

# Analyse de différentes approches d'apprentissage machine quantique

Rapport rédigé dans le cadre du cours de BSQ201

William Pomerleau, Sahar Saoudi, Yannis Ménard et Ludovic Marcotte

October 18, 2024

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Objectifs du projet</b>	<b>2</b>
<b>3</b>	<b>Théorie et méthodes</b>	<b>2</b>
3.1	Notions de base en informatique quantique . . . . .	2
3.2	Ansatz . . . . .	3
3.3	Problèmes de classification binaire . . . . .	3
3.4	Encodage Quantique De Données . . . . .	4
3.5	Réseau Neuronal Convolutif Quantique ( <b>QCNN</b> ) : . . . . .	4
3.6	Classificateur quantique variationnel ( <b>VQC</b> ) : . . . . .	5
3.7	Machine à vecteur de support quantique ( <b>QSVM</b> ) : . . . . .	6
3.7.1	Matrices de corrélation . . . . .	6
3.8	Méthode Classique - Perception Multi-Couche ( <b>MLP</b> ) : . . . . .	7
3.8.1	Matrice de confusion . . . . .	8
3.9	Ensembles de données . . . . .	8
3.9.1	HTRU2 : High Time Resolution Universe Survey . . . . .	8
3.9.2	Dry bean . . . . .	8
<b>4</b>	<b>Résultats</b>	<b>9</b>
<b>5</b>	<b>Discussion</b>	<b>10</b>
<b>6</b>	<b>Conclusion</b>	<b>11</b>
<b>7</b>	<b>Bibliographie</b>	<b>12</b>

# 1 Introduction

L'apprentissage machine classique, comme l'affirme Joey Gonzalez, professeur à UC Berkeley, est un outil puissant qui a démontré son efficacité dans de nombreux domaines de l'informatique au cours des dernières années [6]. Ce rapport explore différentes approches de l'apprentissage machine quantique (QML) afin de comparer les méthodes classiques aux approches quantiques potentiellement prometteuses. L'objectif est de vérifier si ces techniques de QML pourraient révolutionner l'analyse des données, tel que dans le contexte des missions de l'Agence de Recherche d'Exploration Spatiale (ARES).

## 2 Objectifs du projet

Ce projet vise à explorer et appliquer différentes techniques d'apprentissage machine quantique (QML), notamment sur l'ensemble 'HTRU2', décrits en détail dans la section 3.8.1. L'objectif principal du projet est d'implémenter des méthodes permettant de distinguer des données de signaux radio représentant des pulsars parmi d'autres signaux radio. Le projet vise ainsi à l'acquisition d'une meilleure compréhension de ces techniques de QML et de la comparaison des résultats obtenus par chacune d'entre et les résultats obtenus par les méthodes d'apprentissage machine classique afin de déterminer si celles-ci pourraient être pertinentes pour la résolution de problèmes auxquels pourrait faire face l'ARES. Enfin, le projet a également pour but de présenter clairement, par moyen de ce rapport, les résultats et conclusions obtenus par l'analyse de ces méthodes.

## 3 Théorie et méthodes

### 3.1 Notions de base en informatique quantique

Dans le cadre de notre projet, nous utilisons plusieurs concepts fondamentaux de l'informatique quantique. Ces concepts incluent entre autres les qubits, les circuits quantiques, et les portes quantiques.

Les qubits sont l'unité de base de l'informatique quantique sur lequel diverses opérations peuvent être effectuées, tout comme les bits classiques. Les qubits peuvent, par contre, exister dans des états de superposition, ce qui signifie qu'ils peuvent représenter simultanément 0 et 1, selon diverses proportions. Un ensemble de qubits sur lesquels des opérations sont effectuées forment un circuit quantique. Les portes quantiques permettent d'effectuer des transformations sur ces qubits, similairement aux portes logiques présentes en informatique classique.

