

BSQ201 - Projet 1

Apprentissage machine quantique



Karlheinz Fedorlensky Forvil

Ugo Massé

Louis-Félix Vigneux

18 octobre 2024

Table des matières

1	Introduction	2
2	Théorie	2
2.1	Apprentissage supervisé - classification	2
2.2	ARES et l'apprentissage machine quantique	3
2.2.1	Pulsars	3
2.3	Encodage de données	3
2.4	Apprentissage machine	4
2.4.1	Méthodes à noyaux classique et quantique	4
2.4.2	Classificateur variationnel quantique	5
2.4.3	Réseau de neurones convolutif	6
2.5	Implémentation classique	8
2.6	Implémentation quantique	8
3	Résultats	8
3.1	Méthodologie	8
3.2	Métriques de comparaison	9
4	Discussion	9
4.1	Comparaison des méthodes	9
4.1.1	Méthodes à noyau classique et quantique	9
4.1.2	Méthodes variationnelles	10
4.1.3	Meilleure méthode	10
4.2	Classique vs quantique	10
4.3	Impact sur les opérations de l'ARES	11
5	Conclusion	12
6	Références	12
7	Annexe	15
7.1	Tableaux des résultats	15

1 Introduction

La classification des pulsars, des étoiles à neutrons fortement magnétisés et en rotation rapide, représente un défi scientifique majeur dans l'astrophysique moderne. Leur détection à partir de signaux radio nécessite des algorithmes de classification robustes et efficaces. Traditionnellement, cette tâche est accomplie à l'aide d'algorithmes classiques d'apprentissage machine automatique.

Avec l'avènement de l'informatique quantique, une nouvelle classe d'algorithmes d'apprentissage automatique, appelée « quantum machine learning » (QML), a émergé. Ceux-ci exploitent les principes de la mécanique quantique, tels que la superposition et l'intrication, pour fonctionner. Bien que les ordinateurs quantiques soient encore en phase de développement, des simulateurs et des algorithmes hybrides, combinant composants classiques et quantiques, offrent aujourd'hui un environnement permettant d'explorer leur potentiel.

Ce rapport vise à comparer les performances des algorithmes classiques et quantiques dans la tâche spécifique de classification de pulsars et dans la détection d'ondes gamma de haute énergie dans un télescope. Tous les algorithmes, qu'ils soient classiques ou quantiques, seront exécutés sur des ordinateurs classiques via des simulateurs. Cette étude se concentrera sur l'analyse de la précision de classification de méthodes quantiques par rapport à leurs rivaux classiques.

2 Théorie

L'apprentissage machine se focalise sur le développement de modèles statistiques dans le but de faire des prédictions. En autres mots, un système informatique d'apprentissage machine évalue un système et ses tendances sur un système données antérieures et est ensuite, idéalement, en mesure de prédire leur évolution.[1]

Plusieurs subdivisions de l'apprentissage automatique existent. Dans le cadre de ce projet, nous allons nous concentrer sur un classificateur à apprentissage supervisé.

2.1 Apprentissage supervisé - classification

Dans un scénario d'apprentissage supervisé, une structure est entraînée à reconnaître des sous-ensembles étiquetés parmi un ensemble de données. Celle-ci apprend les interactions entre les caractéristiques qui définissent les étiquettes. Une fois entraînée, la structure peut être utilisée pour trier des données non-étiquetées[2]. L'apprentissage supervisé répond à deux types de problèmes lors de l'exploration de données, la classification et la régression. Nous nous arrêterons uniquement sur la classification.

Plus précisément, on dispose d'un domaine d'entrée X et d'un domaine de sortie Y , ainsi que d'un ensemble de données contenant des paires de valeurs (\vec{x}, y) , où \vec{x} représente les entrées(caractéristiques), qui est un vecteur de dimension (n) appelé vecteur de caractéristiques. y , quant à lui, représente l'étiquette associée à l'entrée.

Dans les prochaines sections, nous présenterons les principales caractéristiques de certaines méthodes de classification quantique en faisant ressortir leurs particularités vis-à-vis de leurs homologues classiques.

2.2 ARES et l'apprentissage machine quantique

Dans le domaine de l'astronomie, une grande quantité de données sur différents types d'objets astronomiques est recueillie chaque année. Il est alors important de pouvoir classer ces objets célestes. Les logiciels traditionnels pourront difficilement classer tous ces objets célestes. L'analyse des données en temps réel et une classification performante sont des objectifs clés en astronomie et en science des données. Bien que les méthodes de classification classiques soient suffisantes pour l'instant, elles deviendront inadéquates à mesure que la quantité de données augmentera.

C'est pourquoi que l'Agence de Recherche d'Exploration Spatiale (ARES) nous a engagés pour explorer l'utilisation de l'information quantique pour classer des objets célestes plus efficacement. Différents classificateurs quantiques seront donc implémentés pour trouver la méthode la plus appropriée pour l'ARES. Pour tester la performance de ces nouvelles technologies en comparant aux classificateurs classiques déjà existants, une classification de pulsars sera effectuée.

2.2.1 Pulsars

Les pulsars sont un type rare d'étoile à neutrons qui émet des ondes radio détectables depuis la Terre. Ils sont d'un grand intérêt scientifique, servant à explorer l'espace-temps, le milieu interstellaire et différents états de la matière. En tournant, le faisceau d'émission des pulsars balaie le ciel et produit un motif de signaux radio récurrents. Comme les pulsars tournent rapidement, ce motif se répète périodiquement, ce qui permet de les détecter avec de grands radiotélescopes[3].

Chaque pulsar émet un motif légèrement différent, qui peut varier à chaque rotation. Par conséquent, un signal potentiel, appelé "candidat", est une moyenne des émissions sur plusieurs rotations, en fonction de la durée de l'observation. Cependant, en pratique, la plupart des détections sont causées par des interférences, radiofréquences et du bruit, rendant difficile l'identification de signaux authentiques de pulsars[3].

Pour ce projet nous allons classer des Pulsars en utilisant des données de la base HTRU_2 décrivant un échantillon de candidats pulsars, collectés lors de l'enquête universelle à haute résolution temporelle. Ce jeu de données contient 8 variables continues que nous appellerons caractéristiques. Des détails supplémentaires se trouvent ici[3].

