Lecture 1: Linear Algebra and Quantum States

From linear algebra to tensor network representation of quantum states.

Contents

1. Basic linear algebra

I suggest you to read the bibles of linear algebra, one is **Introduction to Linear Algebra**[1] by Gilbert Strang and another is **Matrix Computation**[2] by Gene H. Golub and Charles F. Van Loan.

1.1. Singular Value Decomposition

Answer: Let us define a complex matrix $A \in \mathbb{C}^{m \times n}$, and let its singular value decomposition be

$$A = USV^\dagger$$

where U and V are unitary matrices and S is a diagonal matrix with non-negative real numbers on the diagonal. The columns of U and V are the orthonormal bases $\{|i\rangle_A\}$ and $\{|j\rangle_B\}$, respectively. If S has more than one non-zero elements, then the state is entangled.

In quantum physics, this is equivalent to the Schmidt decomposition, which is unique up to a global phase.

$$|\psi_{AB}\rangle = \sum_i \lambda_i |i\rangle_A \otimes |i\rangle_B$$

where λ_i are non-negative real numbers and $\sum_i \lambda_i^2 = 1$. It is easy to verify that the Schmidt coefficients λ_i correspond to the non-zero elements on the diagonal of S.

In the $|\psi\rangle_{\text{uniform}}$ example, the matrix A is

$$A = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Its singular value decomposition is

$$A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^\dagger$$

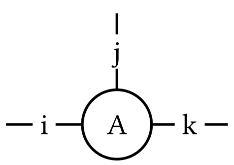
Only one singular value is non-zero, so the state is not entangled.

1.2. Eigenvalue Decomposition

1.3. QR Decomposition

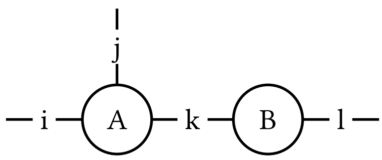
2. Tensor and tensor contraction

A rank 3 tensor A_{ijk} can be represented as



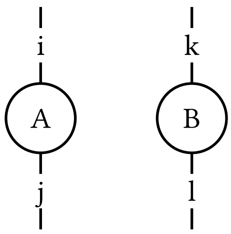
→ `ijk`

The contraction of two tensors A_{ijk} and B_{kl}, i.e. $\sum_k A_{ijk} B_{kl}$, can be diagrammatically represented as



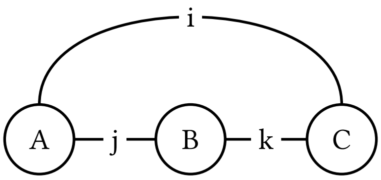
$\rightarrow$ `ijk,kl->ijl`

The kronecker product of two matrices A_{ij} and B_{kl}, i.e. $A_{ij} \otimes B_{kl}$, can be diagrammatically represented as



$\rightarrow$ `ij,kl->ijkl`

The operation $\mathrm{tr}(ABC)$ can be diagrammatically represented as



$\rightarrow$ `ij,jk,ki->`

From the diagram, we can see the trace permutation rule: $\mathrm{tr}(ABC) = \mathrm{tr}(CAB) = \mathrm{tr}(BCA)$ directly.

Definition (Tensor Network):
A tensor network[3], [4] is a mathematical framework for defining multilinear maps, which can be represented by a triple $\mathcal{N} = (\Lambda, \mathcal{T}, V_0)$, where:

- Λ is the set of variables present in the network $\mathcal{N}$.
- $\mathcal{T} = \left\{T_{V_k}\right\}_{k=1}^{K}$ is the set of input tensors, where each tensor T_{V_k} is associated with the labels V_k.
- V_0 specifies the labels of the output tensor.

Specifically, each tensor $T_{V_k} \in \mathcal{T}$ is labeled by a set of variables $V_k \subseteq \Lambda$, where the cardinality $|V_k|$ equals the rank of T_{V_k}. The multilinear map, or the **contraction**, applied to this triple is defined as

$$T_{V_0} = \mathrm{contract}(\Lambda, \mathcal{T}, V_0) \overset{\mathrm{def}}{=} \sum_{m \in \mathcal{D}_{\Lambda \setminus V_0}} \prod_{T_V \in \mathcal{T}} T_{V|M=m},$$

where $M = \Lambda \setminus V_0$. $T_{V|M=m}$ denotes a slicing of the tensor T_V with the variables M fixed to the values m. The summation runs over all possible configurations of the variables in M.

