Matrix Product State and Tensor Network

Lu Wei

ustc

2021

Table of Contents

Why Tensor Networks and its idea

- Dimension of Hilbert space grows exponentially with sites of lattice and space dimension.
- Local gap system (notice: decrete systems are born to have gap, it's for continuous systems to have gap)
- It's tedious to write a high-order tensor as T_{ijklm}. An interesting way to deal with it is to represent in in diagrammatic notation: One circle and many legs with the number of leg representing the order of the tensor and the length of leg representing the dimension of that leg.

Table of Contents

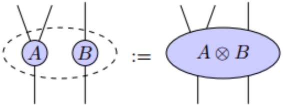
Notations of Tensor Networks

- Tensors

$$R^{\rho}{}_{\sigma\mu\nu} \implies$$

- Tensor operations

$$[A \otimes B]_{i_1,\dots,i_r,j_1,\dots,j_s} := A_{i_1,\dots,i_r} \cdot B_{j_1,\dots,j_s} \tag{1}$$

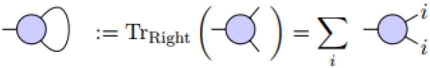


Notations of Tensor Networks

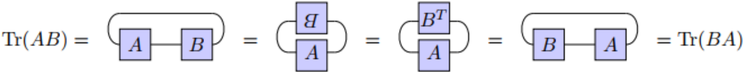
- Trace

$$[\mathrm{Tr}_{x,y}\, A]_{i_1,\ldots,i_{x-1},i_{x+1},\ldots,i_{y-1},i_{y+1},\ldots,i_r} = \sum_{\alpha=1}^{d_x} A_{i_1,\ldots,i_{x-1},\alpha,i_{x+1},\ldots,i_{y-1},\alpha,i_{y+1},\ldots,i_r} \tag{2}$$

Rank-3 for example:

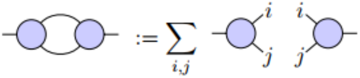


Property of trace in TN:

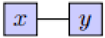
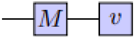
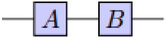


Notations of Tensor Networks

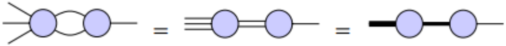
- Contraction



Examples of contraction:

Conventional	Einstein	TNN
$\langle \vec{x}, \vec{y} \rangle$	$x_\alpha y^\alpha$	
$M\vec{v}$	$M^\alpha{}_\beta v^\beta$	
AB	$A^\alpha{}_\beta B^\beta{}_\gamma$	
$\mathrm{Tr}(X)$	$X^\alpha{}_\alpha$	

- Grouping and Splitting



The space of tensors $\mathbb{C}^{a_1 \times \cdots \times a_n}$ and $\mathbb{C}^{b_1 \times \cdots \times b_m}$ are isomorphic as vector spaces whenever the overall dimensions match $(\prod_i a_i = \prod_i b_i)$. If we take a rank $n+m$ tensor, and group its first n indices and last m indices together to form a matrix

$$T_{I,J} := T_{i_1,\ldots,i_n;j_1,\ldots,j_m}$$

where we have defined our grouped indices as

$$\begin{aligned} I &:= i_1 + d_1^{(i)} \cdot i_2 + d_1^{(i)} d_2^{(i)} \cdot i_3 + \cdots + d_1^{(i)} \ldots d_{n-1}^{(i)} \cdot i_n \\ J &:= j_1 + d_1^{(j)} \cdot j_2 + d_1^{(j)} d_2^{(j)} \cdot j_3 + \cdots + d_1^{(j)} \ldots d_{m-1}^{(j)} \cdot j_m \end{aligned}$$

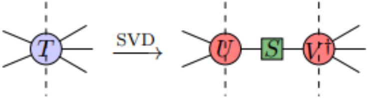
where $d_x^{(i)}$ $\left(d_x^{(j)}\right)$ is the dimension of the x th index of type $i(j)$. When such a grouping is given, we can now treat this tensor as a matrix, performing standard matrix operations.

