Quantum TensorNet Simulator on GPU architectures

Luca Venerando Greco, Leonardo Ignazio Pagliochini

June 27, 2024

Abstract

Enter a short summary here. What topic do you want to investigate and why? What experiment did you perform? What were your main results and conclusion?

1 Introduction

Quantum computing stands at the forefront of computational innovation, promising unprecedented processing power for solving complex problems that are intractable for classical computers. At the heart of this revolutionary paradigm lies the quantum circuit conceptual and practical framework that orchestrates the manipulation of quantum information. Quantum circuits serve as the blueprint for quantum algorithms, encoding the sequence of operations that transform quantum states to solve computational tasks. However, the very properties that give quantum computing its power—superposition and entanglement—also make it notoriously difficult to simulate and analyze using classical computational methods. In response to these challenges, researchers have developed various strategies to optimize quantum circuit simulation. GPU-accelerated simulations harness the parallel processing capabilities of graphics hardware to model larger quantum systems more efficiently. This approach has enabled the simulation of quantum circuits at scales that were previously out of reach for conventional CPU-based methods.

1.1 Quantum Circuits

Quantum circuits are the fundamental framework for describing quantum computations and algorithms. They consist of a sequence of quantum operations applied to a register of qubits. Unlike classical bits, qubits can exist in superposition states, allowing for the manipulation of exponentially large state spaces. Quantum circuits are composed of quantum gates, which are unitary operations that transform qubit states. Common gates include single-qubit rotations (such as Hadamard and phase gates) and multi-qubit entangling operations (like CNOT gates). The circuit model allows for the implementation of quantum algorithms by orchestrating these gates to perform specific computational tasks. Quantum measurements, typically performed at the end of the circuit, collapse the quantum state and provide classical output. The power of quantum circuits lies in their ability to exploit quantum phenomena like superposition and entanglement, potentially offering exponential speedups for certain problems compared to classical computation.

1.2 Quantum Circuit Simulation

Quantum circuit simulation is a critical tool in the development and validation of quantum algorithms, serving as a bridge between theoretical quantum computing and practical implementation on quantum hardware

Two main techniques exist for simulating quantum circuits: state-vector evolution and the circuit-equivalent unitary matrix calculation.

The state vector evolution method involves updating the state vector step-by-step as each quantum gate is applied. This approach is particularly efficient for simulating the evolution of a specific quantum state through the circuit, as it only requires maintaining and transforming a single vector of dimension 2^n, where n is the number of qubits. In contrast, calculating the unitary matrix of the entire circuit involves constructing a $2^n \times 2^n$ matrix that represents the combined effect of all the gates in the circuit. This unitary matrix can then be applied to any initial state vector to determine its evolution. While this method provides a comprehensive representation of the circuit's behavior and can be useful for theoretical analysis and verification, it is computationally intensive, especially for large numbers of qubits, due to the exponential growth in the size of the matrix. Therefore, state vector evolution is generally preferred for practical simulations, whereas unitary matrix calculation is more suited for complete characterizations of small circuits or for deriving analytical insights.

1.3 Tensor Network representation of a Quantum circuits

Innovative representation techniques have emerged to tackle the complexity of quantum circuit simulation. Among these, tensor network methods stand out as a powerful tool for representing and manipulating quantum states and operations.

Considering each gate as a tensor of appropriate rank we could model all the interaction between gates as contractions acting on the ranks related to the quantum lanes shared by the gates to contract. This allows to harness the flexibility in designing an appropriate bracketing of the contraction operations.

2 Tensor contraction

Tensor contraction is a fundamental operation in the realm of tensor calculus, ubiquitous across various fields including physics, engineering, and machine learning. This chapter delves into the core principles of tensor contraction, elucidating its significance and applications through a comprehensive exploration of Einstein notation. By embracing Einstein's summation convention, we will streamline the often cumbersome tensor operations, transforming complex multi-dimensional arrays into more manageable forms. This approach not only simplifies calculations but also unveils the intrinsic elegance of tensor algebra. As we define and dissect the concept of tensor contraction, readers will gain a robust understanding of its theoretical underpinnings and practical utility, setting the stage for advanced applications in subsequent chapters. Firstly, it is essential to explain how tensors will be represented throughout this report. A tensor is a mathematical object that generalizes the concept of scalars, vectors, and matrices to higher dimensions. It can be thought of as a multi-dimensional array of numbers that transforms under specific rules when changing coordinate systems. Tensors are fundamental in various fields of physics, engineering, and data science, allowing for the representation and manipulation of complex multi-dimensional data and physical quantities.

