Discovering faster matrix multiplication algorithms with reinforcement learning

Eddie Sung b09201029
BSc in Mathematics

Overview of Topics

- Introduction
- Basics of Matrix Multiplication
- AlphaTensor
  - Tensor Concepts
  - Tensor Game with Deep Reinforcement Learning (DRL)
- Comparisons on AlphaTensor and Algorithm
- Conclusion
- Q&A Session

Introduction

Goal: To minimize the computational steps involved in matrix multiplication.

Solution: Utilizing AI and reinforcement learning to achieve this goal.

Matrix Multiplication

Consider a multiplication of two 2 × 2 matrices :

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \times \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{bmatrix}$$

Standard algorithm

$h_1 = a_{1,1}\, b_{1,1}$

$h_2 = a_{1,1}\, b_{1,2}$

$h_3 = a_{1,2}\, b_{2,1}$

$h_4 = a_{1,2}\, b_{2,2}$

$h_5 = a_{2,1}\, b_{1,1}$

$h_6 = a_{2,1}\, b_{1,2}$

$h_7 = a_{2,2}\, b_{2,1}$

$h_8 = a_{2,2}\, b_{2,2}$

$c_{1,1} = h_1 + h_3$

$c_{1,2} = h_2 + h_4$

$c_{2,1} = h_5 + h_7$

$c_{2,2} = h_6 + h_8$

Strassen's algorithm :