Pour ceux qui ne sont pas familiers avec ces concepts, nous vous recommandons de consulter le livre de Schuld et al [4], qui fournit une introduction approfondie à la théorie et aux applications de l'apprentissage machine quantique.

### 3.2 Ansatz

Ansatz est un mot allemand qui désigne les premiers signes de quelque chose ou une approche menant potentiellement à la solution d'un problème [2]. Dans un contexte de physique ou de mathématique, un ansatz est une hypothèse initiale pouvant potentiellement mener à une solution. L'utilisation d'un bon ansatz pour résoudre un problème permet ainsi de réduire grandement la complexité d'un problème. En effet, un bon ansatz donne la forme que devrait avoir la solution, ce qui oriente la recherche de la solution. Un ansatz n'est pas une solution mais sert d'esquisse pouvant ensuite être raffinée et optimisée afin de répondre aux critères du problème. Dans le cadre de ce projet d'apprentissage machine quantique, un ansatz est donc une forme préétablie de circuit ou algorithme qui est une piste vers la solution [7]. Si l'ansatz est bon, on peut ensuite obtenir une solution au problème en optimisant les différents paramètres qui forment l'ansatz. Par exemple, dans la méthode du classificateur quantique variationnel (présentée plus tard en section 3.6), l'ansatz est un circuit qui contient plusieurs portes de rotation en  $y$  qui dépendent d'angles  $\theta$ . L'ansatz est donc optimisé en optimisant les angles de ces rotations  $R_y(\theta)$

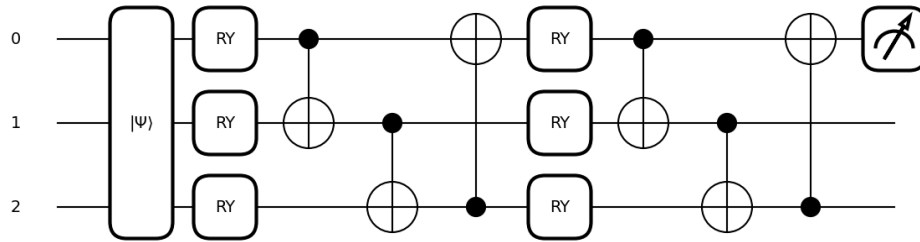


Figure 1: Circuit du ansatz pour la méthode du classificateur quantique variationnel (VQC) sur 3 qubits avec deux couches

### 3.3 Problèmes de classification binaire

La classification de données en utilisant l'apprentissage machine consiste à utiliser une banque de données déjà classifiée afin d'entraîner notre machine pour qu'elle puisse classer des données futures. Les données sont tous attachées à des étiquettes. Ces étiquettes peuvent être, par exemple, de valeur 0 ou 1, vrai ou faux, -1 et 1, et autres, en fonction du cas. Dans le contexte de ce projet, l'apprentissage machine quantique est utilisé afin d'assigner ces étiquettes suite à un apprentissage effectué sur de nombreux points d'entrées distincts ayant préalablement été étiquetés. Un scénario où on reçoit des pommes ou des oranges et nous désirons attribuer l'étiquette 'Pomme' sur les pommes et l'étiquette 'Orange' sur les oranges. L'algorithme aurait ici pour bût d'apprendre à distinguer les pommes et les oranges et d'être en mesure de donner la bonne étiquette à un ensemble de données pris en entrée.

### 3.4 Encodage Quantique De Données

Afin de traiter quantiquement les différents ensembles de données, on doit avoir une manière de représenter ces données sous la forme de circuits quantiques. Le processus d'encodage est donc une étape cruciale dans l'apprentissage machine quantique puisqu'il permet de traduire les données classiques en états quantiques.