A tensor

$$T \in V \otimes V \otimes V \otimes V^* \otimes V^* \otimes V^*$$

where V^* is the dual space of V, will be represented in the following way T^{ijk}_{lmn}, where:

- The superscript indices i, j, and k correspond to the three V spaces in the tensor product
- The subscript indices l, m, and n correspond to the three V^* (dual) spaces in the tensor product

If V is finite-dimensional with dimension d, then each index would typically range from 0 to $d-1$. In our case, since we will deal with quantum circuits, the dimensions will be equal to two, therefore index will take values of either 0 or 1 corresponding to the two basis states of a qubit, often denoted $|0\rangle$ and $|1\rangle$. From now on, a tensor will be graphically represented as in Figure 1.

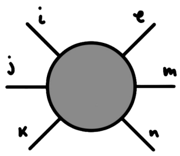

Figure 1: Graphical representation of a rank 6 tensor

In an environment like the one we are working in, where the dimension of the space is 2,

As previously stated, in the context of quantum circuits, gates can be represented as tensor having only two component along each dimension, under these considerations we can define a matrix representation of the tensor using the notation at Figure 2 that explains how the value of each index variates over the matrix.
We need also to define the concept of connections between two tensor

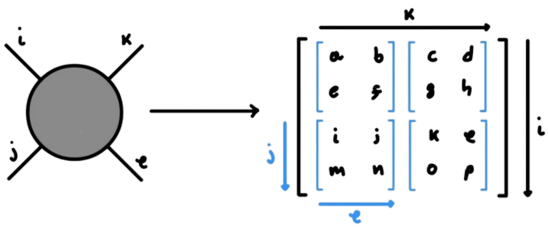

Figure 2: Matrix representation of a rank 4 tensor

If two tensor A^{ijk}_{lmn} and B^{opq}_{rst} are connected over the indexes $n - o$, there is a constraint over the values of the dimensions corresponding to those indexes. Specifically, the dimension of the index n in tensor A must be equal to the dimension of index o in tensor B to allow proper contraction along that shared dimension. This constraint ensures compatibility in the tensor network and enables valid mathematical operations between the two tensors. Graphically the connection between two tensor can be represented as in Figure 3.

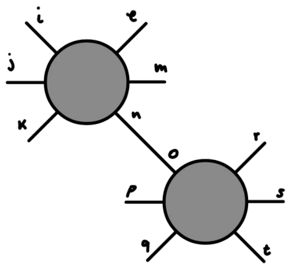

Figure 3: Graphical representation of a Connection between two tensors

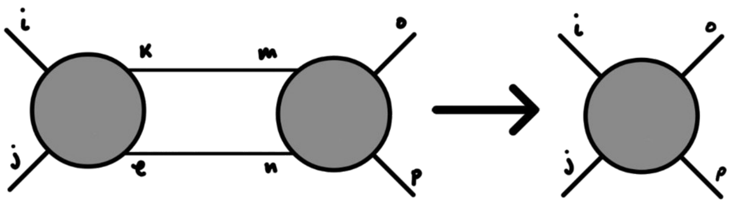

Figure 4: Graphical representation of a tensor contraction over two axes

The contraction of two connected tensors is shown in Figure 4. Given the two tensors A_{ij}^{kl} and B_{mn}^{op} connected over the axes (k, m) and (l, n), each element of the resulting tensor can be calculated as follows:

$$C_{ij}^{op} = \sum_{x,y} A_{ij}^{xy} \cdot B_{xy}^{op}$$

This is the mathematical description of the contraction between two tensors.

3 Tensor contraction in quantum circuits

Starting from the standard representation of a quantum gate (Figure 5), the first step will be give a tensor representation of this latter; this is done using the matrix representation of the tensor given at Figure 2 applying it to the characteristic matrix of the gate.

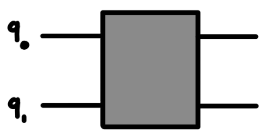

Figure 5: Graphical representation of a generic quantum gate

4 Baseline

In order to provide a reference baseline for our implementation we chose to use the available qiskit tools to test the performance of available sota algorithm for quantum circuit simulations.