Matrix Product States

Glenn LeBlanc, Karthik Siva *

May 10, 2021

Introduction

Many-Body Wavefunctions as Tensors

Consider a spin-$\frac{1}{2}$ particle. The particle's state is given by $|\psi\rangle \in \mathbb{C}^2$, and for some computational basis $\{|0\rangle, |1\rangle\}$ we can write

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

with

$$|\alpha|^2 + |\beta|^2 = 1.$$

This is the principle of superpostion– the particle is superposed between the two basis states $|0\rangle$ and $|1\rangle$. Now if we add a second spin-$\frac{1}{2}$ particle, the many-body system $|\Psi\rangle$ is in some superposition of the four states

$$|\Psi\rangle = \alpha |00\rangle + \beta |01\rangle + \gamma |10\rangle + \delta |11\rangle .$$

Generally, for N qubits (qudits) the system is fully parameterized by 2^N (d^N) complex numbers: $|\Psi\rangle \in \mathbb{C}^{2^N}$. Notice the exponential scaling here. It is useful also to notice the natural bijection between the two spaces

$$\mathbb{C}^{2^N} \longleftrightarrow \mathbb{C}^{2\times\cdots\times 2}$$

meaning we can instead conceptualize $|\Psi\rangle$ as a *tensor*:

$$|\Psi\rangle \in \mathbb{C}^{2\times\cdots\times 2}$$

where in this paper a tensor Ψ is just a multidimensional array with some number of indices such that plugging in an assignment for each index spits out a complex number. More succinctly,

$$\Psi_{i_1,i_2,\cdots,i_N} \in \mathbb{C}$$

A *contraction* between two tensors Ψ and Φ is a summation over a shared index:

$$T_{i,j,l,m} = \sum_k \Psi_{i,j,k}\Phi_{l,k,m}$$

is an example of a contraction. Note that dot products, matrix multiplication, and trace are all different vestiges of tensor contraction:

$$a \cdot b = \sum_k a_k b_k$$
$$(Ax)_i = \sum_k A_{ik}x_k$$
$$\mathrm{Tr}(A) = \sum_k A_{kk}$$

Tensor Networks

A *tensor network* is an undirected graph whose nodes represent tensors and whose edges correspond to tensor indices. An edge between two tensors corresponds to a contraction along the depicted axis of each tensor. Use of this graphical language for representing quantum systems is attractive since it unveils relevant entanglement properties [1].

*This project was part of the Berkeley Physics Directed Reading Program, which allows undergraduates to explore novel material under the auspices of a graduate student mentor. K. Siva very graciously directed this project.

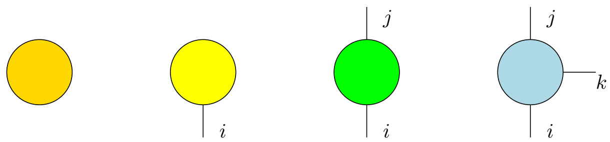

Figure 1: *A graphical depiction of a scalar c, vector v_i, matrix M_{ij}, and a tensor T_{ijk} of rank three.*

See figure 1 for a graphical depiction of tensors of rank zero to three. Each tensor is denoted as a node with free edges representing each index.

Figure 2 depicts the previous examples (dot product, matrix multiplication, trace) of tensor contraction using this graphical language.

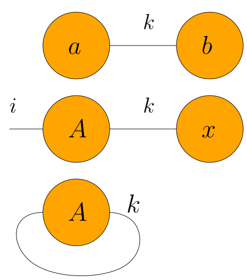

Figure 2: *A graphical depiction of three examples of tensor contraction.*

An example of an insight this graphical language provides is in proving trace cyclicality. The standard proof is as follows:

$$\begin{aligned}
\mathrm{Tr}(ABC) &= \sum_{ijk} A_{ij} B_{jk} C_{ki} \\
&= \sum_{ijk} C_{ki} A_{ij} B_{jk} \\
&= \mathrm{Tr}(CAB)
\end{aligned}$$

The tensor network proof is depicted in figure 3:

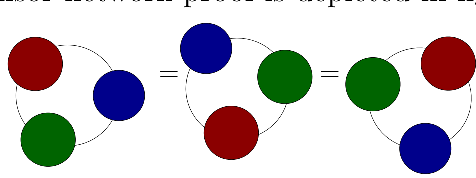

Figure 3: *Proof of trace cyclicality.*

The graphical depiction of trace provides simpler insight into the cyclic structure of the underlying tensor contractions required to calculate $\mathrm{Tr}(ABC)$, and thus allows for a totally visual and immediately obvious proof of the invariance of trace under cyclic permutations of A, B, and C.

