Project plans

Contents

1 Interface

1.1 Optimizers refactor

The optimization module will be extended, introducing more options like state-of-art heuristic algorithm and gradient based optimizers. The current shape of Qibo relies on TensorFlow to compute automatic differentiation, in particular in Quantum Machine Learning routines. We will upgrade this aspect by using hardware-compatible differentiation rules to make the calculations on the quantum devices possible, integrating this strategy with the current optimization layout.

Deliverables and milestones

- add new optimizers (Basin-Hopping, Simulated Annealing, hardware-compatible gradient descent, etc) [by Nov. 2023];
- write custom derivation rules for quantum circuits, integrating them with the TensorFlow backend. These will be hardware-compatible by deploying the Parameter Shift Rules [by Dec. 2023];
- make Qibo's interface compatible with other frameworks (JaX, Pytorch) [by Apr. 2024]
- lighten the access interface to optimisers [by Feb. 2023];

1.2 Qiboml

A new Qibo module will be developed to perform Quantum Machine Learning tasks. It will provide the users with a simple interface to write QML prototypes and algorithms using Qibo and to freely define custom loss functions and optimizers. We will focus on a full-stack approach to the QML routines, having as our strength the possibility to operate from the highest to the lowest level of the quantum computing thanks to the cooperation between Qibo, Qibolab and Qibocal.

Deliverables

- define the QML pipeline and thus the interface [by Dec. 2023];
- specialize the Qibo circuit for QML applications [by Dec. 2023];
- integrate the Qibo's optimization module with the Qiboml's one [By Feb. 2024];
- write a collection of loss function prototypes [by Mar. 2024];
- release Qiboml [by Sep. 2024].

2 Simulation upgrades

2.1 Qibotn

A new tensor network simulation package will be provided, which will built on top of Qibo. This is crucial to increase the number of simulated qubits, in order to treat problems of high interest in Quantum Computing, Quantum Machine Learning and Quantum Chemistry.

Deliverables

- del 1 [by Jan 1990];
- del 2 [by Jan 1990];
- del 3 [by Jan 1990].

2.2 Quantum Federated Learning

Federated Learning is a cooperative approach to computational problems. Widely used in Machine Learning, it implements a decentralized training of the models, making more than one process unit cooperating to get the final common result. Thanks to the agnostic structure of Qibo, Qibolab and Qibocal, we aim to implement a quantum version of the FL strategy, in which different quantum devices cooperates to a single QML training.

Deliverables and milestones

- define the QFL interface [by Nov. 2023];
- define the test problem and prepare the qubits [By Dec. 2023];
- run the QFL process [By Feb. 2024].

2.3 Multi-node

A brief abstract here.

Deliverables and milestones

- del 1 [by Jan 1990];
- del 2 [by Jan 1990];
- del 3 [by Jan 1990].

3 Validation algorithms

3.1 Determining the proton content with real superconducting qubits

Quantum computers has been proved to be able to successfully fit the proton content in the recent years. We aim to deploy the entire fitting process on a real superconducting chip.

Deliverables and milestones

- adapt the simulation problem to the hardware execution [by Sep 2024];
- run the experiment [by Nov 2024].

3.2 Physical pulses to train a Style-Based Quantum GAN

An hybrid classical-quantum generative model has been presented in 2021. The model was trained using the Qibo's tensorflow backend in simulation mode. The new goal is to prove the training on the real chip is possible, with the addition of a new hardware noise source. In fact, we are going to use two superconducting qubits to compute the task: one qubit will generate the noise, which will contaminate the pulses executed on the second qubit. This second qubit will be in charge of learning the target distribution.

Deliverables and milestones

- adapt the simulation problem to the hardware execution [by Nov. 2023];
- train a dummy model on hardware with the old algorithm [by Dec. 2023];
- run the experiment [by Dec 2024].

3.3 Boosting VQE with Double Braket Flow

Variational Quantum Eigensolvers (VQEs) are well-known quantum computing algorithms, which are used to find the ground state of a physical system. We are going to boost the VQE execution in Qibo using the Double Braket Flow (DBF) algorithm to diagonalize the generator of the system's dynamics.

Deliverables and milestones

- implement a validation VQE prototype [by Dec. 2023];
- add the DBF procedure to see if the problem scales better with the number of qubits [by Apr. 2024].

3.4 Noise Resistend Quantum Neural Networks

Variational Quantum Algorithms (VQAs) are proved to react particularly well to the noise impact during the training process. With this work we aim to study many techniques to improve the robustness of Quantum Neural Network if used in a noisy landscape. We will focus on adversarial learning and quantum error mitigation strategies to define a "good practice" way to initialize a VQA problem.

Deliverables and milestones

- implement a validation VQE prototype [by Dec. 2023];
- add the DBF procedure to see if the problem scales better with the number of qubits [by Dec. 2023].