PERSISTENT HOMOLOGY AND INTERACTION GRAPHS

AMELIE SCHREIBER

Abstract. The optimization of quantum circuits for specific hardware backends remains a major challenge in the field of quantum computing. In this paper, we present a comprehensive study of the techniques for optimizing quantum circuits using transpiling and persistent homology. We first introduce the concept of transpiling and its challenges, and discuss the impact of different basis gate sets on circuit performance. We then explore the concept of weighted interaction graphs, which are used to represent the connectivity and entanglement properties of a quantum circuit. We present two methods for computing the weights of the edges in a weighted interaction graph: two-qubit gate count between each pair of qubits and von Neumann entanglement entropy. We discuss the strengths and weaknesses of each method and demonstrate the utility of weighted interaction graphs for understanding the topological structure of quantum circuits. Finally, we delve into the application of persistent homology for analyzing the topological properties of weighted interaction graphs. We demonstrate the use of persistent homology for optimizing circuit transpiling and hardware design and present experimental results on the impact of persistent homology on circuit performance. Our study provides a comprehensive overview of transpiling, weighted interaction graphs, and persistent homology for quantum computing and highlights their potential for optimizing quantum circuits and hardware design.

Contents

1. Introduction

Quantum computing is a rapidly evolving field that has the potential to revolutionize computing as we know it. However, the design and optimization of quantum circuits remain a major challenge due to the complex and fragile nature of quantum systems. To address this challenge, researchers have developed techniques for transpiling quantum circuits from one basis gate set to another, and for analyzing the topological properties of quantum circuits using persistent homology.

In this paper, we present a comprehensive study of these techniques and their applications for designing quantum hardware backends and optimizing circuit transpiling. We begin by introducing

Date: March 3, 2023.
2020 *Mathematics Subject Classification.* Primary 16E05 05E10 14H57 Secondary 11G32 05E16 .
Key words and phrases. Persistent Homology, Interaction Graph, Quantum Circuit Transpiling.

the concept of transpiling, which involves converting a quantum circuit from one basis gate set to another in order to optimize its performance on a specific quantum hardware platform. We discuss the challenges and strategies involved in transpiling quantum circuits.

Next, we explore the concept of weighted interaction graphs, which are used to represent the connectivity and entanglement properties of a quantum circuit. We describe two methods for computing the weights of the edges in a weighted interaction graph: two-qubit gate count between each pair of qubits, and von Neumann entanglement entropy. We present a comparison of these methods and their respective strengths and weaknesses, and demonstrate the utility of weighted interaction graphs for understanding the topological structure of quantum circuits.

Finally, we delve into the application of persistent homology for analyzing the topological properties of weighted interaction graphs. We discuss the concept of persistent homology and its use for identifying the most significant topological features of a graph at different scales. We demonstrate the application of persistent homology for optimizing circuit transpiling and hardware design, and present experimental results on the impact of persistent homology on circuit performance.

Overall, our study provides a comprehensive overview of transpiling, weighted interaction graphs, and persistent homology for quantum computing, and highlights the potential of these techniques for optimizing quantum circuits and hardware design.

2. Transpiling a Quantum Circuit to Chosen Basis Gates

Transpiling a quantum circuit to a chosen basis gate set means converting the circuit, which is specified in terms of a set of quantum gates, into an equivalent circuit that uses only a different set of gates, known as the basis gates. This is done to make the circuit more compatible with a specific quantum hardware or simulator, as different quantum systems may have different sets of native gates that can be implemented directly.

The process of transpiling a circuit involves decomposing each gate in the original circuit into a sequence of basis gates, which can be implemented directly on the target quantum system. The decomposition is done using a gate synthesis algorithm, which searches for a sequence of basis gates that approximates the original gate to a desired level of accuracy.

The choice of basis gates can depend on various factors, such as the hardware architecture, gate error rates, and gate fidelities. Commonly used basis gate sets include the Clifford+T gate set, which consists of the Hadamard, CNOT, and T gates, and the universal gate set, which consists of any two-qubit gate and a single-qubit gate that can generate arbitrary single-qubit rotations.

