計算科学における情報圧縮

Information Compression in Computational Science

2022.1.6

#12: Application of MPS to time evolution and data science

理学系研究科　大久保　毅

Graduate School of Science, **Tsuyoshi Okubo**

Outline

- Application to time evolution of quantum system
  - Time evolution using tensor network representation
  - TEBD algorithm
  - iTEBD and (i)TEBD for eigenvalue problems
- Application of MPS to data science
  - Classification problem
  - Generative models
  - Compressing (deep) neural network

Application to time evolutions of quantum system

*Similar algorithm can be used to simulate quantum circuit.

A. McCaskey et al, PLoS ONE **13**, e0206704 (2018)

Related reference:

C. Guo et al, Phys. Rev. Lett. **123**, 190501 (2019)

Time evolution of a quantum system

Schrödinger equation: $i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = \mathcal{H}|\psi(t)\rangle$

Formal solution:

$$|\psi(t)\rangle = e^{-it\mathcal{H}/\hbar}|\psi(0)\rangle$$

Time evolution operator
(時間発展演算子)

Time evolution using MPS:

1. Multiply the time evolution operator to a MPS.
2. Find an approximate MPS representation for it.

When the time step (t) is small,
we can perform the above step efficiently.

Time evolution of a quantum system using MPS

Target: (Basically) one-dimensional quantum system with short range interaction

Typical example: Chain of qbits or quantum spins

Transverse field Ising model

$$\mathcal{H} = -\sum_{i=1}^{N-1} S_i^z S_{i+1}^z - h \sum_{i=1}^{N} S_i^x$$

Heisenberg model

$$\mathcal{H} = \sum_{i=1}^{N-1} (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + S_i^z S_{i+1}^z) - h \sum_{i=1}^{N} S_i^z$$

Typical situation: Quantum quench

Initial state: Ground state of a Hamiltonian which well approximated by MPS

t > 0: Hamiltonian suddenly changes from the initial one.

For a "short" time interval, evolving state is approximated by MPS efficiently.

Suzuki-Trotter decomposition

Suzuki-Trotter decomposition: (M. Suzuki, Commun. Math. Phys. **51**, 183 (1976))

Systematic approximation of exponential operator

$$e^{\tau(\mathcal{A}+\mathcal{B})} = e^{\tau\mathcal{A}}e^{\tau\mathcal{B}} + O(\tau^2) \qquad \text{(1st order)}$$

$$(\mathcal{A}\mathcal{B} \neq \mathcal{B}\mathcal{A}) \qquad = e^{\tau/2\mathcal{A}}e^{\tau\mathcal{B}}e^{\tau/2\mathcal{A}} + O(\tau^3) \quad \text{(2nd order)}$$

$$= e^{\tau/2\mathcal{B}}e^{\tau\mathcal{A}}e^{\tau/2\mathcal{B}} + O(\tau^3) \quad \text{(2nd order)}$$

If our Hamiltonian is represented as a sum of "local" operators,

$$\mathcal{H} = \sum_i H_i$$

E.g. transverse field Ising model

$$H_i = -S_i^z S_{i+1}^z - \frac{h}{2}(S_i^x + S_{i+1}^x)$$

Time evolution operator can be approximated as

$$e^{-it\mathcal{H}/\hbar} = \left(e^{-i\delta\mathcal{H}}\right)^M = \left(\prod_j e^{-i\delta H_j}\right)^M + O(\delta) \qquad \text{(1st order)}$$

$$\delta \equiv t/(M\hbar)$$

Graphical representation of Suzuki-Trotter decomposition

Suppose the Hamiltonian can be decomposed into the sum of two-body local terms

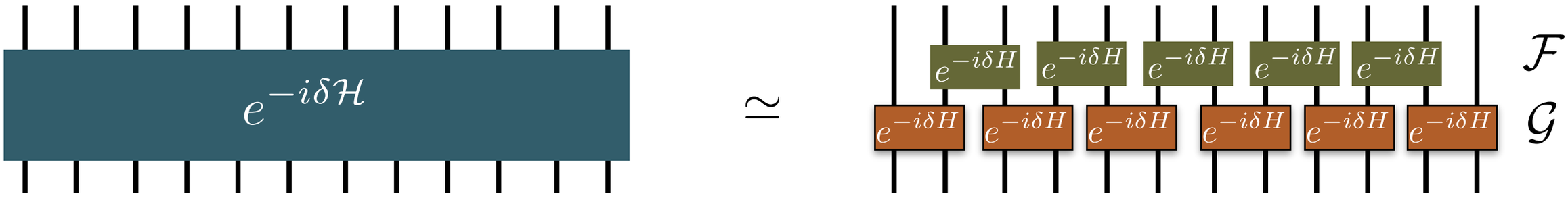
$$\mathcal{H} = \sum_i H_i = \sum_{i \in \text{even}} H_i + \sum_{i \in \text{odd}} H_i$$

$$= \mathcal{F} + \mathcal{G} \qquad\qquad [\mathcal{F}, \mathcal{G}] \neq 0$$

Suzuki-Trotter decomposition of time evolution operator

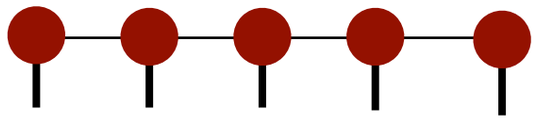
$$e^{-i\delta\mathcal{H}} = e^{-i\delta\mathcal{F}} e^{-i\delta\mathcal{G}} + O(\delta^2)$$

(1st order)

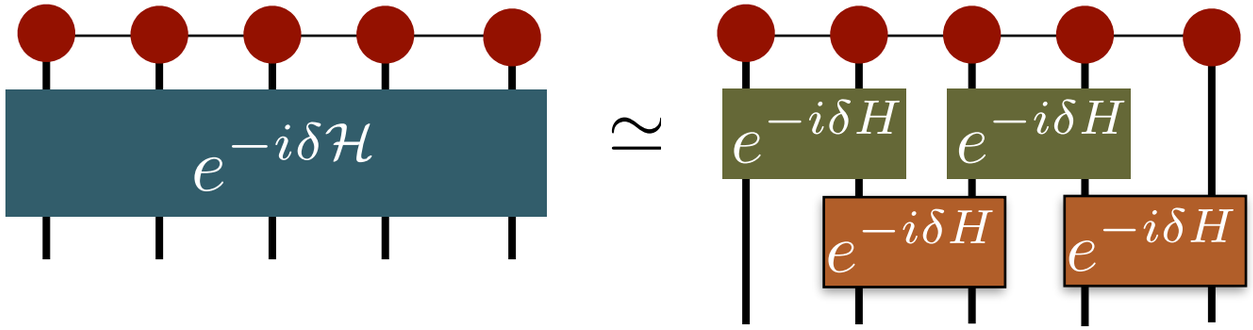


Multiplication of time evolution operator

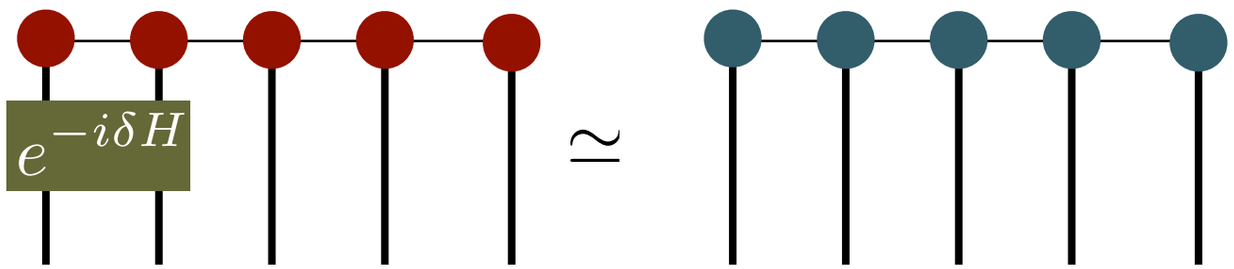
If we have MPS representation of $|\psi\rangle$

$|\psi\rangle =$ 

Multiplying the time evolution operator is represented as

$e^{-i\delta H}|\psi\rangle =$ 

If we can perform the transformation



(Generally, all matrices change for better approximation)

We continue the time evolution repeatedly.