Notations of Tensor Networks

An example of grouping and splitting is the singular value decomposition (SVD):

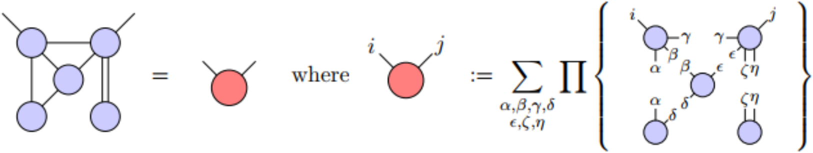
$$T_{i_1,\ldots,i_n;j_1,\ldots,j_m} = \sum_{\alpha} U_{i_1,\ldots,i_n,\alpha} S_{\alpha,\alpha} \bar{V}_{j_1,\ldots,j_m,\alpha} \tag{3}$$

Graphically the above SVD will simply be denoted



Notations of Tensor Networks

- Tensor networks
  A tensor network is a diagram which tells us how to combine several tensors into a single composite tensor:



- Bubbling

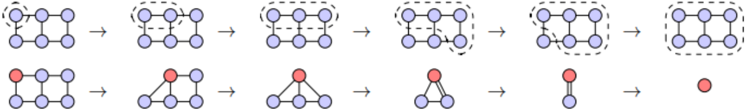


Table of Contents

Quantum information examples

- Bell states and the Bell basis

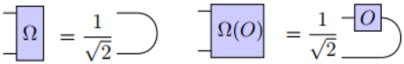
$$|\Phi^{\pm}\rangle := (|0\rangle \otimes |0\rangle \pm |1\rangle \otimes |1\rangle)/\sqrt{2} \tag{4}$$

$$|\Psi^{\pm}\rangle := (|0\rangle \otimes |1\rangle \pm |1\rangle \otimes |0\rangle)/\sqrt{2} \tag{5}$$

- Bell states in TN representation

$$|\Phi^{+}\rangle = |\Omega(I)\rangle, \quad |\Phi^{-}\rangle = |\Omega(Z)\rangle, \quad |\Psi^{+}\rangle = |\Omega(X)\rangle, \quad |\Psi^{-}\rangle \propto |\Omega(Y)\rangle \tag{6}$$

Where



and

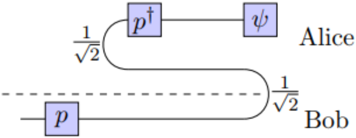
$$|\Omega\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \underset{\text{Matricise}}{\overset{\text{Vectorise}}{\rightleftarrows}} \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I/\sqrt{2}.$$

Quantum information examples

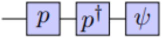
- Quantum teleportation

$$|\phi\rangle = \overbrace{\left(p_B\right)}^{\text{Correction}} \overbrace{\left(\langle\Omega_{A_1A_2}(p)|\right)}^{\text{Teleportation}} \overbrace{\left(|\psi_{A_1}\rangle \otimes |\Omega_{A_2B}\rangle\right)}^{\text{Setup}} = |\psi\rangle/2$$

$|\phi\rangle =$



$= \frac{1}{2}$



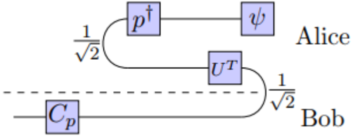
$= |\psi\rangle/2$

Quantum information examples

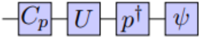
- Gate teleportation

$$|\phi\rangle = \overbrace{\left(C_p\right)}^{\text{Correction}} \overbrace{\left(\langle\Omega_{A_1A_2}(p)|\right)}^{\text{Teleportation}} \overbrace{\left(|\psi_{A_1}\rangle \otimes |\Omega_{A_2B}(U^T)\rangle\right)}^{\text{Setup}}$$

$=$



$= \frac{1}{2}$



$$= C_p U p^\dagger |\psi\rangle / 2$$

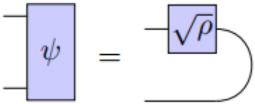
Quantum information examples

- Purification by Choi-isomorphism:

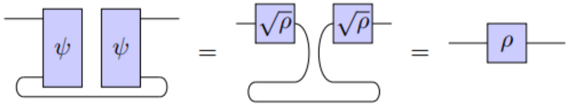
$$|\psi\rangle \propto (\sqrt{\rho} \otimes I)|\Omega\rangle = |\Omega(\sqrt{\rho})\rangle \tag{7}$$

$$(\sqrt{\rho} \otimes U)|\Omega\rangle \tag{8}$$

Corresponding tensor network:



and reduced density:

