Quantum Computing and Machine Learning (614551008)

Practice 1.2 (2023-2024)
Quantum Neural Networks

INSTRUCTIONS:

- **Deadline:** December 4^{th}, 23:59.
- **Objectives**
  - In this practice we will develop a quantum neural network model for the *wine* dataset.
- **Dataset**

  - "*Wine*" is a dataset with data about the chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.
  - The dataset is composed of 178 instances, each of them composed of 13 features (alcohol, malic acid, ash, etc.) and a target label representing one of each of the three possible classes of wine.
  - The dataset can be easily downloaded from *scikit-learn* using the following instructions:

```
from sklearn.datasets import load_wine
wine_data = load_wine()
features = wine_data.data
labels = wine_data.target
```

- **Tasks to be carried out**

  1. **Preprocessing**.
     - Preprocess the dataset to prepare it to feed the quantum neural network: Since all the data is numeric we have only to normalize then using, for example, the `MinMaxScaler` from `scikit-learn`.
     - Divide the data into train and test, using, for example, the `train_test_split()` method from `scikit-learn`.
  2. **Develop a quantum neural network to predict class of each image**.
     - Choose a quantum feature map to represent the wine data from the data encoding circuits available in Qiskit. (`https://docs.quantum-computing.ibm.com/api/qiskit/circuit_library#data-encoding-circuits`)
     - Choose an ansatz (the classifier circuit) from the circuit library (`https://qiskit.org/documentation/apidoc/circuit_library.html`), such as `TwoLocal` or `RealAmplitudes`.
     - Choose an optimizer from the several available in `https://qiskit.org/documentation/stubs/qiskit.algorithms.optimizers.html`
     - Build a Variational Quantum Classifier (VQC) using this feature map, ansatz and optimizer.
  3. **Run the VQC with the train data**.
     - Recording the time elapsed in the training and the intermediate of the objective function (represented in a chart).
  4. **Variations**.
     - Run the experiment several times with different combinations of feature map, ansatz and optimizer.
     - Compare the results obtained.
  5. **Conclusions**.
     - Reach some final conclusions for the experiment carried out.

- **Submission**

  - The exercises will be developed using Jupyter Notebooks.
  - **The notebook should include**:
    - The practice can be carried out alone or in pairs, so the first cell of the notebook must be the full names of the authors.
    - Include all the code developed.
    - The code shall be accompanied by cells with an explanatory report containing a description of the process followed, detailing the results obtained and justifying the decisions taken.
    - The notebook will be saved with the results of its execution included.
  - **Submission process**
    - The exercises will be submitted shared with the professor (Eduardo Mosqueira Rey) using the Teams platform of the master's degree.
    - There is a strict deadline for each assignment. Past due submissions will be rejected.