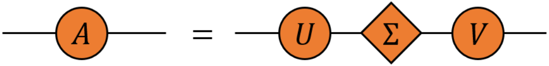


Examples

Matrix trace

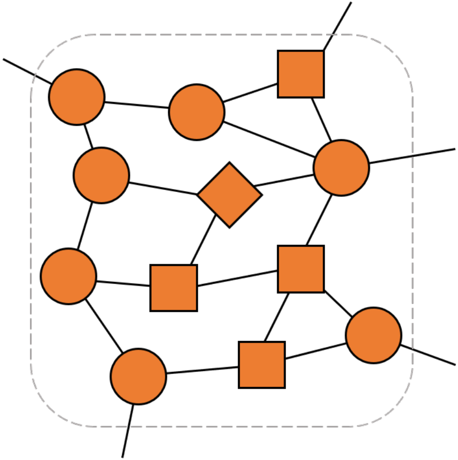


SVD



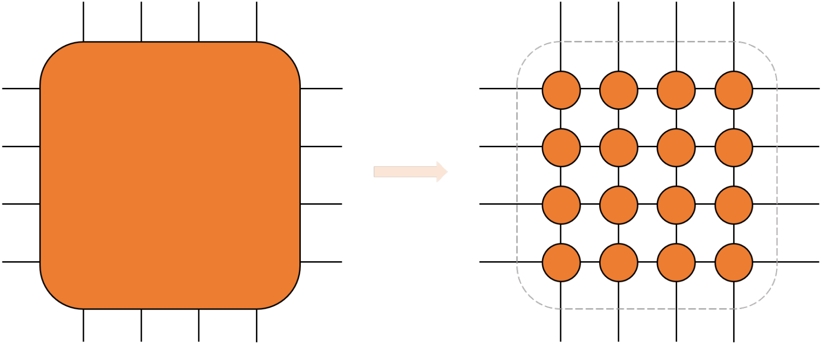
Tensor Networks

Tensor networks are widely used tool of studying quantum system.



Tensor Networks

Key concept: approximate high order tensor with low order tensors.



Approximate one order-16 tensor with sixteen order-4 tensors.
Number of parameter: $d^{16} \longrightarrow 16d^4$.

Apply to Machine Learning

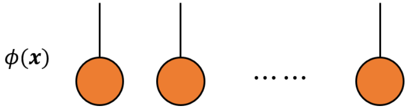
Recall the feature map in kernel method:

$$\phi(\boldsymbol{x}) : \text{low dim space} \rightarrow \text{high dim space}$$

Tensor product: a natural way to obtain high dimensional space.

$$\phi_{s_1 s_2 \cdots s_N}(\boldsymbol{x}) = \varphi_{s_1}(x_1) \otimes \varphi_{s_2}(x_2) \otimes \cdots \varphi_{s_N}(x_N)$$

$\varphi : \mathbb{R} \rightarrow \mathbb{R}^d$ is called local feature map. N is the dimension of each input.



Dimension of $\phi(\boldsymbol{x})$: d^N. Example:

$$\varphi(x) = \left[\cos\left(\frac{\pi}{2}x\right), \sin\left(\frac{\pi}{2}x\right)\right].$$

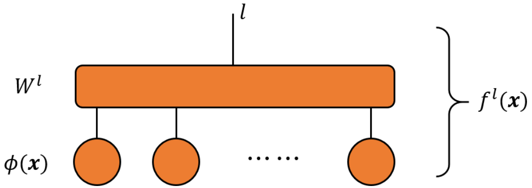
Classification

Classify the input within L different classes.

$$f^l(\boldsymbol{x}) = W^l \cdot \phi(\boldsymbol{x})$$

Take $\text{argmax}_l\, f^l(\boldsymbol{x})$ to be the output.

View W^l as an order-(N+1) tensor:

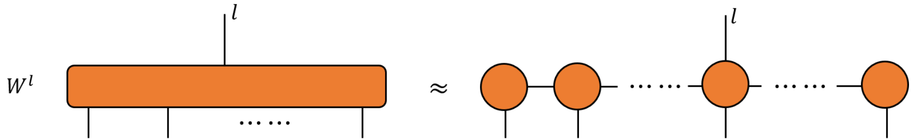


Dimension of W^l: $L \times d^N$.

Classification

Approximate by tensor network to keep the problem tractable.