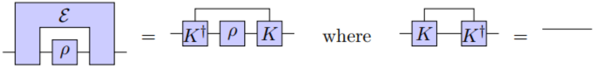
$\mathrm{Tr}_2\Big(|\psi\rangle\langle\psi|\Big) =$ 

Quantum information examples

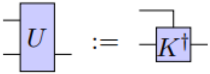
- Stinespring's Dilation Theorem

$$\mathcal{E}(\rho) = \mathrm{Tr}_1 \left[V^\dagger (\rho \otimes |0\rangle\langle 0|) V \right] \tag{9}$$

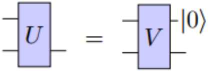
In TNN:



define

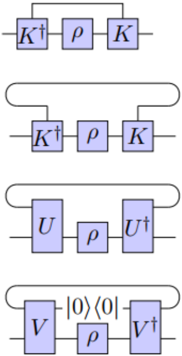


redefine



- Stinespring's Dilation Theorem

$$\mathcal{E}(\rho) = \sum_i K_i^\dagger \rho K_i =$$



$$= \mathrm{Tr}_1 \left[V^\dagger \left(\rho \otimes |0\rangle\langle 0| \right) V \right]$$

Table of Contents

Matrix product states

- Target – strongly interacting quantum many body systems; 1D quantum low energy states
- If you want to store 50 two-dim vectors' outcome T, you need a memory of petabytes. A wise solution is that you can represent T into many small tensors' contraction and one of the popular ways is matrix product state. The key ingredient is the recursive application of the singular value decomposition (SVD)
- Notice: SVD can not assure that the parameters grows linearly with n, also it's intrinsically a canonical form. A wave function of Ising model is itself a state of MPS.

Matrix product states

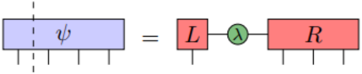
- General N-site spin system:

$$|\psi\rangle = \sum_{j_1 j_2 \ldots j_N = 0}^{d-1} C_{j_1 j_2 \ldots j_N} |j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_N\rangle \tag{10}$$

- Perform SVD and split the first index to get the Schmidt decomposition

$$|\psi\rangle = \sum_i \lambda_i |L_i\rangle \otimes |R_i\rangle \tag{11}$$

and graphically

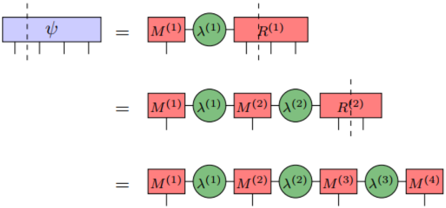


Matrix product states

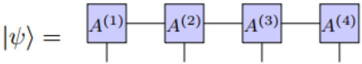
- Entanglement between L and R is given by

$$S_\alpha(\rho) = \frac{1}{1-\alpha} \log \operatorname{Tr} \rho^\alpha \tag{12}$$

- Continue SVD:



- Finally



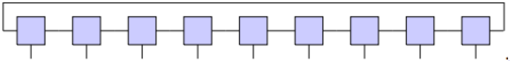
- What kind of state can be expressed into MPS?
  Any state with a so-called strong area law such that $S_0 \leq \log c$ for some constant c along any bipartition can be expressed using an MPS with only $\mathcal{O}\left(dNc^2\right)$ coefficients.
- Add periodic condition:

$$\left|\psi\left[A^{(1)}, A^{(2)}, \ldots, A^{(N)}\right]\right\rangle = \sum_{i_1 i_2 \ldots i_N} \operatorname{Tr}\left[A_{i_1}^{(1)} A_{i_2}^{(2)} \ldots A_{i_N}^{(N)}\right] |i_1 i_2 \ldots i_N\rangle \tag{13}$$

or in the translationally invariant case

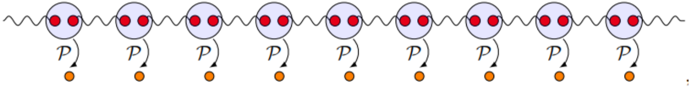
$$|\psi[A]\rangle = \sum_{i_1 i_2 \ldots i_N} \operatorname{Tr}\left[A_{i_1} A_{i_2} \ldots A_{i_N}\right] |i_1 i_2 \ldots i_N\rangle \tag{14}$$