Notice: we want to keep the bond dimension χ constant along time evolution.

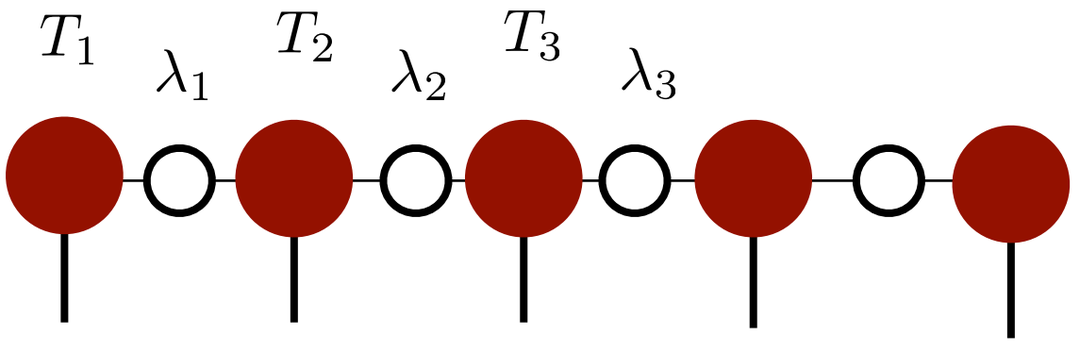
TEBD algorithm

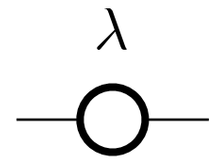
Canonical form of MPS

Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

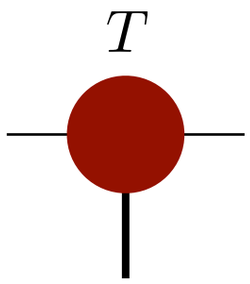
Vidal canonical form

(G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003)



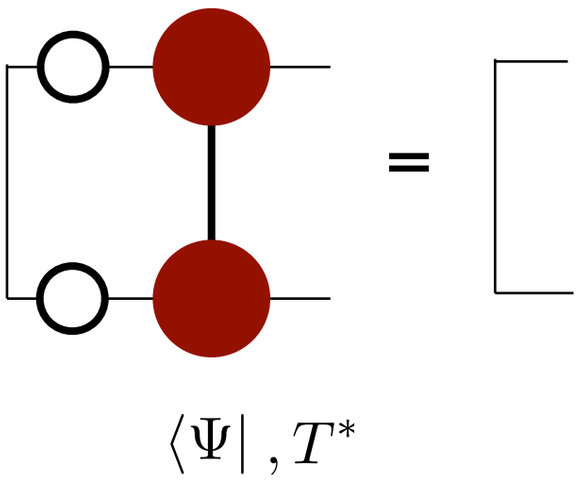


:Diagonal matrix corresponding to Schmidt coefficient

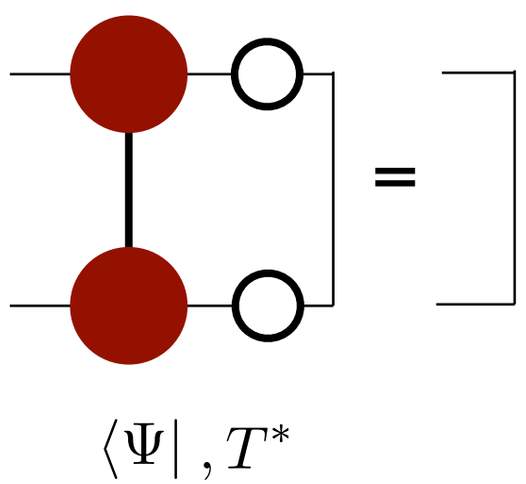


:Virtual indices corresponding to Schmidt basis

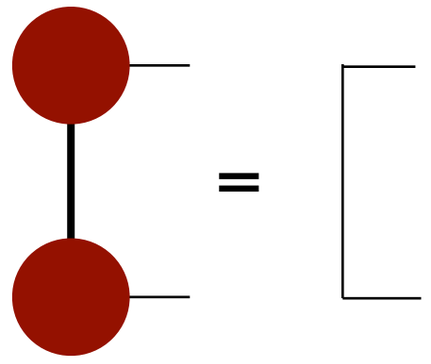
Left canonical condition:



Right canonical condition:



(Boundary)

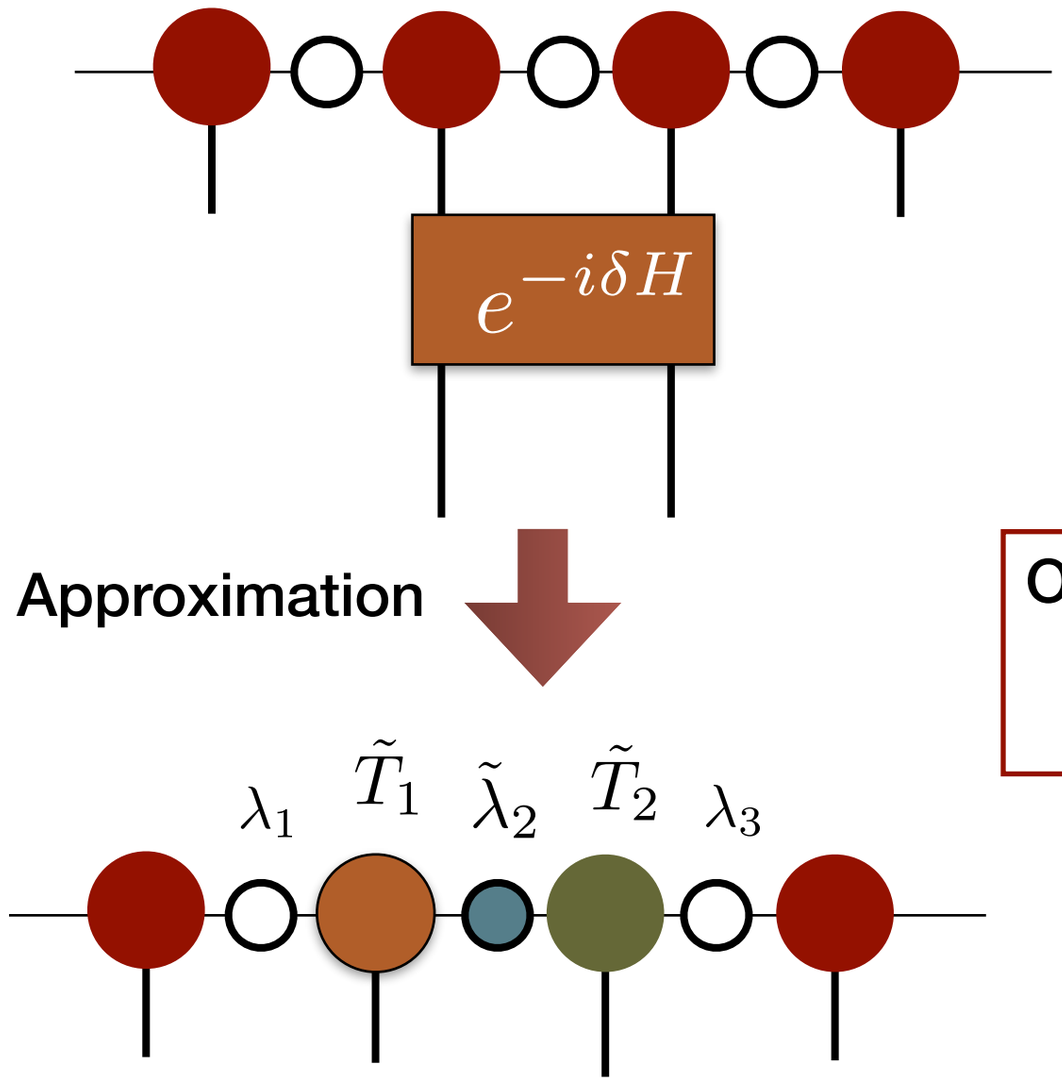


TEBD algorithm:

(G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003))

Time Evolving Block Decimation (TEBD)

We can perform the accurate transformation locally by using canonical MPS.

