Christian B. Mendl, Richard M. Milbradt due: 27 May 2022 (08:15)

Tutorial 4 (Canonical Polyadic (CP) decomposition for optimized matrix-matrix multiplication)
Using the outer product of vectors, the SVD of an $n \times n$ matrix $A = USV^\dagger$ can be represented as

$$A = \sum_{j=1}^{n} \sigma_j \, u_j \circ v_j^*,$$

where u_j and v_j are the column vectors of U and V, respectively. Note that the summation only needs to include the non-zero singular values, i.e., run from 1 to $r = \mathrm{rank}(A)$. Similarly, a tensor $T \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ can be decomposed as

$$T = \sum_{j=1}^{r} u_j \circ v_j \circ w_j =: [\![U, V, W]\!],$$

with $U \in \mathbb{C}^{n_1 \times r}$, $V \in \mathbb{C}^{n_2 \times r}$, $W \in \mathbb{C}^{n_3 \times r}$ (not necessarily isometries). This is denoted the Canonical Polyadic (CP) decomposition. The minimal possible r defines the rank of the tensor.

(a) Does the CP decomposition always exist?

One usage of the CP decomposition is optimizing matrix-matrix multiplication, $C = AB$, with $A, B \in \mathbb{C}^{n \times n}$. A literal implementation of $c_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk}$ (for $i, k = 1, \dots, n$) has runtime $\mathcal{O}(n^3)$, but it turns out that this can be further improved. In the following, we assume that $n = 2^k$ is a power of 2, and use a recursive block partitioning into four blocks of size $2^{k-1} \times 2^{k-1}$ each, such that

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}.$$

(b) Working with this block partitioning, write out the algorithmic complexity of the naive matrix multiplication. What is the computationally most expensive step in this algorithm?

(c) We can enumerate the block indexing of the submatrices as $(1,1) \to 1$, $(1,2) \to 2$, $(2,1) \to 3$, $(2,2) \to 4$, allowing for a one-hot encoding e_1 to e_4. Based on that, assemble a tensor of degree 3, with the first two dimensions corresponding to the input block indices and the third to the output index.

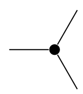
(d) How does the rank of this tensor affect the complexity of the matrix-matrix multiplication?

Remark: It turns out that the tensor in (c) has rank 7. This leads to the *Strassen algorithm*, exploiting that only 7 block matrix multiplications are necessary at each level of recursion to perform the same operation. The algorithmic complexity is thus reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(7^k) = \mathcal{O}(n^{\log_2 7}) \approx \mathcal{O}(n^{2.80735})$.[1]

Exercise 4.1 (Bell circuit)
The delta tensor of degree d is defined via the Kronecker delta, with entries

$$\delta_{i_1,\dots,i_d} = \begin{cases} 1, & i_1 = i_2 = \dots = i_d \\ 0, & \text{otherwise} \end{cases}$$

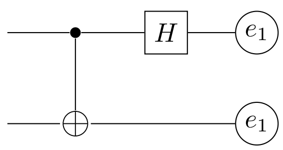
It can be regarded as generalization of the identity matrix, and is visualized by a filled dot, e.g., for $d = 3$:



The XOR tensor $\oplus \in \mathbb{C}^{2 \times 2 \times 2}$ has degree 3 and is given by (using zero-based indices $i, j, k \in \{0, 1\}$):

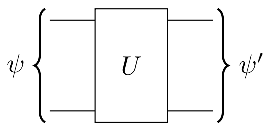
$$\oplus_{i,j,k} \begin{cases} 1, & i + j + k = 0 \mod 2 \text{ (even number of 1s)} \\ 0, & \text{otherwise} \end{cases}$$

Evaluate and simplify the following diagram as far as possible, where $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is the Hadamard matrix and $e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$:



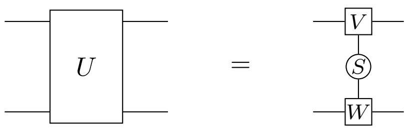
[1] The up to date smallest-known exponent is $\mathcal{O}(n^{2.37286})$, see arxiv.org/abs/2010.05846.