- graphically

$|\psi[A]\rangle =$ 

Matrix product states

- 1D Projected Entangled Pair States

$|\psi\rangle =$ 

where

$|\phi\rangle =$ 

and

$$\sum_{j=0}^{d-1} |dd\rangle \tag{15}$$

- Linear projection

$$\mathcal{P} = \sum_{i,\alpha,\beta} A_{i;\alpha,\beta} |i\rangle\langle\alpha\beta| \tag{16}$$

Matrix product states

- Linear projection

$$
\begin{aligned}
&\mathcal{P}^{(1)} \otimes \mathcal{P}^{(2)} |\phi\rangle_{2,3} \\
&= \sum_{i_1,i_2;\alpha_1,\beta_1,\alpha_2,\beta_2,j} A^{(1)}_{i_1;\alpha_1,\beta_1} A^{(2)}_{i_2;\alpha_2,\beta_2} |i_1 i_2\rangle \langle \alpha_1\beta_1\alpha_2\beta_2| (\mathbb{I} \otimes |jj\rangle \otimes \mathbb{I}) \\
&= \sum_{i_1,i_2;\alpha_1,\beta_1,\beta_2} A^{(1)}_{i_1;\alpha_1,\beta_1} A^{(2)}_{i_2;\beta_1,\beta_2} |i_1 i_2\rangle \langle \alpha_1\beta_2|
\end{aligned} \tag{17}
$$

Matrix product states

- Some MPS states

  –Product State $|00\ldots0\rangle$

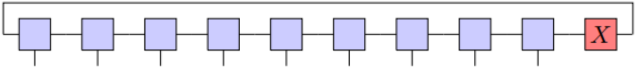
  $|\phi\rangle = |00\rangle + |11\rangle$ where $\begin{array}{l} A_0 = (1) \\ A_1 = (0) \end{array}$

  or $A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

  –W state $|W\rangle = \sum_{j=1}^{N} |000\ldots01_j000\ldots0\rangle$

  with $A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$

  and

  $|\psi[A]\rangle =$ 

  also $|\phi\rangle = |00\rangle + |11\rangle$

- Some MPS states

  –GHZ State $|GHZ\rangle = |00\ldots0\rangle + |11\ldots1\rangle$

  $|\phi\rangle = |00\rangle + |11\rangle$

  $\mathcal{P} = |0\rangle\langle 00| + |1\rangle\langle 11|$ $\quad A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$

  –AKLT State

  $\mathrm{SU}(2)$ spin- $1/2$ singlet as entanglement pairs $\quad |\phi\rangle = |01\rangle - |10\rangle$

  Projection onto Spin-1 subspace $\mathcal{P} : \mathbb{C}^{2\times 2} \to \mathbb{C}^3$

$$\mathcal{P} = |\tilde{1}\rangle\langle 00| + |\tilde{0}\rangle \frac{\langle 01| + \langle 10|}{\sqrt{2}} + |-\tilde{1}\rangle\langle 11|$$

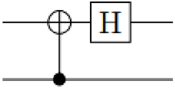
Matrix product states

- Some MPS states
  –Cluster state
  here $|\phi\rangle = |00\rangle + |11\rangle$

$$A_{00} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad A_{10} = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}$$
$$A_{01} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \quad A_{11} = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}$$

  or equivalently the map from virtual to physical spin-1/2 particles

$$\mathcal{P} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \Rightarrow$$

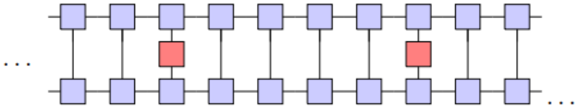


- Some MPS states
  –Cluster state
  The way to understand Cluster state:
  Initial state $\prod|\phi\rangle_{2j,2j+1}$ as unique ground state of $H = -\sum_i (X_{2j}X_{2j+1} + Z_{2j}Z_{2j+1})$
  Transformed by circuits P as

$$
\begin{aligned}
H' &= -\sum_j (Z_{2j-1}X_{2j}Z_{2j+1} + Z_{2j}X_{2j+1}Z_{2j+2}) \\
&= -\sum_k Z_{k-1}X_kZ_{k+1}. \quad \text{Cluster Hamiltonian}
\end{aligned}
$$