Matrix Product State (MPS) approximation:



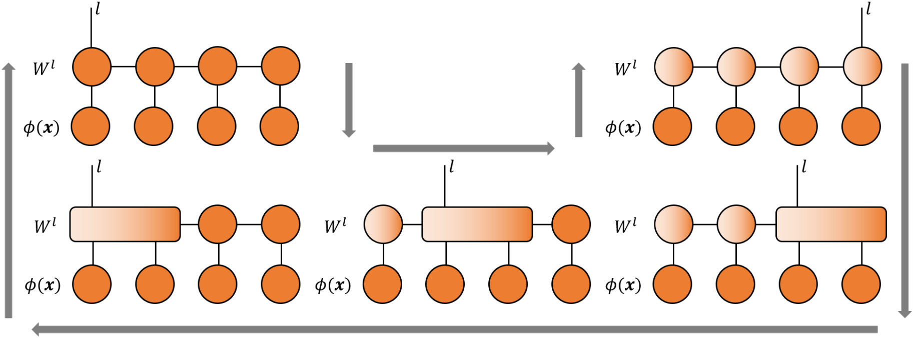
$$W^l_{s_1 s_2 \cdots s_N} \approx \sum_{\{\alpha\}} A^{\alpha_1}_{s_1} A^{\alpha_1 \alpha_2}_{s_2} \cdots A^{l;\alpha_j \alpha_{j+1}}_{s_j} \cdots A^{\alpha_{N-1}}_{s_N}$$

Goal: minimize the loss function over smaller tensors.

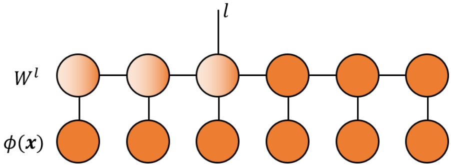
$$C(\{\boldsymbol{x}_n\}; W^l) = \frac{1}{2} \sum_{n=1}^{N_{\text{train}}} \sum_{l=1}^{L} (f^l(\boldsymbol{x}_n) - \delta^l_{L_n})^2$$

Optimization

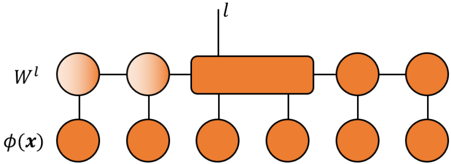
Algorithm: Density Matrix Renormalization Group (DMRG) in physics.



Iterative step 0: start from the following configuration:

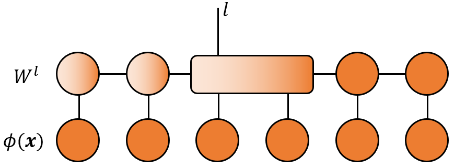


Step 1: group two consecutive tensors with l index on the left tensor;

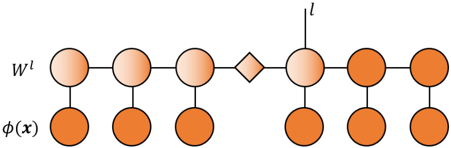


Optimization

Step 2: use gradient descent to update the parameter of this tensor;

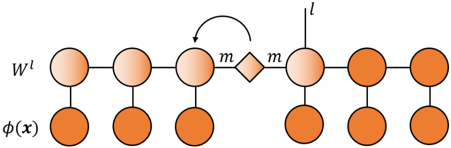


Step 3: perform SVD on updated tensor;

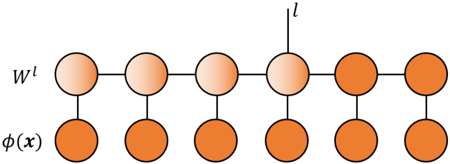


Optimization

Step 4: **keep the m largest** singular values and corresponding vectors;

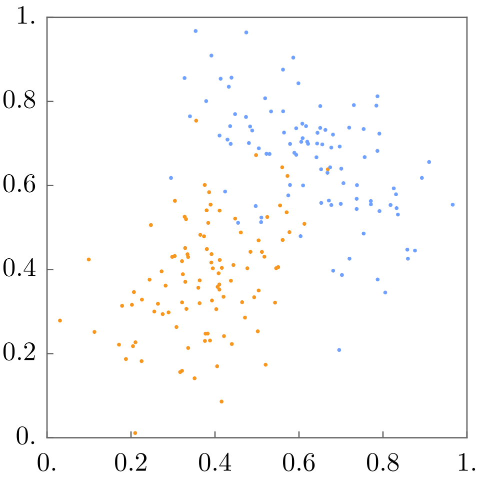


Step 5: merge and now the l index is on the right.



Iterate these steps back and forth.