Différentes approches d'encodage quantiques existent. Dans le cadre de ce projet de recherche, les approches qui ont été explorées sont l'encodage par amplitude, l'encodage de base et l'encodage par angle. Dans la discussion de ce rapport, ces différentes méthodes seront comparées afin de vérifier si elles sont efficaces, voir même pertinentes à implémenter. [12]

- **encodage par amplitude:**

L'encodage par amplitude, comme son nom l'indique, permet d'encoder des données. L'idée derrière est de représenter un vecteur classique en un état quantique normalisé en plaçant  $n$  qubits en une superposition des états  $|0\rangle$  et  $|1\rangle$ . Un vecteur de données de longueur  $N$  peut être encodé sur un registre de  $\log_2(N)$  qubits

- **L'encodage par base:**

L'encodage par base comporte 2 étapes: D'abord, on encode notre vecteur de données en un nombre binaire d'une longueur  $n$ . Ensuite, on initialise un circuit quantique de  $n$  qubits, avec des états  $|1\rangle$  ou  $|0\rangle$  permettant de représenter le nombre binaire. Par exemple, si on obtient 00101 comme nombre binaire représentant nos données, le registre quantique comportera 5 qubits et sera initialisé dans l'état  $|00101\rangle$ . Pour effectuer cet état à l'aide de portes quantiques, on applique des portes X à tous les qubits qui sont à une position où le vecteur de bit vaut 1.

- **L'encodage par angle:**

L'encodage par angle est une méthode d'encodage où chaque variable dans le vecteur de données classique est représentée par une rotation du qubit associé à cette valeur. Les rotations sont effectuées avec une normalisation qui convertit chaque donnée en un angle correspondant, typiquement entre 0 et  $2\pi$ . Ensuite, il suffit d'appliquer une porte de rotation selon l'axe désiré avec l'angle obtenu. Par exemple, pour encoder une donnée  $x_i$  avec une rotation en  $X$ , on obtient l'angle  $\theta_i$  à partir de la valeur de  $x_i$  et applique une porte  $R_x(\theta_i)$  sur le  $i$ ème qubit du registre.

### 3.5 Réseau Neuronal Convolutif Quantique (QCNN) :

Cette méthode, introduite en 2018 dans l'article [1] est basée sur l'approche des réseaux neuronaux convolutifs (CNN). La méthode CNN est une méthode en apprentissage machine classique qui est déjà considérée comme efficace et qui est largement utilisée dans des domaines tels que la vision ordonnée [10].

Le QCNN est une version quantique du CNN. Le QCNN prend un circuit dans lequel un vecteur de donnée est préalablement encodé et y ajoute plusieurs couches de convolution et couches de regroupement afin de ramener l'information encodée à une représentation plus compacte, tout en préservant les caractéristiques des données initiales. Les couches de convolution ont pour but d'extraire des corrélations entre les qubits du registre. Elle s'assurent d'encoder les relations quantiques par moyen, entre autres, de portes de rotations et portes CNOT afin de préparer le système pour la couche de regroupement qui suit. Chaque cycle de convolution et regroupement regroupe l'information sur la première moitié des qubits. Pour un registre de  $2^N$  qubits, il faudra donc  $n$  itérations afin de ramener complètement l'information sur le premier qubit. Pour un registre impair de qubits, on peut démontrer mathématiquement qu'en omettant un qubit lors de la phase de regroupement, on atteindra un registre pair pour finalement arriver à un seul qubit restant. Suite a ces  $n$  itérations, on effectue donc une mesure sur le premier qubit. La couche de convolution utilise un vecteur de poids pour déterminer les angles des rotations. On peut donc utiliser un optimiseur classique afin d'entraîner notre circuit et de trouver des poids optimaux avec lesquels on peut mieux classifier des données.

Les couches de regroupement encodent l'ensemble des données sur la moitié des qubits du registre. Pour se faire, différentes méthodes existent. Il est possible d'effectuer une mesure sur la moitié des qubits et utiliser les résultats obtenus pour modifier l'autre moitié des qubits. Pour l'implémentation dans ce projet, les couches de regroupement comportent des portes contrôle-X (CNOT) qui relient le premier et le dernier qubit, le deuxième et l'avant dernier qubit et ainsi de suite jusqu'au 2 qubits centraux (Voir figure 1). Pour un ensemble de données encodé sur un registre comportant un nombre de qubits n'étant pas une puissance de 2, il est possible d'adapter le processus itératif ou tout simplement ajouter des qubits supplémentaires au registre afin d'atteindre la puissance de 2 la plus près [11].