2.3 Encodage de données

Les algorithmes utilisés dans l'apprentissage automatique quantique sont des algorithmes hybrides. Puisque les données que nous voulons classer sont des données classiques, nous devons donc être capables de les adapter pour qu'elles soient utilisables par une architecture d'ordinateur quantique. Pour le calcul quantique, l'encodage classique des données, qui consiste à charger les données dans l'état des qubits (unités de base de l'information quantique), est une étape cruciale pour la conception globale et l'efficacité des algorithmes. Il existe différentes options pour ce faire.

Encodage par amplitude

Cet encodage utilise les amplitudes $(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ des états quantiques pour représenter les données classiques. Les données classiques sont alors encodées sur un état quantique normalisé. Cela permet d'encoder un grand volume de données en utilisant un nombre logarithmique de qubits par rapport à la taille des données.

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle \quad (1)$$

TABLE 1 – $|\psi\rangle$ représente un état (un vecteur) qui est une combinaison d'états de base. En rouge les α_i pour l'encodage par amplitude.

Encodage par angle

Dans cet encodage, nous représentons un qubit par une sphère de Bloch, et les données classiques peuvent être encodées dans les angles de rotation des qubits autour de certains axes (comme X, Y ou Z). Donc, les caractéristiques sont directement utilisées comme angle de rotation de chacun des qubits.

Encodage IQP

Cet encodage est déjà implémenté dans la librairie de PennyLane [4]. Comme mentionné sur cette page, l'implémentation est basée sur un circuit décrit dans [5]. Cet encodage nécessite le même nombre de qubits que de caractéristiques.

Chacun de ces encodages a ses propres avantages et inconvénients en fonction du problème à résoudre et des ressources quantiques disponibles. Il existe d'autres types d'encodage comme l'encodage par évolution temporelle, par hamiltonien etc. Des détails sur les techniques d'encodage sont donnés ici [6].

2.4 Apprentissage machine

Trois méthodes d'apprentissage machine sont présentées. Elles seront ensuite testées pour déterminer celle qui sera préférable pour classifier des étoiles, plus précisément pour le jeu de données *HTRU_2*.

2.4.1 Méthodes à noyaux classique et quantique

Ce classificateur a pour but de séparer linéairement les données. Par contre, ce n'est pas toujours une tâche facile.

De ce fait, une augmentation de l'espace de représentation est de mise. Au lieu de tout représenter, les données d'entraînement dans ce nouvel espace (très coûteux en nombre d'opérations machine), le truc du noyau sera utilisé. En effet, seulement une mesure de similarité dans ce nouvel espace sera effectuée sur les données.[7] Pour un classificateur classique, cette mesure de comparaison se fera à l'aide de fonctions de noyaux bien définies. Pour en voir quelques exemples, voir cet article : [8]. Avec cette fonction, une machine à vecteur de support pourra créer la séparation des données et prédire les étiquettes.

Pour ce qui est des noyaux quantiques, la même logique d'augmentation de l'espace de représentation est appliquée. Par contre, en encodant les données pour être utilisables sur un ordinateur ou un simulateur quantique, l'espace des représentations des données est directement augmenté. De plus, pour évaluer la similarité des données dans l'espace des états quantiques, nous avons

simplement besoin de réappliquer l’encodage pour une autre donnée. En effet, sachant que les portes utilisées dans un circuit quantique sont hermitiques, si deux données identiques sont encodées, leurs circuits seront identiques et s’annuleront. Ainsi, la mesure de similarité sera tout simplement la probabilité de mesurer la chaîne de bits de 0.

2.4.2 Classificateur variationnel quantique

Les algorithmes quantiques variationnels combinent des éléments classiques et quantiques : un circuit quantique déterminé par certains paramètres et un algorithme classique, qui met à jour ces paramètres. Ces circuits fonctionnent dans une boucle. En effet, le circuit quantique produit initialement des résultats. Ensuite, l’algorithme classique tente de se rapprocher d’un objectif sur ces résultats en réglant les paramètres du circuit. Puis, le processus recommence, cette fois avec les nouveaux paramètres, et cela est répété jusqu’à ce qu’une borne d’arrêt soit atteinte[9]. Un classificateur variationnel quantique (Variational Quantum Classifier, VQC) est donc une variation de ces algorithmes.[10]

Après avoir procédé à l’encodage des données, le classificateur variationnel quantique les envoie au circuit quantique variationnel. Ce circuit est composé d’une série de portes quantiques paramétrées par un ensemble de variables classiques. Plus précisément, les portes quantiques paramétrées sont des blocs effectuant des opérations logiques sur un ou sur plusieurs qubits qui dépendent d’un ou plusieurs paramètres réglables. Dans ce cas-ci ces portes font des rotations soient selon l’axe des x, y ou z sur la sphère de Bloch. C’est la valeur du paramètre qui donnera l’amplitude de cette rotation. [11]

Ces portes sont choisies de manière à transformer l’état quantique de manière non-linéaire, une propriété clé pour permettre à un modèle d’apprendre des relations complexes entre les données.[10] Les paramètres des portes sont ajustés lors du processus d’entraînement du classificateur pour minimiser une fonction de coût.

La fonction de coût fournit une mesure de la performance d’un modèle en calculant l’erreur entre la sortie prédite et la sortie attendue via un optimiseur. [12]

Il existe une multitude de circuits variationnels (mieux connu sous le nom d’ansatz)[13, 14, 15, 16, 17]. Cependant, ils ne s’équivalent pas tous et il est difficile de sélectionner le circuit le plus approprié au problème. Un bon circuit variationnel doit en partie, pouvoir construire n’importe quel état quantique arbitraire et pouvoir utiliser un fort phénomène d’intrication entre les qubits.[18, 19, 20]

L’un de ces circuits variationnels génériques est le *two-local*. [21] On peut ensuite l’affiner pour qu’il convienne mieux à notre type de problématique ou d’ensemble de données. Par exemple, dans ce rapport, nous avons utilisé une variante déjà existante du *two-local* [13] représenté à la figure 1. Dans les résultats, ce circuit sera nommé «sur mesure».

Il est également possible de faire des circuits variationnels quasiment aléatoirement. En effet, il suffit de séparer le circuit variationnel en couches de paramètres et d’intrications aléatoires. [22] Dans les résultats, ce circuit sera nommé «*Random_layers*».