Toy problem



2D Normal distributed data

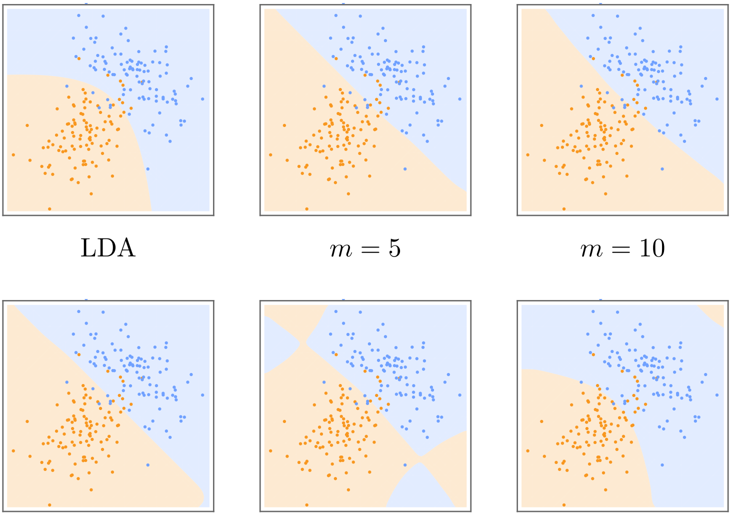
$$\boldsymbol{\mu}_1 = \begin{pmatrix} 0.4 \\ 0.4 \end{pmatrix}$$

$$\Sigma_1 = \begin{pmatrix} 0.02 & 0.01 \\ 0.01 & 0.02 \end{pmatrix}$$

$$\boldsymbol{\mu}_2 = \begin{pmatrix} 0.6 \\ 0.7 \end{pmatrix}$$

$$\Sigma_2 = \begin{pmatrix} 0.02 & -0.01 \\ -0.01 & 0.02 \end{pmatrix}$$

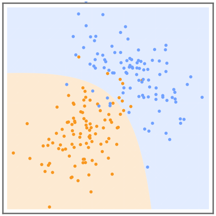
Varying bond dimension m ($d = 2$)



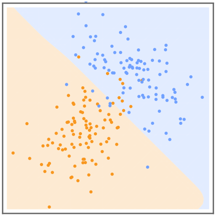
LDA $m = 5$ $m = 10$

$m = 15$ $m = 20$ $m = 40$

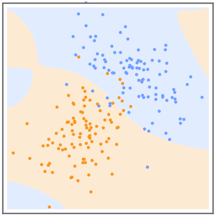
Varying local feature dimension d ($m = 15$)



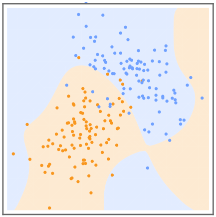
LDA



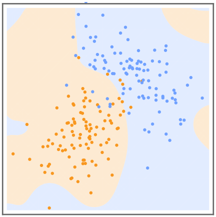
$d = 2$



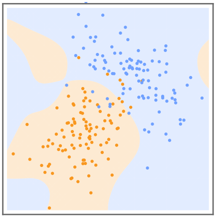
$d = 3$



$d = 5$

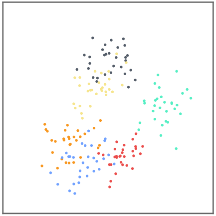


$d = 7$

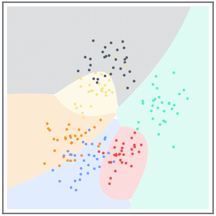


$d = 9$

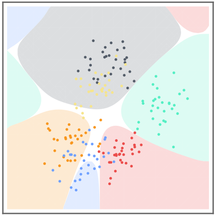
Multi-class classification



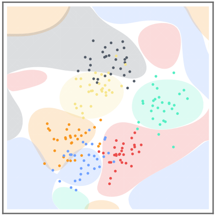
Dataset



LDA

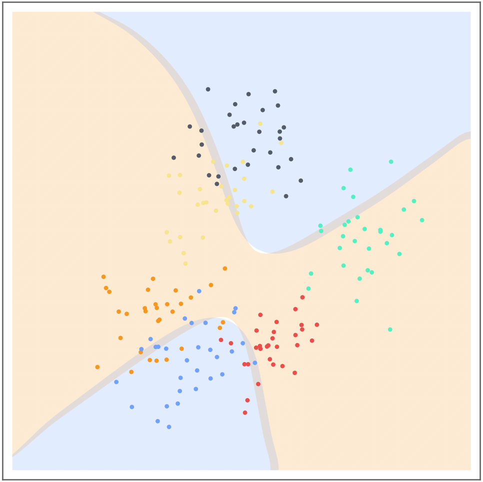


$m = 5, d = 2$



$m = 5, d = 9$

Multi-class classification



step $= 1$

MNIST

Result from [1]

28×28 binary images of hand written digits

Training set: 60000 images

Testing set: 10000 images

- $m = 10$, error on train and test data $\sim 5\%$;
- $m = 20$, error on train and test data $\sim 2\%$;
- $m = 120$, error on train data $= 0.05\%$, on test data $= 0.97\%$.

