

ENSIMAG & NEOVISION

PROJET DE SPÉCIALITÉ

---

# Traitement d'images par réseaux de neurones quantiques

---



ACHOUHAD REDA  
ADIOUANE MUSTAPHA  
AGHIL AYMANE  
AKDIM YASSER  
BAKIR WALID

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Phases de projets</b>	<b>2</b>
2.1	Se familiariser avec le projet . . . . .	2
2.2	Déroulement . . . . .	3
2.2.1	Premier pas vers l'informatique quantique . . . . .	3
2.2.2	Recherche bibliographique . . . . .	3
<b>3</b>	<b>CNN + Quantum layer</b>	<b>4</b>
3.1	Jeu de donnée . . . . .	4
3.2	modèle classique . . . . .	4
3.3	Introduction à l'informatique quantique . . . . .	5
3.3.1	un qubit un n-qubit . . . . .	5
3.3.2	une porte quantique . . . . .	6
3.3.3	sphère de bloch . . . . .	7
3.3.4	circuit quantique . . . . .	7
3.3.5	état intriqué . . . . .	8
3.4	Réseau de neurones quantiques . . . . .	8
3.4.1	Méthode 1 . . . . .	8
3.4.2	Méthode 2 . . . . .	10
<b>4</b>	<b>Réseaux de neurones quanvolutionnels (QNN) :</b>	<b>12</b>
4.1	Circuit non paramétré : . . . . .	12
4.2	Circuit paramétré : . . . . .	14
<b>5</b>	<b>Transfer Learning</b>	<b>14</b>

## 1 Introduction

L'informatique quantique est une méthode moderne de calcul basée sur la science de la mécanique quantique. C'est une belle combinaison de physique, de mathématiques, d'informatique et de théorie de l'information. Il offre une grande puissance de calcul, une consommation d'énergie réduite et une vitesse exponentielle par rapport aux ordinateurs classiques.

Dans ce projet, nous tenterons d'appliquer l'informatique quantique à la reconnaissance d'images. Nous étudierons des implémentations de réseaux de neurones, dont une partie est codée avec des circuits quantiques ainsi que d'étudier le prétraitement quantique sur l'image en amont du réseau.

**Problematique :** Quelle est la contribution des réseaux quantiques à la reconnaissance des images ?

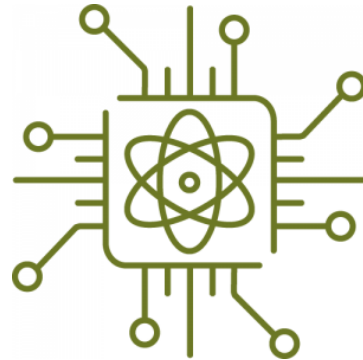
## 2 Phases de projets

Ce projet traite en grande partie de l'informatique quantique, un sujet qui était nouveau pour tous les membres de l'équipe. Ainsi, la partie apprentissage du projet a pris un temps considérable. Se familiariser avec l'informatique quantique et les circuits quantiques, ainsi qu'avec Qiskit et certaines bibliothèques de Python a été une priorité pendant la majeure partie du premier mois.

### 2.1 Se familiariser avec le projet

Cette phase a commencé après notre première rencontre avec messieurs Bernard et Mnacho. Elle a duré environ trois ou quatre semaines.

À cette étape, nous nous sommes largement appuyés sur un tutoriel d'IBM [5].

The PyTorch logo, featuring the word "PYTORCH" in a bold, sans-serif font. The letter "O" is replaced by a stylized orange flame icon.

**Qiskit**



**TensorFlow**

## 2.2 D roulement

### 2.2.1 Premier pas vers l'informatique quantique

Apr s avoir pris un temps raisonnable pour se renseigner sur l'informatique quantique et les diff rents outils qui pourraient  tre utiles tout au long du travail sur le projet, nous avons commenc  par le Notebook [2] fourni par Qiskit.

Ce Notebook comprenait le code source pour la reconnaissance d'images   l'aide de l'informatique quantique en utilisant un ensemble de donn es qui ne contient que des images  chantillons pour 0 et 1.   partir de ce code, nous avons essay  de g n raliser leur mise en  uvre pour former et pr dire en utilisant un ensemble de donn es pour tous les chiffres entre 0 et 9. Pour cela, nous avons utilis  deux m thodes et les avons compar es.