For instance, matrix multiplication can be described as the contraction of a tensor network given by

$$(AB)_{\{i,k\}} = \text{contract}\left(\{i, j, k\}, \left\{A_{\{i,j\}}, B_{\{j,k\}}\right\}, \{i, k\}\right),$$

where matrices A and B are input tensors containing the variable sets $\{i, j\}, \{j, k\}$, respectively, which are subsets of $\Lambda = \{i, j, k\}$. The output tensor is comprised of variables $\{i, k\}$ and the summation runs over variables $\Lambda \setminus \{i, k\} = \{j\}$. The contraction corresponds to

$$(AB)_{\{i,k\}} = \sum_j A_{\{i,j\}} B_{\{j,k\}}.$$

Diagrammatically, a tensor network can be represented as an **open hypergraph**, where each tensor is mapped to a vertex and each variable is mapped to a hyperedge. Two vertices are connected by the same hyperedge if and only if they share a common variable. The diagrammatic representation of the matrix multiplication is given as follows:

Here, we use different colors to denote different hyperedges. Hyperedges for i and k are left open to denote variables of the output tensor. A slightly more complex example of this is the star contraction:

$$\begin{aligned}&\text{contract}\left(\{i, j, k, l\}, \left\{A_{\{i,l\}}, B_{\{j,l\}}, C_{\{k,l\}}\right\}, \{i, j, k\}\right)\\&= \sum_l A_{\{i,l\}} B_{\{j,l\}} C_{\{k,l\}}.\end{aligned}$$

Note that the variable l is shared by all three tensors, making regular edges, which by definition connect two nodes, insufficient for its representation. This motivates the need for hyperedges, which can connect a single variable to any number of nodes.

3. Quantum state

Single qubit

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

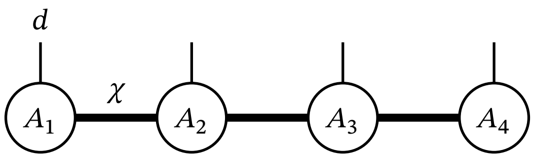
$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Quiz A — Representing a quantum state with tensor networks

1. *How to represent the product state $|0\rangle \otimes |0\rangle$ with tensor network diagram?*
2. *How to represent the GHZ state $\frac{|001\rangle + |111\rangle}{\sqrt{2}}$ with tensor network diagram?*

4. Matrix product state

1. Matrix product states



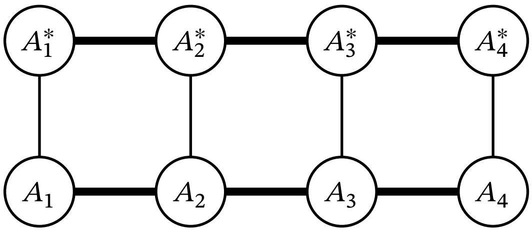
Quiz B — Data compression with matrix product states

What is the data compression ratio of a n-site matrix product state with bond dimension χ and local dimension d?