$h_1 = (a_{1,1} + a_{2,2})\,(b_{1,1} + b_{2,2})$

$h_2 = (a_{2,1} + a_{2,2})\, b_{1,1}$

$h_3 = a_{1,1}\,(b_{1,2} - b_{2,2})$

$h_4 = a_{2,2}\,(-b_{1,1} + b_{2,1})$

$h_5 = (a_{1,1} + a_{1,2})\, b_{2,2}$

$h_6 = (-a_{1,1} + a_{2,1})\,(b_{1,1} + b_{1,2})$

$h_7 = (a_{1,2} - a_{2,2})(b_{2,1} + b_{2,2})$

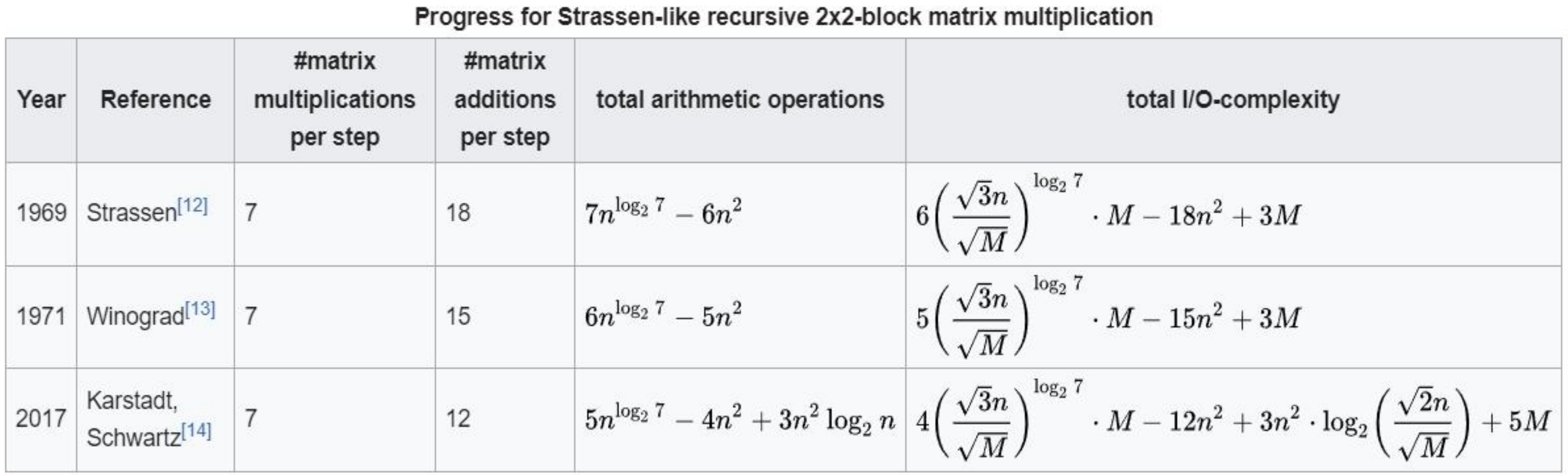
$c_{1,1} = h_1 + h_4 - h_5 + h_7$

$c_{1,2} = h_3 + h_5$

$c_{2,1} = h_2 + h_4$

$c_{2,2} = h_1 - h_2 + h_3 + h_6$

Matrix Multiplication



Progress for Strassen-like recursive 2x2-block matrix multiplication

Year	Reference	#matrix multiplications per step	#matrix additions per step	total arithmetic operations	total I/O-complexity
1969	Strassen[12]	7	18	$7n^{\log_2 7} - 6n^2$	$6\left(\frac{\sqrt{3}n}{\sqrt{M}}\right)^{\log_2 7} \cdot M - 18n^2 + 3M$
1971	Winograd[13]	7	15	$6n^{\log_2 7} - 5n^2$	$5\left(\frac{\sqrt{3}n}{\sqrt{M}}\right)^{\log_2 7} \cdot M - 15n^2 + 3M$
2017	Karstadt, Schwartz[14]	7	12	$5n^{\log_2 7} - 4n^2 + 3n^2 \log_2 n$	$4\left(\frac{\sqrt{3}n}{\sqrt{M}}\right)^{\log_2 7} \cdot M - 12n^2 + 3n^2 \cdot \log_2\left(\frac{\sqrt{2}n}{\sqrt{M}}\right) + 5M$

Alpha Tensor

- The Alpha series was developed by Google DeepMind in London starting in 2014, aiming to create artificial intelligence Go software.

- AlphaGo took two years to defeat human players. Alpha Zero, with even stronger learning capabilities, beat human players in just 21 days.

- The hope is to use Alpha Tensor, based on Alpha Zero, to solve the matrix multiplication problem.

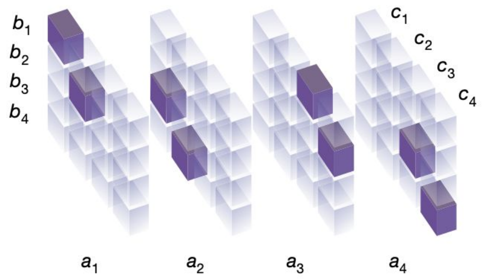
- Computer programs have previously been used to solve the Four Color Problem, which states that any map drawn on a plane or sphere can be colored using no more than four colors, ensuring no two adjacent regions share the same color.

Tensor Concepts

Matrix Multiplication Tensor

The operation of multiplying two N x N matrices can be represented by a tensor of dimension N^(2) x N^(2) x N^(2).There is an example:

$$\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}$$



The figure shows the product of two 2 x 2 matrices representing by a tensor of dim 4 x 4 x 4 .

Tensor Concepts

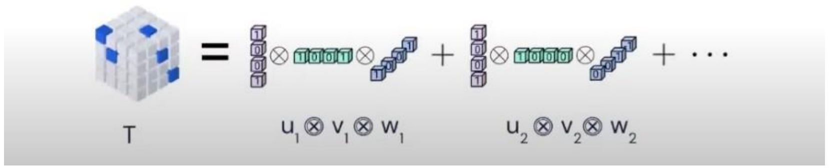
As matrix multiplication (A, B) ↦ AB is bilinear, it can be fully represented by a 3D-tensor.

Define T_n as the 3D tensor describing n × n matrix multiplication has entries in {0, 1}.