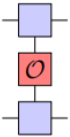
Matrix product states

- MPS properties
  –Decay of Correlations
  $\langle\psi[A]\,|\mathcal{O}_0\mathcal{O}_{j+1}|\,\psi[A]\rangle$:



Define O-transfer matrix :

$$\mathbb{E}_{\mathcal{O}} = \sum_{i,j=0}^{d-1} \mathcal{O}_{i,j} A_i \otimes \bar{A}_j =$$



Matrix product states

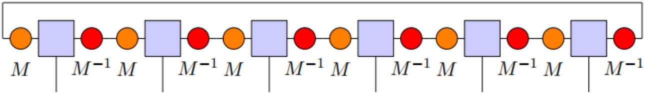
- MPS properties
  –Decay of Correlations
  The correlator (in the thermodynamic limit) can then be written as

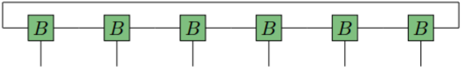
$$\begin{aligned}\langle \psi[A] | \mathcal{O}_0 \mathcal{O}_{j+1} | \psi[A] \rangle &= \mathrm{Tr}\left(\mathbb{E}^{\infty} \mathbb{E}_{\mathcal{O}_0} \mathbb{E}^{j} \mathbb{E}_{\mathcal{O}_{j+1}} \mathbb{E}^{\infty}\right) \\ &\propto V_L^{\dagger} \mathbb{E}^{j} V_R\end{aligned} \tag{18}$$

where V_L and V_R are the dominant left and right eigenvectors of $\mathbb{E}$ respectively.

Matrix product states

- MPS properties
  –Gauge freedom

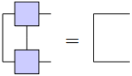
$|\psi[A]\rangle =$ 

$=$ 

$= |\psi[B]\rangle,$

where $B_j = MA_jM^{-1}$
M is only required to have a left inverse, so can be rectangular and enlarge the bond dimension:

$$\sum_{j=0}^{d-1} A_j^\dagger A_j = \mathbb{1}_{D\times D};$$



Matrix product states

- MPS properties
  –Block freedom Combine several MPS tensors $A_{i_1}, A_{i_2}, \ldots$, effective tensor B_k, on a larger physical region.

Table of Contents

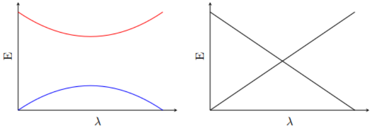
Matrix product states

- Quantum phase

  –Classical phase transition: nonanalytic behaviour of the free energy density

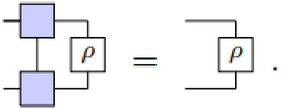
$$f(\beta, \mathrm{v}) = -\frac{\log \operatorname{tr} e^{-\beta H(\mathrm{v})}}{\beta} \quad \beta \to \infty$$

  Correlation will become long range at critical point
  $\langle \mathcal{O}_0 \mathcal{O}_x \rangle - \langle \mathcal{O}_0 \rangle \langle \mathcal{O}_x \rangle \sim |x|^{-\nu}$
  Connected by local Unitary transformation(can be expressed by some circuit)
  We say two quantum state $|\phi_0\rangle$ and $|\phi_1\rangle$ lie in the same quantum phase if there exist a continuous family $H(\lambda)$ with ground state $|\phi_0\rangle$ for $H(\lambda = 0)$ and $|\phi_1\rangle$ for $H(\lambda = 1)$ without gap closing for all $\lambda \in [0, 1]$:



- Injective MPS
  –If we assume the MPS is in left canonical form then injective MPS are those for which the identity is the unique eigenvalue 1 left eigenvector of the transfer matrix. Moreover this means that there exists a unique full-rank density matrix ρ which is a 1 right eigenvector:

$$\sum_{j=0}^{d-1} A_j \rho A_j^\dagger =: \mathcal{E}(\rho) = \rho$$

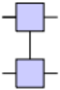


These MPS correspond to unique gapped ground states of local Hamiltonians.