Après que l’information ait traversé les couches du circuit variationnel, les états des qubits sont mesurés. La valeur moyenne de l’observable Z sur chacun des qubits est évaluée pour l’état sortant du circuit quantique variationnel.[10]

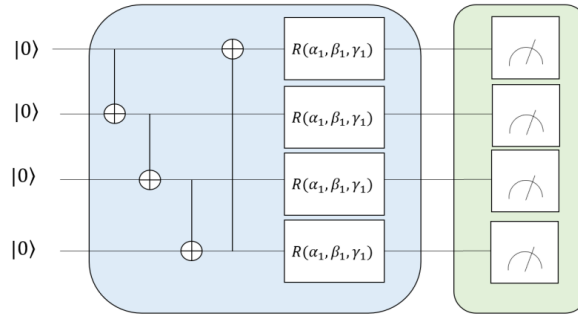


FIGURE 1 – Variante du circuit variationnel two-local utilisé lors de la classification variationnelle quantique. Tiré de l'article[13].

2.4.3 Réseau de neurones convolutif

Un réseau de neurones convolutif (CNN) est un type de réseau de neurones artificiels conçu principalement pour traiter des données structurées en grille. Il est particulièrement efficace dans les tâches de vision par ordinateur, tel que la classification, la détection d'objets ou la segmentation d'images.[1]

Le CNN repose sur plusieurs couches spécifiques qui permettent d'extraire et de traiter des caractéristiques locales de données structurées de manière hiérarchique. Voici les couches typiquement utilisées :[1, 23]

1) **Couches convolutives** : Les couches convolutives appliquent un ensemble de filtres appelés noyaux à la donnée d'entrée pour capturer des motifs locaux. Cette étape met donc en valeur des interactions entre les données d'entrées dans le but de distinguer plus facilement leur étiquette.

2) **Couches de regroupement (pooling)** : Le but du regroupement est de réduire la dimension des cartes de caractéristiques tout en conservant les informations les plus importantes.

3) **Couches de correction** : Après chaque couche convolutive, une fonction d'activation est appliquée pour introduire de la non-linéarité. Ainsi, ces couches transforment les signaux entrants d'un neurone pour déterminer s'il doit être activé ou non (s'il doit transmettre des informations aux neurones suivants).

4) **Couches complètement connectées** : Ces couches ressemblent aux réseaux de neurones traditionnels où chaque neurone est connecté à tous ceux de la couche précédente. Elles sont souvent utilisées en fin de réseau pour effectuer la classification finale.

Une donnée passe successivement à travers les couches convolutives, de regroupement et d'activation, extrayant des caractéristiques de plus en plus abstraites à chaque étape. À la fin, les couches complètement connectées et une fonction qui transforment les résultats en probabilités sont utilisées pour prédire la classe de la donnée.

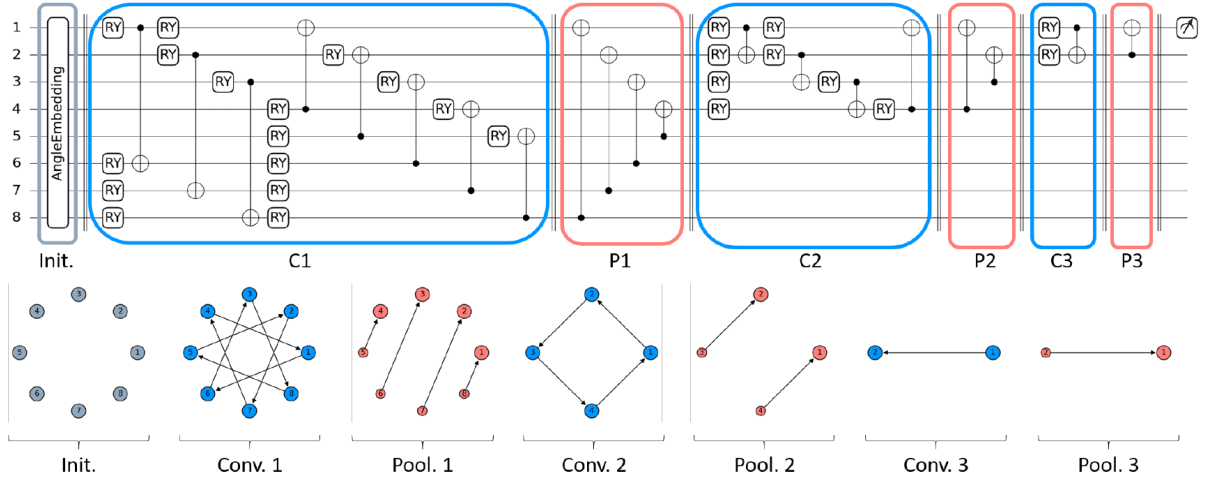


FIGURE 2 – Circuit pour le QCNN avec 8 qubits, tiré de l'article [25] : encodage par angle, circuits de convolution en bleu, circuits de regroupement en rouge. En dessous : des graphes représentateurs pour illustrer le circuit en entier.

Pour que les noyaux bon fonctionnement du noyau, il faut entraîner le réseau convolutif. En effet, pendant l'entraînement du réseau, la rétropropagation est utilisée pour ajuster les poids des filtres et des connexions dans les couches complètement connectées. L'erreur entre la sortie prédite et la sortie attendue est calculée via une fonction de coût. Cette erreur est ensuite rétropropagée dans le réseau pour ajuster les poids en utilisant un optimiseur.[24]

Réseau de neurones convolutif quantique

Plusieurs propositions ont été faites concernant l'implémentation du réseau de neurones convolutif quantique (QCNN). Nous implémenterons celui qui se trouve dans le papier de Slabbert [25]. Tout comme le réseau de neurones convolutif (classique), le QCNN implémente, de façon analogue, les couches convolutives ainsi que les couches de regroupement dans un cadre de circuit quantique.

Dans cette implémentation, une convolution consiste en des opérations unitaires appliquées à tous les qubits qui sont disponibles dans une couche donnée. Les opérations unitaires choisies sont arbitraires, mais identiques afin de favoriser le partage de poids ainsi que de conserver la capacité d'apprentissage des modèles à travers les données.

