Presentation QAOA – QAR-Lab

Script, Matthias Pawlik, 10.04.2024

Welcome to my quick presentation on the Quantum Approximate Optimization Algorithm. So, since I was not sure who to expect in the audience, I prepared my talk in English, so I hope that's fine. For the interview after the presentation we can also switch back to German if you prefer.

(click) In a first instance, let me classify the QAOA in the family of quantum algorithms. C As you can expect from the name, the QAOA aims to solve optimization problems approximately. C It is a heuristic quantum algorithm that may produce acceptable solutions fast in practice, but in the worst-case scenario might still scale exponentially. C It applies variational methods and can be thought of as an approximation to adiabatic quantum computation. C I will explain each of these buzz words step by step, elaborating more specifically on the concept of VQEs, QUBOs and AQC. This then allows us to understand the QAOA. C Let me already state the key idea of this optimization algorithm: We try to reformulate the optimization problem in terms of a Hamiltonian whose ground state corresponds to the optimal solution.

(click) First, we will discuss Variational Quantum Eigensolvers. C From quantum mechanics, we know how to compute the energy expectation value of a quantum state. C We now restrict our Hilbert space to a family of quantum states that only depends on a variational parameter theta. C Most importantly, we can find the approximate ground state of a Hamiltonian by minimizing the energy expectation value with respect to theta. The ground state energy E star constitutes the upper bound to the actual ground state energy. C With the help of Variational Quantum Circuits one successively determines the optimal value theta star. Now when we apply the concept of a VQE in the context of QAOA, this will look like the following: We prepare an initial quantum state and evolve with some time evolution operator, then measure the energy, update the theta parameter with a classical optimizer and repeat this procedure several times, until we are happy with the approximation.

(click) Another important concept is the so-called Quadratic Unconstrained Binary Optimization Problem (or short QUBO). C In this setting, the goal is to extremize the value of a quadratic form with some n-dimensional vector x. C Furthermore we only allow for zero and one entries in the components of x and impose no other constraints. C A prominent example is the MaxCut problem: We try to cut a fully connected graph in two halves where the cut weight corresponds to sum of the weights of edges that connect both sets. C That is for a given weight matrix W, we want to find the maximal cut C x. The x1 to x4 in the example can either be zero or one, which encodes to which set each vertex of the graph belongs. In fact, when considering the cost function C, one realizes that we can reformulate the MaxCut problem into a QUBO with these two relations: C . C Another important step then consists of encoding the cost function into a Hamiltonian operator, where Z denotes the Pauli-Z-operator. This can quickly be derived from the time-independent Schrödinger equation.

(click) Last but not least, we quickly have to talk about Adiabatic Quantum Computing. C The time evolution of a quantum state is given by the time dependent Schrödinger equation C whose solution for time independent Hamiltonians is C . The adiabatic theorem now states C that we remain in the ground state of a quantum system if the latter is perturbed only slow enough which is essential for preparing our cost Hamiltonian later on. C Furthermore, we can discretize the time evolution relying on the concept of Trotterization for the numerical implementation of our algorithm. C Now the QAOA is an adiabatic schedule: We start of from an initial easy-to-prepare eigenstate of the so-called mixer Hamiltonian HM which is then adiabatically changed into the ground state of the cost Hamiltonian HC over a time span T which we are mainly looking for. In that way, we obtain the ground state of a potentially very difficult cost Hamiltonian if we only proceed slowly enough.

(click) Let us have a closer look on how to construct the variational quantum circuit for the QAOA. C The pattern was introduced in 2014 in a paper of Farhi, Goldstone and Gutmann. C As mentioned before the idea is to propose an approximate solution for QUBO instances using quantum mechanics. C . As a special case of the VQE, we have to deal with a layerized variational form. C Typically the algorithm starts with preparing the equal superposition state corresponding to the highest energy eigenstate of the mixer Hamiltonian HM. So when evolving towards the cost Hamiltonian HC we find the approximate solution up to a minus sign. C In our example of four qubits in the ground state, such superposition state is generated by applying Hadamard gates on each qubit. We then alternately apply the time evolution operators of the cost and mixer Hamiltonian over p layers with 2p variational parameters gamma 1 to gamma p and beta 1 to beta p. Finally, we measure in the computational basis to evaluate the cost function and update the variational parameters with a classical optimizer accordingly. C We can have a closer look at the cost time evolution operator UC, which encodes the specific optimization problem, and the mixing time evolution operator UM on the next slide.

(click) C When exponentiating HM we obtain single qubit rotational gates along the x axis with some angle dependent on the beta parameters. C Inserting the expression of the cost Hamiltonian from the QUBO slide into the matrix exponential yields a more complex result: Here we have to deal with single rotational gates around the z axis and then with two qubit rotational z-gates in a second step. C In fact one can further decompose the two qubit gates into a construction with CNOT gates and single rotational gates around the z axis. So in total, we have found an explicit variational circuit representation of the QAOA in terms of one and two qubit gates ready for execution.

(click) C Finally, I want to give a short overview on implementation details as far as I have understood them without any programming experience with qiskit, but having reviewed the code base on the IBM Quantum Learning Platform. C So they mainly work with numpy and qiskit libraries in a qiskit runtime. For the graph representation one may use rustworkx and for the classical optimization one can implement a routine from the SciPy library. C The first step consists of constructing the cost Hamiltonian and generating the variational quantum circuit with the QAOAAnsatz method, whose correctness can be verified by respective plotting features. C Secondly, the problem needs to be optimized for execution with a transpiler. For example, the circuit may still be modified in a way that we only have two qubit gates acting on neighboring qubits for less error rates. C Then qiskit primitives come into play, such as an estimator that computes the expectation value for the cost function. With the help of the SciPy routine, we eventually obtain the optimized variational parameters. C Lastly, the solution vector of variational parameters needs to be plugged into the ansatz circuit again several times to generate a probability distribution of the most probable bit-strings which can again be interpreted as solutions to the MaxCut problem.

(click) Here are some important sources that I based my research on. Thank you for your attention.