

Ideas in Science

Tensor Networks

Matthew Thomson, University of Alberta

Talk Outline

- What are Tensors?
- What do Tensors do?
- Making Tensor Networks
- How Computers Compute Tensors
- Applications
- Summary

What are Tensors?

For our purposes today,

Tensors $\equiv$ Multi-dimensional Arrays $\equiv$ Matrix-like Objects With Many Indices

You've all used some basic tensors: Matrices!

$$\begin{array}{c} A \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \\ \begin{array}{c} i \\ j \end{array} \end{array} = i \text{ --- } \boxed{A} \text{ --- } j = A_{ij}$$

What are Tensors?

They appear in many contexts in physics:

- Electromagnetic Field Strength Tensor $F^{\mu\nu}$
- Cauchy Stress Tensor σ_{ij}
- Levi-Civita Symbol ϵ_{ijk}
- Quadrupole Tensor Q_{ij}

Technically the strict physics definition of a tensor (field) involves how the object transforms under coordinate transformations.

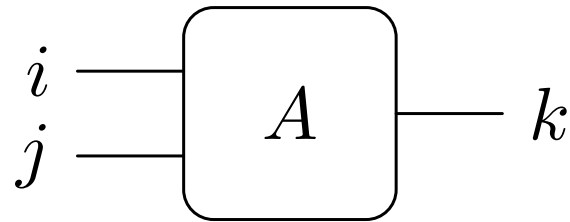
But for the talk today my definition will be loose: Any object that you can index and get numbers out of is a tensor.

What do Tensors do?

Tensors are a natural extension of matrices from linear algebra.

Many matrix operations can be extended to tensors.

First, let's get the hang of manipulating tensor indices. Here's A_{ijk} with three:



Each index has a particular dimension, i.e.

$$i = 1, 2, \dots, N$$

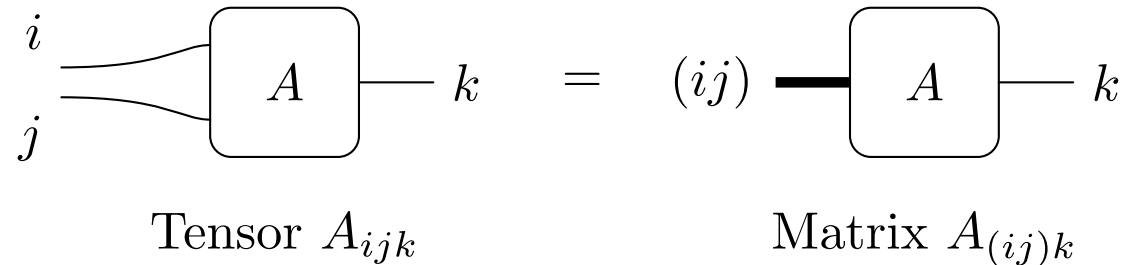
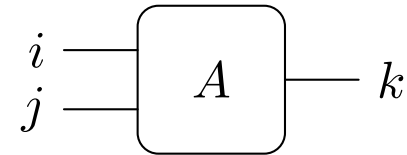
$$j = 1, 2, \dots, M$$

etc.

What do Tensors do? Tensor Indices

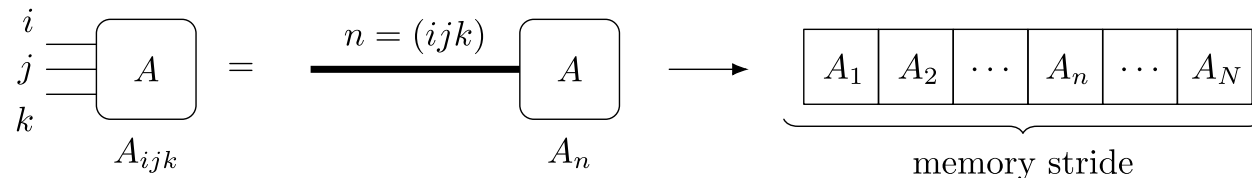
The rank of a tensor = # of indices

- Scalar: rank 0
- Vector: rank 1
- Matrix: rank 2
- Tensor: rank ≥ 3



Indices can be combined or grouped together.

