Efficient tensor network contraction algorithms

Linjian Ma and Edgar Solomonik

Department of Computer Science
University of Illinois Urbana-Champaign

Workshop on Sparse Tensor Computations

October 19th, 2023

Presentation overview

Summary of contributions: introduce two approximate tensor network contraction algorithms that accelerate applications in statistical physics and quantum circuit simulation

Outline of the presentation:

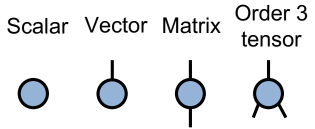
- Introduction to tensors and (approximate) tensor network contractions
- Cost-efficient contraction tree for approximate contraction[1]
- A flexible and cost-efficient low-rank approximation for approximate contraction[2]

[1] Joint work with Cameron Ibrahim and Ilya Safro from University of Delaware
[2] Joint work with Matt Fishman and Miles Stoudenmire from Flatiron Institute

Tensor and tensor contraction

Tensor: multi-dimensional array of data

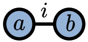
Tensor diagram: an order N tensor is represented by a vertex with N adjacent edges



Tensor contraction: summing element products from two tensors over contracted dimensions

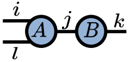
A dimension (edge) is contracted if it has no open end

Examples:



Inner product: $\sum_i a_i b_i$

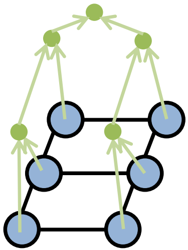
Matrix product : $C_{ik} = \sum_j A_{ij} B_{jk}$



Tensor times matrix: $C_{ilk} = \sum_j A_{ilj} B_{jk}$

Tensor network contraction

Tensor network: denoted by undirected hypergraph $G = (V, E)$



Contraction tree: rooted binary tree T

- A leaf of T represents a tensor in G
- A non-leaf vertex represents its children's contraction output

Find contraction cost-optimal contraction tree: NP-hard[1], many heuristics are used[2,3]

Cost under optimal contraction tree: exponential to the treewidth of G's line graph[4]

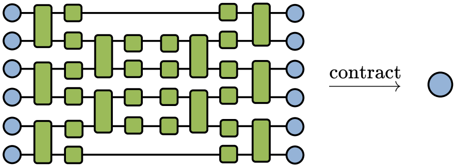
[1] O'Gorman, Parameterization of Tensor Network Contraction, TQC 2019
[2] Gray and Kourtis, Hyper-optimized tensor network contraction, Quantum 2021
[3] Liu et al, Computing solution space properties of combinatorial optimization problems via generic tensor networks, SISC 2023
[4] Markov and Shi, Simulating quantum computation by contracting tensor networks, SIAM Journal on Computing 2008

Applications of tensor network contraction

Quantum computing: simulate quantum algorithm[1]

Statistical physics: evaluate the classical partition function[2]

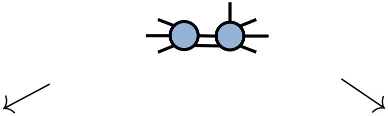
Computer science: constraint satisfaction problems[3]



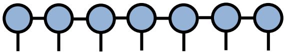
[1] Markov and Shi, Simulating quantum computation by contracting tensor networks, SIAM Journal on Computing 2008
[2] Levin, Michael, and Nave, Tensor renormalization group approach to two-dimensional classical lattice models, PRL 2007
[3] Kourtis, Stefanos, et al, Fast counting with tensor networks, SciPost Physics 2019

Approximate tensor network contractions: previous work

Idea: approximate each contraction output as a bounded-rank tensor network

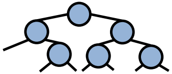


Tensor train/matrix product state (MPS)[1,2]



We propose an algorithm for cost-efficient contraction tree

Binary tree tensor network[3]



We propose to contract with flexible and cost-efficient low-rank approximation

[1] Pan et al, Contracting arbitrary tensor networks: general approximate algorithm and applications in graphical models and quantum circuit simulations, PRL 2020

[2] Chubb, General tensor network decoding of 2D Pauli codes, 2021

[3] Jermyn, Automatic contraction of unstructured tensor networks, SciPost Physics 2020

Presentation overview

Cost-efficient contraction tree for the tensor train-based algorithm[1]

- Solves a linear ordering problem to minimize edge crossings
- Achieves 5.9X speed-up when compared to previous works on contracting an Ising model tensor network

