MNIST Classification with Quantum Phase Estimation

Angelo Caponnetto

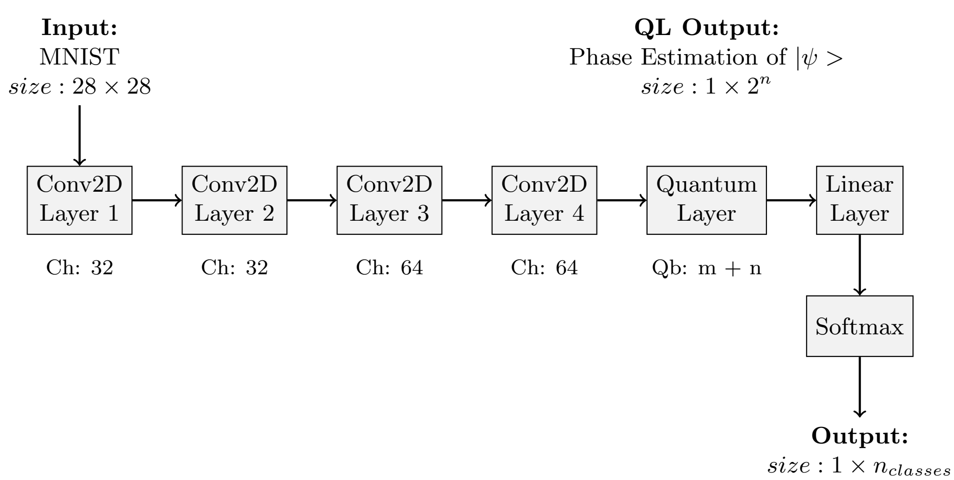
May 2025

1 Introduction

MNIST classification is a well-known benchmark that allows researchers to evaluate their models on a relatively simple task. In this work, the goal is to develop a hybrid neural network capable of classifying MNIST digits by leveraging Quantum Phase Estimation (QPE).

2 Neural Network

The neural network consists of two main components: a classical block and a quantum block. The classical block is responsible for extracting features from the input images and is composed of four convolutional layers.



The quantum block performs the phase estimation. It consists of a single layer with $m + n$ qubits: the first m qubits are used to construct the state $|\psi\rangle$, while the remaining n qubits are used to estimate its phase.

This measurement yields a probability vector in $\mathbb{R}^{2^n}$, where each component corresponds to a specific bit string among the 2^n possible outcomes (since n qubits yield 2^n bit strings).

The final layer maps the output of the quantum layer (which may have a higher dimensionality) to the actual number of classes to be classified.

Quantum Phase Estimation (QPE) involves selecting the bit string with the highest probability and converting it into a phase $\phi \in [0, 1]$, according to Equation (1). The larger the value of m, the smaller the error in estimating ϕ, with an upper bound of $\epsilon_\phi < \frac{1}{2^m}$.

$$\phi = \sum_{\substack{k=1, \\ j \in \{0,1\}^n}}^{n} \frac{j}{2^k} \tag{1}$$

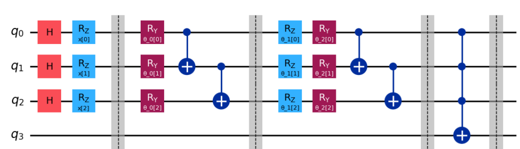
Starting from images of size 28×28, the classical block outputs scalar features x_i, with $i = 1, 2, \ldots, m-1$. Each x_i is then used as a rotation parameter in the quantum layer.

3 Quantum Layer

The quantum layer is made of three main blocks: *Init + Ansatz, U Gates* and *IFT* (inverse Fourier Transform).

3.1 Init + Ansatz

The block has the following structure (illustrated here with 4 qubits for the sake of simplicity):



The quantum block is composed of multiple R_z and R_y gates. The first rotation on each qubit uses the features produced by the classical convolutional layers, while the subsequent rotations use learnable parameters. The sequence $R_z + R_y +$ CNOT can be applied multiple times. Finally, a multi-controlled CNOT is applied to entangle the last qubit with all the others. The state of the last qubit can be expressed as:

$$|\psi >= a|1 > +b|0 > \tag{2}$$

where a and b depend on all the parameters.