Transpiling a circuit to a chosen basis gate set is an important step in implementing a quantum algorithm on a specific quantum system, as it ensures that the circuit can be executed efficiently and accurately on the target hardware or simulator.

3. Computing the Weighted Interaction Graph

Suppose we are given a quantum circuit with density matrix ρ and with qubits $\{q_k\}_{k=0}^{n-1}$ with corresponding reduced density matrices $\{\rho_k\}_{k=0}^{n-1}$. We would like to define a weighted interaction graph with nodes corresponding to qubits, and edge weights corresponding to

(1) The two-qubit gate count between each pair of qubits,
(2) the von Neumann entanglement entropy between qubits,
(3) using the quantum mutual information between pairs of qubits.

3.1. **von Neumann Entanglement Entropy.** To define a weighted interaction graph with nodes corresponding to qubits and edge weights corresponding to the von Neumann entanglement entropy between qubits in a quantum circuit, we can simply compute the reduced density matrix of each two-qubit composite system, and then compute $S(\rho) = -\sum_x \lambda_x \log(\lambda_x)$. Here the λ_x are the eigenvalues of the density operator ρ.

3.2. **Quantum Mutual Information.** To define a weighted interaction graph with nodes corresponding to qubits and edge weights corresponding to the quantum mutual information between qubits in a quantum circuit, you can follow these steps:

Compute the reduced density matrix ρ_k for each qubit q_k in the circuit. The reduced density matrix describes the state of a single qubit in the circuit, obtained by tracing out the other qubits.

For each pair of qubits (q_i, q_j) in the circuit, compute the **quantum mutual information** $M_{i,j} = S(\rho_i) + S(\rho_j) - S(\rho_{i,j})$ between the reduced density matrices ρ_i and ρ_j. Use the quantum mutual information $M_{i,j}$ to define the weights of the edges in the graph. Specifically, the weight of an edge between qubits q_i and q_j is given by $w_{i,j} = S_{i,j}$.

We can now construct the weighted interaction graph with nodes corresponding to qubits and edge weights corresponding to one of the three weight paradigms mentioned above. Specifically, the vertices of the graph correspond to the qubits in the circuit, and the edges correspond to the interactions between the qubits induced by the gates. The weight of an edge between qubits q_i and q_j represents the strength of the entanglement between the two qubits, as quantified by the two-qubit gate count, von Neumann entanglement entropy, or quantum mutual information.

Once the weighted interaction graph is constructed, it can be used to study the properties of the quantum circuit, such as the connectivity of the qubits, the degree of entanglement, and the presence of certain gate sequences. The graph can also be used in the transpiling process to find an efficient decomposition of the original circuit into a sequence of basis gates that can be implemented on a specific quantum hardware or simulator.

4. Persistent Homology of Weighted Interaction Graphs

Persistent homology is a mathematical tool used to study the topological properties of data sets, including weighted graphs. In the case of a weighted interaction graph, we can use persistent homology to study the connectivity and structure of the graph at different scales defined by a filtration given by the weights of the edges.

To study a weighted interaction graph using persistent homology, we need to define a filtration parameter that increases monotonically with the edge weights. This filtration parameter is used to define a sequence of subgraphs of the original graph, where each subgraph includes all edges with weights less than or equal to the filtration parameter.

We can then compute the homology groups of each subgraph in the sequence, which describe the connected components, loops, voids, and higher-dimensional features of the graph at different scales. By tracking the birth and death of these topological features as the filtration parameter increases, we can construct a persistence diagram or barcode that summarizes the persistent homology of the graph.

To apply persistent homology to a weighted interaction graph, we can follow these steps:

(1) Construct the weighted interaction graph as described earlier, with nodes corresponding to qubits and edge weights corresponding to the von Neumann entanglement entropy between qubits (or use two-qubit gate count, or quantum mutual information).
(2) Define a filtration parameter based on the edge weights. For example, we can use the entanglement entropy as the filtration parameter, so that the filtration parameter increases monotonically with the strength of the entanglement between qubits.