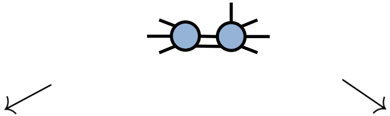
Contraction with a flexible and cost-efficient low-rank approximation[2]

- Uses normal equations to improve efficiency and can flexibly select the environment
- Achieves 9.2X speed-up when compared to previous works on contracting an Ising model tensor network

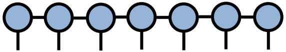
[1] **Ma**, Ibrahim, Safro, and Solomonik, Tensor network contraction with an efficient swap-based algorithm, in preparation
[2] **Ma**, Fishman, Stoudenmire, and Solomonik, Tensor network contraction with a flexible and cost-efficient density matrix algorithm for tree approximation, in preparation

Accelerate tensor train-based algorithm

Idea: approximate each contraction output as a bounded-rank tensor network



Tensor train/matrix product state (MPS)[1,2]



We propose an algorithm for cost-efficient contraction tree

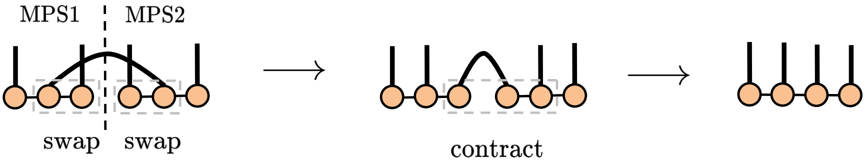
[1] Pan et al, Contracting arbitrary tensor networks: general approximate algorithm and applications in graphical models and quantum circuit simulations, PRL 2020

[2] Chubb, General tensor network decoding of 2D Pauli codes, 2021

Contraction of two tensor trains into a tensor train

Algorithm: move contracted edges to the center through adjacent swaps, then eliminate them[1]

- Each swap uses low-rank approximation to maintain a bounded rank



Observation: The total number of swaps is lower bounded by the convex crossing number[2]

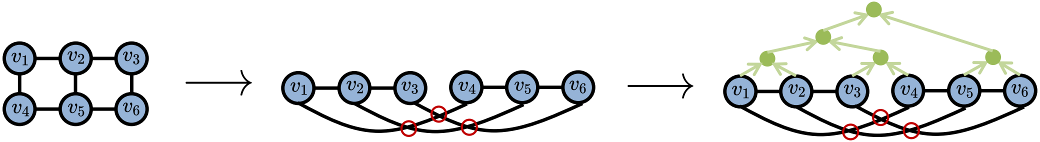
[1] Pan et al, Contracting arbitrary tensor networks: general approximate algorithm and applications in graphical models and quantum circuit simulations, PRL 2020

[2] Shahrokhi et al, Book embeddings and crossing numbers, WG'94

CATN-GO: build contraction tree constrained by a vertex ordering

Our approach: find a vertex ordering that minimizes edge crossings, then find a contraction tree constrained by the ordering

- Inspired by prior work on building exact tensor network contraction trees[1]

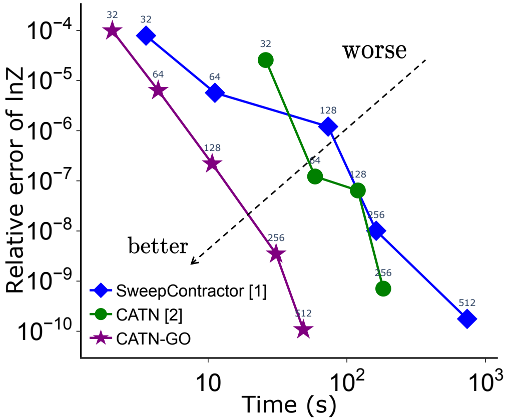


Find the optimal vertex ordering: NP-hard problem, heuristics are used[2]

Contraction tree optimization: minimize the cost using dynamic programming

[1] Ibrahim et al, Constructing Optimal Contraction Trees for Tensor Network Quantum Circuit Simulation, HPEC 2022
[2] Shahrokhi et al, Book embeddings and crossing numbers, WG'94