Une couche de regroupement consiste à utiliser une partie des qubits disponibles comme qubits de contrôles en appliquant des rotations contrôlées sur l'autre partie. Cela conduit à une réduction de la taille du système et permet une évolution logarithmique du nombre de paramètres. En réduisant la taille du système, l'information des qubits de contrôle sera transférée aux qubits cibles.

On répète ces opérations de convolution et de regroupement jusqu'à ce qu'il reste qu'un seul qubit impliqué. Nous pourrions ensuite le mesurer.

2.5 Implémentation classique

Deux implémentations de classificateurs classiques ont été utilisées, soit un réseau de neurones convolutif et un algorithme utilisant une machine à vecteurs de support, SVM.

2.6 Implémentation quantique

Trois encodages de données sont proposés (par angle, amplitude ou iqp).

Le kernel utilise le SVM linéaire de `sk.learn` pour la classification des données à partir de la matrice de similitude.

Pour le QCNN, le circuit utilisé est celui de [25].

Pour le VQC, un ansatz a été créé à partir d'essais-erreurs et un autre à l'aide des «`random_layers`» de `pennylane`.

Pour les deux méthodes variationnelles, deux optimiseurs différents seront testés : COBYLA de `scipy` et un optimiseur nommé Nesterov de la librairie `optimize` de `pennylane`. Leurs impacts seront discutés plus tard.

La fonction de coût pour ces deux méthodes est laissée au choix de l'utilisateur. Pour les tests, l'erreur quadratique moyenne sera utilisée.

$$MSE = \frac{\sum_i^n (prediction_i - valeur_attendue_i)^2}{2n}$$

où n est le nombre de données prédites.

Les deux méthodes variationnelles utilisent toutes les données d'entraînement à chaque itération de l'optimiseur.

3 Résultats

3.1 Méthodologie

Pour les tests, nous allons utiliser le jeu de données *HTRU_2* défini dans la théorie. De plus, nous allons aussi tester les données sur un autre jeu de données astronomique nommé «MAGIC Gamma Telescope» [26]. Dans ce cas-ci, c'est la classification de particule gamma de haute énergie. Pour plus d'information sur la création de ce jeu de données de 10 caractéristiques, voir cet article : [27]. Nous allons nommer ce jeu de données *Télescope*.

200 données prises aléatoirement dans les jeux de données seront choisies. Ces partitions comportent autant de vecteurs de caractéristique d'une étiquette que de l'autre. Ces vecteurs choisis sont ensuite arrangés dans un ordre aléatoire. Un nouveau choix de partition se fait à chaque test. 160 données serviront pour entraîner les classificateurs et les 40 dernières seront prédites.

Les encodages d'angles se font sur un nombre de qubits identique au nombre des caractéristiques du jeu de données. Ils seront décrits par leur rotation selon l'axe des x , y ou z .

L'optimiseur sans gradient COBYLA est utilisé. En effet, il est souvent utilisé dans des problèmes d'optimisation quantique. Un optimiseur à gradient Nesterov sera aussi employé. Ces deux optimiseurs exécuteront 100 itérations au maximum.

Bien qu’elles peuvent en faire beaucoup plus, les algorithmes d’apprentissage automatique classiques sont limités au même nombre de données entrantes ainsi qu’au même ratio de données qui sont utilisées pour l’entraînement que leur équivalent quantique.

3.2 Métriques de comparaison

Exactitude : Le rapport entre les étiquettes bien prédites sur le nombre total d’étiquettes à prédire.

Les autres tests utiliseront la notion de vrai ou faux positifs ou négatifs. En prenant l’exemple du jeu de données de pulsar, un vrai positif est défini lorsque le classificateur prédit correctement que l’étoile analysée est un pulsar. Un faux positif, quant à lui, est lorsque l’algorithme affirme qu’une étoile est un pulsar mais, qu’en réalité, elle ne l’est pas. La même logique permet de déduire le sens des vrais et faux négatifs. Nous attribuerons les noms suivants à ces définitions :

- VP : Vrai positif
- FP : Faux positif
- VN : Vrai négatif
- FN : Faux négatif

Précision : Cette métrique consiste à évaluer la fréquence de vrai positif que l’on obtient. Autrement dit, si le classificateur affirme un résultat positif, la précision donne la fréquence que cette prédiction soit correcte.

$$Precision = \frac{VP}{VP + FP}$$

Précision négative : C’est le même principe que la précision sauf, en utilisant les vrais et faux négatifs.

$$Precision_negative = \frac{VN}{VN + FN}$$

Rétention de l’information Cette métrique permet de mesurer à quel point le modèle prédit bien les étiquettes. Les détails sur ce calcul sont disponibles sur cet article[3].

Les tableaux des résultats seront en annexe. La valeur div0 indique que le résultat n’est pas disponible puisqu’une division par 0 a été effectuée.

4 Discussion

4.1 Comparaison des méthodes

4.1.1 Méthodes à noyau classique et quantique

Nous remarquons qu’avec les figures 3 et 4 que la méthode à noyau classique a de meilleures performances. En effet, à l’exception de la précision que la méthode quantique a un léger avantage d’environ 0,03 en moyenne, la moyenne des métriques donne l’avantage au noyau classique. Il est intéressant de noter que la méthode d’encodage par amplitude donne de bien meilleurs résultats sur toute la ligne. Même si la méthode classique demeure meilleure en général, l’encodage par

amplitude d'un noyau quantique donne de très bons résultats pour le jeu de données de pulsar avec une précision de 1.

4.1.2 Méthodes variationnelles

Les figures 5, 6 et 7 nous indiquent aussi clairement qu'en moyenne, la méthode classique donne des meilleurs résultats, et ce, même si elle était limitée. L'algorithme classique fonctionne aussi plus efficacement avec l'optimiseur à gradient Nesterov.

Pour ce qui est du classificateur à circuit variationnel quantique, nous remarquons que l'ansatz sur mesure décrit dans la théorie a de meilleurs résultats pour la classification des pulsars que celui composé des *Random layers* de PennyLane. Cette méthode donne des meilleurs résultats sur la classification de pulsar et avec un encodage par amplitude.