### 2.2.2 Recherche bibliographique

Cette partie est venue comme une succession naturelle de ce que nous faisions avant d'utiliser l'ordinateur portable IBM. Jusqu'  ce moment, nous avons assist    un certain point   l'effet de l'informatique quantique. Nous nous sommes appuy s sur une poign e d'articles et de recherches pour mieux comprendre ce qui peut  tre fait ensuite en termes de mise en  uvre. Nous avons essay  de rechercher d'autres circuits quantiques et leur mise en  uvre potentielle et leur utilisation dans la reconnaissance d'images, ainsi que de mener une comparaison approfondie entre ces m thodes et approches dans le but de trouver la meilleure mise en  uvre possible en

termes de complexité temporelle et de précision.

**Quantum Convolutional Neural Networks** Nous avons essayé d’extraire des caractéristiques utiles dans les données qui peuvent être exploitées à des fins de classification. Les couches convolutives d’une pile CNN sont chacune composées de  $n$  filtres convolutifs. [4]

**Trainable circuits [QCNN]** Le réseau de neurones convolutionnel quantique (QCNN) n’utilise que  $\mathcal{O}(\log N)$  pour des tailles d’entrée de  $N$  qubits, ce qui permet un apprentissage et une mise en œuvre efficaces sur des dispositifs quantiques réalistes à court terme.[3]

Nous avons essayé d’utiliser cet article pour implémenter des circuits quantiques avec des paramètres entraînables.

**Transfer Learning** Cet article tente d’appliquer ce simple constat : les connaissances acquises dans un contexte spécifique peuvent être transférées dans un autre domaine.[1]

### 3 CNN + Quantum layer

#### 3.1 Jeu de donnée

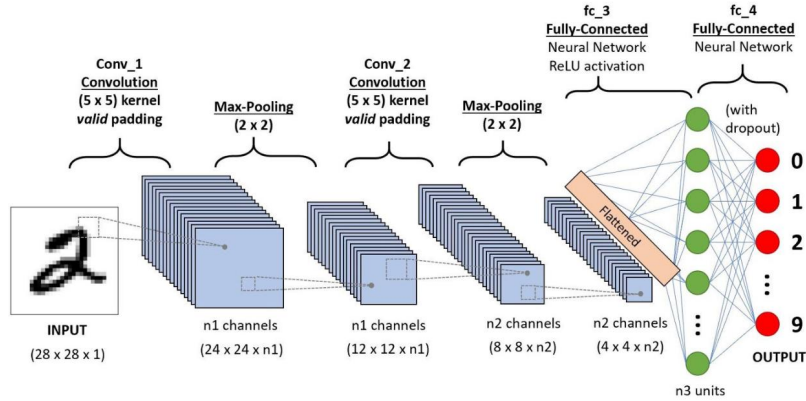
Comme les calculs sont assez complexes et que l’ensemble de données MNIST est assez vaste, nous avons choisi de travailler sur un sous-ensemble de données.

$$4000 \text{ images} = \begin{cases} 1000 & \text{training} \\ 3000 & \text{testing} \end{cases}$$

Chaque nombre dans  $0, 1, \dots, 9$  a exactement 400 images échantillons avec la même distribution que précédemment (300 pour le test et 100 pour l’entraînement). Nos données sont équiréparties sur les données cibles.

#### 3.2 modèle classique

Ce modèle sera la base de la comparaison avec les résultats que nous obtiendrons en utilisant l’informatique quantique.