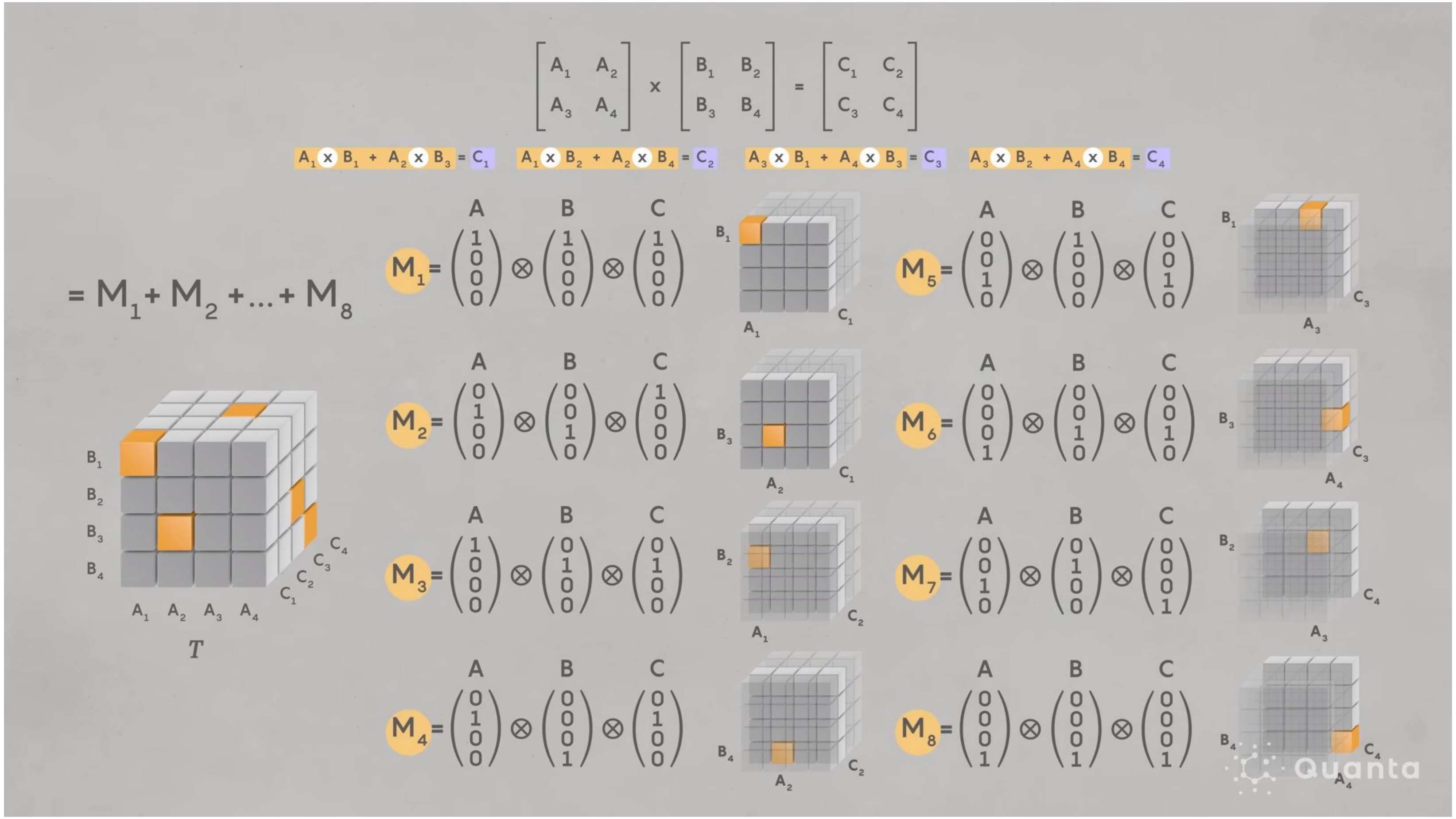
By a decomposition of T_n into R rank-one terms, we mean

$$\mathcal{T}_n = \sum_{r=1}^{R} \mathbf{u}^{(r)} \otimes \mathbf{v}^{(r)} \otimes \mathbf{w}^{(r)}, \qquad (1)$$

where $\otimes$ denotes the outer (tensor) product, and $\mathbf{u}^{(r)}$, $\mathbf{v}^{(r)}$ and $\mathbf{w}^{(r)}$ are all vectors. If a tensor $\mathcal{T}$ can be decomposed into R rank-one terms, we say the rank of $\mathcal{T}$ is at most R, or Rank $(\mathcal{T}) \leq R$. This is a natural extension from the matrix rank, where a matrix is decomposed into $\sum_{r=1}^{R} \mathbf{u}^{(r)} \otimes \mathbf{v}^{(r)}$.