Pour le classificateur à réseau de neurones convolutif quantique, il est possible de remarquer qu'il a beaucoup de division par 0. Cela indique qu'il n'est pas en mesure de bien séparer les données (les étiquettes de sorties sont tous les mêmes). Il ne fonctionne donc pas avec tous les encodages. En effet, c'est celui par amplitude qui donne de meilleurs résultats avec cette méthode. Pour cet encodage, les résultats entre les deux jeux de données sont semblables. L'utilisation des deux optimiseurs donne des résultats assez semblables pour l'encodage par amplitude.

4.1.3 Meilleure méthode

Nous avons conclu que dans les deux sous-sections précédentes que les méthodes classiques étaient plus performantes en moyennes sur toutes les métriques évaluées sur les deux ensembles de données. Ils sont plus généralisables à des ensembles de données différents.

De plus, sachant que pour la classification de pulsar, il est préférable de réduire le nombre de faux positifs [3], la méthode de réseau neuronal classique serait préférable. En moyenne, elle est plus exacte, détecte mieux les résultats positifs et a un meilleur ratio de rétention de l'information.

Pour ce qui est des méthodes quantiques, l'encodage par amplitude reste le meilleur pour les quatre métriques évaluées. Pour le jeu de données des pulsars, c'est le QCNN qui a donné la meilleure performance des trois méthodes avec COBYLA comme optimiseur. Par contre, la méthode à noyau avec l'encodage par amplitude a donné la même exactitude et précision tout en ayant une précision négative un peu meilleure. En revanche, la rétention de l'information est meilleure pour le classificateur à réseau de neurones. Ces deux méthodes offriraient une exactitude et précision des résultats semblables aux méthodes classiques pour l'ensemble de données de pulsar.

Lorsque les données des particules gamma sont utilisées, les performances des algorithmes quantiques diminuent. Par contre, le circuit variationnel donne des résultats assez constants pour les jeux de données différents. Par contre, le QCNN donne tout de même de meilleurs résultats.

Alors, si une méthode quantique devait être choisie, nous suggérons à l'ARES d'utiliser la méthode de réseau de neurones convolutifs quantique avec un encodage par amplitude et l'utilisation de l'optimiseur Nesterov qui est plus rapide et donne des résultats un peu plus précis.

4.2 Classique vs quantique

Les algorithmes classiques d'apprentissage automatique sont bien établis et offrent une efficacité robuste pour une grande variété de tâches de classification. Leurs capacités à gérer des

ensembles de données volumineux, couplées à des techniques d’optimisation, font des algorithmes classiques des outils très polyvalents.

Cependant, les méthodes classiques souffrent souvent de limitations liées à la complexité algorithmique. À mesure que la taille des données et la dimensionnalité augmentent, les coûts computationnels peuvent devenir prohibitifs, ce qui nécessite l’utilisation de ressources importantes en termes de puissance de calcul et de temps.

Dans quelques cas particuliers, les algorithmes quantiques ont pour objectif de surmonter ces limitations en exploitant la superposition et l’intrication quantiques pour résoudre certains problèmes de manière plus efficace. [28]

Pour la classification des pulsars, les approches classiques ont montré de très bonnes performances, atteignant des niveaux élevés de précision en moyenne pour les deux jeux de données. Cela peut s’expliquer par la nature relativement bien structurée des données du pulsar, qui sont composées de caractéristiques permettant une séparation claire des classes.

En revanche, les algorithmes quantiques appliqués à des jeux de données réels sont encore limités par l’encodage des données. En effet, certains algorithmes d’apprentissage automatique quantiques devraient être plus rapides que leur équivalent classique. Or, cette affirmation ne tient pas en prenant en compte les méthodes d’encodage des données actuelles. [29].

De plus, bien que des résultats prometteurs aient été obtenus pour quelques méthodes quantiques particulières sur des ensembles de données de petite taille, l’utilisation de ces classificateurs reste trop lente en simulation. En outre, la petite taille des ensembles de données que les algorithmes quantiques peuvent prendre en entraînement fait en sorte que ces algorithmes s’entraînent moins et, ainsi, ont de moins bonnes performances.

Étant beaucoup plus rapides que les méthodes quantiques simulées, les méthodes classiques ont naturellement plus de temps pour prendre davantage de données pour s’entraîner et, ainsi, avoir de meilleurs résultats. De plus, les résultats obtenus indiquent que les méthodes classiques étaient meilleures en moyenne même en étant limitées dans leur performance (voir la méthodologie).

La maturité technologique de l’apprentissage automatique quantique n’est pas assez avancée pour faire quelconque comparaison sur le temps et la précision lorsque nous laissons les algorithmes classiques utiliser leur plein potentiel (par exemple, utiliser plus de neurones).

4.3 Impact sur les opérations de l’ARES

Les différents tests effectués montrent que certaines méthodes quantiques ont des performances similaires en termes de précision comparées aux méthodes classiques. À noter que nous n’avons effectué nos tests que sur une partie (200 candidats) de nos bases de données et le temps particulièrement élevé que les méthodes quantiques aient eu besoin pour classer les données. Sachant que les données astronomiques sont de plus en plus volumineuses, l’un des objectifs de ces algorithmes d’apprentissage quantique est de fournir des outils assez puissants pour pouvoir classer ce type de données. On s’attendrait donc à un certain avantage. Néanmoins, nous n’avons roulé nos algorithmes qu’à partir de simulateurs sur des ordinateurs classiques, ce qui explique en partie le temps très long pris pour classer les données.

Le fait que les tests aient montré que ce type d’algorithmes peuvent être quasiment aussi précis que leur homologue classique nous pousse à envisager des pistes de solutions pour approfondir les méthodes prometteuses. Nous avons mentionné plus haut des problèmes concernant les méthodes d’encodages, par conséquent, une deuxième phase à ce projet pourrait être d’explorer d’autres méthodes afin d’améliorer la précision de nos modèles. Exécuter nos algorithmes sur de vrais

ordinateurs quantiques pour bien évaluer le temps de traitement des données est une autre piste de recherche. Le but global reste tout de même classer de grandes quantités de données.

Une autre avenue à explorer serait d'utiliser la fragmentation d'entraînement de données. En effet, ce processus consiste à utiliser les données d'entraînement une seule fois dans le réseau de neurones convolutif quantique. Chaque itération de l'optimiseur utilise de nouvelles données. Il y aura alors un gain de temps d'exécution. Plus de détails sur cette avenue sont données dans ce papier [25].