Your computer stores (dense) tensors in memory like this! The two indices are reshaped into a single vector index.



What do Tensors do? Tensor Contractions

Basic tensor contraction: just matrix multiplication

$$\sum_j A_{ij} B_{jk} = i \text{ --- } \boxed{A} \text{ --- }^j \boxed{B} \text{ --- } k = i \text{ --- } \boxed{C} \text{ --- } k = C_{ik}$$

Contraction with more indices:

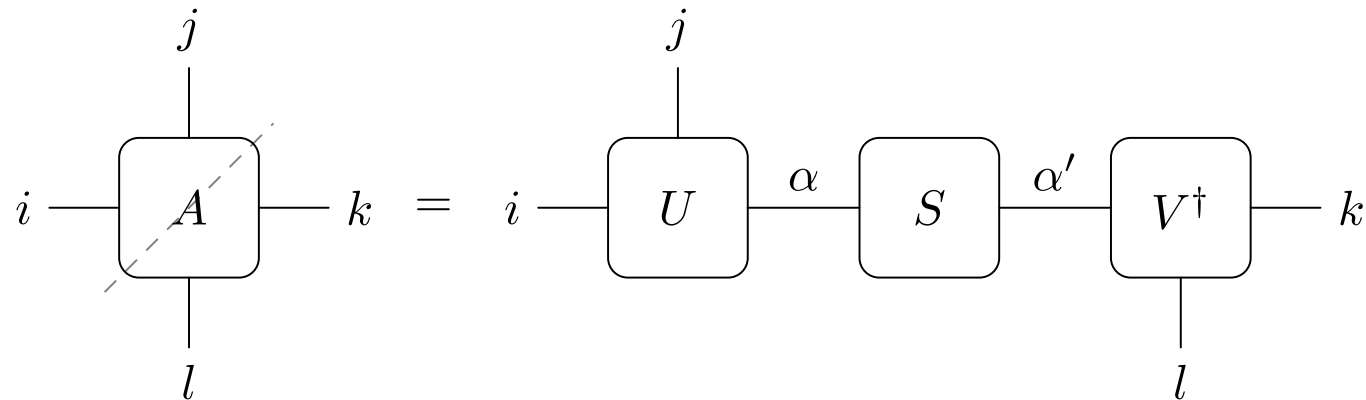
$$\sum_k A_{ijk} B_{klm} = i \text{ --- } \boxed{A} \text{ --- }^j \boxed{B} \text{ --- }^l \text{ --- } m = i \text{ --- } \boxed{C} \text{ --- } m = C_{ijlm}$$

You can always combine indices to make a particular contraction into a matrix product.

What do Tensors do? Tensor Decompositions

Any of your favourite matrix decompositions work with tensors.

For example, Singular Value Decomposition (SVD):



S contains the singular values of the decomposition, U and $V^\dagger$ are unitary matrices.

All it takes is combining indices until the tensor is a matrix, then doing the decomposition.

Making Networks

A tensor network is just a collection of tensors with common indices contracted over.

For example, in QM a "Matrix" Product State (MPS) is stored as a collection of tensors, but is contracted over to compute things.

We represent the network with shared indices joined together.



The contraction order is important! Doing it inefficiently can be extremely slow.

How Computers Compute Tensors

Ever heard of Nvidia's tensor cores?