4.1. Tensor operations

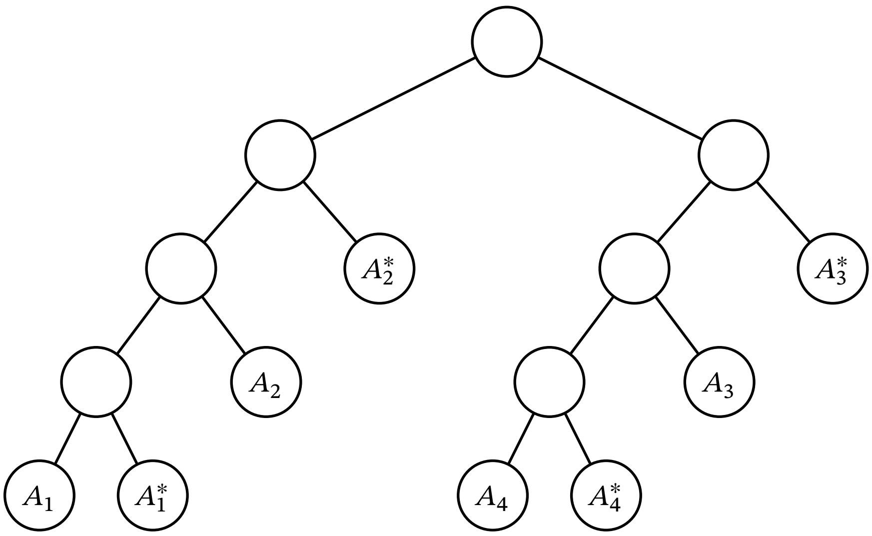
- Algorithm: convert a quantum state to a tensor network
- Convert a tensor network to a canonical form

4.2. Inner product and optimal contraction order



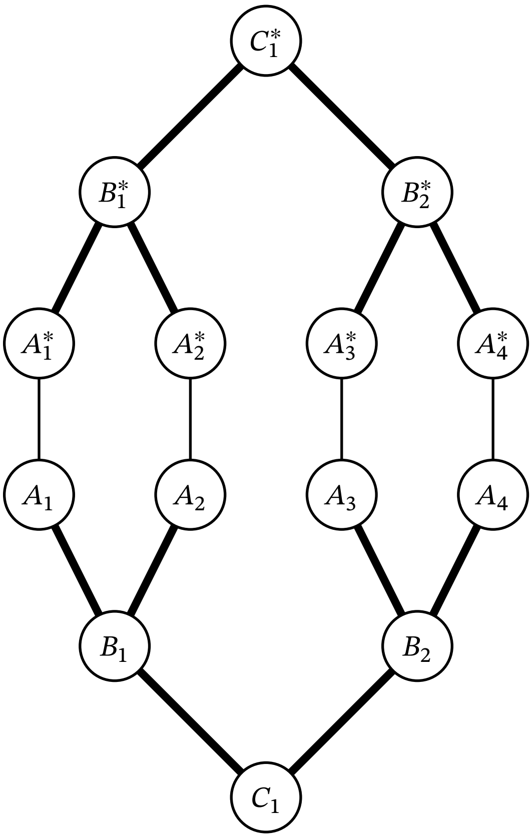
The contraction order can be represented as a contraction tree, where the leaves are the tensors and the internal nodes are the contractions. The goal of contraction order optimization is to minimize the computational cost, including the time complexity, the memory usage and the read/write operations. Multiple algorithms[5], [6] to find optimal contraction orders could be found in [this blog post](#).

One of the optimal (in space) contraciton order for the inner product of the two states is



Quiz C — Optimal contraction order of a tree tensor network

1. *What is the time complexity, memory usage and read/write operations of the above contraction order?*
2. *Given a tree tensor network, what is the optimal contraction order to compute the inner product of the two states? The inner product of a tree tensor network is diagrammatically represented as*



3. Hamiltonian and expectation value

Bibliography

[1] G. Strang, *Introduction to linear algebra*. SIAM, 2022.

[2] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU press, 2013. doi: 10.2307/3621013.

[3] J.-G. Liu, X. Gao, M. Cain, M. D. Lukin, and S.-T. Wang, "Computing solution space properties of combinatorial optimization problems via generic tensor networks." [Online]. Available: https://arxiv.org/abs/2205.03718

[4] M. Roa-Villescas, X. Gao, S. Stuijk, H. Corporaal, and J.-G. Liu, "Probabilistic Inference in the Era of Tensor Networks and Differential Programming," *arXiv preprint arXiv:2405.14060*, 2024, [Online]. Available: https://arxiv.org/abs/2405.14060

[5] G. Kalachev, P. Panteleev, and M.-H. Yung, "Recursive Multi-Tensor Contraction for XEB Verification of Quantum Circuits." 2021.

[6] J. Gray and S. Kourtis, "Hyper-optimized tensor network contraction," *Quantum*, vol. 5, p. 410–411, Mar. 2021, doi: 10.22331/q-2021-03-15-410.