Où  $n1 = 6$ ,  $n2 = 16$ , et  $n3 = 64$   
Après avoir tester ce modèle de base sur notre jeu de donnée, on a eu les performances suivantes :

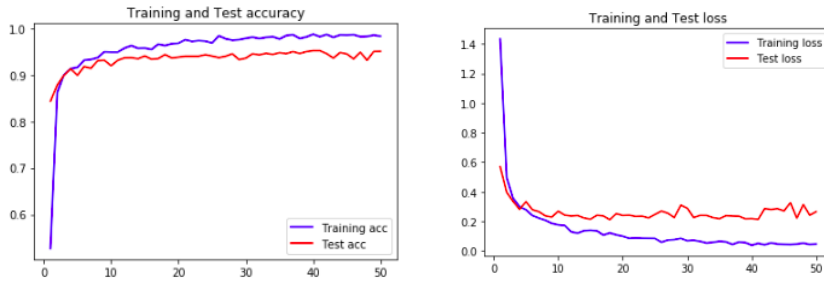


FIGURE 1 – performance CNN

Remarque : MNIST est une base de données facile à apprendre par un modèle, c'est pour cela que même avec peu de donnée, on obtient des performances acceptables.

### 3.3 Introduction à l'informatique quantique

Cette introduction présente quelques définitions fondamentales et cruciales de notions qui sont fréquemment utilisées dans ce projet.

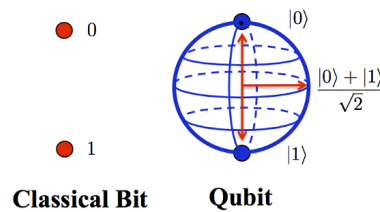
#### 3.3.1 un qubit un n-qubit

Un qubit est un bit quantique, l'équivalent dans l'informatique quantique du chiffre binaire ou bit de l'informatique classique. Le qubit est l'équivalent en informatique quantique d'un bit binaire (0 et 1), que l'on trouve sur les ordinateurs classiques. Le qubit est la base de l'informatique

quantique en raison de ses propriétés mécaniques quantiques : superposition, intrication et effet tunnel quantique.

L'équation d'un qubit en superposition est représentée comme une combinaison linéaire de  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , où  $\alpha$  et  $\beta$  sont les amplitudes de probabilité pour l'état de superposition, et  $|0\rangle$  et  $|1\rangle$  sont les résultats possibles (en termes d'états). En raison du théorème de Pythagore,  $|\alpha|^2 + |\beta|^2 = 1$  puisque la sphère de Bloch est une sphère unitaire. Un qubit en superposition aurait l'équation suivante :  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ .

on peut vérifier que la superposition donne une chance sur deux d'être dans l'état  $|0\rangle$  et l'état  $|1\rangle$ .



Remarques :

- Les Qubits peuvent être dans un état de superposition où ils sont 0 et 1 en même temps, et peuvent être représentés par une sphère de Bloch.
- $|\alpha|^2 + |\beta|^2 = 1$  doit être égal à 1, ce qui signifie que  $|\psi\rangle$  doit aussi être égal à 1.

### 3.3.2 une porte quantique

En informatique quantique, et plus précisément dans le modèle de calcul par circuits quantiques, une porte logique quantique est un circuit quantique de base fonctionnant sur un petit nombre de qubits. Ce sont les blocs de construction des circuits quantiques, comme les portes logiques classiques le sont pour les circuits numériques conventionnels, parmi quoi on énumère la négation, l'opérateur "ET", l'opérateur "OU", ... . De même, les qubits subissent des opérations de même effet, comme par exemple, la porte NOT, ou bien la porte X, qui bascule tout simplement la valeur positionnée en entrée.

Une autre porte quantique est d'utilité dans nos implémentations de code, et de circuits quantiques, est la **CNOT**, Controlled-NOT, qui joue le rôle de la transformation classique XOR.

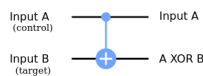


FIGURE 2 – Porte quantique CNOT

La porte de Hadamard est une opération quantique fondamentale. Elle permet de réaliser une superposition entre les deux états fondamentaux d'un qubit :  $|0\rangle$  and  $|1\rangle$ , ce qui représente le coeur de la quantique (et son utilité également). En appliquant la matrice  $H$  tel que :

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

pour une entrée  $|0\rangle$  ou bien  $|1\rangle$ , on trouve alors  $H|0\rangle = |+\rangle$  et  $H|1\rangle = |-\rangle$ , avec :

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

### 3.3.3 sphère de bloch

Nous avons vu plus tôt dans ce chapitre que l'état général d'un qubit est  $|q\rangle = \alpha|0\rangle + \beta|1\rangle$

