



Stage d'été – Semaine 1

QGNN-TimeCausality

Avancement du projet




Stage d'été – Semaine 1

01

Outils créés

Tracker des lectures et ressources consultés


Contient tous les tutoriels, papiers, videos et githubs consultés pour ce projet avec les takeaways et les principaux apprentissages de chaque lecture.

 Lien d'accès

Lectures					Main learnings	
Tr	Lecture	Type	From	Status	Start date	
	Intro to GNN	video	Sahar Saoudi	Read	5/13/2025	<ul style="list-style-type: none">- La différence entre les NNs et les GNNs en termes d'inputs, de structure, etc.- Le concept de message passing et les mathématiques derrière ça.- L'embedding space, les node embeddings, et leur rôle dans la représentation des graphes.- Comment les GNNs apprennent en faisant circuler des messages entre les nœuds connectés.- Comment les embeddings dépendent d'objectifs mesurables.- La loss function et son rôle.- Les différentes tâches que les GNNs peuvent accomplir, comme la classification de nœuds, le clustering, la classification de graphes.- La vidéo montre un excellent exemple d'une tâche appelée link prediction.- L'approche matricielle du message passing.
	GNN Explanation	video	Sahar Saoudi	Read	5/13/2025	<ul style="list-style-type: none">- explication claire de ce qu'est un graphe et le fait que c'est utilisé pour modéliser des relations entre entités.- Comment les GNNs exploitent la structure de graphe pour effectuer des tâches d'apprentissage sur des données non-euclidiennes, comme les réseaux sociaux ou les molécules- Les GNNs permettent de propager l'information entre les nœuds d'un graphe, en prenant en compte la structure du graphe et les caractéristiques des nœuds- Le concept de message passing est central : chaque nœud met à jour son état en agrégeant les informations de ses voisins.- Comment les GNNs peuvent être utilisés pour des tâches comme la classification de nœuds, la prédiction de liens ou la classification de graphes entiers.- Les défis liés à l'entraînement des GNNs, notamment le surajustement et la complexité computationnelle, et les techniques pour les atténuer.- L'importance croissante des GNNs dans divers domaines, tels que la biologie, la chimie, les réseaux sociaux et la recommandation de contenu

To-do list

Regroupe les tâches à faire chaque semaine, où on en est rendu et quelques notes au besoin.

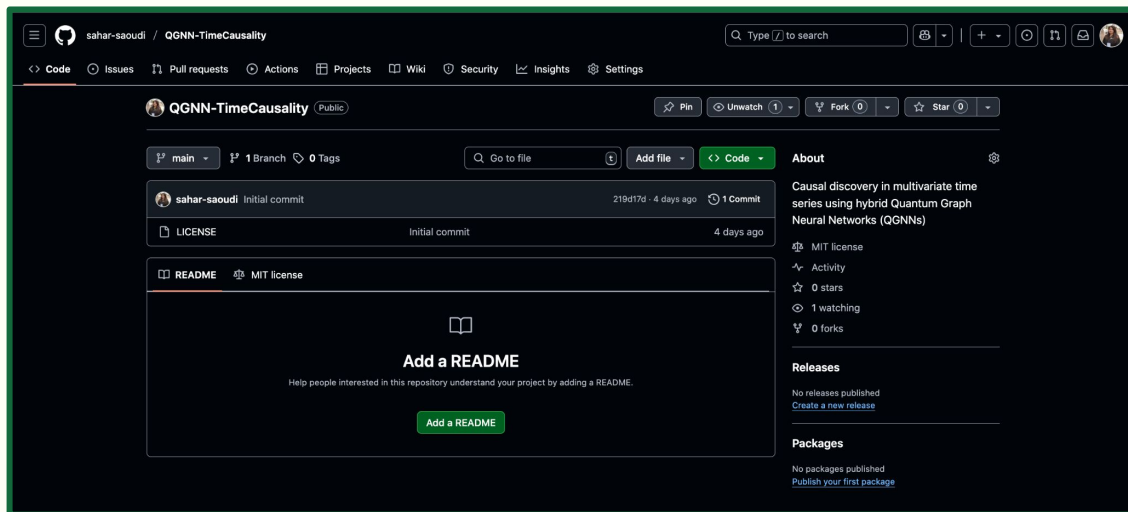
 Lien d'accès

[illegible]

Repository sur Github Créé

Je n'ai encore rien ajouté pour le moment, mais ce repo contiendra tous les fichiers nécessaires au projet. Je pense actuellement à la structure à adopter pour le repo (e.g uniquement du code, ou peut-être aussi un tutoriel en Jupyter Notebook ou d'autres éléments) Je suis preneuse de toute suggestion à ce sujet !