3.2 U Gates

QPE requires a unitary gates such that:

$$U|\psi >= e^{-i2\pi\phi}|\psi > \tag{3}$$

$$U = V\Lambda V^{\dagger} \quad \text{where} \quad V = \begin{bmatrix} b & -a^* \\ a & b^* \end{bmatrix}, \quad \Lambda = \begin{bmatrix} e^{-i2\pi\phi} & 0 \\ 0 & e^{i2\pi\phi} \end{bmatrix} \tag{4}$$

Here ϕ is the phase we want to estimate.
By calculating, the unitary U becomes:

$$U = \begin{bmatrix} \cos(2\pi\phi) - ic\sin(2\pi\phi) & 2ia^*b\sin(2\pi\phi) \\ \\ 2iab^*\sin(2\pi\phi) & \cos(2\pi\phi) + ic\sin(2\pi\phi) \end{bmatrix} \tag{5}$$

where $c = |b|^2 - |a|^2 = \frac{2^m - 2}{2^m}$.

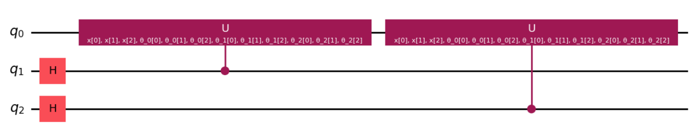
U can also be written as a composition of standard gates, $R_z + U_3$:

$$U = U_3 R_z = \begin{bmatrix} e^{-i\alpha}\cos\theta & -e^{i(\alpha+\lambda)}\sin\theta \\ e^{i(\beta-\alpha)}\sin\theta & e^{i(\alpha+\beta+\lambda)}\cos\theta \end{bmatrix} \tag{6}$$

Comparing equations 5 and 6, all the angles can be written as:

$$\begin{aligned} &\alpha = \arctan\left[c\tan(2\pi\phi)\right], \quad \theta = \arccos\left(\frac{\cos(2\pi\phi)}{\cos\alpha}\right) \\ &\lambda = -\alpha + \arctan(\cot\phi), \quad \beta = -\lambda \end{aligned} \tag{7}$$

Knowing U allows us to construct the U-gate block. It consists of n qubits (2 in the figure below for simplicity).



Note that q_0 is the qubit where $|\psi >$ was prepared.

3.3 IFT

At the end of the circuit, a IFT gate is applied to measure the global phase ϕ. This gate produces a state (on the last m qubits) of the form:

$$|\psi_{IFT}>= \sum_x \alpha_x |x> \quad where \quad \alpha_x = \frac{1}{2^m} \sum_k e^{-i2\pi k(\phi-\phi_x)} \tag{8}$$

The probability of measuring the state $|x>$ is then:

$$P_x = \frac{1}{2^{2m}} \left(\frac{\sin(2^m \pi(\phi-\phi_x))}{\sin(\pi(\phi-\phi_x))} \right)^2 \tag{9}$$

In Equations (8) and (9), ϕ is the real (continuous) phase of $|\psi\rangle$, while ϕ_x is the quantized phase associated with the state $|x\rangle$, which is evaluated using Equation (1).

Any arbitrary function can be chosen for the global phase of $|\psi\rangle$. Here:

$$\phi = \sum_{weights} \omega_i^2 + \sum_{inputs} x_i^2 \tag{10}$$

where x_i are the features that come from convolutions, while ω_i are the learnable weights of the quantum layer.

4 Training

Once the neural network is built, we can focus on its training.
The model is trained using **cross-entropy** loss between its output and the image label, thereby forcing it to map each input to one of the available quantized phases ($\mathbb{R}^{2^n}$) corresponding to a single label.

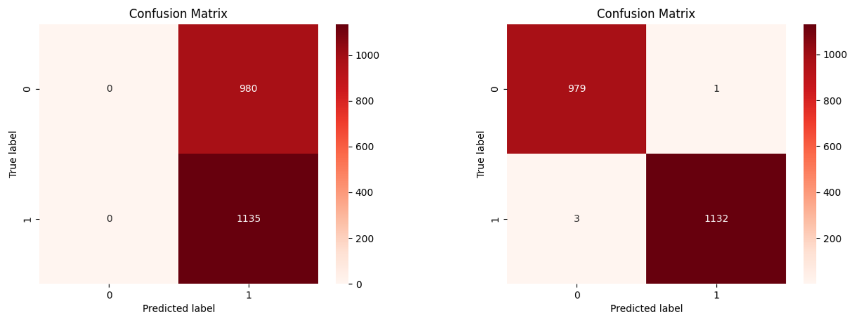
5 Results

Different experiments have been conducted, with the neural network being trained to classify a different number of digits in each experiment. For each configuration $m = 5$ and $n\_iter = 5$ where n_iter is the number of repetition of the block $R_z + R_y + CNOT$. The number of qubits for phase estimation n changes in each configuration.