Puisque nous ne pouvons pas mesurer la phase globale, nous pouvons seulement mesurer la différence de phase entre les états  $|0\rangle$  et  $|1\rangle$ . Au lieu d'avoir  $\alpha$  et  $\beta$  être complexes, nous pouvons les confiner aux nombres réels et ajouter un terme pour nous indiquer la phase relative entre eux :  $|q\rangle = \alpha|0\rangle + e^{i\phi}\beta|1\rangle$  où  $\alpha, \beta, \phi \in \mathbf{R}$

$$|\alpha|^2 + |\beta|^2 = 1 \Rightarrow \boxed{|q\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle} \quad \theta, \phi \in \mathbf{R}$$

Si nous interprétons et comme des coordonnées sphériques nous pouvons représenter n'importe quel état du qubit sur la surface d'une sphère, appelée sphère de Bloch.

### 3.3.4 circuit quantique

Comme sera vu dans la suite du projet, l'intégration de l'aspect quantique dans les modèles classiques de Machine Learning, ou bien les réseaux de neurones profonds, est de la forme des circuits quantiques, ou un oracle, qui réalisent une fonction paramétrée ou pas, sur une donnée en entrée. Cet oracle nécessite les opérateurs dont on a parlé plus haut. Des mesures seront alors nécessaires pour bien évaluer les sorties des circuits, sous forme des bits classiques, chacune a sa propre probabilité, qui est approximé grâce à la notion de *shots* où l'on exécute le circuit un nombre de fois, et calcule les différents sorties possibles.



### 3.3.5 état intriqué

Un état produit est un état quantique qui peut s'écrire comme une séquence de qubit à un seul état. L'état  $|\Phi^+\rangle$  qui représenté par :

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

est un deux qubit état dont on ne pourra jamais trouvé une combinaison de deux états simples. Ce sont les états intriqués. Leur utilité réside sur le fait que les deux qubits dans l'exemple donné ci-dessus sont liés entre eux, et contiennent une information partagée ce qui pourrait être utile dans des situations particulières.

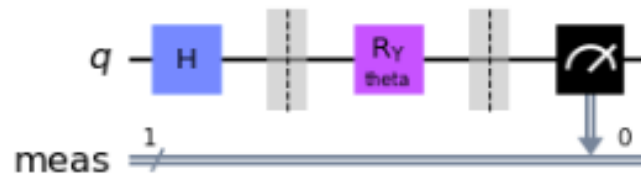
### 3.4 Réseau de neurones quantiques

Après avoir vu le modèle classique de Deep Learning, et défini les notions de base de l'informatique quantique, l'étape suivante logique serait d'intégrer les deux dans un nouveau modèle qui s'appuie sur les réseaux de neurones de classification des images (principalement MINST et Image-Net).

## Intégration circuit quantiques dans le modèle classique

#### 3.4.1 Méthode 1

Nous commençons par un cas simple, qui est une généralisation de ce que nous avons vu dans le cahier Qiskit. Dans Qiskit, on utilisait un circuit de la forme (avec un paramètre  $\theta$ ) :



Principe :

La seule option pour l'entrée est un 0. Nous utiliserons donc une rotation avec une variable  $\theta$  qui est définie sur  $\pi$  ou un 0 en fonction de l'information d'origine.

Le circuit calcule la probabilité d'obtenir 1 comme sortie. La probabilité de mesurer est élevée signifie que l'image d'entrée est un 1 (Nos données à ce stade sont juste des images échantillons de 1 et 0).

C'était l'idée derrière l'implémentation de Qiskit. Notre modèle, quant à lui, utilisera le même principe mais au lieu d'utiliser un circuit, nous en utiliserons 10. Chaque circuit effectuera la même tâche que celle décrite précédemment (calculer la probabilité d'obtenir un 1 en sortie).

L'idée derrière cette implémentation est que nous calculerons toutes les probabilités (10 probabilités, chaque circuit en fournira une) et la probabilité la plus élevée aura été fournie par le circuit indexé  $i$ . Nous adapterons notre circuit de sorte que l'image d'entrée dans ce cas soit le nombre  $i$ .