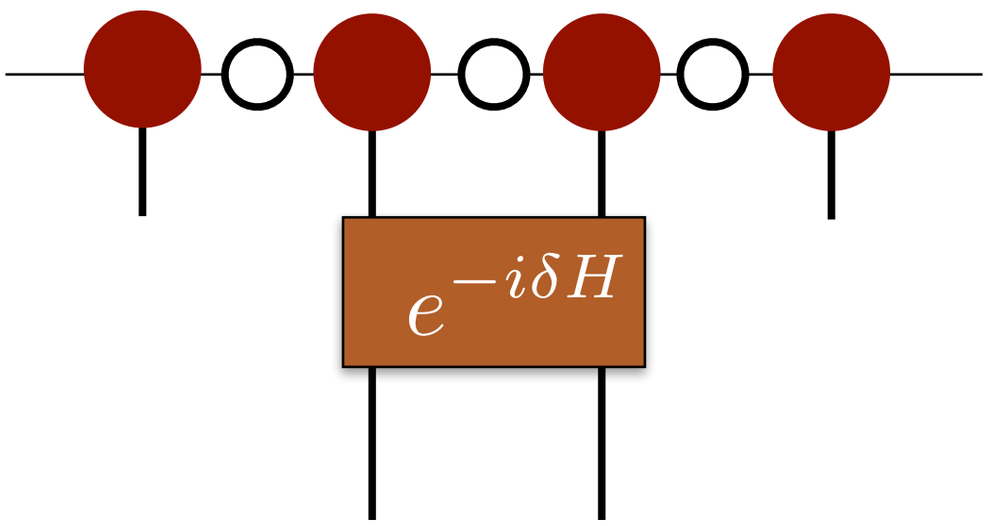


Only the two matrices which are directly applied TE operator changes in MPS.

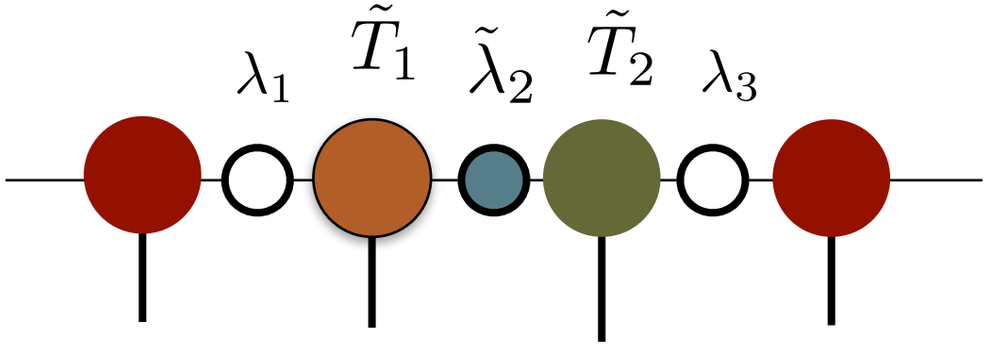
TEBD algorithm:

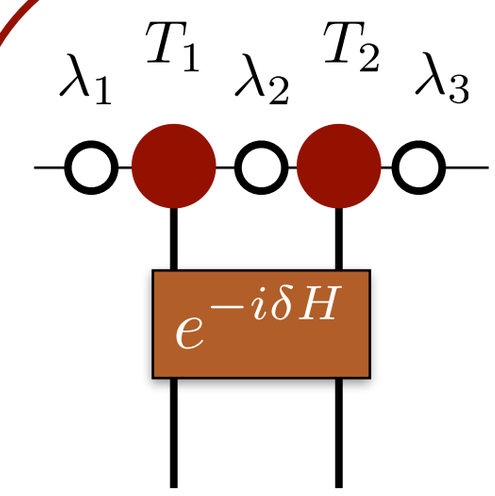
(G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003))