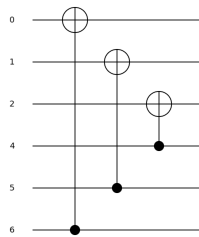


Figure 2: Circuit d'une couche de regroupement permettant de passer de 7 à 4 qubits (le 4ème qubit n'est pas affiché)

### 3.6 Classificateur quantique variationnel (VQC) :

La méthode basée sur un classificateur quantique variationnel (ou VQC) est une méthode qui représente une alternative quantique aux réseaux neuronaux classiques visant à effectuer des tâches de classification. Cette méthode diffère du QCNN en sa structure. En fait, pour faire un

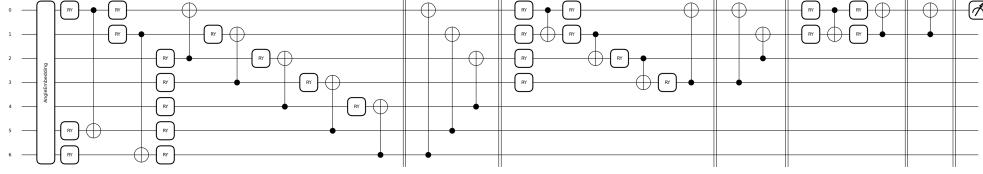


Figure 3: Circuit du QCNN avec portes CNOT sur un registre de 7 qubits

classificateur quantique variationnel, on a besoin en premier lieu d'encoder notre information tout comme le QSVM. On doit ensuite créer des couches d'ansatz selon le nombre voulu en utilisant des poids. On doit ensuite utiliser un optimiseur classique pour déterminer les poids optimaux que devrait utiliser l'ansatz de notre VQC afin qu'il classifie les données avec une bonne précision. Dans ce projet, nous avons testés deux ansatz. Il y a un ansatz composé de couches aléatoires de portes quantiques et un ansatz qui fait des rotations autour de l'axe des  $y$  en utilisant les poids et qui connecte ensuite les qubits à l'aide d'une structure de CNOT en escalier. [5]

### 3.7 Machine à vecteur de support quantique (QSVM) :

La machine à vecteur de support quantique (ou QSVM) est semblable à la machine à vecteur de support classique (SVM). La SVM est une approche qui tente de trouver un hyperplan qui va séparer les données, avec la meilleure exactitude possible, en fonction de l'étiquette associée à chacune des données. Pour des cas simples où un ensemble de donnée est linéairement séparable d'un second ensemble de donnée, cette méthode consisterait à trouver la droite qui sépare ces deux ensembles en maximisant la marge entre les deux classes. Le QSVM utilise la même approche, mais fait usage aux principes de l'informatique quantique en représentant les données dans des espaces d'ordres supérieurs sous la forme d'espaces de Hilbert.

Les données sont projetées dans des espaces d'ordres supérieurs par des noyaux quantiques (ou quantum kernels). L'utilisation de cette méthode quantique simplifie les manipulations nécessaires pour représenter des données dans des espaces de très hautes dimensions et facilite les opérations et les transformations effectuées sur les données. Effectuer des mesures sur les noyaux quantiques permet d'obtenir de l'information sur la similarité entre des données. En fait, on encode une donnée  $x$  de notre ensemble de données et on tente de faire l'inverse en utilisant une donnée  $y$  de notre ensemble de données. On peut alors mesurer l'état du système pour déterminer si les deux données semblent appartenir à la même classe. On peut alors ensuite créer un hyperplan de décision en utilisant des données d'entraînement. Notre algorithme est alors capable de classifier les données prises en entrée selon le côté de l'hyperplan où elles figurent.