Nous testons la performance de ce modèle. Nous avons obtenu les résultats suivants :

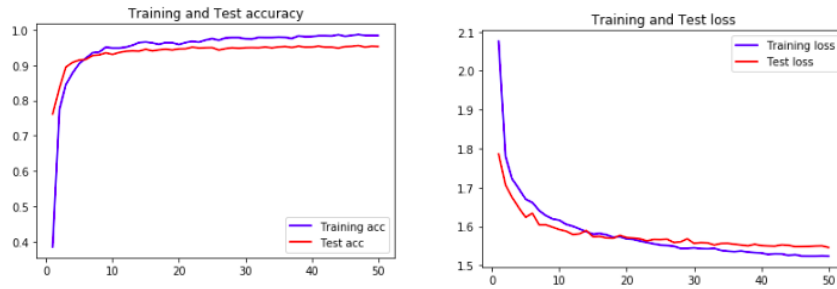
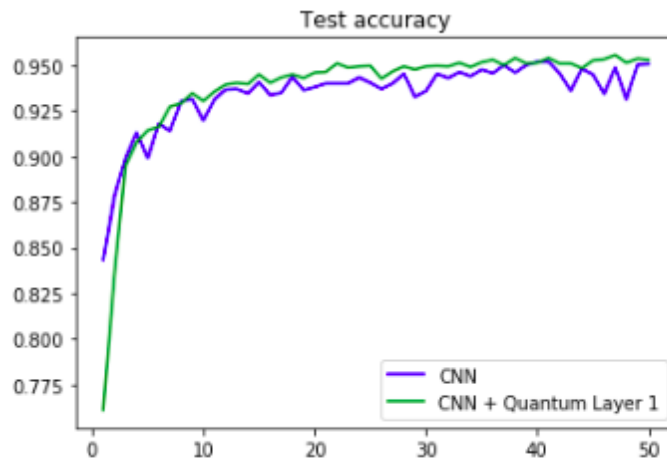


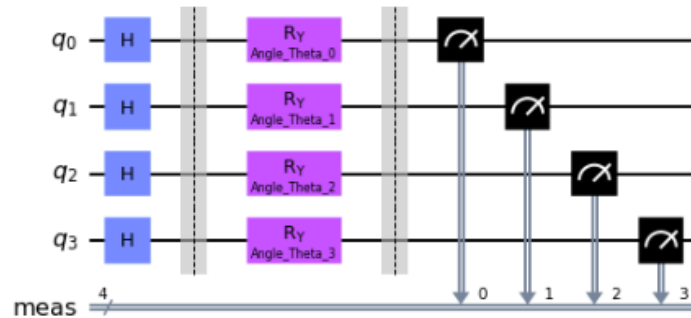
FIGURE 3 – performance modèle avec 1 layer

## Comparaison avec CNN



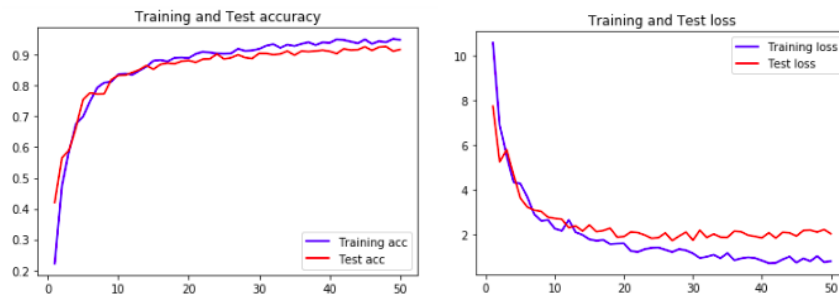
### 3.4.2 Méthode 2

Principe : 1 circuit à la fin du modèle de la forme :