Apply TE operator

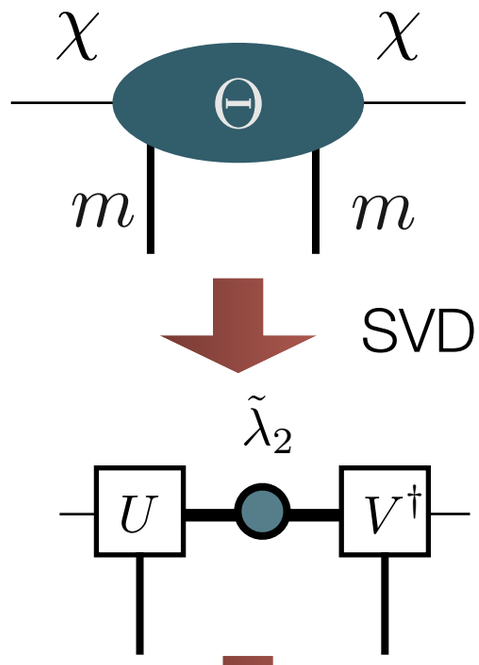


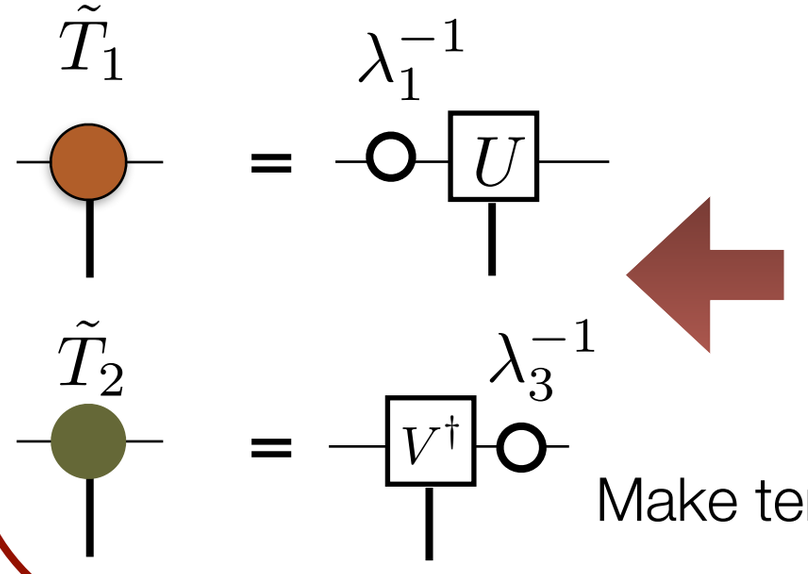
Approximation





Combine and make matrix

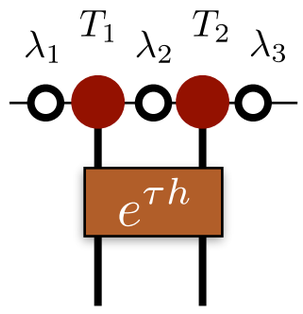




Make tensor

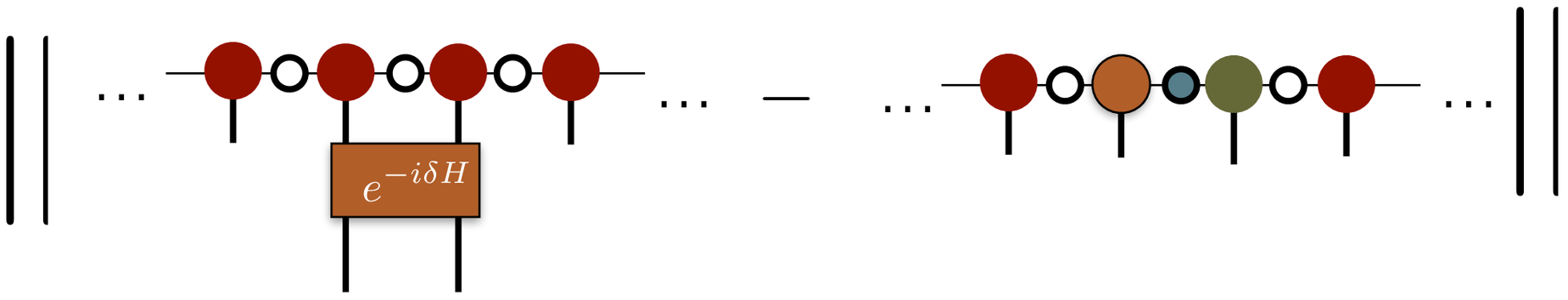
Why TEBD is accurate?

1. For accurate calculation, the canonical form is important.



If λ is equal to the Schmidt coefficient, it contains
all information of the remaining part of the system.

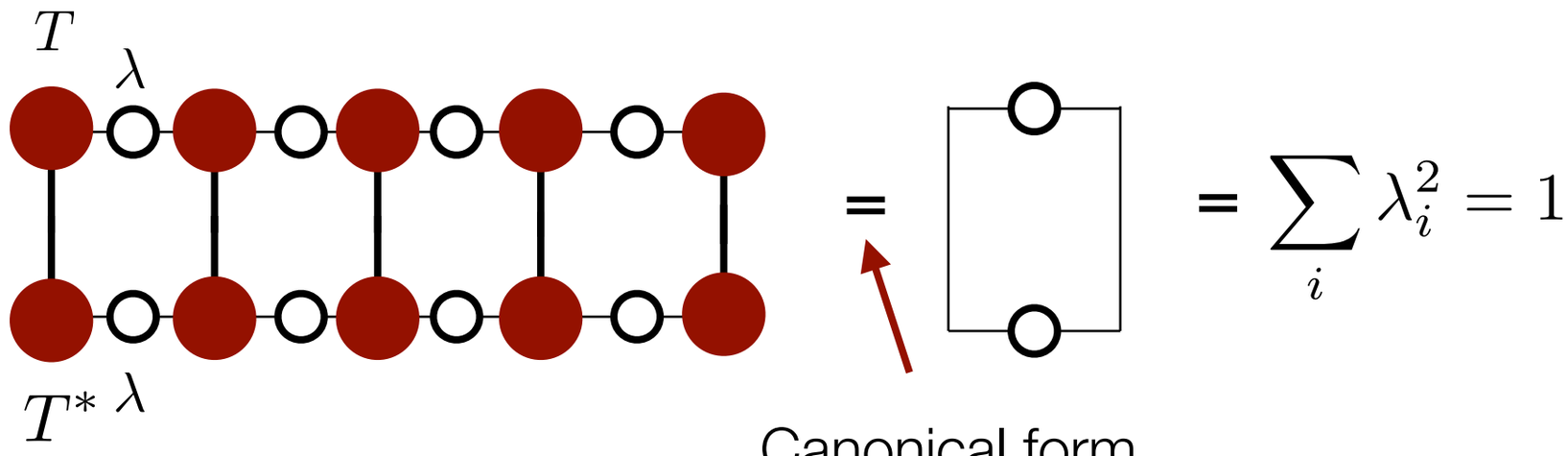
Due to the canonical form, we can prove that TEBD algorithm minimize the distance of two quantum states:



Truncation based on local SVD can be globally optimal,
even if we look at a part of the MPS.