5 Conclusion

En conclusion, nous avons exploré et comparé les performances des méthodes d'apprentissage automatique classiques et quantiques appliquées à la classification de pulsars.

D'un côté, les algorithmes classiques, déjà bien établis et optimisés, montrent une efficacité et une robustesse pour de nombreux cas d'usage grâce à leur maturité, leur flexibilité[30] et leurs applications dans de multiples secteurs. D'un autre côté, les méthodes quantiques, bien que toujours en phase exploratoire et limitée par les ressources matérielles actuelles, révèlent des avantages prometteurs, notamment en ayant des résultats quasiment aussi précis en simulation. Alors, en ayant accès, dans un futur proche, à un ordinateur quantique fiable, une possibilité d'accélération du calcul pourra aussi se présenter.

Bien que les approches quantiques n'offrent pas encore de gains significatifs pour la classification des pulsars par rapport aux méthodes classiques en termes de précision, les perspectives de progrès technologique dans le domaine des ordinateurs quantiques pourraient inverser cette tendance à mesure que les dispositifs et les algorithmes quantiques mûrissent.

Ainsi, les méthodes classiques demeurent des outils fiables et largement accessibles, mais l'apprentissage automatique quantique ouvre de nouvelles perspectives pour l'avenir de la classification des données astrophysiques et autres. Il sera essentiel de continuer à surveiller les avancées technologiques afin de déterminer si et à quel moment les algorithmes quantiques surpasseront leurs équivalents classiques.

6 Références

- [1] Yann LECUN, Y. BENGIO et Geoffrey HINTON. “Deep Learning”. In : *Nature* 521 (mai 2015), p. 436-44. DOI : 10.1038/nature14539.
- [2] Jonathan ALLCOCK et al. *Quantum algorithms for feedforward neural networks*. 2019. arXiv : 1812.03089 [quant-ph]. URL : <https://arxiv.org/abs/1812.03089>.
- [3] Robert LYON. *HTRU2*. UCI Machine Learning Repository. DOI : <https://doi.org/10.24432/C5DK6R>. 2015.
- [4] PENNYLANE. *qml.IQPEmbedding*. URL : <https://docs.pennylane.ai/en/stable/code/api/pennylane.IQPEmbedding.html> (visité le 17/10/2024).
- [5] Vojtěch HAVLÍČEK et al. “Supervised learning with quantum-enhanced feature spaces”. In : *Nature* 567.7747 (mars 2019), p. 209-212. ISSN : 1476-4687. DOI : 10.1038/s41586-019-0980-2. URL : <http://dx.doi.org/10.1038/s41586-019-0980-2>.
- [6] Francesco Petruccione MARIA SCHULD. *Machine Learning with Quantum Computers*. DOI : <https://doi.org/10.1007/978-3-030-83098-4>. Springer Cham, 2021.

- [7] Thomas HUBREGTSEN et al. “Training quantum embedding kernels on near-term quantum computers”. In : *Physical Review A* 106.4 (oct. 2022). ISSN : 2469-9934. DOI : 10.1103/PhysRevA.106.042431. URL : <http://dx.doi.org/10.1103/PhysRevA.106.042431>.
- [8] Seth LLOYD et al. *Quantum embeddings for machine learning*. 2020. arXiv : 2001.03622 [quant-ph]. URL : <https://arxiv.org/abs/2001.03622>.
- [9] Jarrod R MCCLEAN et al. “The theory of variational hybrid quantum-classical algorithms”. In : *New Journal of Physics* 18.2 (fév. 2016), p. 023023. ISSN : 1367-2630. DOI : 10.1088/1367-2630/18/2/023023. URL : <http://dx.doi.org/10.1088/1367-2630/18/2/023023>.
- [10] Elies M. Gil FUSTER et José Ignacio LATORRE. “Variational Quantum Classifier”. In : 2019. URL : <https://api.semanticscholar.org/CorpusID:203595818>.
- [11] Colin P. WILLIAMS. “Quantum Gates”. In : *Explorations in Quantum Computing*. London : Springer London, 2011, p. 51-122. ISBN : 978-1-84628-887-6. DOI : 10.1007/978-1-84628-887-6_2. URL : https://doi.org/10.1007/978-1-84628-887-6_2.
- [12] Trevor HASTIE, Robert TIBSHIRANI et Jerome FRIEDMAN. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer, 2009.
- [13] Dheeraj PEDDIREDDY et al. *Tensor Ring Parametrized Variational Quantum Circuits for Large Scale Quantum Machine Learning*. Jan. 2022. DOI : 10.48550/arXiv.2201.08878.
- [14] Vasilis BELIS et al. “Higgs analysis with quantum classifiers”. In : *EPJ Web of Conferences* 251 (2021). Sous la dir. de C. BISCARAT et al., p. 03070. ISSN : 2100-014X. DOI : 10.1051/epjconf/202125103070. URL : <http://dx.doi.org/10.1051/epjconf/202125103070>.
- [15] Sukin SIM, Peter D. JOHNSON et Alán ASPURU-GUZIĆ. “Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms”. In : *Advanced Quantum Technologies* 2.12 (oct. 2019). ISSN : 2511-9044. DOI : 10.1002/qute.201900070. URL : <http://dx.doi.org/10.1002/qute.201900070>.
- [16] Lena FUNCKE et al. “Dimensional Expressivity Analysis of Parametric Quantum Circuits”. In : *Quantum* 5 (mars 2021), p. 422. ISSN : 2521-327X. DOI : 10.22331/q-2021-03-29-422. URL : <https://doi.org/10.22331/q-2021-03-29-422>.
- [17] Juneseo LEE et al. “Progress toward favorable landscapes in quantum combinatorial optimization”. In : *Physical Review A* 104.3 (sept. 2021). ISSN : 2469-9934. DOI : 10.1103/PhysRevA.104.032401. URL : <http://dx.doi.org/10.1103/PhysRevA.104.032401>.
- [18] Michał STECHLY. *Introduction to Variational Quantum Algorithms*. 2024. arXiv : 2402.15879 [quant-ph]. URL : <https://arxiv.org/abs/2402.15879>.
- [19] Tobias HAUG, Kishor BHARTI et M.S. KIM. “Capacity and Quantum Geometry of Parametrized Quantum Circuits”. In : *PRX Quantum* 2.4 (oct. 2021). ISSN : 2691-3399. DOI : 10.1103/prxquantum.2.040309. URL : <http://dx.doi.org/10.1103/PRXQuantum.2.040309>.
- [20] M. CEREZO et al. “Cost function dependent barren plateaus in shallow parametrized quantum circuits”. In : *Nature Communications* 12.1 (mars 2021). ISSN : 2041-1723. DOI : 10.1038/s41467-021-21728-w. URL : <http://dx.doi.org/10.1038/s41467-021-21728-w>.
- [21] IBM QUANTUM. *TwoLocal*. URL : <https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.TwoLocal#twolocal> (visité le 17/10/2024).