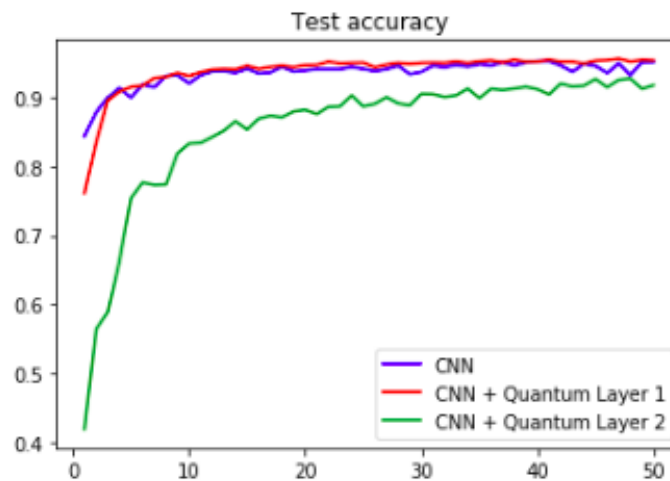
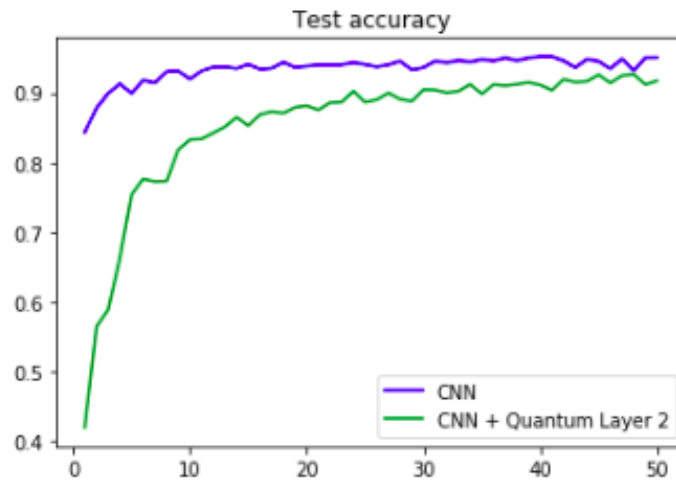


Le principe de cette deuxième méthode réside sur le fait que l'on peut représenter nos données, qui sont des images de chiffres allant de 0 à 9 (donc un total de 10 classes) par 4 bits, ou bien 4 qubits. En associant à chaque chiffre, le résultat de mesure correspondant, que l'on prend bien la représentation binaire du chiffre pour des raisons de clarté ( par exemple les images du chiffre 5 est associé à la mesure 0101).

### Performance de ce modèle



## Comparaison avec CNN et avec Modele 1



## Remarque

la deuxième méthode est moins performante car on utilise 4 qubits pour coder les chiffres de 0 à 9 mais il y a 16 états quantiques résultants d'un circuit quantique à 4 qubits. La non-surjectivité d'une telle application ou fonction entre nos entrées et sorties implique des marges d'erreurs supplémentaires au niveau du réseau implémenté. Exemple concret d'un tel conflit est d'avoir comme résultat pour une image "0", la mesure 1110, qui n'est pas associé à une entrée bien déterminée. Une autre image "9" donne par exemple 1100. Comme on ne traite pas des tels résultats (c'est qui est le cas des résultats allant de 1010 à 1111). Ainsi, on confondra les deux résultats,

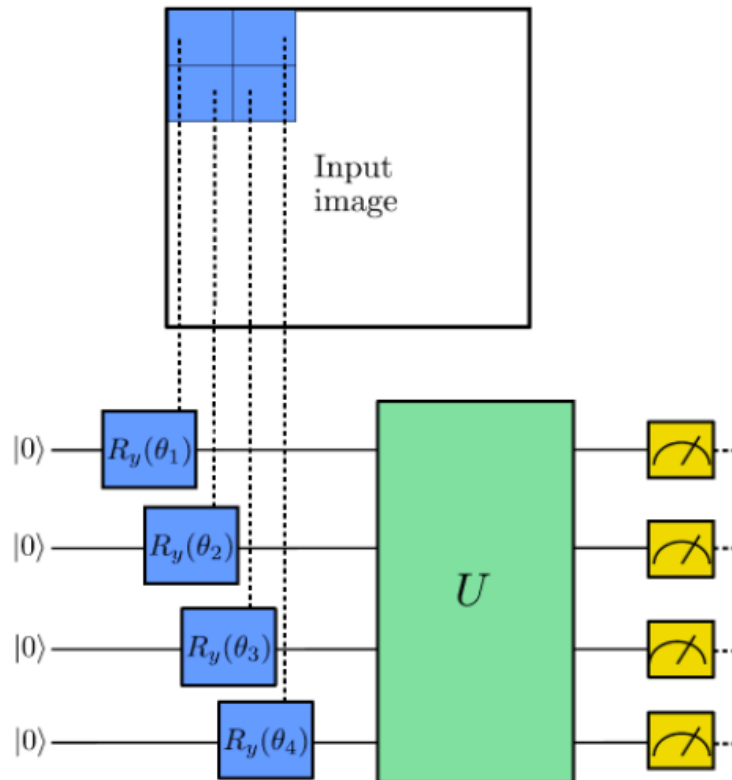
et dont les deux images, ce qui bien évidemment va réduire la précision du modèle en totalité.