Summary and future directions

Summary

- Tensor networks are tools to study high order tensors while still keep the problem tractable by endowing internal structures;
- it provides natural and effective representations of high dimensional parameter space, which is beneficial for machine learning tasks.

Summary and future directions

Summary

- Tensor networks are tools to study high order tensors while still keep the problem tractable by endowing internal structures;
- it provides natural and effective representations of high dimensional parameter space, which is beneficial for machine learning tasks.

Future directions

- different tensor factorizations other than SVD;
- different tensor network structures;

Tensor network structures

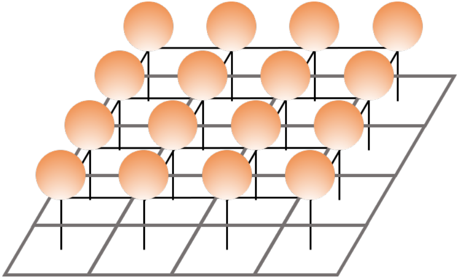


Figure: Projected Entangled Pair States (PEPS)

Tensor network structures

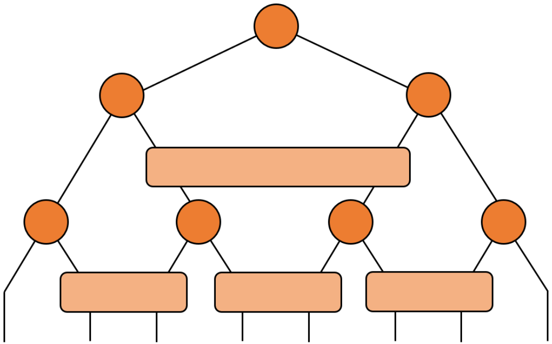


Figure: Multi-scale Entanglement Renormalization Ansatz (MERA)

Summary and future directions

Summary

- Tensor networks are tools to study high order tensors while still keep the problem tractable by endowing internal structures;
- it provides natural and effective representations of high dimensional parameter space, which is beneficial for machine learning tasks.

Future directions

- different tensor factorizations other than SVD;
- different tensor network structures;
- generative tensor network.

Generative tensor network

Given label, perform a 'quantum measurement' on the input arms.

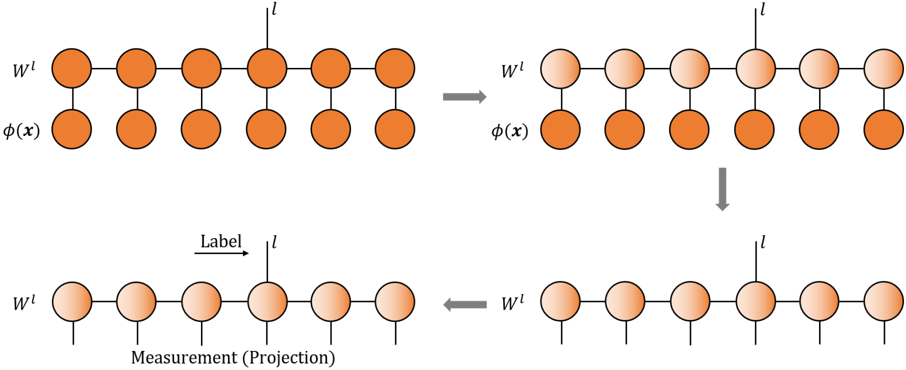


Figure: Scheme for generative tensor network

References

1. Stoudenmire, Edwin, and David J. Schwab. *Supervised learning with tensor networks*. Advances in Neural Information Processing Systems, 2016.
2. Biamonte, J., and V. Bergholm. *Quantum Tensor Networks in a Nutshell.* arXiv 1708, 2017.
3. Milsted, Ashley, et al. *TensorNetwork on TensorFlow: A Spin Chain Application Using Tree Tensor Networks.* arXiv preprint arXiv:1905.01331, 2019.
4. Roberts, Chase, et al. *TensorNetwork: A Library for Physics and Machine Learning.* arXiv preprint arXiv:1905.01330, 2019.

Thank You! & Questions