 Lien





02

Ressources consultées et principaux apprentissages

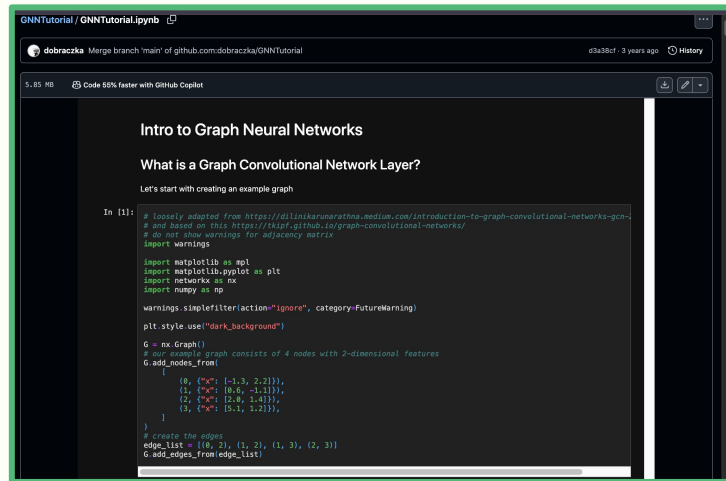
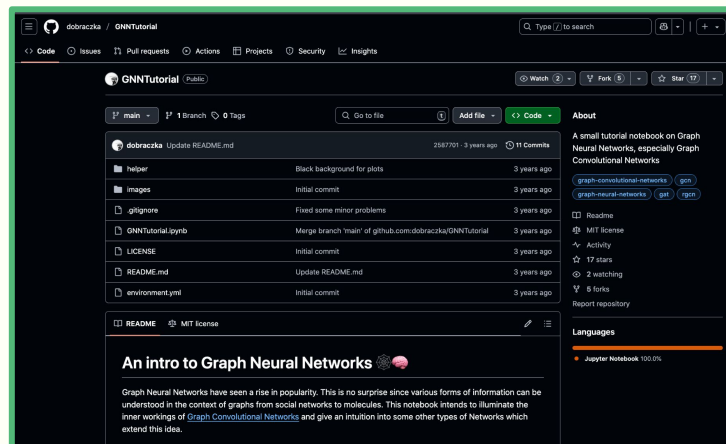
Vidéos sur les GNNs

- Un graphe permet de modéliser des relations entre entités (réseaux sociaux, molécules, etc.).
- Les GNNs exploitent la structure du graphe et les caractéristiques des nœuds pour apprendre.
- Le message passing est au cœur du fonctionnement : les nœuds échangent des infos avec leurs voisins.
- Chaque nœud met à jour son état (embedding) à partir des messages reçus.
- Le modèle est entraîné en minimisant une loss function, comme dans les NNs classiques.
- Les GNNs sont utiles pour plusieurs tâches : classification de nœuds, prédiction de liens, classification de graphes, clustering.
- Le message passing peut être formulé de façon matricielle, ce qui le rend compatible avec des implémentations vectorisées.
- L'entraînement des GNNs peut poser des défis : surapprentissage, complexité, etc.
- Ils sont de plus en plus utilisés en biologie, chimie, réseaux sociaux, etc..



Repo Github sur les GNNs

- explication visuelle de ce qu'est un graphe et comment le représenter avec networkx en Python
- l'utilisation de matrice d'adjacence pour représenter les connexions entre les nœuds
- On attribue à chaque nœud un vecteur de caractéristiques (features), ce qui permet de passer de simples graphes à des graphes avec informations locales.
- concept de couche de convolution sur graphe (GCN layer), qui est une généralisation des CNNs classiques
- La propagation de l'information se fait couche par couche : plus on a de couches, plus un nœud reçoit d'information de ses voisins lointains
- comment visualiser les graphes et les features et comment ça aide à mieux comprendre ce que fait chaque étape du modèle.
- Le tutoriel permet de bien comprendre les bases mathématiques des GNNs et comment les implémenter étape par étape