## 4 Réseaux de neurones quantvolutionnels (QNN) :

### 4.1 Circuit non paramétré :

L'idée de ce modèle repose sur le fait de remplacer, ou bien ajouter, une couche quantique, qui sera à la forme d'un circuit quantique, dans la partie convolutionnelle des réseaux neurones dédiés à la classification des images. D'abord, parce que cette partie est considéré une des bases de ce type de réseau. Ensuite, pour voir comment l'informatique quantique, et plus précisément nos circuits élémentaires, peuvent conduire à l'amélioration de cette intelligence artificielle.

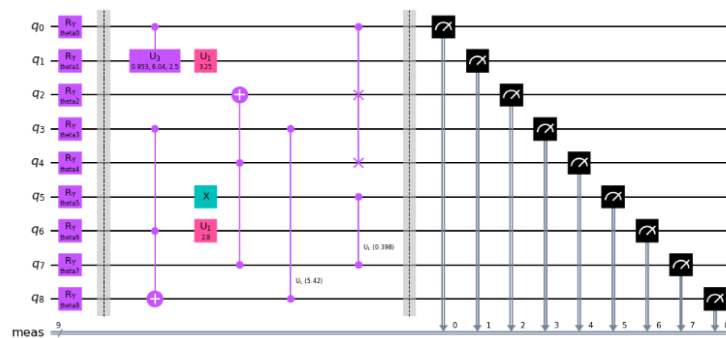
En se basant sur les articles scientifiques fournies par le sujet, et bien d'autres, un circuit qui remplacera un filtre convolutionnel doit contenir trois grandes blocs : une partie de codage, le circuit qui réalise l'opération du filtre, et une partie de décodage des mesures issues du circuit :



Codage : En faite, à ce niveau, il existe plusieurs possibilités de coder les images placées en entrée. Une est de placée au niveau de chaque qubit une porte quantique rotationnel d'un paramètre  $\theta$  où est codé les pixels de

nos images. Dans notre circuit, les pixels sont quantifiés entre la valeur 0 et 1 (valeurs de 0 à 0.5 sont traduites par 0 et les valeurs de 0.5 à 1 par 1), et donc on se ramène qu'à coder par les valeurs 0 et  $\pi$ .

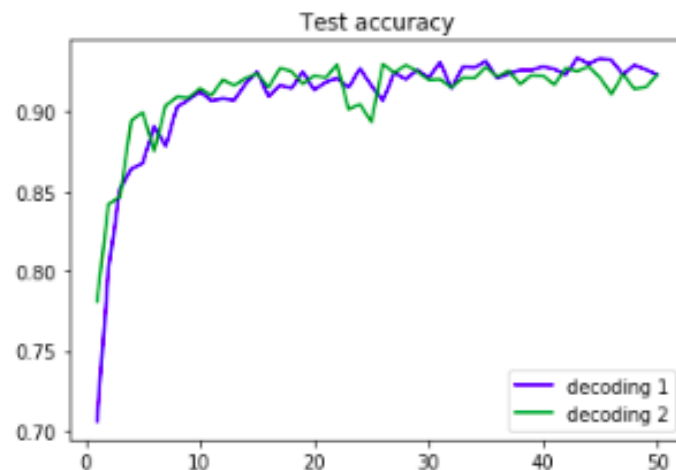
Circuit : Dans un premier temps, nous réalisons un circuit d'une manière aléatoire qui prend en entrée les images codées, et qui produit une sortie de taille de nombre des qubits, donc 4, à travers des mesures logiques. Un circuit aléatoire contient alors des portes quantiques avec paramètres bien déterminées. Un exemple d'un tel circuit généré à l'aide de Qiskit est donné ci-dessous.