Calculation distance: remember the calculations

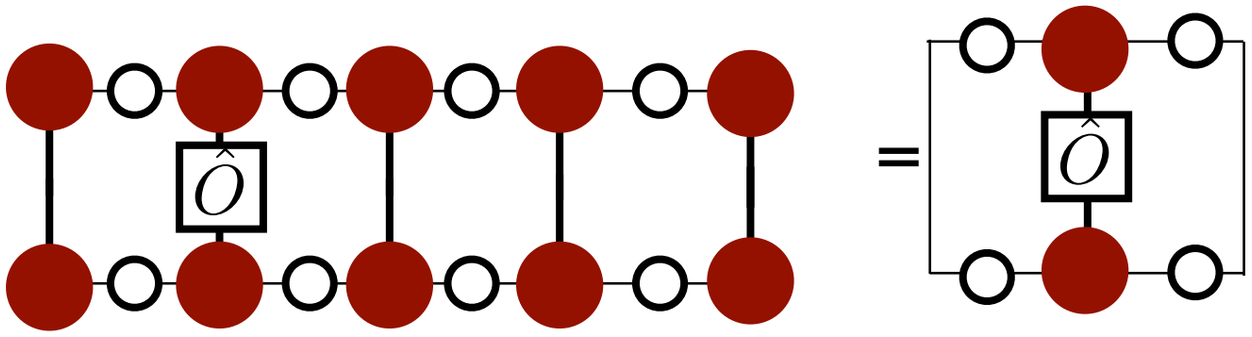
$$\langle \Psi | \Psi \rangle =$$



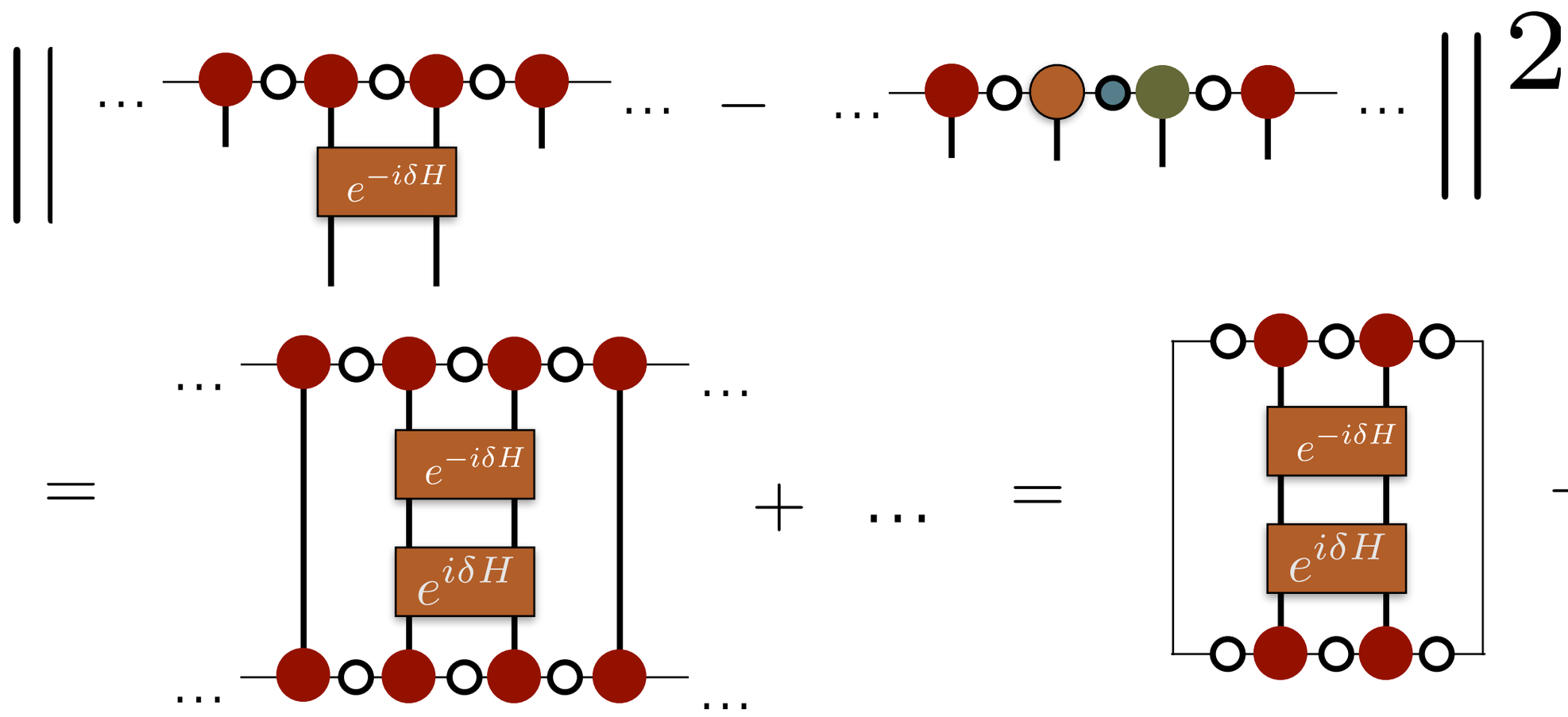
$$= \sum_i \lambda_i^2 = 1$$

Canonical form

$$\langle \Psi | \hat{O} | \Psi \rangle =$$



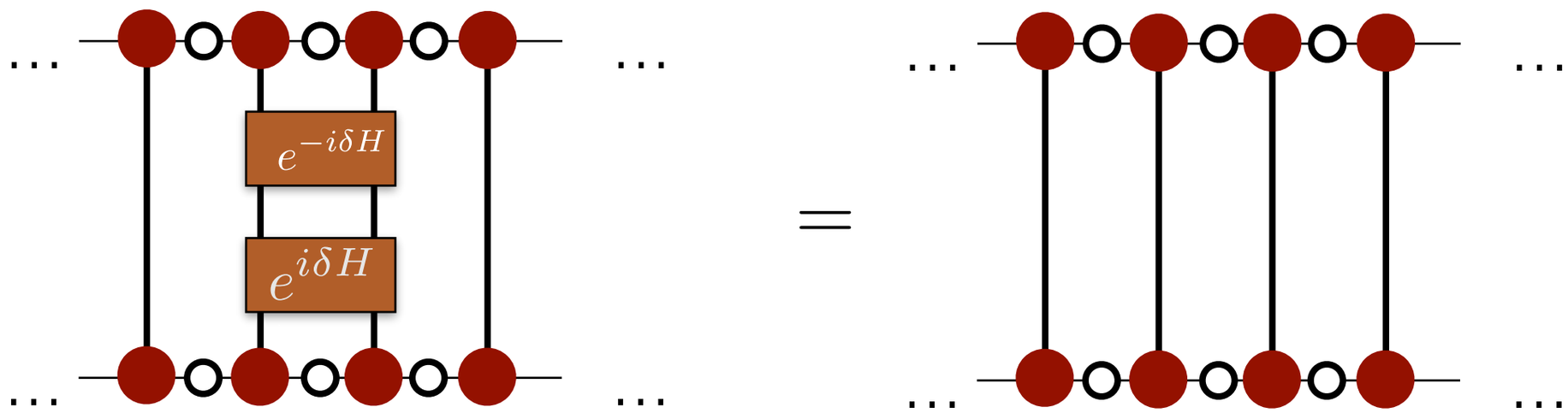
Calculation of the distance



*This is minimized by the matrix Θ

Why TEBD is accurate?

2. If the operator is unitary, MPS keeps canonical form within truncation error



*Unitary operator does not affect to the other Schmidt coefficients

If we chose the initial MPS as the canonical form,
TEBD algorithm almost keep it.
(So, TEBD is almost "globally optimal")

iTEBD and (i)TEBD for eigenvalue problems

Extension to infinite sytem iTEBD:

Finite system: TEBD

(G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007))

(R. Orús and G. Vidal, Phys. Rev. B **78**, 155117 (2008))

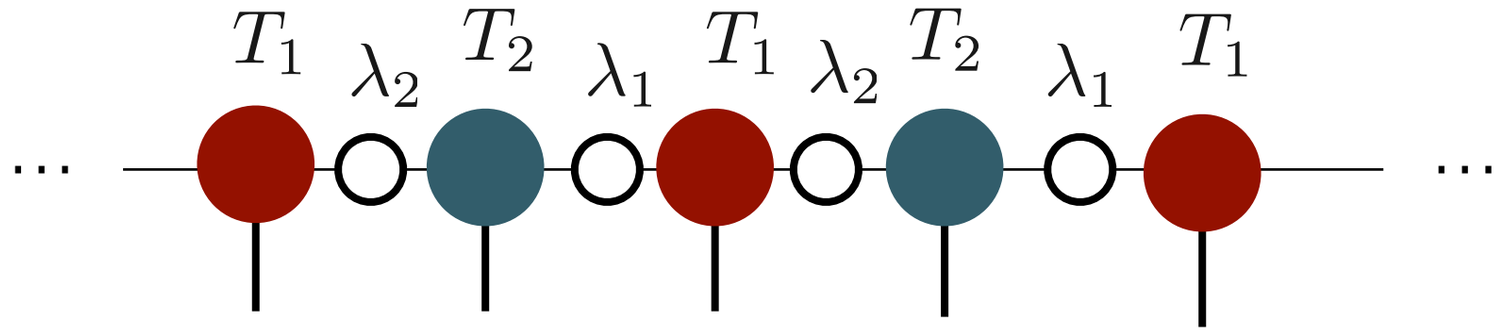
Sequentially apply ITE operators → $O(N)$ SVD for each step

Infinite system: iTEBD

Due to the translational invariance, all SVD are equivalent. → $O(1)$ SVD for each step

*Note

Because of SVD in iTEBD algorithm, we need at least two independent tensors even in translationally invariant system



(i)TEBD can be used for eigenvalue problem

Method to optimize MPS for GS of a specific Hamiltonian

1. Variational optimization

   Change matrix elements to reduce the energy: $\min_{T,\lambda} \dfrac{\langle\Psi|\mathcal{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle}$

2. Imaginary time evolution

Simulate imaginary time evolution: $|\Psi_{\mathrm{GS}}\rangle \propto \lim_{\beta\to\infty} e^{-\beta\mathcal{H}}|\Psi_0\rangle$
（虚時間発展）

For a initial state $\langle\Psi_{\mathrm{GS}}|\Psi_0\rangle \neq 0$

→ By replacing the time evolution operator to the imaginary time evolution operator,

$$e^{-i\mathcal{H}t} \to e^{-\tau\mathcal{H}} \qquad (t \to -i\tau)$$

We can use (TEBD) algorithm for eigenvalue problem.