They're just parallelized matrix multiplication units!

$$\begin{array}{c} A \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \end{array} \times \begin{array}{c} B \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \end{array} + \begin{array}{c} C \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \end{array} = \begin{array}{c} D \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \end{array}$$

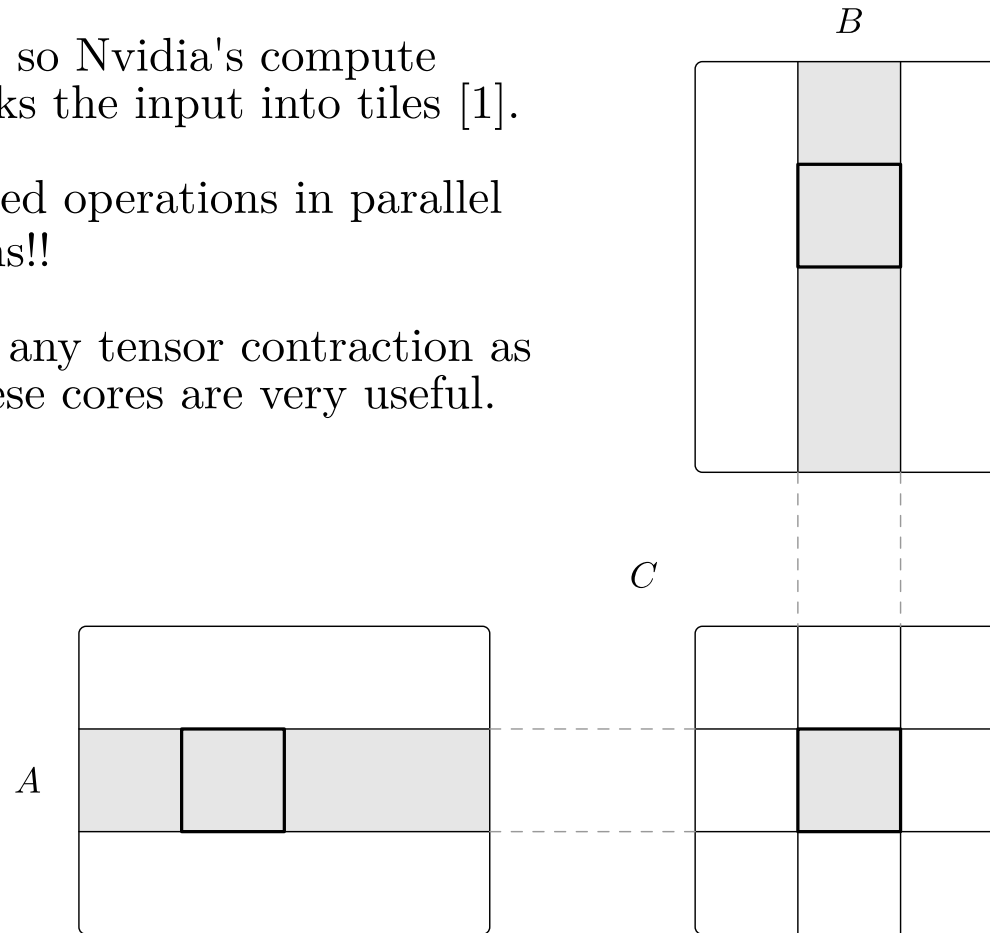
They do 4×4 generalized matrix products $AB + C = D$ in a single clock cycle [1].

How Computers Compute Tensors

4×4 is pretty small, so Nvidia's compute interface CUDA breaks the input into tiles [1].

A GPU does these tiled operations in parallel
→ huge accelerations!!

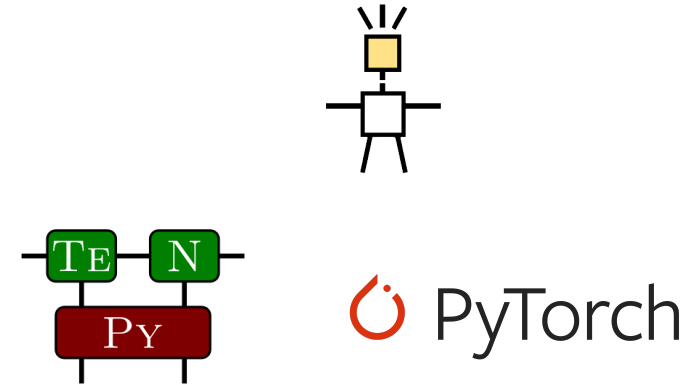
Since we can reshape any tensor contraction as a matrix product, these cores are very useful.