5.1 0 - 1 Digits

The model was trained to distinguish 0 and 1 digits. In this configuration:

- $n = 1 \quad \Rightarrow \quad \phi_x = \{0, 0.5\}$ allowed phases (from eq 1)
- **accuracy**: 0.998



(a) Classification before training. (b) Classification after training.

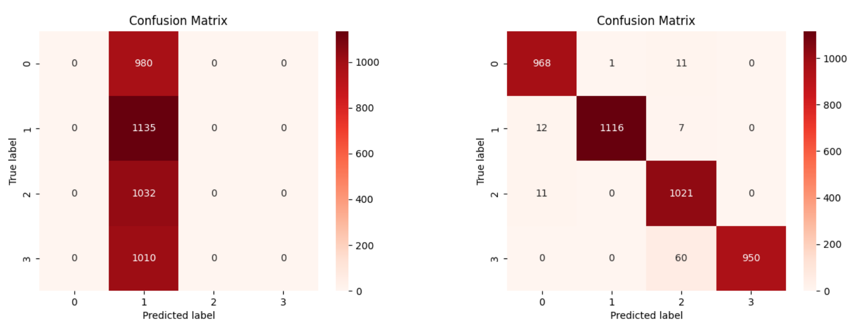
Figure 1: Classifier confusion matrices computed on the test set for digits 0, 1.

As shown in Figure 1, before training, all the images were classified with the same label (phase). After training, each digit has its own phase.

5.2 0 - 3 Digits

The model was trained to distinguish digits from 0 to 3. In this configuration:

- $n = 2 \quad \Rightarrow \quad \phi_x = \{0, 0.5, 0.25, 0.75\}$ allowed phases (from eq 1)
- **accuracy**: 0.975

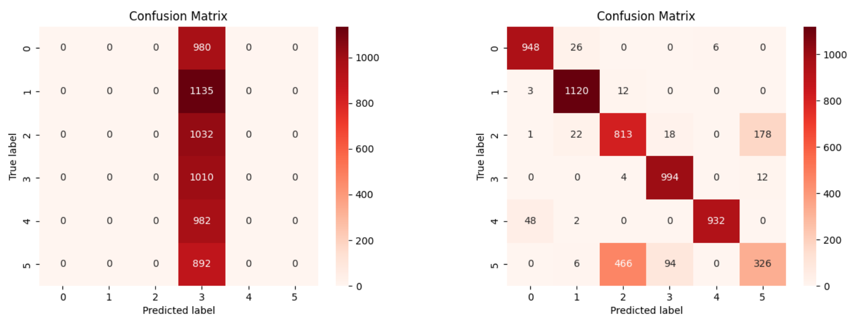


(a) Classification before training. (b) Classification after training.

5.3 0 - 5 Digits

The model was trained to distinguish digits from 0 to 5. In this configuration:

- $n = 3 \quad \Rightarrow \quad \phi_x = \{0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875\}$
- **accuracy**: 0.851

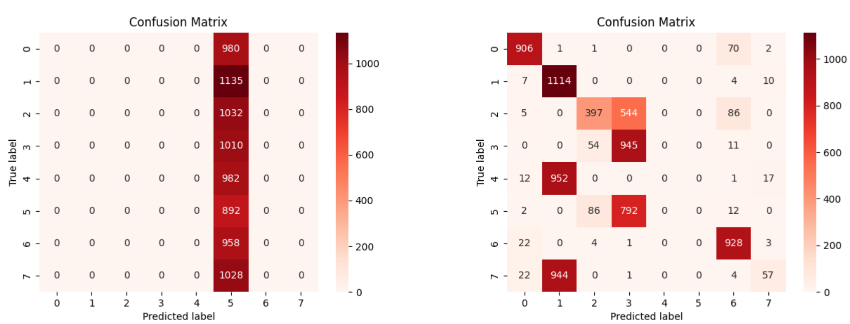


(a) Classification before training. (b) Classification after training.

5.4 0 - 7 Digits

The model was trained to distinguish digits from 0 to 7. In this configuration:

- $n = 3 \quad \Rightarrow \quad \phi_x = \{0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875\}$
- **accuracy**: 0.542



(a) Classification before training. (b) Classification after training.

6 Conclusions

The confusion matrices in Section 5 indicate that the classifier performs well when the number of labels is small. However, as the number of labels grows, classification errors increase. This is likely because the quantum layer (and the linear layer) have very few learnable parameters, which may limit the model's ability to capture all the relevant features across all classes.

Additionally, using all digits (0–9) would require 4 qubits, capable of representing up to 16 classes. Because only 10 classes are actually needed, this leads to unnecessary use of computational resources.