Papiers scientifiques sur les QGNNs

- Les différentes variantes de QGNN : QGRNN, QGCNN, et QSGCNN
- QGRNN : Utilise des paramètres partagés à travers les itérations, Convient pour modéliser l'évolution temporelle de systèmes quantiques (ex. modèle d'Ising), Simule l'évolution hamiltonienne via la Trotterisation.
- QGCNN : L'ansatz doit être invariant par permutation des nœuds, Utilise des paramètres globaux partagés, non spécifiques aux nœuds
- QSGCNN : Variante de QGCNN inspirée du modèle à variables continues (référence QAOA), Intègre les opérateurs de position et de moment pour simuler le message passing.
- Applications clés mentionnées : Modélisation de dynamiques hamiltoniennes (QGRNN) / Préparation d'états GHZ pour capteurs quantiques (QGCNN) / Clustering de graphes (non supervisé, QSGCNN) / Classification d'isomorphismes de graphes (supervisé, QSGCNN)

Quantum Graph Neural Networks

Guillaume Verdon
X, The Moonshot Factory
Mountain View, CA
gverdon@x.team

Trevor McCourt
Google Research
Venice, CA
trevormcrt@google.com

Enxhell Luzhnica, Vikash Singh,
Stefan Leichenauer, Jack Hriday
X, The Moonshot Factory
Mountain View, CA
{enxhell,singvikash,
sleichenauer,hriday}@x.team

Abstract

We introduce **Quantum Graph Neural Networks (QGNN)**, a new class of quantum neural network ansatz which are tailored to represent **quantum processes** which have a **graph structure**, and are particularly suitable to be executed on distributed quantum systems over a quantum network. Along with this general class of ansatz, we introduce further **specialized architectures**, namely, **Quantum Graph Recurrent Neural Networks (QGRNN)** and **Quantum Graph Convolutional Neural Networks (QGCNN)**. We provide four example applications of QGNNs: learning Hamiltonian dynamics of quantum systems, learning how to create multipartite entanglement in a quantum network, unsupervised learning for spectral clustering, and supervised learning for graph isomorphism classification.

1 Introduction


Variational Quantum Algorithms are a promising class of algorithms that are rapidly emerging as a central subfield of Quantum Computing [1, 2, 3]. Similar to parameterized transformations encountered in deep learning, these parameterized quantum circuits are often referred to as **Quantum Neural Networks (QNNs)**. Recently, it was shown that QNNs that have no prior on their structure suffer from a quantum version of the no-free lunch theorem [4] and are **exponentially difficult to train** via gradient descent. Thus, there is a **need for better QNN ansatz**. One popular class of QNNs has been Trotter-based ansatz [2, 5]. The optimization of these ansatz has been extensively studied in recent works, and efficient optimization methods have been found [6, 7]. On the classical side, graph-based neural networks leveraging data geometry have seen some recent successes in deep learning, finding applications in biophysics and chemistry [8]. Inspired from this success, we propose a new class of Quantum Neural Network ansatz which allows for **both quantum inference and classical probabilistic inference** for data with a graph-geometric structure. In the sections below, we introduce the general framework of the QGNN ansatz as well as several more specialized variants and showcase four potential applications via numerical implementation.

Tutoriel de pennylane sur le QGRNN

- application concrète de la QGRNN, notamment comment elle peut apprendre la dynamique d'un modèle d'Ising
- Le modèle repose sur le partage des mêmes paramètres à chaque étape temporelle, ce qui permet de modéliser efficacement l'évolution dynamique.
- Chaque nœud du graphe est encodé dans un état quantique, ce qui permet de capturer la structure du graphe dans le circuit.
- Le QGRNN applique les mêmes blocs quantiques à plusieurs reprises, à la manière d'un RNN classique.
- J'ai testé le tutoriel moi-même, ce qui m'a permis de mieux comprendre les bibliothèques utilisées : networkx pour manipuler les graphes, scipy pour les opérations d'algèbre linéaire, pennylane pour appliquer les portes quantiques et la préparation des états.
- La visualisation de l'apprentissage à travers la matrice de Hamiltonien (initiale, cible et apprise) permet de bien suivre le progrès du modèle
- Le test de swap est utilisé pour mesurer la similarité entre l'état final appris et l'état cible