(3) Use a persistent homology software package, such as Giotto-TDA see [Tauzin et al.](or Scikit-TDA), to compute the homology groups of each subgraph in the sequence defined by the filtration parameter.
(4) Construct a persistence diagram or barcode that summarizes the persistent homology of the graph. This diagram shows the birth and death of topological features, such as connected components, loops, and voids, as the filtration parameter increases. The persistence diagram can be used to identify the most persistent topological features of the graph, which correspond to the most significant structural features of the circuit.

By studying the persistence diagram of a weighted interaction graph, we can gain insights into the topological structure of the graph at different scales defined by the weights of the edges. This can help us understand the connectivity and entanglement properties of the quantum circuit, and guide the design of more efficient and accurate quantum algorithms and circuits.

5. 2-parameter Persistent Homology Using a Discrete Time Bifiltration

Dividing the quantum circuit into time-steps and computing the reduced density matrix operator for each time step allows us to construct a weighted interaction graph, where the weights are given by the von Neumann entanglement entropy between each pair of qubits (or alternatively quantum mutual information or two-qubit gate count). By considering the entanglement entropy (or the quantum mutual information) as a parameter, we obtain a bifiltration of the weighted interaction graph.

This bifiltration can be used to compute the 2-parameter persistent homology of the weighted interaction graph. The first parameter is time, and the second parameter is the entanglement entropy. The persistent homology captures the topological features of the graph that persist across different time-steps and entanglement entropy values.

The 2-parameter persistent homology of the weighted interaction graph can be used to study the evolution of the entanglement structure of the quantum circuit. This can provide insights into the quantum information processing that is happening in the circuit, and can help us understand the role of entanglement in quantum computation. This can again be utilized in understanding and improving the transpiling process, especially when hardware backend constraints are topological in nature. Also note, that time between two-qubit gates between two fixed qubits q_i and q_j should be captured by 2-parameter persistence. In particular, if sufficient time is elapsed between two-qubit gates, this should show up in two-parameter persistence in some way. This is important as error rates may be affected as two-qubit gate frequency increases.

Overall, computing the 2-parameter persistent homology of a bifiltration constructed from a quantum circuit can be a useful tool for studying the entanglement structure of the circuit and gaining insights into the quantum information processing that is happening. We can then use [Rivet] to study the minimal presentations of persistence modules and their diagrams.

6. Comparing to Hardware Topology

The interaction graph notebooks [AS4] shows how to transpile a quantum circuit from one gate set to a chosen basis gate set, then compute the interaction graph and weighted interaction graphs, using one of the three paradigms listed above, in order to analyze interaction topology. This can help in choosing a hardware backend that is best suited to run the circuit and can be used in the design of quantum ASICs. The notebook ends with a comment on how one should apply persistent homology to study the weighted interaction graph. Circuits which transpile to a circuit with weighted interaction graph such that interactions have low persistence may imply less topological constraints on the hardware backend. In particular, if there is weak entanglement between two qubits due to low two-qubit gate count between those two qubits, then this will show up in the persistent homology of the weighted interaction graph and will imply that direct connectivity of those qubits is less necessary than two qubits with a high amount of interaction. This is quantifiable in terms of persistence diagrams and Betti numbers. The next step we believe to be important is generalizing the computations to the generalizations of surface codes in [AS3], known as *chimera codes*, also sometimes referred to as *hybrid qudit surface codes*.

References

[AS3] Amelie Schreiber, *Hybrid Qudit Surface Codes in Google Cirq*, https://github.com/Amelie-Schreiber/quantum_surface_codes

[AS4] Amelie Schreiber, *Persistent Homology of Weighted Interaction Graphs Notebooks*, https://github.com/Amelie-Schreiber/persistent_homology_of_entanglement

[Rivet] *Rivet*, https://github.com/rivetTDA/rivet

[Tauzin et al.] Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Wojciech Reise, Anibal Medina-Mardones, Alberto Dassatti, Kathryn Hess, *giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration*, https://arxiv.org/abs/2004.02551

Email address: amelie.schreiber.math@gmail.com