Hybrid Quantum Neural Network for binary classification with qiskit

Orazio Pontorno

University of Catania
Dipartimento di Matematica e Informatica

July 10, 2023

Quantum Computing(1/2)

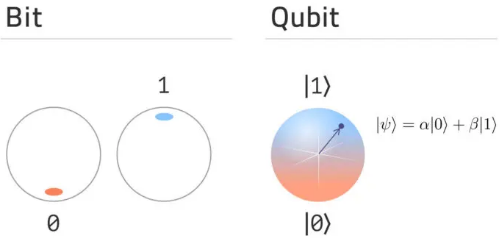
Quantum Computing

Quantum computing is a type of computing that uses quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. Instead of using classical bits that can only be in a state of 0 or 1, quantum computing uses quantum bits (qubits) that can be in multiple states at once. This allows quantum computers to process certain types of problems exponentially faster than classical computers.

Quantum Computing(1/2)

Quantum Computing

Quantum computing is a type of computing that uses quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. Instead of using classical bits that can only be in a state of 0 or 1, quantum computing uses quantum bits (qubits) that can be in multiple states at once. This allows quantum computers to process certain types of problems exponentially faster than classical computers.



Quantum Computing(2/2)

What are the advantages?

The computational power of a quantum computer is much higher than that of a classical computer. Quantum computers are not just as powerful as digital computers today but are even more powerful.

- Fastest Calculations
- Storing and Retrieving Data
- Solve Complex Problems
- Faster Computations
- New Technologies
- High Privacy
- Medicine Creation
- Internet Security
- Artificial intelligence
- Deep Learning

and so on...

Quantum Machine Learning

What is Quantum Machine Learning?

Quantum Machine Learning (QML) is a relatively novel discipline that brings together concepts from MachineLearni ng (ML), Quantum Computing (QC) and Quantum Information (QI). The great development experienced by QC, partly due to the involvement of giant technological companies as well as the popularity and success of ML have been responsible of making QML one of the main streams for researchers working on fuzzy borders between Physics, Mathematics and Computer Science.

Quantum Machine Learning

What is Quantum Machine Learning?

Quantum Machine Learning (QML) is a relatively novel discipline that brings together concepts from MachineLearni ng (ML), Quantum Computing (QC) and Quantum Information (QI). The great development experienced by QC, partly due to the involvement of giant technological companies as well as the popularity and success of ML have been responsible of making QML one of the main streams for researchers working on fuzzy borders between Physics, Mathematics and Computer Science.

Before we delve into the construction of the hybrid network, let us expose the advantages and main differences between a classical neural network and a network making use of quantum computation.

Classical Neural Networks (1/2)

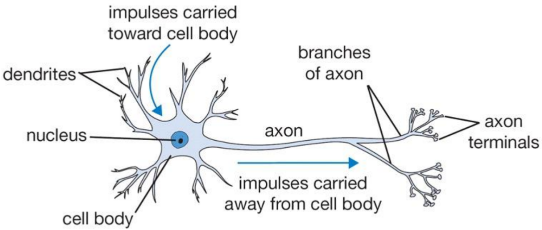
Neural Network

A neural network is a computational model that is inspired by the structure of the human brain. It is composed of a massive number of nerve cells, called neurons. Neurons have a simple three part structure consisting of: a cell body, a set of fibers called dendrites, and a single long fiber called an axon.

Classical Neural Networks (1/2)

Neural Network

A neural network is a computational model that is inspired by the structure of the human brain. It is composed of a massive number of nerve cells, called neurons. Neurons have a simple three part structure consisting of: a cell body, a set of fibers called dendrites, and a single long fiber called an axon.