Standard Algorithm Tesnor Decomposition



$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \times \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}$$

$A_1 \times B_1 + A_2 \times B_3 = C_1$ $A_1 \times B_2 + A_2 \times B_4 = C_2$ $A_3 \times B_1 + A_4 \times B_3 = C_3$ $A_3 \times B_2 + A_4 \times B_4 = C_4$

$$= M_1 + M_2 + \ldots + M_8$$

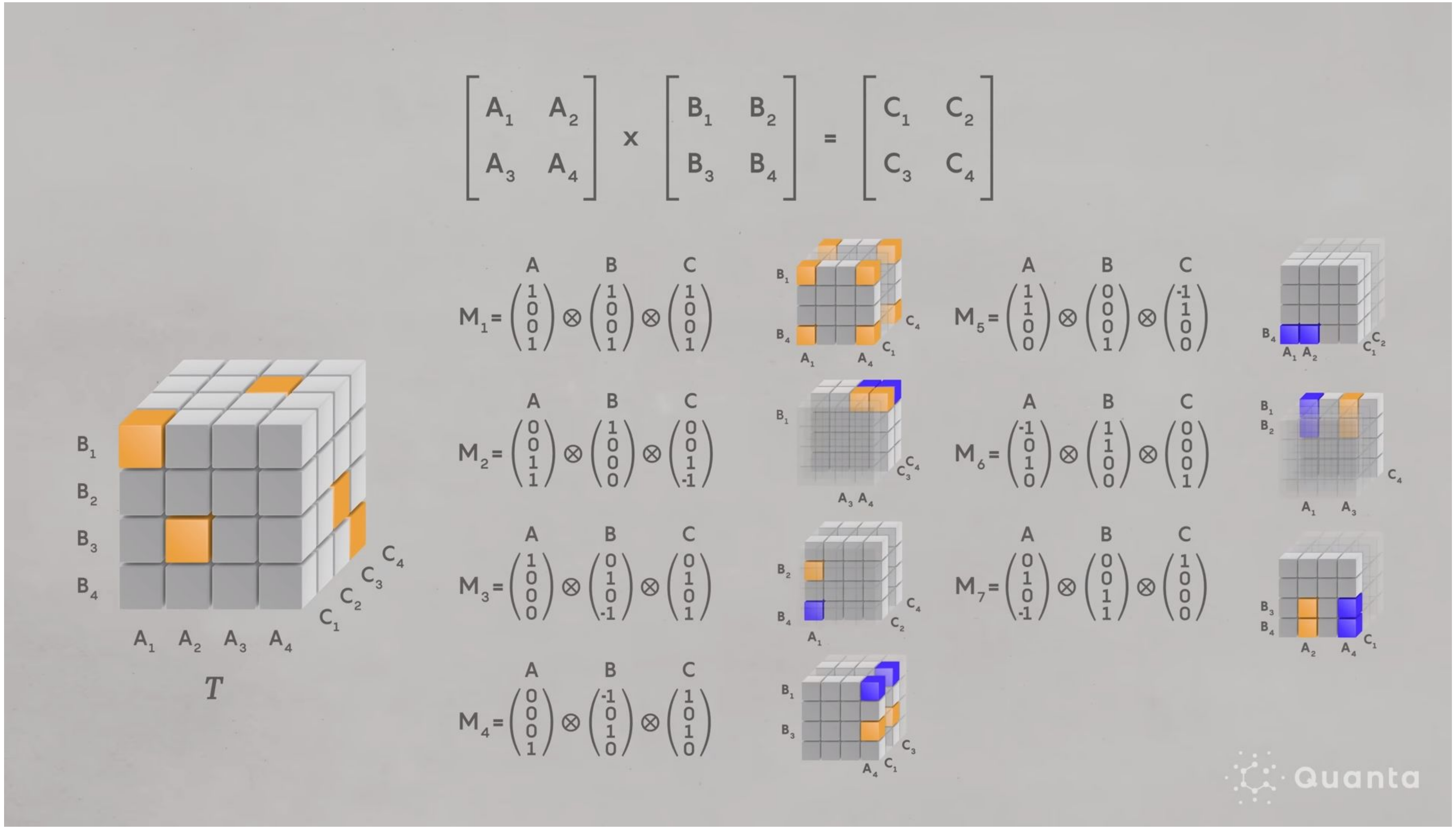
$$M_1 = \begin{array}{c} A \\ \begin{pmatrix}1\\0\\0\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} B \\ \begin{pmatrix}1\\0\\0\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} C \\ \begin{pmatrix}1\\0\\0\\0\end{pmatrix}\end{array} \qquad M_5 = \begin{array}{c} A \\ \begin{pmatrix}0\\0\\1\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} B \\ \begin{pmatrix}1\\0\\0\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} C \\ \begin{pmatrix}0\\0\\1\\0\end{pmatrix}\end{array}$$