Difference between TE and ITE

$e^{-i\mathcal{H}t}$:Time evolution operator is unitary

$e^{-\mathcal{H}\tau}$:Imaginary time evolution operator is not unitary

In general, by multiplying imaginary time evolution operator to MPS, the canonical form is destroyed and TEBD becomes less accurate.

However, when τ is small the operator is almost unitary.

(Because it is almost identity matrix)

If we chose the initial MPS as the canonical form,
TEBD algorithm almost keep it.
(So, TEBD is almost "globally optimal" even in the case of the imaginary time evolution.)

*Instead, we can transform the MPS into the canonical form after multiplying ITE operator in every steps.

Exercise 3: (TEBD and) iTEBD simulation (ITE)

3-1: TEBD simulation

Simulate small finite size system and compare energy with ED

Sample code: Ex3-1.py or Ex3-1.ipynb

3-2: iTEBD simulation

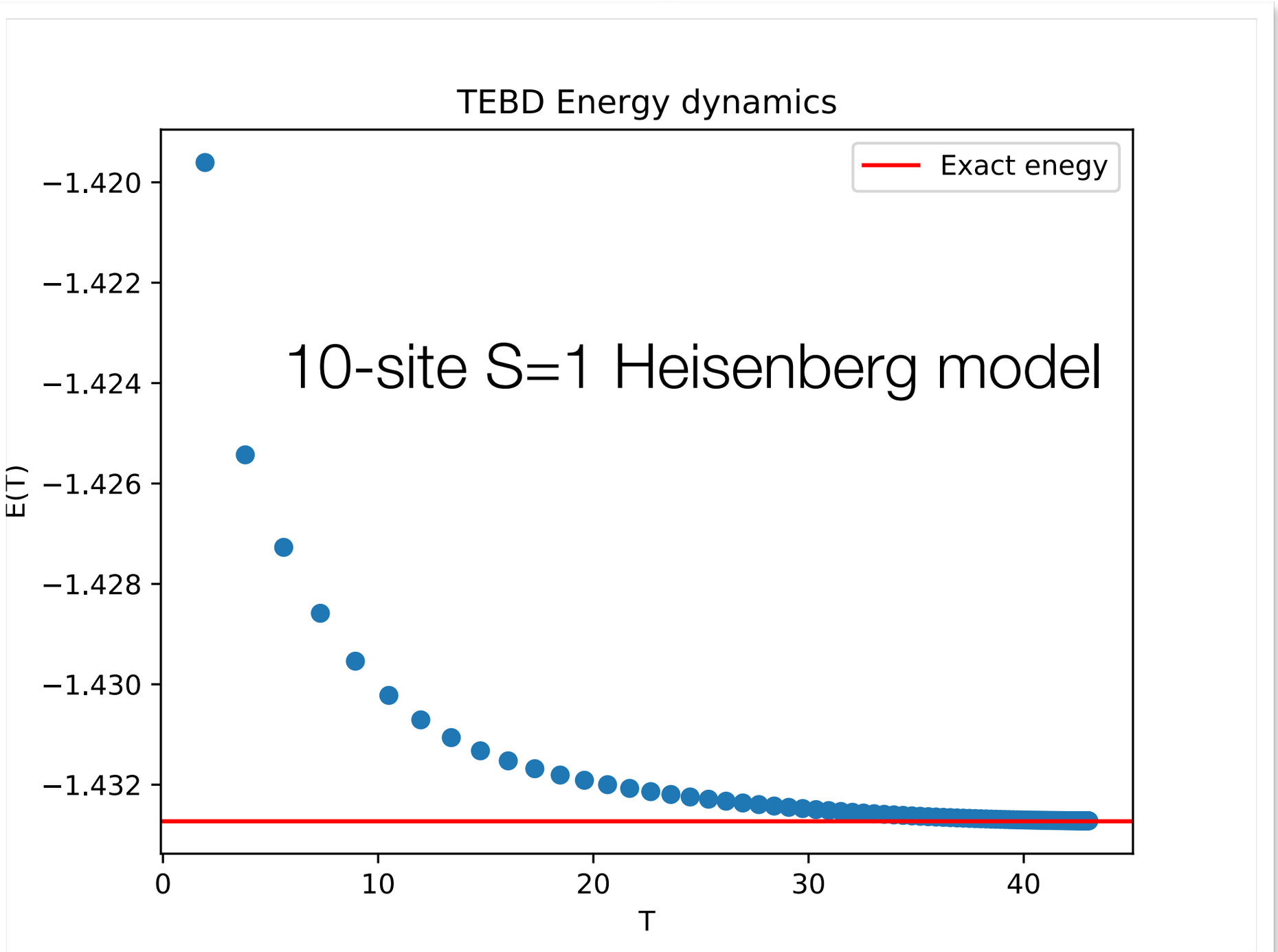
Simulate infinite system and calculate energy

Sample code: Ex3-2.py or Ex3-2.ipynb

*** Try simulation with different "chi_max", "T_step"**

*If you run them at Goole Colab, please upload ED.py and TEBD.py for Ex3-1.ipynb,
and please upload TEBD.py and iTEBD.py for Ex3-2.ipynb.

3-1: Energy dynamics in TEBD



Application to data science

E. Miles Stoudenmire and D. J. Schwab, NIPS 2016

Z.-Y. Han et al, Phys. Rev. X **8**, 031012 (2018).

Z.-F. Gao et al, Phys. Rev. Research **2**, 023300 (2020).

Machine learning for classification problem

Problem: we want to classify an input vector by several labels

E.g Classification of handwriting images

Standard procedure:

First, input vector $\boldsymbol{x}$ is mapped onto higher dimensional space

$$\vec{\psi}(\boldsymbol{x})$$ (non-linear feature map)

Then it is transformed to labels through a linear map

$$f^l = W^l \vec{\psi}(\boldsymbol{x})$$

In the case of supervised machine learning, we optimize W based on the correct labels of a lot of input vectors.

MPS representation of the classification problem

E. Miles Stoudenmire and D. J. Schwab, NIPS 2016

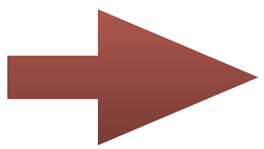
If we select a "product state" as a feature map

$$\psi_{i_1,i_2,\ldots,i_N}(\boldsymbol{x}) = \phi_{i_1}(x_1) \otimes \phi_{i_2}(x_2) \otimes \cdots \otimes \phi_{i_N}(x_N)$$

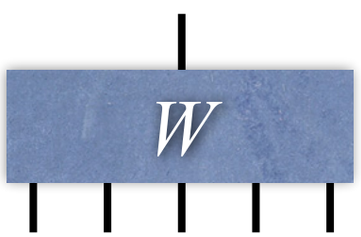
$$\vec{\psi} =$$

$$\vec{\phi}(x_1) \quad \cdots \quad \vec{\phi}(x_N)$$

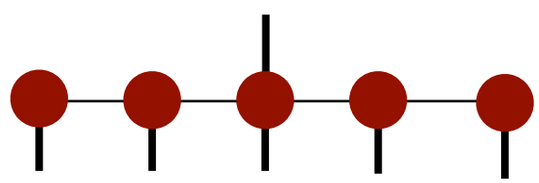
The dimension of vector space is a^N



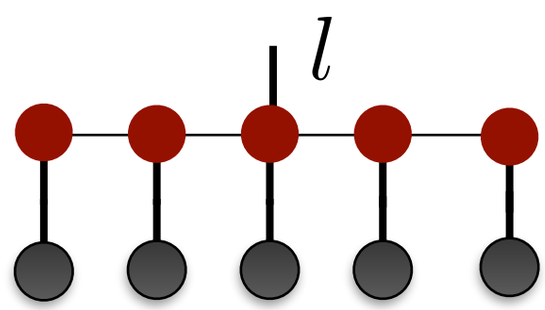
Then we can apply MPS approximation for W