Matrix pcroduct states

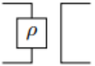
- No Topological Order in 1d
  Let A_j define some injective MPS, and construct the transfer matrix $\mathbb{E}$:

$$\mathbb{E} =$$



then

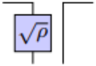
$$\mathbb{E}^k =$$



$$+ \tilde{\mathcal{O}}\left(|\lambda_2|^k\right)$$

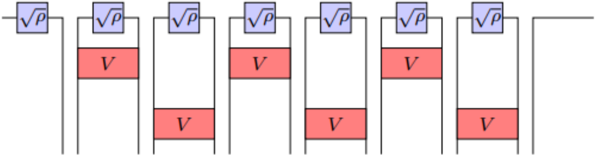
where $|\lambda_2| < 1$ is the second eigenvalue of the transfer matrix ρ, is the fixed point of the channel
Decompose to give a new effective MPS tensor describing the long wavelength physics

$$\tilde{A} =$$



Matrix pcroduct states

- No Topological Order in 1d
  let V be some unitary which acts as $\sum_{j,k} \sqrt{\rho}_{j,k} |j,k\rangle \to |0,0\rangle$



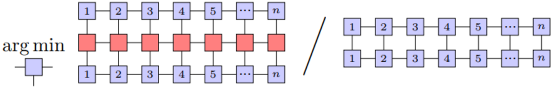
$\Rightarrow |00\cdots 0\rangle$

Table of Contents

Matrix pcroduct states

- Density Matrix Renormalization Group

$$|\Gamma\rangle := \underset{|\psi\rangle\in\mathcal{D}}{\arg\min}\frac{\langle\psi|H|\psi\rangle}{\langle\psi\mid\psi\rangle} \Rightarrow$$



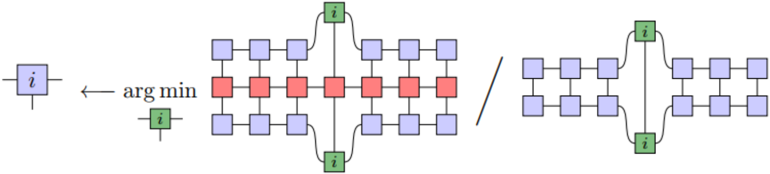
The key heuristic behind DMRG is to exploit the simplicity of these local problems, approximating the multivariate (multi-tensor) optimisation by iterated univariate (single tensor) optimisations.

Matrix pcroduct states

- Density Matrix Renormalization Group –One site(DMRG1)
  For a fixed site i, the sub-step involves fixing all but a single MPS tensor, which is in turn optimised over

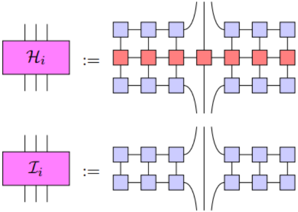
$$A_i \longleftarrow \underset{A_i}{\arg\min} \frac{\langle \psi(A_i) | H | \psi(A_i) \rangle}{\langle \psi(A_i) \mid \psi(A_i) \rangle}$$

$$\Downarrow$$

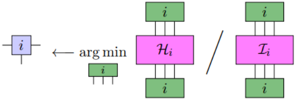


Matrix pcroduct states

- Density Matrix Renormalization Group –One site(DMRG1)
  Define the environment



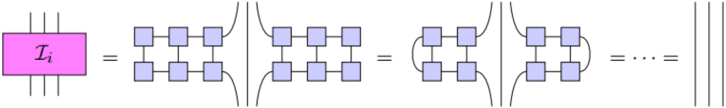
then we get



Vectorising this equation yields

$$A_i \longleftarrow \underset{Ac}{\arg\min} \frac{\langle A_i | \mathcal{H}_i | A_i \rangle}{\langle A_i | \mathcal{I}_i | A_i \rangle}$$

- Density Matrix Renormalization Group –One site(DMRG1)
  Fix the matrix left of our site in left-canonical form and the right of our site right-canonical form