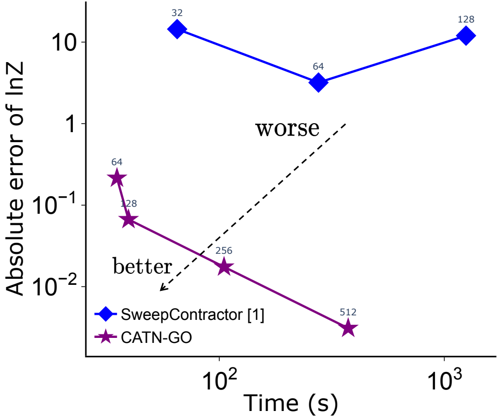
Experimental results: Ising model



Results for contracting an Ising model tensor network defined on a $5 \times 5 \times 5$ lattice

- Number on each point: maximum tensor train rank
- Achieve 5.9X speed-up relative to previous works to reach a relative error of 10^{-8}

[1]Chubb, General tensor network decoding of 2D Pauli codes, 2021
[2]Pan et al, Contracting arbitrary tensor networks: general approximate algorithm and applications in graphical models and quantum circuit simulations, PRL 2020



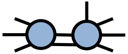
Results for simulating 2D random quantum circuits[2]

- The initial state $|\psi\rangle$ is defined on a 6×6 grid
- 5 layers of random quantum gates (U) are applied to $|\psi\rangle$
- We approximately contract the tensor network that represents $\langle\psi| U^T U |\psi\rangle$

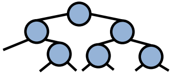
[1] Chubb, General tensor network decoding of 2D Pauli codes, 2021
[2] Arute et al, Quantum supremacy using a programmable superconducting processor, Nature 2019

Efficient low-rank approximation for tensor network contraction

Idea: approximate each contraction output as a bounded-rank tensor network



Binary tree tensor network[3]

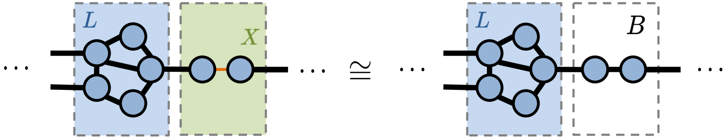


We propose to contract with flexible and cost-efficient low-rank approximation

[3] Jermyn, Automatic contraction of unstructured tensor networks, SciPost Physics 2020

Motivation for a new low-rank approximation subroutine

$$\min_{X,\text{rank}(X)\leq R} \| L \; X - L \; B \|_F$$



Accuracy: environment (L) typically comprises a small part of the whole tensor network[1,2]

- Small $L \rightarrow$ minimizes local rather than global error

Efficiency: Orthogonalization (via implicit QR factorization) on L is performed

- QR factorization can be expensive when L is not a tree

[1] Pan et al, Contracting arbitrary tensor networks: general approximate algorithm and applications in graphical models and quantum circuit simulations, PRL 2020

[2] Chubb, General tensor network decoding of 2D Pauli codes, 2021

Normal equations for low-rank approximation

$$X^* = \underset{X, \text{rank}(X) \leq r}{\text{argmin}} \|LX - LB\|_F$$

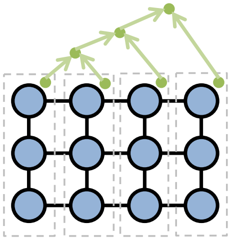
Orthogonalization-based: $Q_L, R_L \leftarrow \texttt{QR}(L)$, then use the rank-$r$ approximation of $R_L B$ to update solution

Normal equations-based: compute the leading r eigenvectors of $B^T L^T LB$, and $X^* = BVV^T$

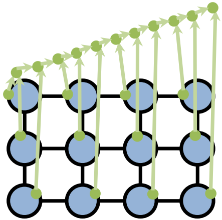
The asymptotic cost to form normal equations ($B^T L^T LB$) is upper-bounded by doing QR

Partitioned Contract: use partial contraction tree for flexible environment

Contraction tree over partitions



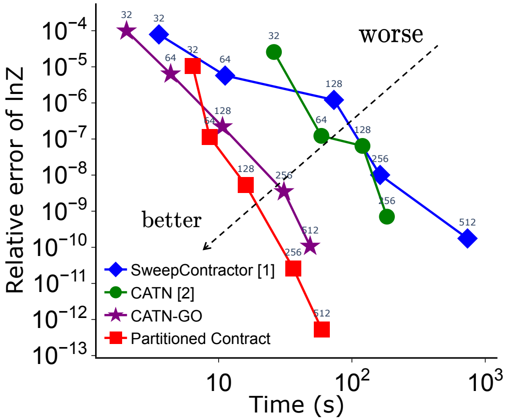
Complete contraction tree



Each contraction outputs a binary tree tensor network

- The input pair of partitions are considered the environment
- Larger partition implies larger environment $\rightarrow$ minimizes the global error