$$M_2 = \begin{array}{c} A \\ \begin{pmatrix}0\\1\\0\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} B \\ \begin{pmatrix}0\\0\\1\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} C \\ \begin{pmatrix}1\\0\\0\\0\end{pmatrix}\end{array} \qquad M_6 = \begin{array}{c} A \\ \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\end{array} \otimes \begin{array}{c} B \\ \begin{pmatrix}0\\0\\1\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} C \\ \begin{pmatrix}0\\0\\1\\0\end{pmatrix}\end{array}$$

$$M_3 = \begin{array}{c} A \\ \begin{pmatrix}1\\0\\0\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} B \\ \begin{pmatrix}0\\1\\0\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} C \\ \begin{pmatrix}0\\1\\0\\0\end{pmatrix}\end{array} \qquad M_7 = \begin{array}{c} A \\ \begin{pmatrix}0\\0\\1\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} B \\ \begin{pmatrix}0\\1\\0\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} C \\ \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\end{array}$$

$$M_4 = \begin{array}{c} A \\ \begin{pmatrix}0\\1\\0\\0\end{pmatrix}\end{array} \otimes \begin{array}{c} B \\ \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\end{array} \otimes \begin{array}{c} C \\ \begin{pmatrix}0\\1\\0\\0\end{pmatrix}\end{array} \qquad M_8 = \begin{array}{c} A \\ \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\end{array} \otimes \begin{array}{c} B \\ \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\end{array} \otimes \begin{array}{c} C \\ \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\end{array}$$

Strassen’s Algorithm Tesnor Decomposition



$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \times \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}$$

$$M_1 = \overset{A}{\begin{pmatrix}1\\0\\0\\1\end{pmatrix}} \otimes \overset{B}{\begin{pmatrix}1\\0\\0\\1\end{pmatrix}} \otimes \overset{C}{\begin{pmatrix}1\\0\\0\\1\end{pmatrix}}$$

$$M_2 = \overset{A}{\begin{pmatrix}0\\0\\1\\1\end{pmatrix}} \otimes \overset{B}{\begin{pmatrix}1\\0\\0\\0\end{pmatrix}} \otimes \overset{C}{\begin{pmatrix}0\\0\\1\\-1\end{pmatrix}}$$

$$M_3 = \overset{A}{\begin{pmatrix}1\\0\\0\\0\end{pmatrix}} \otimes \overset{B}{\begin{pmatrix}0\\1\\0\\-1\end{pmatrix}} \otimes \overset{C}{\begin{pmatrix}0\\1\\0\\1\end{pmatrix}}$$

$$M_4 = \overset{A}{\begin{pmatrix}0\\0\\0\\1\end{pmatrix}} \otimes \overset{B}{\begin{pmatrix}-1\\0\\1\\0\end{pmatrix}} \otimes \overset{C}{\begin{pmatrix}1\\0\\1\\0\end{pmatrix}}$$

$$M_5 = \overset{A}{\begin{pmatrix}1\\1\\0\\0\end{pmatrix}} \otimes \overset{B}{\begin{pmatrix}0\\0\\0\\1\end{pmatrix}} \otimes \overset{C}{\begin{pmatrix}-1\\1\\0\\0\end{pmatrix}}$$

$$M_6 = \overset{A}{\begin{pmatrix}-1\\0\\1\\0\end{pmatrix}} \otimes \overset{B}{\begin{pmatrix}1\\1\\0\\0\end{pmatrix}} \otimes \overset{C}{\begin{pmatrix}0\\0\\0\\1\end{pmatrix}}$$

$$M_7 = \overset{A}{\begin{pmatrix}0\\1\\0\\-1\end{pmatrix}} \otimes \overset{B}{\begin{pmatrix}0\\0\\1\\1\end{pmatrix}} \otimes \overset{C}{\begin{pmatrix}1\\0\\0\\0\end{pmatrix}}$$

Tensor Game

Goal: Decompose the 3D-tensor using as few unique rank-1 tensors as possible.