Classical Neural Networks (2/2)

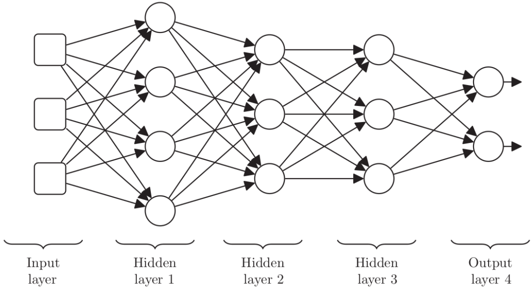
Artificial Neural Network

An artificial neural network consists of a network of simple information processing units, called neurons. The power of neural networks to model complex relationships is not the result of complex mathematical models, but rather emerges from the interactions between a large set of simple neurons.

Classical Neural Networks (2/2)

Artificial Neural Network

An artificial neural network consists of a network of simple information processing units, called neurons. The power of neural networks to model complex relationships is not the result of complex mathematical models, but rather emerges from the interactions between a large set of simple neurons.



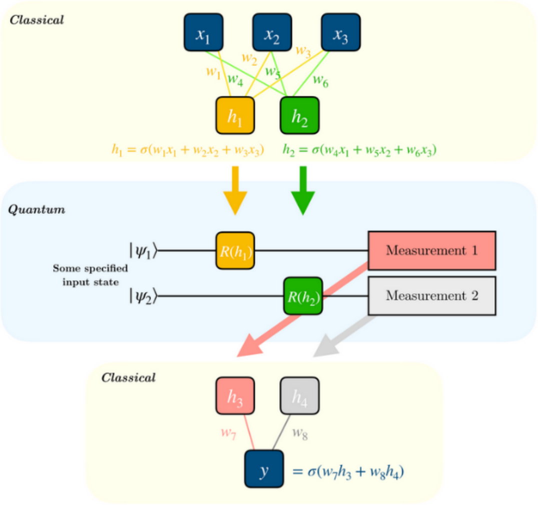
Hybrid Neural Networks (1/2)

Quantum Neural Network

A quantum neural network (QNN) is a type of neural network that uses quantum mechanics to perform computations. Like classical neural networks, QNNs consist of layers of interconnected nodes or neurons, where each neuron receives input from other neurons and produces output to be fed to the next layer. However, in QNNs, these neurons are implemented using quantum bits, or qubits, which can exist in multiple states at the same time, unlike classical bits which can only exist in one state (either 0 or 1).

Hybrid Neural Networks (2/2)

Here, σ is a nonlinear function and h_i is the value of neuron i at each hidden layer. $R(h_i)$ represents any rotation gate about an angle equal to h_i and y is the final prediction value generated from the hybrid network.



What about Backpropagation? (1/3)

How do we choose the best configuration of network weights?

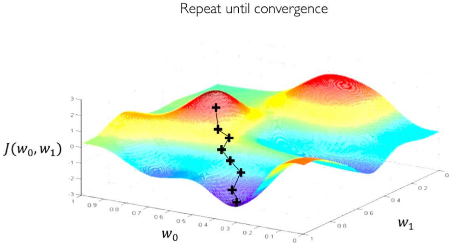
As in a classical neural network, the best configuration is sought by trying to minimise the error function, intuitively the better the configuration of weights the lower the value of the loss function. This search is performed through the use of an iterative algorithm called '*Backpropagation*'. This consists of two steps: the '*foward pass*', in which the value of the loss function is calculated using the current weights; the '*backward pass*', in which the weights are updated by minimising the loss function.

What about Backpropagation? (2/3)

Gradient Descent

The mathematical iterative algorithm that is used in the weight update phase is called '*Gradient descent*'. As the name implies, it uses the gradient calculated at a point in the function to be minimised, which shows the 'descent direction' of the function.

$$W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$$



What about Backpropagation? (3/3)

How do we calculate gradients when quantum circuits are involved?