#### 3.7.1 Matrices de corrélation

une matrice de corrélation est un tableau qui met en relation toutes les paires possibles de points de données afin de montrer le degré de corrélation entre les différentes variables. Chaque élément de la matrice représente la corrélation entre deux variables. Tous les éléments

sont comparées avec tout autre élément; les éléments sur la diagonale de la matrice doivent donc tous avoir une valeur de 1 puisque chaque variable est directement liée à elle même.

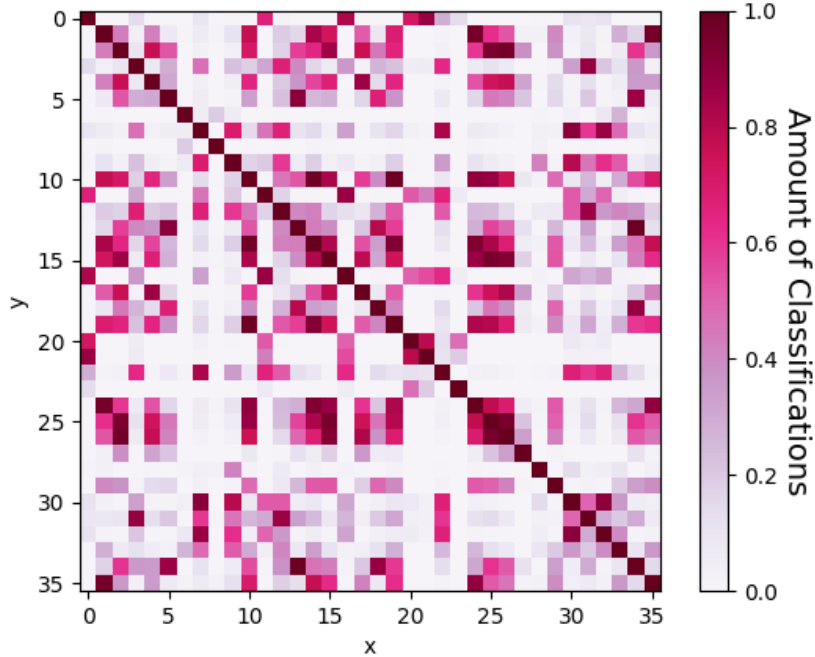


Figure 4: Exemple de matrice de corrélation qui compare 35 variables entre elles

### 3.8 Méthode Classique - Perception Multi-Couche (MLP) :

Pour évaluer l'efficacité des techniques d'apprentissage quantique, il est essentiel de disposer d'une référence classique. Nous avons donc opté d'utiliser un classificateur perceptron multicouche (MLPC) comme méthode classique de référence.

Le MLPC est un réseau neuronal artificiel qui peut traiter des tâches de classifications et modéliser des relations complexes au sein des données. Il se compose de plusieurs couches et chaque couche est entièrement connectée à la suivante. Les nœuds des couches sont des neurones à fonctions d'activation non linéaires, à l'exception des nœuds de la couche d'entrée. Entre la couche d'entrée et la couche de sortie, il peut y avoir une ou plusieurs couches cachées non linéaires.

Dans notre projet, nous avons utilisé l'implémentation de scikit-learn du MLPClassifier [8]. Nous avons configuré un réseau avec deux couches cachées, chacune contenant 200 neurones. L'entraînement a été réalisé avec un maximum de 1000 itérations, en utilisant l'optimiseur adam et la fonction d'activation ReLU. Un paramètre de régularisation alpha de 0.001 a aussi été appliqué.

### 3.8.1 Matrice de confusion

Une matrice de confusion est un tableau qui permet d'évaluer la performance d'un modèle de classification en comparant les prédictions du modèle avec les résultats réels. Dans le cas du modèle du MLPC, la matrice de confusion permet de visualiser les prédictions des classes, par exemple, des pulsars et des non-pulsars. Elle aide à identifier les erreurs de classification, en montrant combien de fois le modèle a confondu les deux catégories.