- [22] PENNYLANE. *qml.RandomLayers*. URL : <https://docs.pennyLane.ai/en/stable/code/api/pennyLane.RandomLayers.html> (visité le 17/10/2024).
- [23] Zewen LI et al. “A Survey of Convolutional Neural Networks : Analysis, Applications, and Prospects”. In : *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2022), p. 6999-7019. DOI : 10.1109/TNNLS.2021.3084827.
- [24] Hongwei QIN et al. “Joint Training of Cascaded CNN for Face Detection”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.
- [25] Donovan SLABBERT, Matt LOURENS et Francesco PETRUCCIONE. “Pulsar classification : comparing quantum convolutional neural networks and quantum support vector machines”. In : *Quantum Machine Intelligence* 6.2 (sept. 2024). ISSN : 2524-4914. DOI : 10.1007/s42484-024-00194-9. URL : <http://dx.doi.org/10.1007/s42484-024-00194-9>.
- [26] R. BOCK. *MAGIC Gamma Telescope*. UCI Machine Learning Repository. DOI : <https://doi.org/10.24432/C52C8B>. 2004.
- [27] D. HECK et al. “CORSIKA : A Monte Carlo code to simulate extensive air showers”. In : (fév. 1998).
- [28] Ashley MONTANARO. “Quantum algorithms : an overview”. In : *npj Quantum information* (2016).
- [29] Joseph BOWLES, Shahnawaz AHMED et Maria SCHULD. *Better than classical? The subtle art of benchmarking quantum machine learning models*. 2024. arXiv : 2403.07059 [quant-ph]. URL : <https://arxiv.org/abs/2403.07059>.
- [30] Sara ALGHUNAIM et Heyam H. AL-BAITY. “On the Scalability of Machine-Learning Algorithms for Breast Cancer Prediction in Big Data Context”. In : *IEEE Access* 7 (2019), p. 91535-91546. DOI : 10.1109/ACCESS.2019.2927080.

7 Annexe

7.1 Tableaux des résultats

QSVM							
Encodage	Optimiseur	Jeu de données	Noyau/circuit variationnel	Exactitude	Précision	Précision négative	Rétention de l'info
Amplitude	NONE	Pulsar	N.A.	0,9250	1,0000	0,8846	0,8235
RX	NONE	Pulsar	N.A.	0,6500	0,9000	0,5667	0,3535
RY	NONE	Pulsar	N.A.	0,8500	0,8750	0,8333	0,6869
RZ	NONE	Pulsar	N.A.	0,4750	div0	0,4750	div0
IQP	NONE	Pulsar	N.A.	0,7500	0,9231	0,6667	0,5188
Amplitude	NONE	Télescope	N.A.	0,6500	0,5294	0,7391	0,2800
RX	NONE	Télescope	N.A.	0,4000	div0	0,4000	div0
RY	NONE	Télescope	N.A.	0,3250	div0	0,3250	div0
RZ	NONE	Télescope	N.A.	0,5750	0,5600	0,6000	0,1500
IQP	NONE	Télescope	N.A.	0,4750	0,4750	div0	div0
Moyenne				0,6075	0,7518	0,6100	0,4688

FIGURE 3 – Résultats des tests sur l'implémentation de la méthode à noyau quantique

SVM							
Encodage	Optimiseur	Jeu de données	Noyau/circuit variationnel	Exactitude	Précision	Précision négative	Rétention de l'info
N.A.	N.A.	Pulsar	rbf	0,9000	0,8513	0,9520	0,8143
N.A.	N.A.	Pulsar	poly	0,8750	0,9267	0,8393	0,7583
N.A.	N.A.	Pulsar	linear	0,9750	0,9803	0,9717	0,9463
N.A.	N.A.	Télescope	rbf	0,7750	0,6513	0,8667	0,5520
N.A.	N.A.	Télescope	poly	0,6833	0,3827	0,9510	0,5287
N.A.	N.A.	Télescope	linear	0,7167	0,5480	0,8797	0,4683
Moyenne				0,8208	0,7234	0,9101	0,6780

FIGURE 4 – Résultats des tests sur l'implémentation de la méthode à noyau classique