Software

Many many libraries out there for working with tensors.

- Julia: ITensor, TensorKit, ...
- Python: TeNPy, PyTorch + TensorFlow, NumPy, ...
- C++: ATen, ITensor (old), ...
- MATLAB: Tensor Toolbox
- + more!



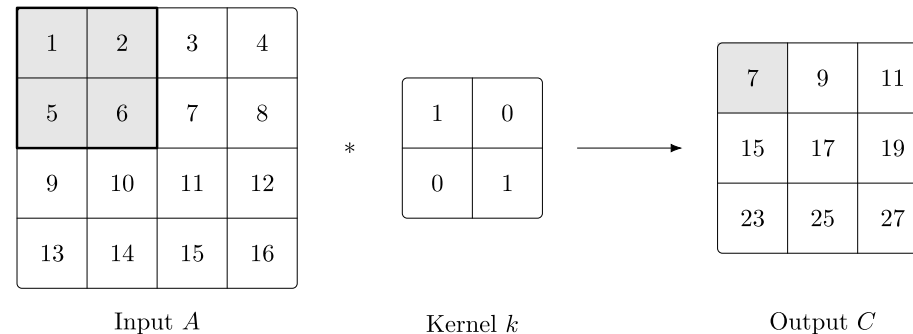
Any language which supports multidimensional arrays, reshaping, and permuting dimensions also works!

Many of these can interface with compute APIs like CUDA to get GPU acceleration.

Applications 1: Machine Learning, CNNs

Preface: I am not an expert! I've only dabbled with convolutional neural networks (CNNs)

CNNs work by passing image data through a series of convolution kernel/filter layers [2].



Don't quote me, but I think the formula for a single output element c of C is

$$c = \text{tr} \left(A_{ij} (k^T)_{jk} \right)$$

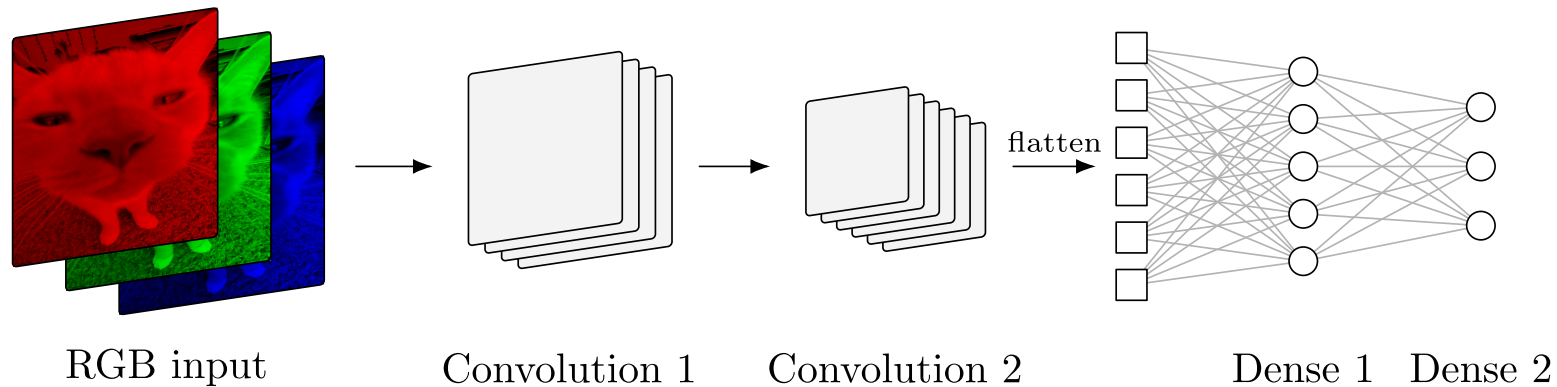
These kernels help distinguish edges, features, etc., just like how the eye works!