Every step t of the game, we subtract the rank-1 tensor, where S_0 = T_n

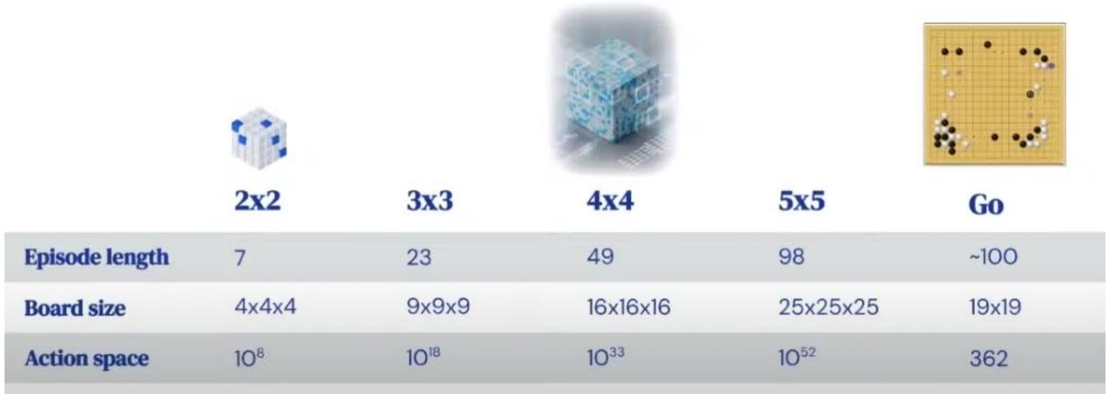
$$\mathcal{S}_t \leftarrow \mathcal{S}_{t-1} - \mathbf{u}^{(t)} \otimes \mathbf{v}^{(t)} \otimes \mathbf{w}^{(t)}.$$

The game ends when the state reaches the zero tensor, S_R = 0.

For every step taken, we provide a reward of −1 to encourage finding the shortest path to the zero tensor.

Deep Reinforcement Learning (DRL)

However, the action space of tensor game is large. The agent is trained using deep reinforcement learning, playing the tensor game repeatedly to receive rewards and update its policy and value functions. Over time, it learns the most efficient strategies.



	2x2	3x3	4x4	5x5	Go
Episode length	7	23	49	98	~100
Board size	4x4x4	9x9x9	16x16x16	25x25x25	19x19
Action space	10^{8}	10^{18}	10^{33}	10^{52}	362

Deep Reinforcement Learning (DRL)

The learning process mainly consists of three parts, Synthetic demonstrations, Change of Basis and Neural network.

Synthetic demonstrations:

- utilized by generating tensor-factorization pairs, where factorizations are randomly generated and formed into tensors.
- Previous games with high scores are added to the demonstration buffer to reinforce successful agent behaviors.

Deep Reinforcement Learning (DRL)

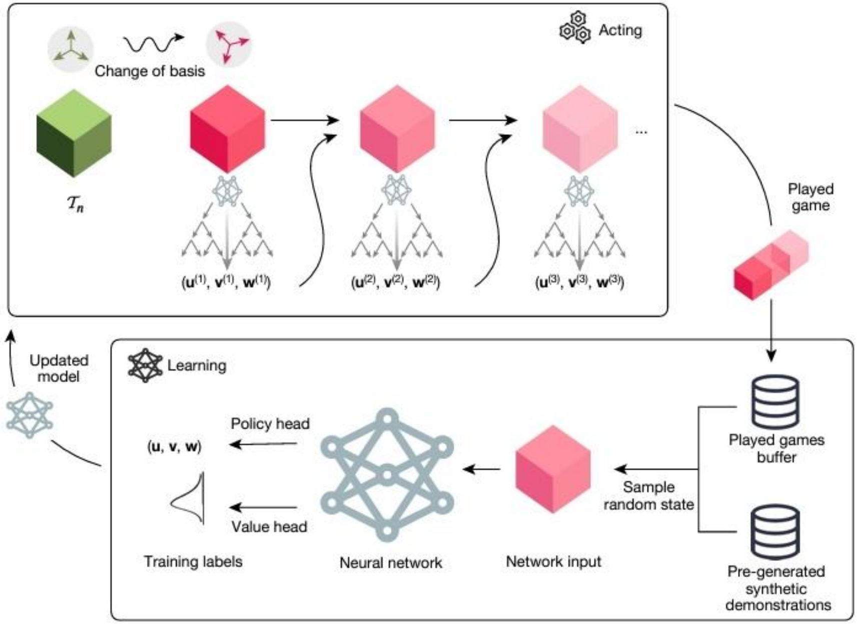
Change of Basis:

- By expressing the tensor in various bases, including randomly generated ones, the exploration space is expanded, enhancing diversity and problem-solving capabilities.
- Randomly sampled bases with imposed restrictions simplify the algorithm while ensuring integer entries and numerical stability in the resulting decomposition.