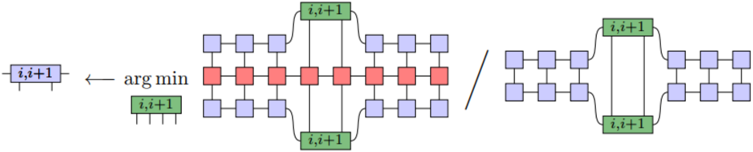


reduces to

$$A_i \longleftarrow \arg\min_{A_i} \frac{\langle A_i|\mathcal{H}_i|A_i\rangle}{\langle A_i|A_i\rangle}$$

As H_i is Hermitian, this optimisation has a closed form solution given by the minimum eigenvector of H_i. By sweeping back and forth along the chain, solving this localised eigenvector problem, and then shifting along the canonicalisation as necessary, we complete our description of the algorithm.

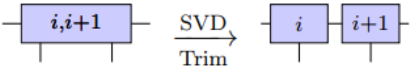
- Density Matrix Renormalization Group
  –two site(DMRG2)
  The idea with DMRG2 is to block two sites together, perform an optimization in the vein DMRG1, then split the sites back out this splitting process gives DMRG2 its power, allowing for dynamic control of bound dimension.



  Once again can be solved by taking the minimum eigenvector of an environment tensor with respect to two sites, $H_{i,i+1}$, once again in mixed canonical form.

Matrix pcroduct states

- Density Matrix Renormalization Group
  –two site(DMRG2)
  Split the 2-site tensor apart by doing SVD and a bond trimming



This trimmed SVD has two key features. Firstly the bond dimension to which we trim could be higher than that we originally started with, allowing us to gently expand out into the space of higher bond dimension MPS. Secondly we can use the truncated singular values to quantify the error associated with this projection back down into the lower bond dimension space, better informing our choice of bond dimension.

- Time-evolving Block Decimation
  Allows the dynamics of 1D spin systems to be simulated. By simulating imaginary-time-evolution low-temperature features such as the ground state may be calculated as well.

$$U(\tau) = e^{-\tau \sum_i h_i} \qquad \text{where} \qquad H = \sum_i h_i$$

  h_i is an interaction term acting on spins i and $i+1$
  Let $H_o\,(H_e)$ denote the sum of terms h_i for odd(even) i. As all the terms within $H_o\,(H_e)$ are commuting, $e^{-\tau H_o}$ $\left(e^{-\tau H_e}\right)$ can be efficiently computed and represented. The problem of approximating $U(\tau)$ can therefore be reduced to the problem of approximating $e^{-t(A+B)}$ when only terms of the form $e^{-\tau A}$ and $e^{-\tau B}$ can be computed.

- Time-evolving Block Decimation
  Exponential approximation

$$e^{-\tau(A+B)} = e^{-\tau A}e^{-\tau B} + \mathcal{O}\left(\tau^2\right)$$

  The TEBD algorithm works by approximating the imaginary-time-evolution operator by the above exponential product formulae, applying it to a given MPS, and trimming the bond dimension to project back down into the space of MPS.

- Time-evolving Block Decimation

  At each time step, we apply the evolution operator to immediatily MPS and update it. Suppose we want to apply an operator U to the spins at i and i+1. The idea is to apply the operator, contract everything into a single tensor, then once again use an SVD trimming to truncate the bond demension back down.

$$U(\tau) = e^{-\tau \sum_i h_i} \qquad \text{where} \qquad H = \sum_i h_i$$

  The TEBD algorithm works by approximating the imaginary-time-evolution operator by the above exponential product formulae, applying it to a given MPS, and trimming the bond dimension to project back down into the space of MPS.

Table of Contents

API of Tensor Network

- Environment Tensorflow and Tensor network
- Create nodes tn.Node(np.ones(10))
- Contract nodes tn.contract(edge)
- Edge-centric connection

```
a[0].is_dangling()
```

- Create a "trace" edge

```
trace_edge = a[0] ^ a[1]
```

- Axis naming

```
a = tn.Node(np.eye(2), axis_names=['alpha', 'beta'])
```

Table of Contents

- Jacob C. Bridgeman, Christopher T. Chubb. Hand-wabving and Interpretive Dance: An Introductory Course on Tensor Networks
- https://github.com/google/tensornetwork
- TensorNetwork: A Library for Physics and Machine Learning