VQC

Encodage	Optimiseur	Jeu de données	Noyau/circuit variationnel	Exactitude	Précision	Précision négative	Rétention de l'info
Amplitude	COBYLA	Pulsar	Sur mesure	0,9333	0,9667	0,9077	0,8613
Amplitude	COBYLA	Pulsar	Random_layers	0,8500	0,9090	0,7780	0,7080
RX	COBYLA	Pulsar	Sur mesure	0,9000	0,9440	0,8640	0,8000
RX	COBYLA	Pulsar	Random_layers	0,7250	1,0000	0,6210	0,5000
RY	COBYLA	Pulsar	Sur mesure	0,6250	1,0000	0,5710	0,2500
RY	COBYLA	Pulsar	Random_layers	0,7210	1,0000	0,6330	0,4760
RZ	COBYLA	Pulsar	Sur mesure	0,0000	0,5500	div0	0,0000
RZ	COBYLA	Pulsar	Random_layers	0,4500	0,4500	div0	0,0000
IQP	COBYLA	Pulsar	Sur mesure	0,5667	0,5570	0,5903	0,1177
IQP	COBYLA	Pulsar	Random_layers	0,5500	0,4670	0,6000	0,0640
Amplitude	Nesterov	Pulsar	Sur mesure	0,8500	0,8820	0,8260	0,6940
Amplitude	Nesterov	Pulsar	Random_layers	0,8500	0,8000	0,9330	0,6890
RX	Nesterov	Pulsar	Sur mesure	0,9000	1,0000	0,8260	0,8100
RX	Nesterov	Pulsar	Random_layers	0,4250	div0	0,4250	0,0000
RY	Nesterov	Pulsar	Sur mesure	0,8500	1,0000	0,7600	0,7140
RY	Nesterov	Pulsar	Random_layers	0,4250	div0	0,4250	0,0000
RZ	Nesterov	Pulsar	Sur mesure	0,0000	0,4750	div0	0,0000
RZ	Nesterov	Pulsar	Random_layers	0,6500	0,6500	div0	0,0000
IQP	Nesterov	Pulsar	Sur mesure	0,5250	0,3850	0,5930	-0,0210
IQP	Nesterov	Pulsar	Random_layers	0,5000	0,6110	0,4090	0,0210
Amplitude	COBYLA	Télescope	Sur mesure	0,6750	0,5710	0,9170	0,4190
Amplitude	COBYLA	Télescope	Random_layers	0,7000	0,6320	0,7620	0,4020
RX	COBYLA	Télescope	Sur mesure	0,5750	0,6110	0,5450	0,1550
RX	COBYLA	Télescope	Random_layers	0,5750	0,6000	0,5330	0,1260
RY	COBYLA	Télescope	Sur mesure	0,6750	0,6670	0,6840	0,3500
RY	COBYLA	Télescope	Random_layers	0,4000	0,3300	0,4050	-0,0740
RZ	COBYLA	Télescope	Sur mesure	0,0000	0,5000	div0	0,0000
RZ	COBYLA	Télescope	Random_layers	0,4500	div0	0,4500	0,0000
IQP	COBYLA	Télescope	Sur mesure	0,3750	0,2500	0,4580	-0,2860
IQP	COBYLA	Télescope	Random_layers	0,4750	0,5000	0,4380	-0,0600
Amplitude	Nesterov	Télescope	Sur mesure	0,7250	0,6670	0,8130	0,4610
Amplitude	Nesterov	Télescope	Random_layers	0,7750	0,7780	0,7730	0,5460
RX	Nesterov	Télescope	Sur mesure	0,4000	div0	0,4000	0,0000
RX	Nesterov	Télescope	Random_layers	0,3750	0,3750	div0	0,0000
RY	Nesterov	Télescope	Sur mesure	0,6250	0,6670	0,5910	0,2560
RY	Nesterov	Télescope	Random_layers	0,5000	0,5000	div0	0,0000
RZ	Nesterov	Télescope	Sur mesure	0,0000	0,5000	div0	0,0000
RZ	Nesterov	Télescope	Random_layers	0,5000	0,5000	div0	0,0000
IQP	Nesterov	Télescope	Sur mesure	0,4750	0,5220	0,4710	-0,0080
IQP	Nesterov	Télescope	Random_layers	0,3250	0,2730	0,3890	-0,3430
Moyenne				0,5474	0,6414	0,6255	0,2157

FIGURE 5 – Résultats des tests sur l'implémentation du classificateur variationnel quantique

QCNN

Encodage	Optimiseur	Jeu de données	Noyau/circuit variationnel	Exactitude	Précision	Précision négative	Rétention de l'info
Amplitude	COBYLA	Pulsar	N.A.	0,9250	1,0000	0,8750	0,8421
RX	COBYLA	Pulsar	N.A.	0,6000	0,5833	0,6250	0,2000
RY	COBYLA	Pulsar	N.A.	0,8500	0,8889	0,8182	0,7000
RZ	COBYLA	Pulsar	N.A.	0,4500	0,4500	div0	0,0000
IQP	COBYLA	Pulsar	N.A.	0,5750	0,8000	0,5429	0,1500
Amplitude	Nesterov	Pulsar	N.A.	0,9250	0,9412	0,9130	0,8434
RX	Nesterov	Pulsar	N.A.	0,4250	div0	0,4250	0,0000
RY	Nesterov	Pulsar	N.A.	0,4500	div0	0,4500	0,0000
RZ	Nesterov	Pulsar	N.A.	0,4750	0,4750	div0	0,0000
IQP	Nesterov	Pulsar	N.A.	0,3750	0,3750	div0	0,0000
Amplitude	COBYLA	Télescope	N.A.	0,7000	0,5714	0,8421	0,4400
RX	COBYLA	Télescope	N.A.	0,4500	0,4500	div0	0,0000
RY	COBYLA	Télescope	N.A.	0,4000	0,4000	div0	0,0000
RZ	COBYLA	Télescope	N.A.	0,4250	div0	0,4250	0,0000
IQP	COBYLA	Télescope	N.A.	0,4750	0,4750	div0	0,0000
Amplitude	Nesterov	Télescope	N.A.	0,7250	0,7083	0,7500	0,4411
RX	Nesterov	Télescope	N.A.	0,6000	div0	0,6000	0,0000
RY	Nesterov	Télescope	N.A.	0,5000	div0	0,5000	0,0000
RZ	Nesterov	Télescope	N.A.	0,3250	div0	0,3250	0,0000
IQP	Nesterov	Télescope	N.A.	0,4750	div0	0,4750	0,0000
Moyenne				0,5563	0,6245	0,6119	0,1808

FIGURE 6 – Résultats des tests sur l'implémentation du classificateur de réseau de neurones convolutif quantique

CNN

Encodage	Optimiseur	Jeu de donnée	Noyau/circuit variationnel	Exactitude	Précision	Précision négative	Rétention de l'info
N.A.	Nesterov	Pulsar	N.A.	0,9333	0,8900	0,9800	0,8707
N.A.	COBYLA	Pulsar	N.A.	0,9167	0,8687	0,9627	0,8410
N.A.	Nesterov	Télescope	N.A.	0,9000	0,9553	0,8667	0,7817
N.A.	COBYLA	Télescope	N.A.	0,7250	0,6673	0,7837	0,4560
Moyenne:				0,8688	0,8453	0,8983	0,7373

FIGURE 7 – Résultats des tests sur l'implémentation du classificateur de réseau de neurones convolutif classique