Neural network:

- The neural network architecture consists of a torso, a policy head, and a value head.

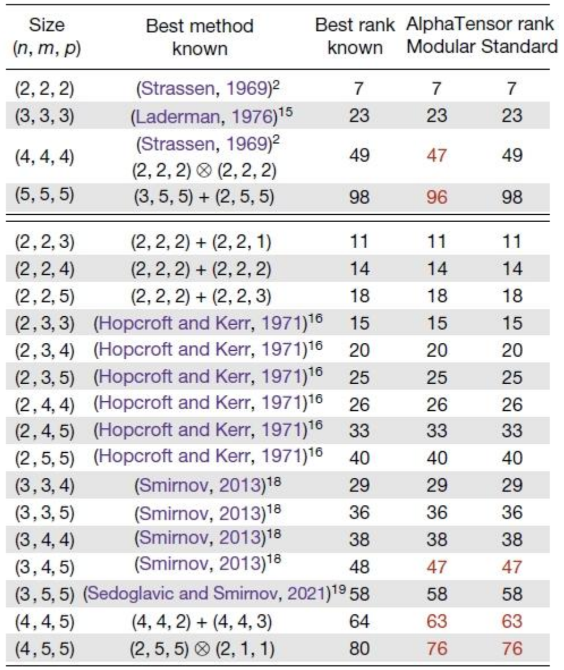
Overview of AlphaTensor



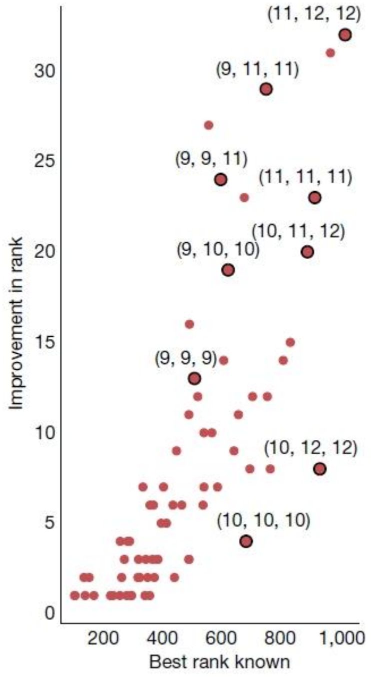
Acting
Change of basis
$\mathcal{T}_n$
...
$(\mathbf{u}^{(1)}, \mathbf{v}^{(1)}, \mathbf{w}^{(1)})$
$(\mathbf{u}^{(2)}, \mathbf{v}^{(2)}, \mathbf{w}^{(2)})$
$(\mathbf{u}^{(3)}, \mathbf{v}^{(3)}, \mathbf{w}^{(3)})$
Played game
Updated model
Learning
Policy head
(u, v, w)
Value head
Training labels
Neural network
Network input
Sample random state
Played games buffer
Pre-generated synthetic demonstrations

Fig. 2 | Overview of AlphaTensor. The neural network (bottom box) takes as input a tensor $\mathcal{S}_t$, and outputs samples $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ from a distribution over potential next actions to play, and an estimate of the future returns (for example, of $-\mathrm{Rank}(\mathcal{S}_t)$). The network is trained on two data sources: previously played games and synthetic demonstrations. The updated network is sent to the actors (top box), where it is used by the MCTS planner to generate new games.

