QML-HEP GSoC 2021 Tasks

Instructions

Below are 5 evaluation tasks that you will perform to demonstrate the skills needed for the projects. All students, regardless of project, will have to complete Tasks I, IV, and V.

If you're interested in "Implementation of Quantum Generative Models to Perform High Energy Physics Analysis at the LHC", you will have to complete Task II. If you're interested in "Quantum Neural Networks for High Energy Physics Analysis at the LHC", you will have to complete Task III. Students interested in both projects will have to complete all tasks.

We *highly* recommend students to complete **all tasks**, regardless of your project of interest, to maximize the chances of being selected.

Submission Instructions

For the exercise, it's recommended to use a Python notebook. There are 2 ways to submit your work:

1. (Recommended) Upload the .ipynb file to Github and send us the link. *Make sure that it is a public repository.*

2. Alternatively, send us the .ipynb file and a PDF version of the .ipynb file with the corresponding cell outputs.

Please send your completed work to ml4-sci@cern.ch with "Evaluation Test: QML-HEP" in the title.

Task I: Quantum Computing Part

1) implement a simple quantum operation with Cirq
   a) With 5 qubits
   b) Apply Hadamard operation on every qubit
   c) Apply CNOT operation on (0, 1), (1,2), (2,3), (3,4)
   d) SWAP (0, 4)
   e) Rotate X with pi/2 on any qubit
   f) Plot the circuit

2) Create a circuit that is a series of small `cirq.Rx` rotations and plot the probability of measuring the state in the |0⟩ state.

Task II: Quantum Generative Adversarial Network (QGAN) Part

You will explore how best to apply a quantum generative adversarial network (QGAN) to solve a High Energy Data analysis issue, more specifically, separating the signal events from the background events. You should use the Google Cirq and Tensorflow Quantum (TFQ) libraries for this task.

A set of input samples (simulated with Delphes) is provided in NumPy NPZ format [Download Input]. In the input file, there are only 100 samples for training and 100 samples for testing so it won't take much computing resources to accomplish this task. The signal events are labeled with 1 while the background events are labeled with 0.

Be sure to show that you understand how to fine tune your machine learning model to improve the performance. The performance can be evaluated with classification accuracy or Area Under ROC Curve (AUC).

Task III: Quantum Convolutional Neural Network (QCNN) Part

Your task is to setup and apply a quantum convolutional neural network (QCNN) on particle physics data to perform binary classification on two types of objects (electrons and photons). You should use TFQ for this task.

The electron-photon dataset (which can be found here) contains 100 samples for training and another 100 for testing, laid out as follows:

- data["x_train"]: Training dataset of 100 32x32 images containing the particles' energy (100, 32, 32)
- data["y_train"]:" Training labels, 0 = "photon", 1 = "electron" (100,)

- data["x_test"]: Test dataset of 100 32x32 images containing the particles' energy (100, 32, 32)
- data["y_test"]:" Test labels, 0 = "photon", 1 = "electron" (100,)

The dataset labels are labelled 0 for photons and 1 for electrons. Your task is to implement a QCNN model in Tensorflow Quantum that uses this dataset's input and performs binary classification. Please feel free to experiment with different ways of encoding the classical data inputs into the qubits.

Specifically, show that the model fits the dataset and that your training loss decreases over time. (Given the small dataset size, we will not be focusing on the accuracy of your model).

Task IV: Classical Graph Neural Network (GNN) Part

For Task IV, you will use ParticleNet's data for Quark/Gluon jet classification available here with its corresponding description.

- Choose 2 Graph-based architectures of your choice to classify jets as being quarks or gluons. Provide a description on what considerations you have taken to project this point-cloud dataset to a set of interconnected nodes and edges.
- Discuss the resulting performance of the 2 chosen architectures.

Task V: Open Task Part

Please comment on quantum computing or quantum machine learning. You can also comment on one quantum algorithm or one quantum software you are familiar with. You can also suggest methods you think are good and you would like to work. Please use your own understanding. Comments copied from the internet will not be considered.