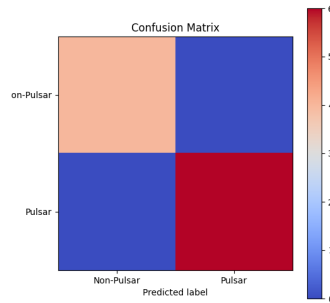


Figure 5: Exemple de matrice de confusion

## 3.9 Ensembles de données

### 3.9.1 HTRU2 : High Time Resolution Universe Survey

Cet ensemble de données, qui contient un total de 17898 points de données, est un bon choix pour tester notre implémentation. Chaque point comporte 8 variables en plus de l'étiquette qui dit si le point correspond, ou non, à un pulsar. Ces 8 variables correspondent à différents aspects des signaux radios captés, tels que `Profile_mean` et `DM_mean`. Les points de données de l'ensemble sont tous des nombres, ce qui facilite l'encodage dans notre cas puisque nous n'avons pas développer d'encodage de paramètres qui pourraient être autre choses que des nombres (par exemple une chaîne de caractère) [9].

### 3.9.2 Dry bean

Ce second ensemble de données est ajouté afin de valider que notre code puisse être utilisés pour diverses bases de données et non seulement sur les données du HTRU2. Cet ensemble de données correspond à des points de données numériques récupérées de 13611 images haute définition de haricots secs. un total de 16 variables sont utilisées pour représenter chaque haricot. Parmi ces 16 variables, on trouve par exemple l'excentricité de l'haricot, sa solidité et son périmètre. Tandis que le HTRU2 utilise des étiquettes d'une valeur de 1 ou de 0 pour la classe du point (soit un pulsar ou autre chose), l'ensemble de données Dry bean contient 7 classes distinctes, représentant chacune une espèce d'haricot différent. [3]



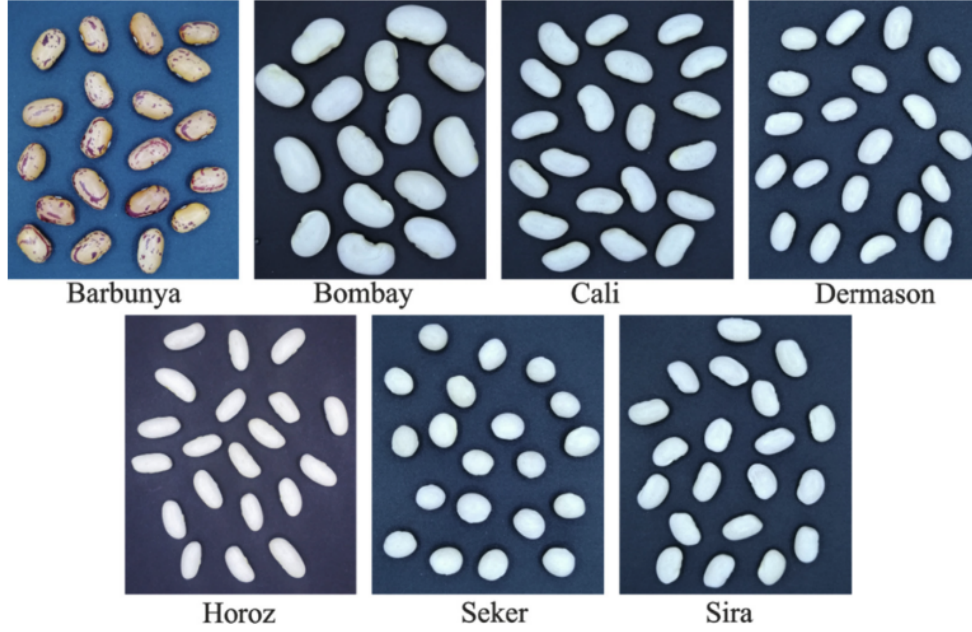


Figure 6: Exemples d'images utilisées pour obtenir les données dans l'ensemble 'Dry bean'  
Image provenant de <https://doi.org/10.1016/j.compag.2020.105507>