Demos / Quantum Machine Learning / The Quantum Graph Recurrent Neural Network

The Quantum Graph Recurrent Neural Network



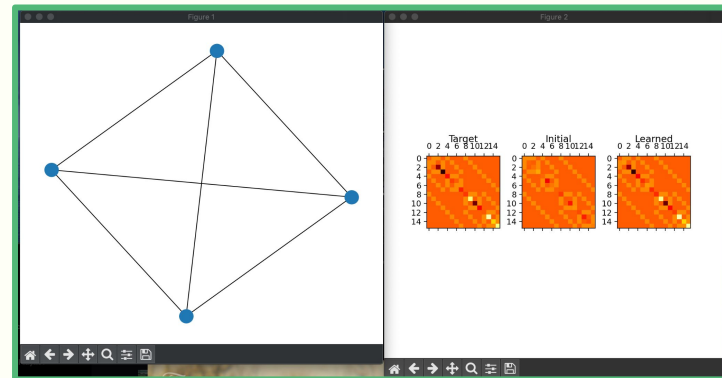
Jack Ceroni

Published: July 26, 2020. Last updated: January 9, 2025.

This demonstration investigates quantum graph recurrent neural networks (QGRNN), which are the quantum analogue of a classical graph recurrent neural network, and a subclass of the more general quantum graph neural network ansatz. Both the QGNN and QGRNN were introduced in [this paper \(2019\)](#).

The Idea

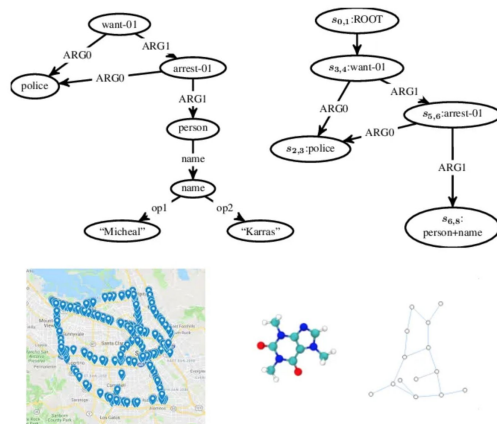
A graph is defined as a set of *nodes* along with a set of *edges*, which represent connections between nodes. Information can be encoded into graphs by assigning numbers to nodes and edges, which we call **weights**. It is usually convenient to think of a graph visually:



Article sur les applications des QGNNs dans la vie réelle

- application des QGNNs pour reconstruire les trajectoires de particules générées lors des collisions (CERN)
- Les collisions produisent énormément de particules, et il est difficile de déterminer quelles traces appartiennent à quelle particule. C'est là que les graphes interviennent.
- Les points de détection sont modélisés comme des nœuds d'un graphe, et les GNNs servent à identifier les liens (arêtes) les plus probables entre eux
- Modèle hybride proposé : il combine des couches classiques (structure du graphe) et des circuits quantiques (traitement des informations)
- Les caractéristiques des nœuds (ex. position, temps) sont encodées dans des états quantiques à l'aide de portes
- profiter des propriétés quantiques (superposition, intrication) pour mieux apprendre les relations complexes entre les nœuds.
- Apprentissage supervisé, avec un ensemble de trajectoires connues servant à entraîner le modèle.
- Le projet parmi les top5 au QHack 2022

• Modelling molecules



Section 1: GraphML meets High Energy Physics

Before diving right into the QML, let's understand the problem we're hoping to solve along with the context it resides upon. That takes us to Geneva.

CERN general operations

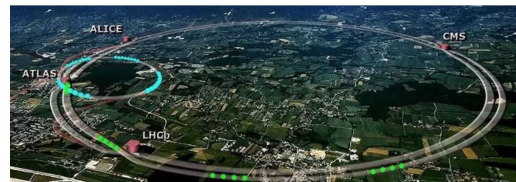


Diagram of the particle accelerator rings at CERN (27km in circumference) | Image credit: CERN

The Large Hadron Collider (LHC) in Geneva, Switzerland, is the world's largest and most powerful particle accelerator. It's a 27-kilometre ring that uses superconducting magnets along with a number of accelerating



03

Questions & Prochaines étapes

Questions

- Je souhaite accéder à mon courriel @univ.telug, mais on me demande d'entrer un numéro de téléphone de bureau pour la vérification.
- Dois-je déjà réfléchir à la structure du dépôt GitHub ou est-ce encore trop tôt ?
- Y a-t-il quelque chose que je pourrais améliorer dans ma présentation ?

Prochaines etapes

- Continuer les lectures, en particulier sur les QGNN, et consulter des dépôts GitHub liés au sujet.
- Explorer les ensembles de données que je peux utiliser pour le projet, et réfléchir à comment les représenter sous forme de graphes.
- Poursuivre les présentations pour communiquer l'avancement du projet.



Merci de votre attention!