Training the network + kernels, you can do image classification and recognition!

Applications 1: Machine Learning, CNNs

Often, the image data is split into a 3D array (2 spatial dimensions $\times$ 3 RGB channels or similar).

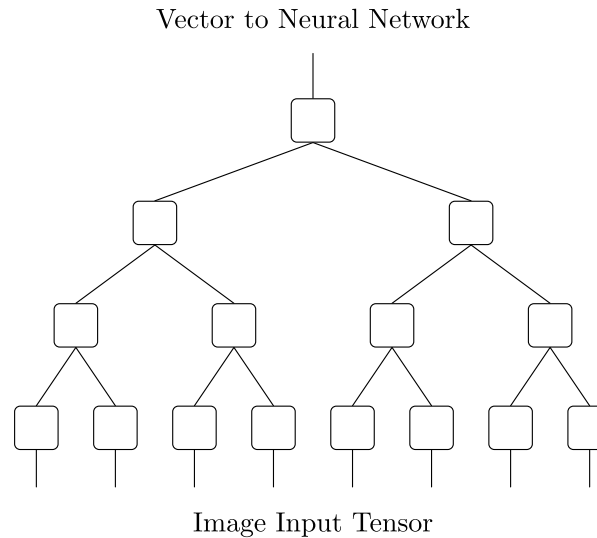
This input data is therefore a tensor.



The convolution kernels can then be thought of as a (messy) sequence of small tensor operations.

Applications 1: Machine Learning, CNNs

Each convolution of the kernel with the input data is a set of tensor contractions. These form a Tree Tensor Network (TTN)[3, 4]:



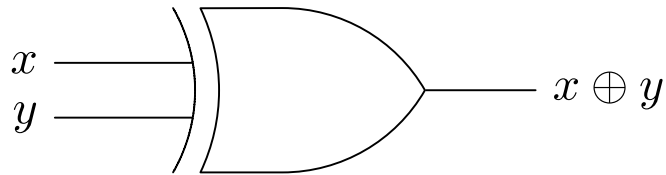
We will see this structure again in a bit.

Applications 2: Quantum Computing and Circuits

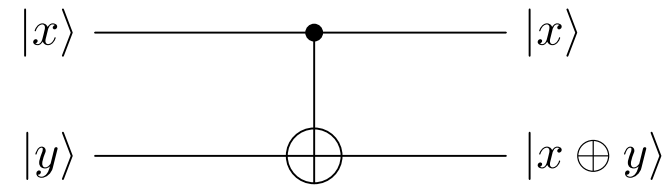
Fundamentally, quantum computing algorithms are sequences of operations.

These operations end up forming circuits, in analogy to classical logic [5]:

Classical XOR



Quantum CNOT



All the quantum mechanical operators appear as tensors in these diagrams.

Simulating quantum computers can be done efficiently with these networks.

Applications 3: Quantum Systems, MPS

Calculating the ground state of large many-body quantum systems is very difficult.

Matrices grow exponentially with the system size!!

In one dimension, it has been shown that many quantum ground states can be approximated by Matrix Product States (MPS).

The (essentially) compressed wavefunction ends up being represented as a tensor network.



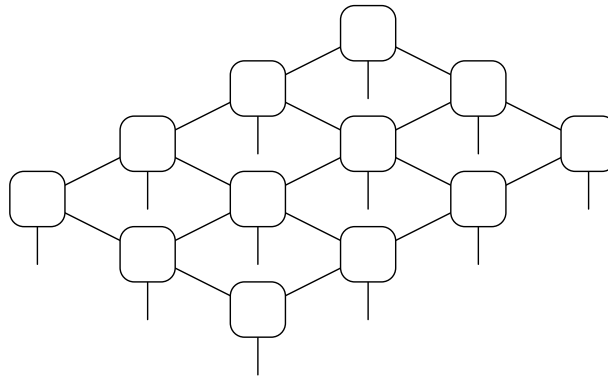
Shared indices are so-called virtual bonds.