## 4 Résultats

En exécutant notre code avec les deux jeux de données, on réussit à extraire des résultats de précision. Nous avons exécuté le code avec des poids pareils pour les différents type d'encodage dans les tableaux et des poids semblables entre les différentes méthodes puisqu'ils devaient être ajustés selon la méthode. Ainsi, on obtient des résultats comparables pour toutes les méthodes selon le type d'encodage de données choisi. Les méthodes VQC et QCNN ont été exécutées sur 20 itérations et nous avons utilisé 20 données d'entraînement pour 50 données de test.

Nous avons choisi de ne pas prendre en compte le temps d'exécution puisque les méthodes quantique sont simulées sur un ordinateur classique. Ainsi, les temps d'exécution ne représentent pas réellement les temps d'exécution qui seraient obtenus avec un ordinateur quantique. D'ailleurs, il est clair qu'il n'y a pas, pour l'instant, d'avantage quantique si nous n'avons pas accès à un ordinateur quantique capable d'effectuer ces mêmes traitements de données puisque nos méthodes sont simplement des simulations classiques de méthodes quantiques.

Voici nos pourcentages de précisions en exécutant les diverses méthodes avec le jeu de données 'HTRU\_2.csv'. Nous classifions les objets comme étant des pulsars ou non:

HTRU_2.csv	QCNN	VQC (random layers)	VQC (Ry et CNOT layers)	QSVM	MLPC (pas d'encodage)
Encodage par angle (Rx)	0.60	0.70	0.88	0.92	0.90
Encodage par angle (Ry)	0.94	0.94	0.90	0.85	0.90
Encodage par angle (Rz)	0.50	0.50	0.50	0.38	0.90
Encodage par amplitude	0.94	0.40	0.90	0.70	0.90

Voici nos pourcentages de précisions en exécutant les diverses méthodes avec le jeu de données 'Dry\_Bean\_Dataset.xlsx'. Nous classifions deux espèces d'haricots: les Dermason et les Sira:

Dry_Bean_Dataset.xlsx	QCNN	VQC (random layers)	VQC (Ry et CNOT layers)	QSVM	MLPC (pas d'encodage)
Encodage par angle (Rx)	0.94	0.96	0.88	0.92	0.90
Encodage par angle (Ry)	0.92	0.88	0.88	0.85	0.90
Encodage par angle (Rz)	0.50	0.50	0.50	0.46	0.90
Encodage par amplitude	0.68	0.52	0.84	0.46	0.90

## 5 Discussion

En analysant nos résultats, on remarque que l'encodage par angle suivant l'axe des z n'est pas un encodage utile puisqu'il mène à des prédictions qui ont une précision de 50%, peu importe la méthode testée. En fait, en analysant plus profondément, on se rend compte que les méthodes prédisent uniquement des étiquettes de -1 pour chaque test ou une étiquette de 1 pour chaque test. En utilisant cet encodage, on obtient alors une précision équivalente à celle d'un algorithme de classification qui met toutes les données dans la même case. Similairement, l'encodage par base n'est pas pertinent pour un projet comme celui-ci. Encoder les vecteurs de données selon une conversion en binaire des valeurs nécessiterait une trop grande quantité de qubits. Par exemple, encoder par base un vecteur de donnée correspondant à un point arbitraire de l'ensemble de données 'HTRU\_2' nécessiterait plus de 80 qubits. D'autres approches à l'encodage binaire peuvent être tentées, comme en utilisant une normalisation pour ramener les valeurs entre 0 et 1 et en utilisant différents seuils pour représenter la valeur de chaque variable sur un plus petit nombre de qubits. Cependant, de telles représentations ne sont pas fidèles aux caractéristiques des données initiales et nos tentatives d'un tel encodage n'a pas mené à des résultats significatifs.