$\simeq$



$$f^l = W^l \vec{\psi}(\boldsymbol{x}) =$$



MPS representation of the classification problem

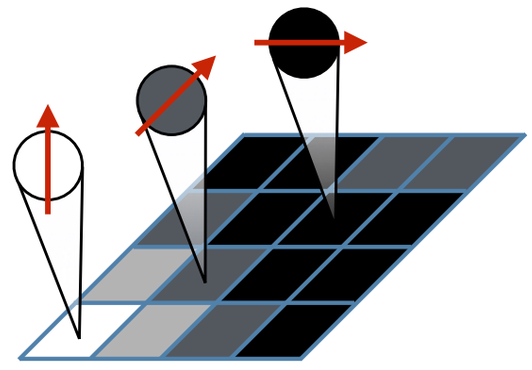
Feature map

E. Miles Stoudenmire and D. J. Schwab, NIPS 2016
https://github.com/emstoudenmire/TNML

$$\psi_{i_1,i_2,\ldots,i_N}(\boldsymbol{x}) = \phi_{i_1}(x_1) \otimes \phi_{i_2}(x_2) \otimes \cdots \otimes \phi_{i_N}(x_N)$$

Their feature map:

$$\phi^{s_j}(x_j) = \left[\cos\left(\frac{\pi}{2}x_j\right),\ \sin\left(\frac{\pi}{2}x_j\right)\right]$$



For the case of grayscale image

Application to MNIST database of handwritten digits

(handwritten numbers from 0 to 9)

χ	Test set error	
10	~5%	500/10000
20	~2%	200/10000
120	~0.97%	97/10000

<1%

It is comparable with the state of the art!

Unsupervised Generative Modeling

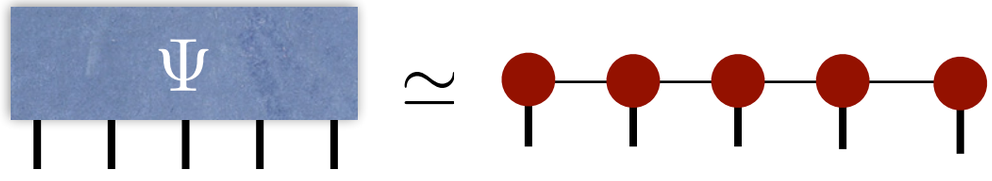
Z.-Y. Han et al, Phys. Rev. X **8**, 031012 (2018).

N pixel grey scale image → Binary data: $\vec{v} \in \mathbb{V} = \{0,1\}^{\otimes N}$

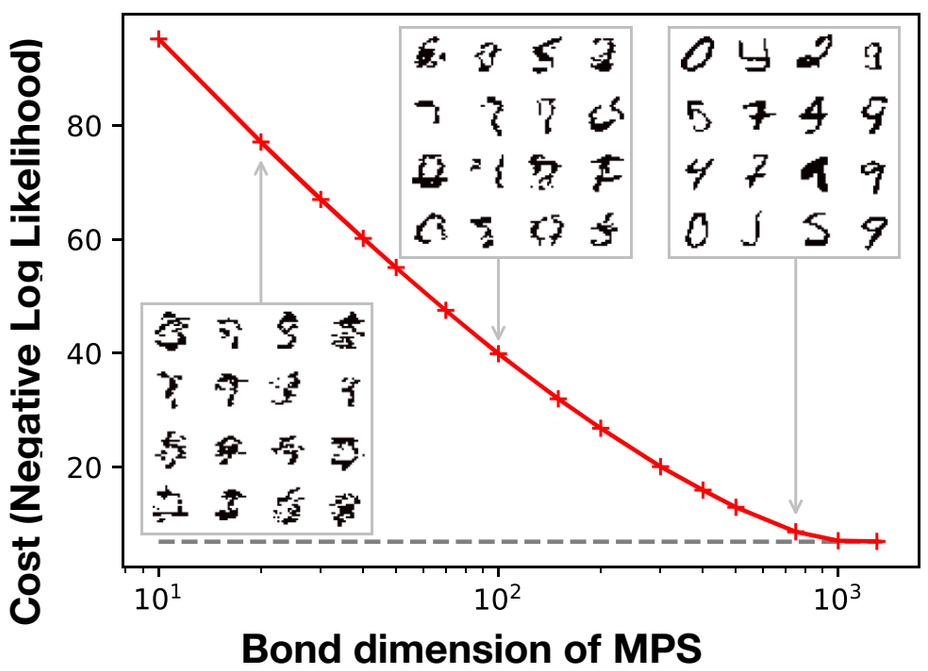
Same structure with qubits

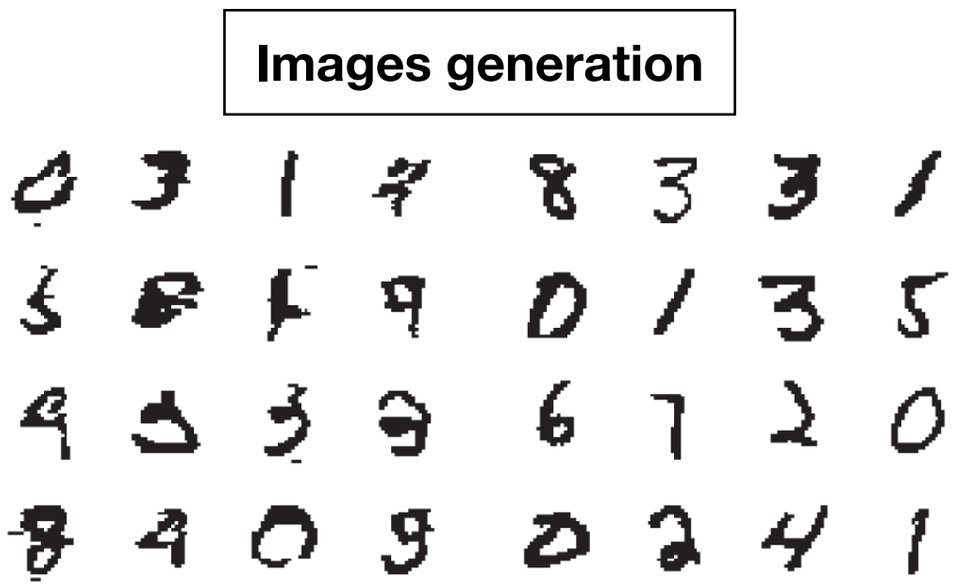
Probability distribution of images

$$P(\vec{v}) = \frac{|\Psi(\vec{v})|^2}{Z} \qquad (Z = \sum_{\vec{v}} |\Psi(\vec{v}_i)|^2)$$



Find optimal MPS to minimize $F = -\frac{1}{\mathbb{T}} \sum_{\vec{v} \in \mathbb{T}} \ln P(\vec{v})$ $\mathbb{T}$: Set of training data





(a) Generated (b) Original

Compressing deep neural network

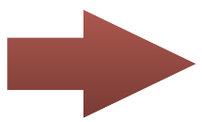
Z.-F. Gao et al, Phys. Rev. Research **2**, 023300 (2020).