These bonds carry information about the entanglement between tensors.

Applications 4: Quantum Systems, PEPS

This sort of decomposition sort of works for higher dimensions.

Projected Entangled Pair States (PEPS) in 2D [6, 4]:

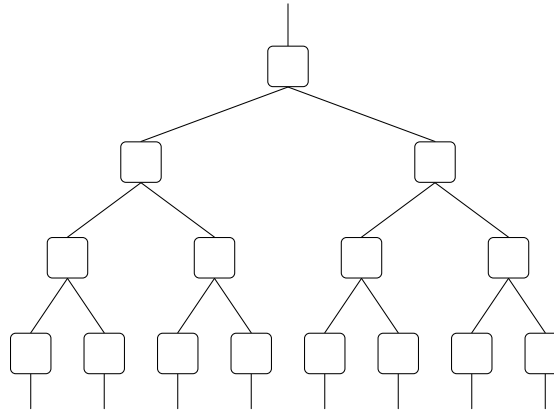


Major problems with scaling though. Contractions become very costly.

Even harder to scale to 3D.

Applications 5: Quantum Systems, TTN

Tree Tensor Networks [7]:



These networks encode a Wilsonian Renormalization Group structure by how they coarse grain degrees of freedom [4].

This is the same form the CNNs had!

There's a relation between certain deep learning techniques and quantum mechanics [3, 4].

Applications 6: Quantum Algorithms, DMRG

In just 1D, the Density Matrix Renormalization Group (DMRG) algorithm lets us find ground states for cheap using Matrix Product States.

Originally formulated in [8, 9], MPS proven in [10, 11], Reviewed well in [12, 13]

Effectively boils down to a variational optimization problem.

At each step, one tensor in the MPS is optimized against the Hamiltonian.



By tweaking all of the tensors multiple times, we converge to the ground state.

This process is mathematically optimal for compressing the wavefunction!

Can also be applied in quantum chemistry for simple molecules!

Applications 7: Quantum Systems, TEBD

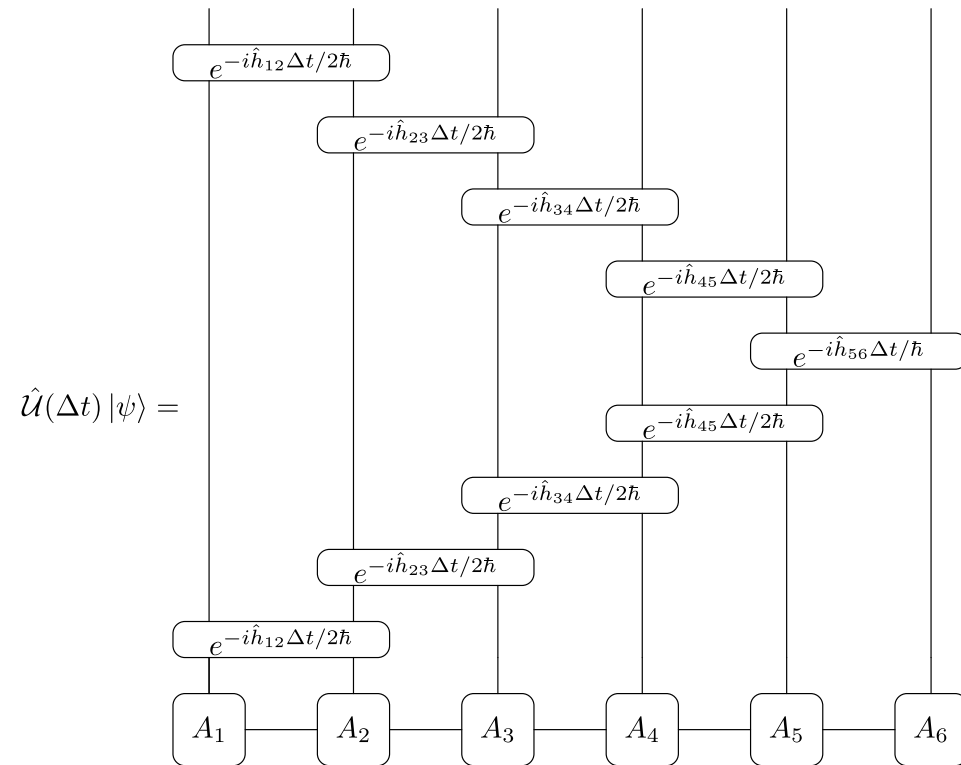
Equipped with a state and Hamiltonian, we can evolve it in time!