Matrix Product States

A *matrix product state* (MPS) is a particular class of tensor network consisting of a chain of tensors each having one dangling edge and a bond between their nearest neighbors. See figure 4 for an example of a MPS with three sites with periodic and nonperiodic boundary conditions. From here we only consider MPS with nonperiodic boundary conditions.

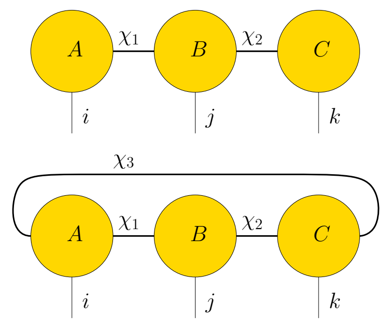

Figure 4: *Two three-site MPS, the first lacking periodic boundary conditions and the second with periodic boundary conditions.*

The indices χ_i in figure 4 are called *bond indices* and are associated with a *bond dimension*. The free indices i, j, k are *site indices*. Algebraically, the MPS decomposition of a tensor Ψ is written as:

$$\Psi_{i_1, i_2, \cdots, i_N} = \sum_{\chi_1, \chi_2, \cdots, \chi_{N-1}} A_{i_1}^{[1]\chi_1} A_{i_2}^{[2]\chi_1, \chi_2} \cdots A_{i_N}^{[N]\chi_{N-1}}$$

where the first and last local tensors $A^{[1]}$ and $A^{[N]}$ matrices and every other tensor in the chain is of rank three. For a given choice of site indices (e.g., $i_1 = 0$, $i_2 = 1$, $i_3 = 0$, $\cdots$), the coefficient $\Psi_{010\cdots}$ is given by a matrix product– hence the name matrix product state.

Generating a MPS from a tensor

The first step of the iterative process for generating a MPS from a tensor is depicted in figure 5 for an initial tensor with three indices, each of dimension d. The input tensor ψ is reshaped into a matrix ψ_{MAT} by squashing indices the two leftmost indices into one index while keeping the rightmost index separated. The singular value decomposition is then performed on ψ_{MAT}. Σ is truncated and renormalized to Σ' such that the bond index between V and Σ is less than or equal to the bond dimension χ. The truncated matrix Σ' is contracted to the left into U, resulting in the $d^2 \times \chi$ matrix ψ'_{MAT}. Finally, the index of dimension d^2 is unsquashed into two indices of dimension d each. The leftmost site in figure 5 is an orthonormal matrix by definition of the SVD. Non-boundary sites in the final chain resulting from iteratively applying this method are *right normal*, which is a loose generalization of orthonormality. Algebraically, if a tensor $T_{L,i,R} \in \mathbb{C}^{b\times d\times b}$ is right normal, then

$$\sum_{i,R} T_{L,i,R} T^*_{L',i,R} = \delta_{L,L'}$$

and likewise if T is left normal then

$$\sum_{i,L} T_{L,i,R} T^*_{L,i,R'} = \delta_{R,R'}.$$

See figure 6 for a graphical depiction of this property.

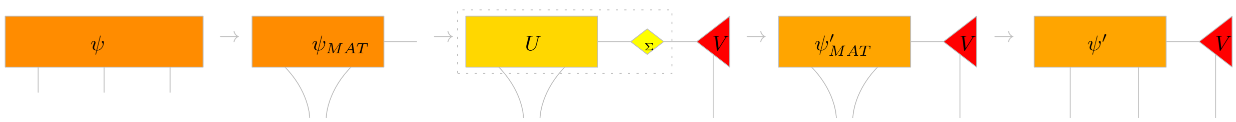

Figure 5: *Separating the first site index from ψ.*

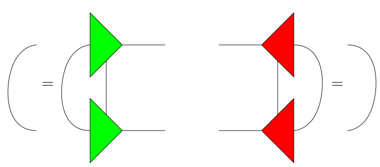

Figure 6: *Left normality (green) and right normality (red).*

Moving the Orthogonality Center

If we have a MPS in

Time Evolving Block Decimation

Package Overview

This section provides an overview of the meat of this project: a Julia package for creating and manipulating matrix product states, located at `https://github.com/gl3nnleblanc/pdrp2021`.

Julia

Julia is a modern programming language incubated at MIT in 2009 and designed from the beginning with high performance in mind [2]. Julia is fast, easy to use, and open source.

Algorithms

Examples

References

[1] R. Orus, A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States, Annals Phys. 349 (2014) 117–158. arXiv:1306.2164, doi:10.1016/j.aop.2014.06.013.

[2] J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, Julia: A fresh approach to numerical computing, SIAM review 59 (1) (2017) 65–98.
URL `https://doi.org/10.1137/141000671`