Quantum Machine Learning with Python

Carlos Cocha

PyHEP.dev Workshop, 2023
July 26, 2023

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

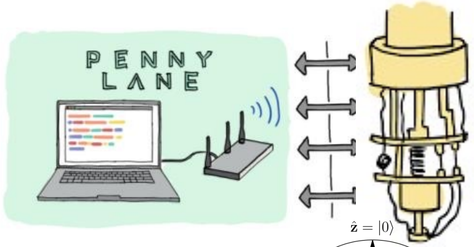
Background Info

- I am Carlos, from Ecuador
- B.Sc. in Physics at Yachay Tech (Ecuador)
- Master in Physics at the University of Padova (Italy):
  - ❑ First steps in HPC
  - ❑ Quantum Machine Learning for jet classification @LHCb
- Ph.D. student at Heidelberg University (Germany):
  - ❑ Search for dark photons from Charm decays @LHCb
    - ➢ HLT2 Trigger Lines development
    - ➢ Mass resolution studies

Quantum Machine Learning (QML)

- QML = Quantum Computing + Machine Learning -> deal with the increased luminosity & limited bandwidth @ HL-LHC
- Quantum Computer is trained as a neural network.
- Quantum hardware is simulated using Pennylane (quantum differentiable programming)

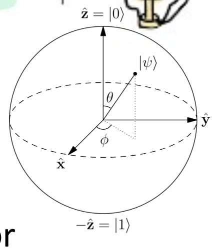
Workflow of a QML algorithm:



1. Preprocess the Data
- Scale the data to be suitable for the Quantum Circuit

2. Choose a QML model →
- QC simulation

3. Embed the data into the model → $x \to |x\rangle$ →



- Feature map into quantum states

4. Train the model → Mean Square Error
- Update the QML parameters up to minimize the cost function.

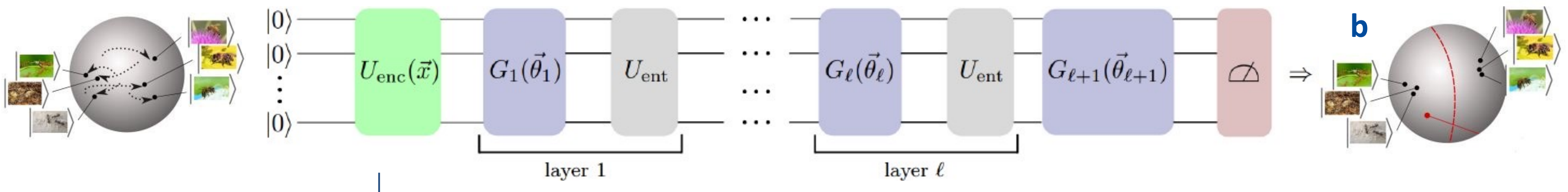
5. Evaluate the model → $\frac{N^{tag}_{b(c)}}{N^{tot}_{b(c)}}$
- Calculate the test accuracy using data not used in the training

6. Tune classical parameters → $W_{new} = W_{old} - \lambda \nabla_{W_{old}} \mathbb{C}$
- Learning rate can be modified to improve the accuracy

7. Measure the performance → QML vs. ML
- Use a scorer and compare it with the classical ML method.

Results

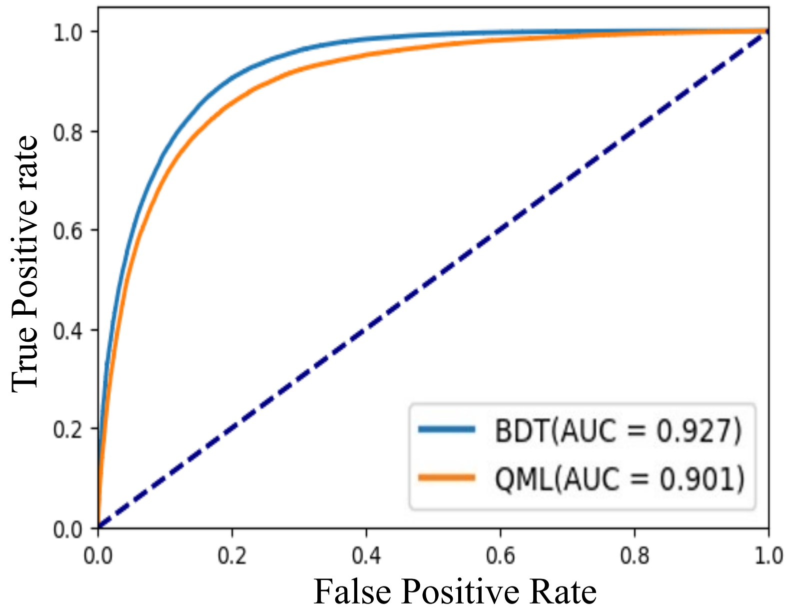
$$n_{features} = n_{qubits}$$



Accuracy

qubits \ Layers	1	2	3	4	5	6	7	8	9	10
13	0.61	0.7652	0.7886	0.8028	0.8087	0.8173	0.8246	0.8291	0.8286	0.8291
12	0.6048	0.7649	0.7895	0.8083	0.8143	0.8168	0.8211	0.8275	0.8266	0.8281
11	0.645	0.7808	0.7863	0.8057	0.8083	0.8199	0.8152	0.8222	0.8205	0.8225
10	0.6195	0.7667	0.798	0.8078	0.816	0.8192	0.8187	0.8202	0.8233	0.8227
9	0.6466	0.7663	0.7972	0.8081	0.8109	0.8183	0.819	0.8231	0.8199	0.8235
8	0.6238	0.7855	0.806	0.8121	0.8128	0.8217	0.824	0.8255	0.8192	0.8261
7	0.604	0.7619	0.794	0.8108	0.8169	0.8145	0.8171	0.8185	0.8143	0.8182
6	0.6129	0.7875	0.7997	0.805	0.8139	0.8177	0.8193	0.8212	0.8199	0.8212
5	0.6073	0.7578	0.7885	0.7971	0.8031	0.8024	0.7964	0.8011	0.8015	0.8041
4	0.7415	0.7555	0.7902	0.7938	0.7948	0.7963	0.8005	0.8023	0.8008	0.8041
3	0.7178	0.7654	0.7901	0.7899	0.7983	0.7962	0.798	0.8015	0.8023	0.8029
2	0.6192	0.7482	0.7616	0.7626	0.7676	0.763	0.7702	0.7679	0.7695	0.771
1	0.5812	0.6859	0.6792	0.677	0.6784	0.679	0.6858	0.6782	0.6855	0.6856

ROC curve



- Big room for improvement.
- New QC can be implemented

Future Prospects:

- QML developments:
  - Apply new QML algorithms and learn how to exploit features correlations.
  - Take profit of Quantum hardware with many qubits (simulations is computational demanding).
  - LHCb ongoing projects:
    - QC for Track Reconstruction
    - QML for b-jet flavour tagging

- About this workshop:
  - Discussions about open source development.
  - Understand the needs in HEP analysis and how developers deal with it.
  - Packaging of tools.

Thanks