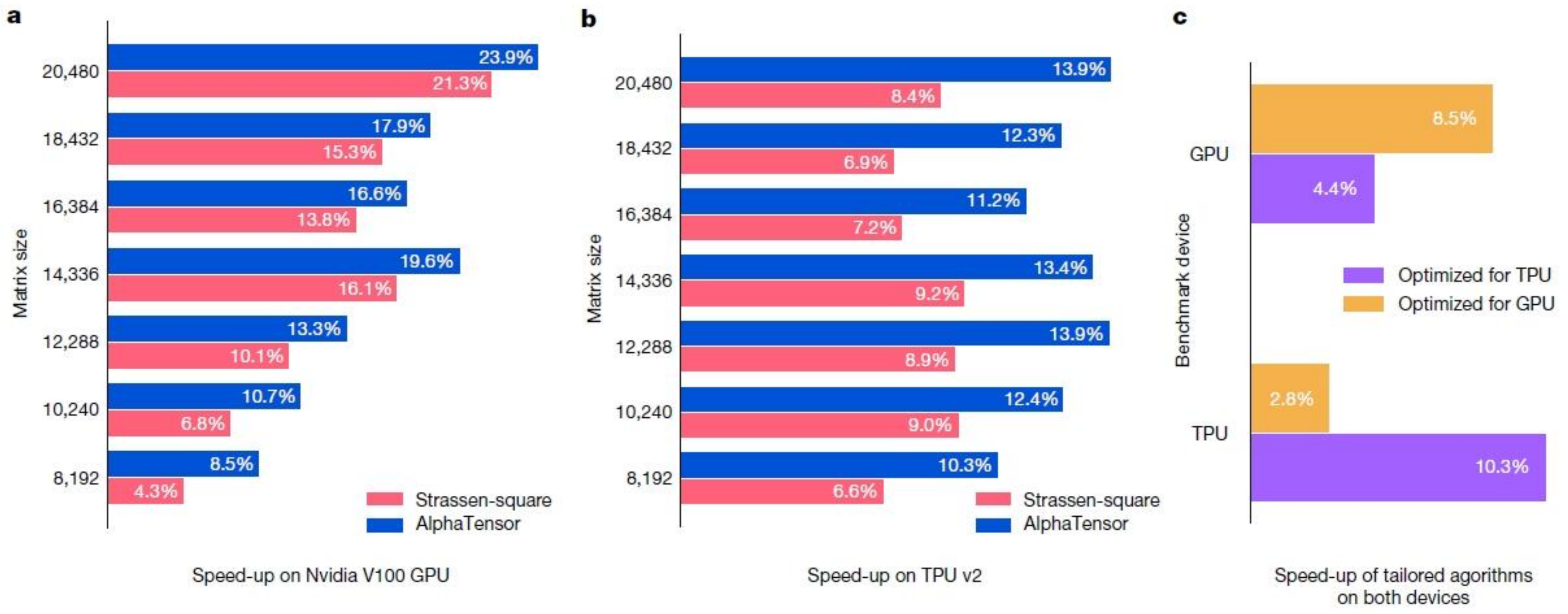
Comparisons on alpha tensor and Algorithm



Size (*n*, *m*, *p*)	Best method known	Best rank known	AlphaTensor rank Modular	AlphaTensor rank Standard
(2, 2, 2)	(Strassen, 1969)[2]	7	7	7
(3, 3, 3)	(Laderman, 1976)[15]	23	23	23
(4, 4, 4)	(Strassen, 1969)[2] (2, 2, 2) ⊗ (2, 2, 2)	49	47	49
(5, 5, 5)	(3, 5, 5) + (2, 5, 5)	98	96	98
(2, 2, 3)	(2, 2, 2) + (2, 2, 1)	11	11	11
(2, 2, 4)	(2, 2, 2) + (2, 2, 2)	14	14	14
(2, 2, 5)	(2, 2, 2) + (2, 2, 3)	18	18	18
(2, 3, 3)	(Hopcroft and Kerr, 1971)[16]	15	15	15
(2, 3, 4)	(Hopcroft and Kerr, 1971)[16]	20	20	20
(2, 3, 5)	(Hopcroft and Kerr, 1971)[16]	25	25	25
(2, 4, 4)	(Hopcroft and Kerr, 1971)[16]	26	26	26
(2, 4, 5)	(Hopcroft and Kerr, 1971)[16]	33	33	33
(2, 5, 5)	(Hopcroft and Kerr, 1971)[16]	40	40	40
(3, 3, 4)	(Smirnov, 2013)[18]	29	29	29
(3, 3, 5)	(Smirnov, 2013)[18]	36	36	36
(3, 4, 4)	(Smirnov, 2013)[18]	38	38	38
(3, 4, 5)	(Smirnov, 2013)[18]	48	47	47
(3, 5, 5)	(Sedoglavic and Smirnov, 2021)[19]	58	58	58
(4, 4, 5)	(4, 4, 2) + (4, 4, 3)	64	63	63
(4, 5, 5)	(2, 5, 5) ⊗ (2, 1, 1)	80	76	76



More Graph on Comparisons



Conclusion

Leveraging AI and reinforcement learning through the tensor game can significantly optimize matrix multiplication, driving advancements in computational efficiency and mathematical problem-solving.