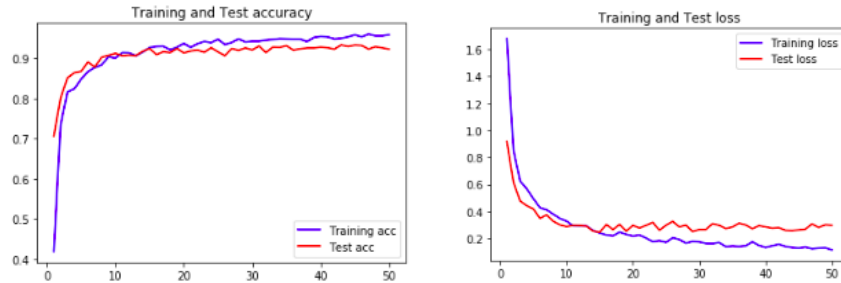
Décodage : On distingue deux méthodes de décodage implémentées :

- + l'argument max (à détailler)
- + nombre de 1 (à détailler)

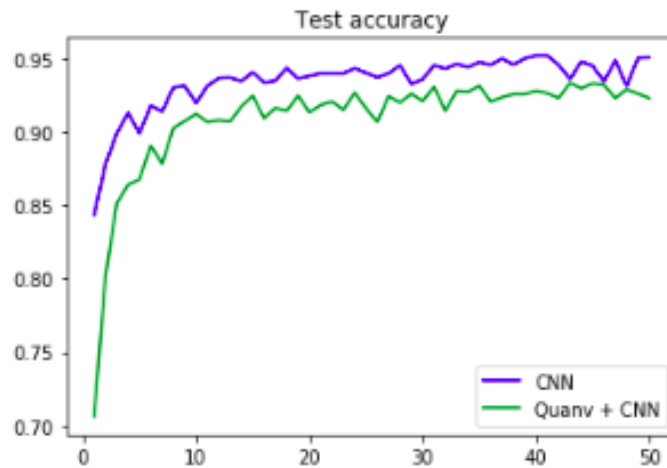
Voilà ce qu'on a eu en comparant les deux logiques de décodage :



Nous testons les performances de ce modèle, et nous obtenons les résultats suivants



## Comparaison avec CNN



**N.B :** Ce modèle quantique est considéré comme un pré-traitement (pre-processing) couche des images de la base MNIST, vu qu'il s'agit de la première étape de propagation du réseau. Donc on ne parle pas encore de paramètres entraînables, ou bien un circuit paramétrée qui est pris en compte au niveau des opérations de back-propagation.

### 4.2 Circuit paramétré :

## 5 Transfer Learning

## Références

- [1] Josh Izaac ANDREA MARI Thomas Bromley. “Transfer learning in hybrid classical-quantum neural networks”. In : *Xanadu* (2020). URL : <https://arxiv.org/pdf/1912.08278.pdf>.
- [2] *Hybrid quantum-classical Neural Networks with PyTorch and Qiskit*. URL : <https://qiskit.org/textbook/ch-machine-learning/machine-learning-qiskit-pytorch.html>.
- [3] Mikhail D. Lukin IRIS CONG Soonwon Choi. “Quantum Convolutional Neural Networks”. In : *Department of Physics, Harvard University* (2019). URL : <https://arxiv.org/pdf/1810.03787.pdf>.
- [4] Shashindra Pradhan MAXWELL HENDERSON Samriddhi Shakya. “Quantum Convolutional Neural Networks”. In : *QxBranch* (2019). URL : <https://arxiv.org/pdf/1904.04767.pdf>.
- [5] *What is Quantum ?* URL : <https://qiskit.org/textbook/what-is-quantum.html>.