Ensuite, on peut voir que l'encodage par angle suivant l'axe des x et l'encodage par angle suivant l'axe des y mènent à des résultats semblables pour chaque méthode testée. Ces résultats sont ceux qui sont le plus concluants face aux résultats classiques. Pour ce qui en est de l'encodage par amplitude, on voit que celui-ci n'est pas préférable dans le cas du classificateur variationnel quantique utilisant l'ansatz 'random.layers' et dans le cas de la machine à vecteur de support quantique. Cependant, cette méthode d'encodage semble fonctionner pour le réseau neuronal convolutif quantique avec le jeu de données 'HTRU\_2.csv'. L'utilisation des ensembles de données HTRU2 et Dry\_bean a pu permettre de vérifier que nos implémentations fonctionnent sur des vecteurs de données comportant différents nombres de valeurs (8 pour HTRU2 et 16 pour Dry\_bean). Enfin, en comparant les méthodes quantiques à la méthode classique que nous avons utilisé, nous remarquons que les méthodes quantiques ne présentent pas d'avantages dans la plupart des cas. On peut ainsi se questionner si y aurait réellement un avantage quantique dans la classification par apprentissage machine. Pour vraiment déterminer la meilleur méthode, on pourrait analyser d'autres facteurs de réussite de la classification comme le temps d'exécution, mais pour cela, il faudra que l'ordinateur quantique fonctionne afin que nous n'ayons plus à simuler les méthodes quantiques.

## 6 Conclusion

Ainsi, différentes approches à l'apprentissage machine quantique ont pu être comparées. Différentes méthodes d'encodage de données, les réseaux neuronaux convolutifs quantiques (QCNN), les classificateurs variationnels (VQC), ainsi que les machines à vecteurs de support quantiques (QSVM) ont été appliquées sur deux ensembles de données afin de vérifier leur validité et de comparer les résultats obtenus pour chacune des méthodes avec la méthode classique. Ces comparaisons et résultats montrent que les différentes méthodes quantiques ont un potentiel selon les données et l'encodage utilisée. Cependant, pour des problèmes de classification, une méthode classique demeure avantageuse pour sa simplicité et sa précision généralement plus fiable que la précision obtenue par les méthodes quantiques. Il est encore trop tôt pour avancer que l'apprentissage machine quantique est plus efficace que l'apprentissage machine classique, puisque certains paramètres comme le temps d'exécution ne peuvent pas encore être calculés dans la version quantique.

## 7 Bibliographie

### References

- [1] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, August 2019.
- [2] Cambridge dictionary. ansatz definition, n.d. consulté le: 2024-10-15.
- [3] M. Koklu et Ilker Ali Özkan. Dry bean, 2020.
- [4] Maria Schuld et Nathan Killoran. *Machine Learning with Quantum Computers*. Springer Nature Switzerland AG, 2nd edition, 2021. Originally published with the title: Supervised Learning with Quantum Computers.
- [5] Elies M. Gil Fuster. Variational quantum classifier, 2019. Consulté le: 2024-10-12.
- [6] Joey Gonzalez. How machine learning became useful, 2022. Consulté le: 2024-10-16.
- [7] Xiaoyu Guo, Takahiro Muta, and Jianjun Zhao. Quantum circuit ansatz: Patterns of abstraction and reuse of quantum algorithm design. *arXiv arXiv:2405.05021v3*, 2024.
- [8] Scikit learn. mlp classifier documentation. Consulté le: 2024-10-12.
- [9] Robert Lyon. Htru2, 2015.
- [10] D. Slabbert, M. Lourens, and F. Petruccione. Pulsar classification: comparing quantum convolutional neural networks and quantum support vector machines. *Quantum Machine Intelligence*, 6:56, 2024.
- [11] Xanadu. Quantum convolutional neural networks, n.d. Consulté le: 2024-10-12.
- [12] Xanadu. tutorial kernel based training, n.d. Consulté le: 2024-10-12.