Exercise 4.2 (Decomposition of two-qubit quantum logic gates)
A two-qubit quantum logic gate is described by a unitary matrix $U \in \mathbb{C}^{4\times 4}$. Applying U to a vector $\psi \in \mathbb{C}^4$, i.e., the linear transformation $\psi' = U\psi$, can be visualized in terms of a quantum circuit diagram as follows:



The input ψ appears on the left since such diagrams are read from left to right. The horizontal lines are associated with single qubits, and can be interpreted as tensor legs. From this perspective, U is the matricization of a $2 \times 2 \times 2 \times 2$ tensor.[2]

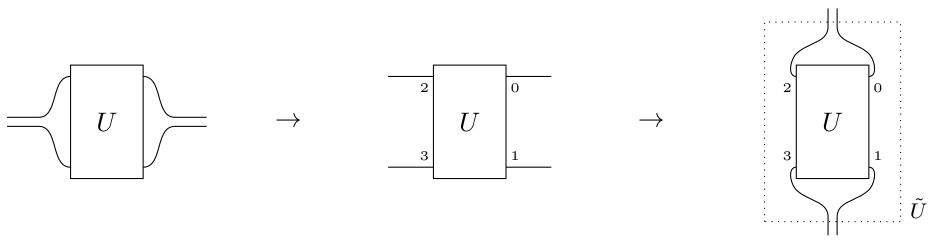
Here our goal is to implement a variant of the singular value decomposition applied to U, graphically summarized as



The diagonal matrix S stores the singular values. In general there are four of them, but for special gates some of the singular values can be zero and thus be omitted. This allows for a more efficient (classical) simulation of such quantum gates.

Implement the following steps using Python/NumPy:

(a) Reshape U (as matrix) into a $2\times2\times2\times2$ tensor, then permute (interchange) dimensions 1 and 2 (counting from zero), then reshape back to a 4×4 matrix (denoted $\tilde{U}$). Graphically:



Here $0, 1, 2, 3$ are dimension (index) labels of U. Note that $\tilde{U}$ is (in general) no longer unitary.[3]

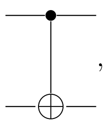
(b) Compute the "compact" singular value decomposition of $\tilde{U}$ (i.e., omitting zero singular values): $\tilde{U} = VSW$, with $S = \mathrm{diag}(\sigma_1, \dots, \sigma_r)$ the diagonal matrix of non-zero singular values (r is the rank of $\tilde{U}$), and isometric matrices $V \in \mathbb{C}^{4\times r}$ and $W \in \mathbb{C}^{r\times 4}$. (Note the convention without taking the adjoint of W in the SVD.)

Hint: Start with a standard SVD, then extract the corresponding submatrices (cf. Exercise 2.2 (b)); r is the number of non-zero singular values.

(c) Reshape V into a $2 \times 2 \times r$ tensor and W into a $r \times 2 \times 2$ tensor.

(d) Apply these steps to decompose the controlled-NOT and fSim gates defined as

$$U_{\mathrm{CNOT}} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{pmatrix} \hat{=}$$



$$, \qquad U_{\mathrm{fSim}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -i\sin(\theta) & 0 \\ 0 & -i\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & \mathrm{e}^{-i\varphi} \end{pmatrix},$$

for the specific choice of rotation angles $\theta = \frac{\pi}{3}$ and $\varphi = \frac{\pi}{4}$. Which singular values do you obtain? You can verify your implementation by comparing the gates to the reconstructed version computed as

$$\sum_{j=1}^{r} \sigma_j \, V_{:,:,j} \otimes W_{j,:,:}.$$

[2]It will be clear from context whether U is regarded as matrix or degree-4 tensor. Rectangular versus circular shapes for drawing tensors have no mathematical significance here.

[3]Requiring that $\tilde{U}$ is likewise unitary is an interesting subject by itself, see B. Bertini, P. Kos, T. Prosen: *Exact correlation functions for dual-unitary lattice models in* $1+1$ *dimensions*, Phys. Rev. Lett. 123, 210601 (2019) (arxiv.org/abs/1904.02140).