Time-Evolving Block Decimation (TEBD) is one tensor network algorithm for this [14, 15].

Very similar to DMRG, but instead uses the unitary time evolution operator.

A Trotter-Suzuki decomposition [16] on suitable Hamiltonians makes this possible.

The four-index "gates" are similar to those found in quantum computing.

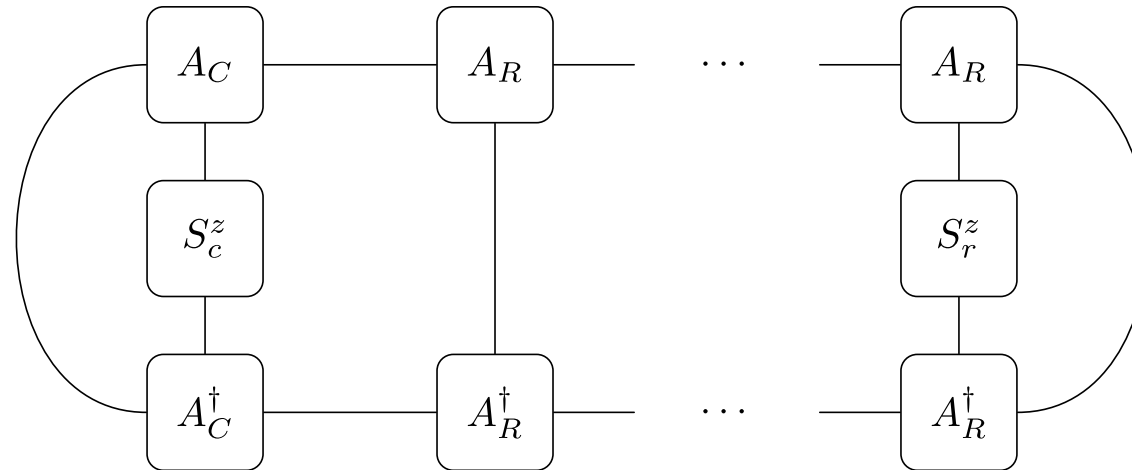


Applications 8: Quantum Systems, VUMPS

In 1D in the thermodynamic limit, we can formulate a special Variational Uniform MPS (VUMPS) [17]

VUMPS can be optimized to get the ground state directly in the thermodynamic limit!

It can also give us correlators at arbitrary separation, ex. $\langle S_c^z S_r^z \rangle$:



Technically requires an infinite bond dimension to capture the true ground state.

Summary

Tensors are:

- Not scary
- Versatile
- Easy to work with

The networks we can create with tensors can:

- Compute quantum mechanical models
- Perform deep learning
- Compress highly entangled systems
- Solve previously computationally impossible problems

I hope this has been inspiring! Let me know if you come up with a novel application using tensor networks!