In his paper, Gavin E. Crooks, showed how it is possible to extend the powerful optimization technique of gradient descent to the quantum case as well. The author calls this technique the **parameter shift rule**. In short, we can view a quantum circuit as a black box and the gradient of this black box with respect to its parameters can be calculated as follows:

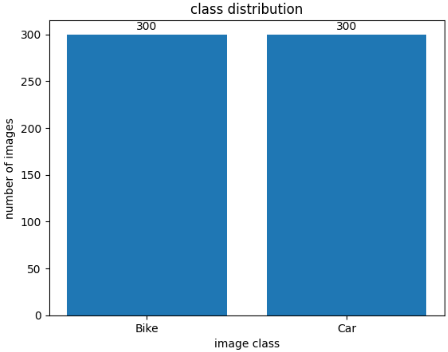
$$\nabla_\theta QC(\theta) = QC(\theta + s) - QC(\theta - s)$$

where θ represents the parameters of the quantum circuit and s is a macroscopic shift. The gradient is then simply the difference between our quantum circuit evaluated at $\theta + s$ and $\theta - s$. Thus, we can systematically differentiate our quantum circuit as part of a larger backpropagation routine.

CASE STUDY

Case study - The Dataset

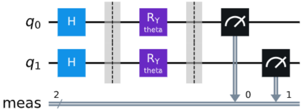
The dataset used for our case study consists images of cars and motorbikes (class 1 and 0 respectively). As a first step, exploratory data analysis (EDA) and preprocessing of the images contained in the dataset were carried out. As can be seen, not all the images contained in the original dataset on the kaggle site were used, but only a portion of them: specifically, 300 images were taken for each class. From this it follows that the dataset used turns out to be balanced with respect to the classes.



Case study - The Hybrid QCNN Architecture

Our hibrid neural network is a typical Convolutional Neural Network with two fully-connected layers at the end.

The value of the last neuron of the fully-connected layer is fed as the parameter θ into our quantum circuit. The circuit measurement then serves as the final prediction for 0 or 1 as provided by a σ_z measurement.

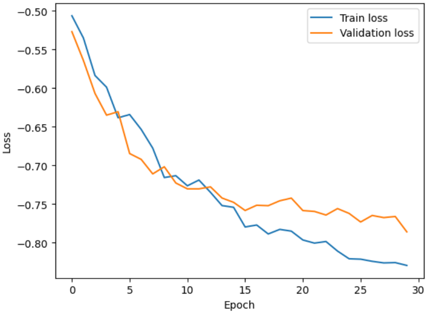



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 6, 124, 124]             456
         MaxPool2d-2         [-1, 6, 123, 123]               0
         Dropout2d-3         [-1, 6, 123, 123]               0
            Conv2d-4          [-1, 15, 62, 62]             825
         MaxPool2d-5          [-1, 15, 61, 61]               0
         Dropout2d-6          [-1, 15, 61, 61]               0
            Linear-7                 [-1, 120]       6,697,920
         Dropout2d-8                 [-1, 120]               0
            Linear-9                  [-1, 84]          10,164
           Linear-10                   [-1, 1]              85
           Hybrid-11                   [-1, 2]               0
================================================================
Total params: 6,709,450
Trainable params: 6,709,450
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.72
Forward/backward pass size (MB): 3.38
Params size (MB): 25.59
Estimated Total Size (MB): 29.69
----------------------------------------------------------------
```

Case study - Training phase

For network training, we used the loss function *Negative Log-Likelihood* and the optimiser *Adam*. All this using the following hyperparameters:

- *Epochs*: 30.
- *batch size*: 1.
- *learning rate*: 10^{-5}
- *weights decay*: 10^{-3}



As can be seen from the graphs, the loss curves have maintained an optimal trend without leaving the suspicion of any possible underfitting and overfitting error. A final value of *NLL* of 0.83 for the training set and 0.79 for the validation set was reached.

Case study - Testing and Results

The good results obtained in the training phase of the model were repeated in the testing phase; in fact, by testing the model on the images of the test set, we were able to obtain a value of NLL equal to $-\mathbf{0.78}$ and a value of accuracy equal to **86.7%**. Some predictions on images from the test set are given below.

References

- learn.qiskit.org
- Quantum Machine Learning: A tutorial

THANKS FOR YOUR ATTENTION!