Experimental results: Ising model



Results for contracting an Ising model tensor network defined on a $5 \times 5 \times 5$ lattice

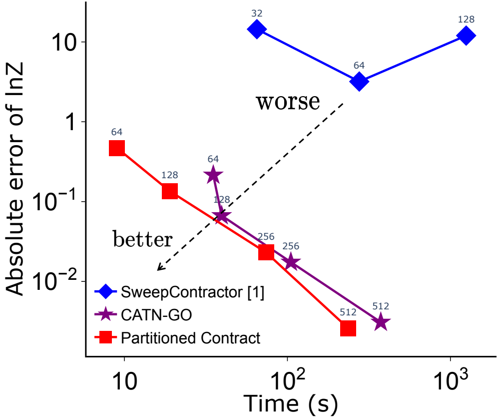
- Number on each point: maximum tensor train rank
- Achieve 9.2X speed-up relative to previous works to reach a relative error of 10^{-9}

---

[1] Pan et al, Contracting arbitrary tensor networks: general approximate algorithm and applications in graphical models and quantum circuit simulations, PRL 2020

[2] Chubb, General tensor network decoding of 2D Pauli codes, 2021

Experimental results: random quantum circuit



Results for simulating 2D random quantum circuits[2]

- The initial state $|\psi\rangle$ is defined on a 6×6 grid
- 5 layers of random quantum gates (U) are applied to $|\psi\rangle$
- We approximately contract the tensor network that represents $\langle\psi| U^T U |\psi\rangle$

[1] Chubb, General tensor network decoding of 2D Pauli codes, 2021

[2] Arute et al, Quantum supremacy using a programmable superconducting processor, Nature 2019

Conclusion

Conclusion

- Introduce efficient approximate tensor network contraction algorithms
- CATN-GO uses an efficient contraction tree for tensor train-based contraction
- Partitioned Contract contains efficient low-rank approximation and can incorporate large environments
- Both works are part of my dissertation at `https://linjianma.github.io/pdf/dissertation.pdf`

Future work

- CATN-GO: devise heuristics for finding vertex orderings with fewer edge crossings
- Partitioned Contract: find efficient partial contraction trees

Acknowledgements

CATN-GO:

Cameron Ibrahim

Ilya Safro

Edgar Solomonik

Partitioned Contract:

Matt Fishman

Miles Stoudenmire

Edgar Solomonik

Backup slides

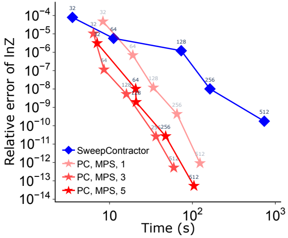
Additional experimental results for CATN-GO

Vertex ordering	$8 \times 8 \times 8$ lattice			$(6, 300)$-rand regular graph		
	# crossings	Time (s)	GFlops	# crossings	Time (s)	GFlops
Baseline	34.6k	2.2k	9.4k	133k	10.8k	52k
Recursive bisection	16.8k	1.0k	4.6k	37.5k	2.8k	13.8k
Relative improvements	2.1X	2.2X	2.1X	3.5X	3.8X	3.8X

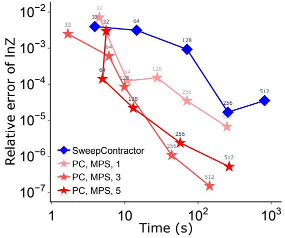
Vertex orderings with fewer edge crossings yield less contraction time

- Baseline: sequential traversal for lattice, and random ordering for a random graph
- Random regular graph has 300 vertices and degree 6

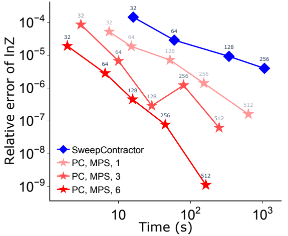
Additional experimental results for Partitioned Contract



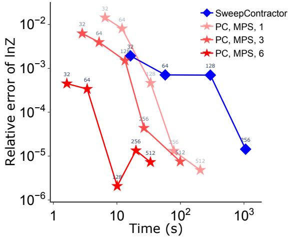
3D cube, Ising model



3D cube, random tensors



random regular graph, Ising model



random regular graph, random tensors