References

- [1] “Matrix Multiplication Background User’s Guide,” NVIDIA Docs. Feb. 2023. Accessed: Sep. 20, 2026. [Online]. Available: <https://docs.nvidia.com/deeplearning/performance/dl-performance-matrix-multiplication/index.html>
- [2] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, “A Review of Convolutional Neural Networks in Computer Vision,” *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, Mar. 2024, doi: 10.1007/s10462-024-10721-6.
- [3] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, “Quantum Entanglement in Deep Learning Architectures,” *Physical Review Letters*, vol. 122, no. 6, p. 065301, Feb. 2019, doi: 10.1103/PhysRevLett.122.065301.
- [4] R. Orús, “Tensor Networks for Complex Quantum Systems,” *Nature Reviews Physics*, vol. 1, no. 9, pp. 538–550, Sep. 2019, doi: 10.1038/s42254-019-0086-7.
- [5] R. P. Feynman, “Quantum Mechanical Computers,” *Foundations of Physics*, vol. 16, no. 6, pp. 507–531, Jun. 1986, doi: 10.1007/BF01886518.

References

- [6] F. Verstraete and J. I. Cirac, “Renormalization Algorithms for Quantum-Many Body Systems in Two and Higher Dimensions.” arXiv, Jul. 2004. doi: 10.48550/arXiv.cond-mat/0407066.
- [7] Y.-Y. Shi, L.-M. Duan, and G. Vidal, “Classical Simulation of Quantum Many-Body Systems with a Tree Tensor Network,” Physical Review A, vol. 74, no. 2, p. 022320, Aug. 2006, doi: 10.1103/PhysRevA.74.022320.
- [8] S. R. White, “Density Matrix Formulation for Quantum Renormalization Groups,” Physical Review Letters, vol. 69, no. 19, pp. 2863–2866, Nov. 1992, doi: 10.1103/PhysRevLett.69.2863.
- [9] S. R. White, “Density-Matrix Algorithms for Quantum Renormalization Groups,” Physical Review B, vol. 48, no. 14, pp. 10345–10356, Oct. 1993, doi: 10.1103/PhysRevB.48.10345.

References

- [10] J. Dukelsky, M. A. Martin-Delgado, T. Nishino, and G. Sierra, “Equivalence of the Variational Matrix Product Method and the Density Matrix Renormalization Group Applied to Spin Chains,” *Europhysics Letters (EPL)*, vol. 43, no. 4, pp. 457–462, Aug. 1998, doi: 10.1209/epl/i1998-00381-x.
- [11] S. Östlund and S. Rommer, “Thermodynamic Limit of Density Matrix Renormalization,” *Physical Review Letters*, vol. 75, no. 19, pp. 3537–3540, Nov. 1995, doi: 10.1103/PhysRevLett.75.3537.
- [12] U. Schollwöck, “The Density-Matrix Renormalization Group,” *Reviews of Modern Physics*, vol. 77, no. 1, pp. 259–315, Apr. 2005, doi: 10.1103/RevModPhys.77.259.
- [13] U. Schollwoeck, “The Density-Matrix Renormalization Group in the Age of Matrix Product States,” *Annals of Physics*, vol. 326, no. 1, pp. 96–192, Jan. 2011, doi: 10.1016/j.aop.2010.09.012.

References

- [14] G. Vidal, “Efficient Classical Simulation of Slightly Entangled Quantum Computations,” *Physical Review Letters*, vol. 91, no. 14, p. 147902, Oct. 2003, doi: 10.1103/PhysRevLett.91.147902.
- [15] G. Vidal, “Efficient Simulation of One-Dimensional Quantum Many-Body Systems,” *Physical Review Letters*, vol. 93, no. 4, p. 040502, Jul. 2004, doi: 10.1103/PhysRevLett.93.040502.
- [16] M. Suzuki, “Generalized Trotter’s Formula and Systematic Approximants of Exponential Operators and Inner Derivations with Applications to Many-Body Problems,” *Communications in Mathematical Physics*, vol. 51, no. 2, pp. 183–190, Jun. 1976, doi: 10.1007/BF01609348.
- [17] V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete, and J. Haegeman, “Variational Optimization Algorithms for Uniform Matrix Product States,” *arXiv.org*. Jan. 2017. doi: 10.1103/PhysRevB.97.045145.