MPO approximation of the weight matrix

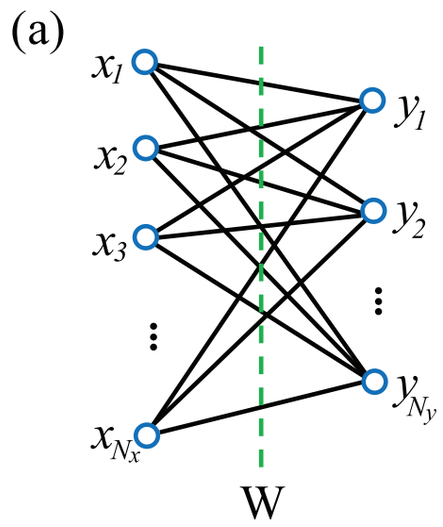
x_i: input neuron (pixel)

y_i: output neuron

W_{ij}: weight matrix connecting x and y



MPO approximation of W



Example: application to classification problems

TABLE I. Test accuracy a and compression ratios ρ obtained in the original and MPO representations of LeNet-5 on MNIST and VGG on CIFAR-10.

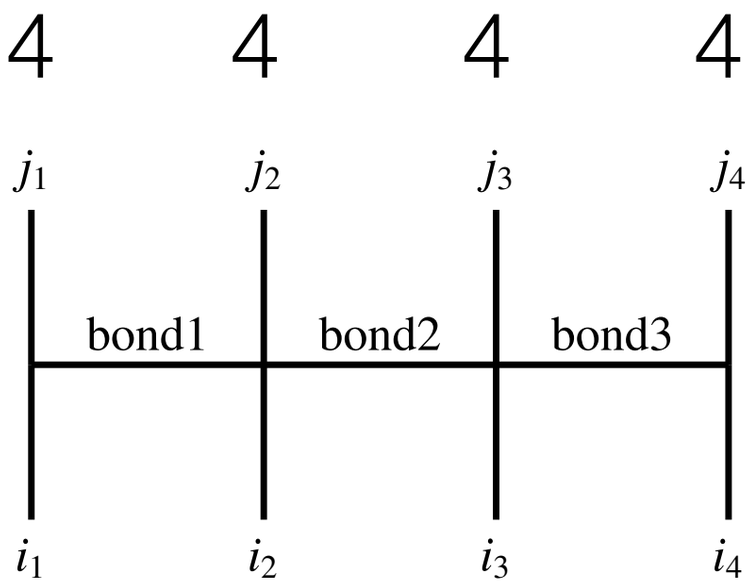
Data set	Network	Original Rep a (%)	MPO-Net a (%)	MPO-Net ρ
MNIST	LeNet-5	99.17 ± 0.04	99.17 ± 0.08	0.05
CIFAR-10	VGG-16	93.13 ± 0.39	93.76 ± 0.16	$\sim$0.0005
	VGG-19	93.36 ± 0.26	93.80 ± 0.09	$\sim$0.0005

a:**accuracy (%)**

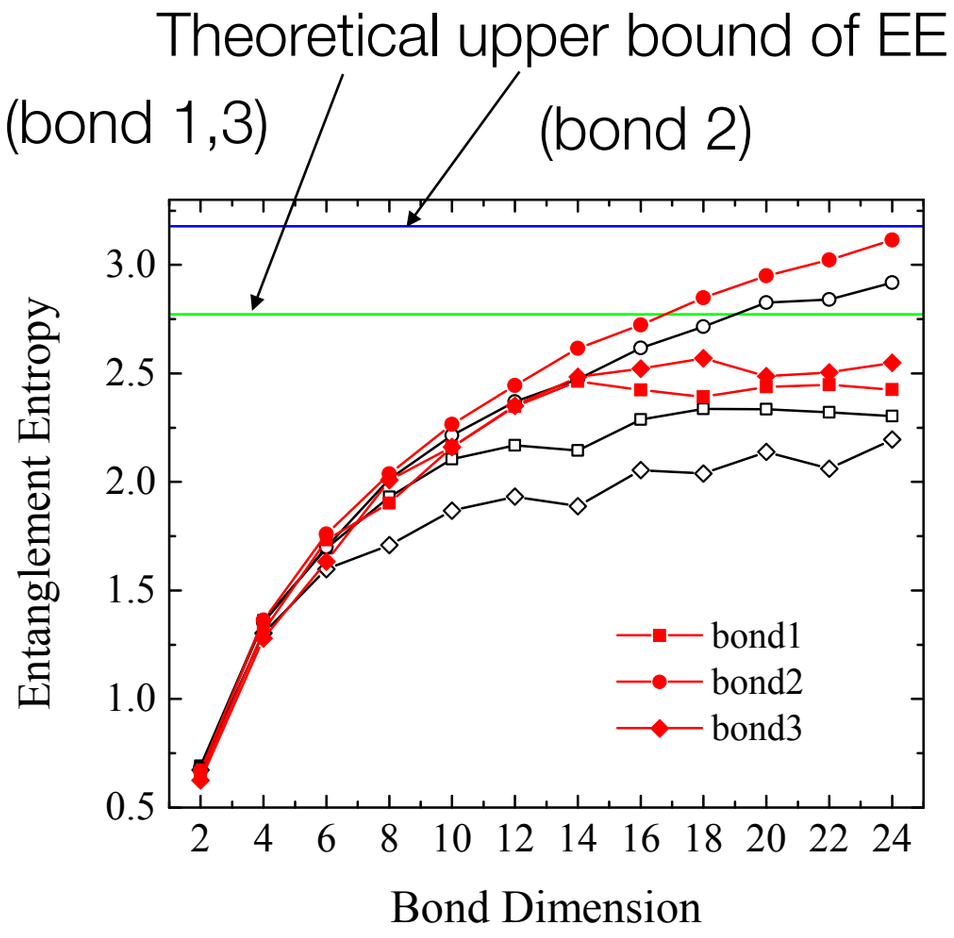
ρ:**compression ratio**

Entanglement entropy of (trained) MPO

Z.-F. Gao et al, Phys. Rev. Research **2**, 023300 (2020).



input: 28 x 28 pixel image



- Fashion-MNIST gives larger EE
  - It might be related to the difficulty of the dataset
- For bond 1 and 3, EE saturated smaller values than the theoretical upper bound
  - It indicate we can use MPO for approximation

Black: MNIST
Red: Fashion-MNIST
(more complicated)

References for application to machine learning

Low-Rank Tensor Networks for Dimensionality Reduction and Large-Scale Optimization Problems: Perspectives and Challenges PART 1

A. Cichocki, N. Lee, I.V. Oseledets, A.-H. Phan, Q. Zhao, D. Mandic

Foundations and Trends in Machine Learning, vol. 9, no. 4-5, pp. 249-429, 2016 (arXiv.1609.00893)

Tensor Networks for Dimensionality Reduction and Large-Scale Optimizations. Part 2 Applications and Future Perspectives

A. Cichocki, A-H. Phan, Q. Zhao, N. Lee, I.V. Oseledets, M. Sugiyama, D. Mandic

Foundations and Trends in Machine Learning: Vol. 9: No. 6, pp 431-673, 2017 (arXiv.1708.09165)

Topics:

- Supervised Learning with Tensors
- Tensor Train Networks for Selected Huge-Scale Optimization Problems
- Tensor Networks for Deep Learning
- ...

Next (Jan. 13th)

1st: Huge data in modern physics (Today)

2nd: Information compression in modern physics

(+review of linear algebra)

3rd: Review of linear algebra (+ singular value decomposition)

4th: Singular value decomposition and low rank approximation

5th: Basics of sparse modeling

6th: Basics of Krylov subspace methods

7th: Information compression in materials science

8th: Accelerating data analysis: Application of sparse modeling

9th: Data compression: Application of Krylov subspace method

10th: Entanglement of information and matrix product states

11th: Matrix product states + Application of MPS to eigenvalue problems

12th: Application of MPS to time evolution and data science

13th: Other tensor network representations

+ (Appendix: Information compression by tensor network renormalization)