

INF6403-G1 - Science des données et intelligence artificielle

Analyse des réseaux sociaux

Thème : La prédiction de liens en utilisant les réseaux de neurones en graphes ("Graph Neural Networks")

Professeur : Missaoui Rokia

Présenté par : Naklan Julien Camara

Été 2024

PLAN

Introduction

Partie 1 : Concepts Fondamentaux des Graphes et de la Prédiction de Liens

Partie 2 : Introduction aux Réseaux de Neurones sur Graphes (GNNs)

Partie 3 : Techniques et Modèles pour la Prédiction de Liens avec GNNs

Partie 4 : Applications et Cas d'Utilisation

Partie 5 : Implémentation

Conclusion

Références

Annexe

Introduction

Contexte

Les graphes sont des structures de données fondamentales qui représentent des relations entre des entités. Ces entités, appelées nœuds, sont connectées par des arêtes qui peuvent être dirigées ou non dirigées, pondérées ou non pondérées. Les graphes sont omniprésents dans divers domaines de la vie réelle :

- **Réseaux sociaux** : Les utilisateurs sont des nœuds et leurs amitiés ou interactions sont des arêtes.
- **Biologie** : Les protéines peuvent être représentées comme des nœuds et leurs interactions comme des arêtes.
- **Systèmes de recommandation** : Les utilisateurs et les produits sont des nœuds, et les achats ou évaluations sont des arêtes.
- **Infrastructure** : Les villes sont des nœuds et les routes ou lignes de communication sont des arêtes.

Dans ces exemples, les graphes capturent les dépendances complexes et les interactions entre les entités, ce qui en fait des structures de données idéales pour de nombreuses applications.

Importance

La **prédiction de liens** est une tâche cruciale dans les graphes. Elle consiste à prédire l'apparition de nouvelles arêtes entre des nœuds dans un graphe donné. Cette tâche a des applications variées et essentielles :

- **Réseaux sociaux** : Recommander des amis ou des connexions professionnelles.
- **Biologie** : Identifier de nouvelles interactions protéine-protéine ou découvrir de nouvelles relations entre gènes.
- **Systèmes de recommandation** : Recommander des produits à des utilisateurs en fonction de leurs préférences et des comportements similaires d'autres utilisateurs.
- **Cybersécurité** : Détecter des relations anormales ou suspectes dans un réseau informatique.

La capacité de prédire de manière précise et efficace ces connexions potentielles peut améliorer les systèmes existants et permettre de découvrir de nouvelles connaissances.

Objectif du Sujet

Les **réseaux de neurones sur graphes (GNN)** ont émergé comme une solution puissante pour traiter des données de graphe de manière plus efficace et plus précise que les méthodes traditionnelles. Les GNNs peuvent capturer les informations structurelles et attributaires des graphes grâce à des mécanismes de propagation de messages et d'agrégation des informations des nœuds voisins.

L'objectif de ce sujet est de :

1. **Présenter les concepts fondamentaux** des graphes et de la prédiction de liens.
2. **Introduire les réseaux de neurones sur graphes (GNNs)**, en expliquant comment ils fonctionnent et en quoi ils diffèrent des méthodes traditionnelles.
3. **Décrire les techniques et modèles de GNN** spécifiquement conçus pour la prédiction de liens.
4. **Illustrer l'implémentation pratique des GNNs** pour la prédiction de liens à l'aide de TensorFlow ou pytorch, un cadre de deep learning populaire.
5. **Explorer les applications réelles** des GNNs pour la prédiction de liens, mettant en évidence leur potentiel et leur impact dans divers domaines.

Partie 1 : Concepts Fondamentaux des Graphes et de la Prédiction de Liens

1.1 Définition et Structure des Graphes

Les graphes sont des structures mathématiques utilisées pour modéliser les relations entre des objets. Ils sont composés de nœuds (ou sommets) et d'arêtes (ou liens).

Éléments d'un Graphe :

- **Nœuds (Sommets)** : Représentent les entités ou objets. Par exemple, dans un réseau social, les utilisateurs sont des nœuds.
- **Arêtes (Liens)** : Représentent les relations ou interactions entre les nœuds. Dans un réseau social, une amitié ou une interaction entre deux utilisateurs est une arête.
- **Matrice d'adjacence** : Une représentation courante des graphes où les nœuds sont indexés et chaque élément (i, j) de la matrice indique s'il existe une arête entre le nœud i et le nœud j .
- **Attributs des Nœuds et des Arêtes** : Des informations supplémentaires peuvent être associées aux nœuds (ex. caractéristiques des utilisateurs) ou aux arêtes (ex. force de l'interaction).

Types de Graphes :

- **Graphes Dirigés** : Les arêtes ont une direction, c'est-à-dire qu'elles vont d'un nœud source à un nœud cible.
- **Graphes Non Dirigés** : Les arêtes n'ont pas de direction. Si une arête existe entre deux nœuds, elle est bidirectionnelle.
- **Graphes Pondérés** : Les arêtes ont des poids, représentant l'importance ou la force de la relation.
- **Graphes Non Pondérés** : Les arêtes n'ont pas de poids.

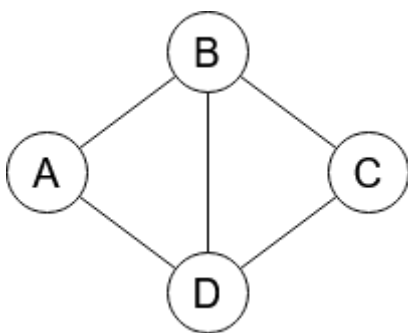


Figure 1 : un graphe non-orienté

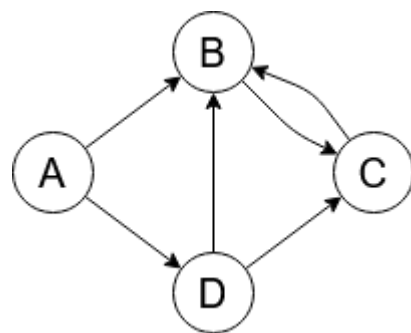


Figure 2 : un graphe orienté

1.2 Prédiction de Liens

La prédiction de liens est la tâche consistant à prédire l'existence ou l'apparition future de nouvelles arêtes dans un graphe. Elle peut être formulée comme un problème de classification binaire où l'objectif est de déterminer si une arête existe ou non entre deux nœuds donnés.

Applications de la Prédiction de Liens :

- **Réseaux Sociaux** : Recommander des amis ou des connexions professionnelles.
- **Biologie** : Identifier de nouvelles interactions entre protéines ou gènes.
- **Systèmes de Recommandation** : Suggérer des produits, des films ou d'autres articles aux utilisateurs.
- **Finance** : Détecter des relations entre des entités économiques ou financières.

Méthodes Traditionnelles de Prédiction de Liens :

- **Basées sur la Similarité** : Ces méthodes utilisent des mesures de similarité pour prédire les liens. Par exemple :
 - **Common Neighbors** : Deux nœuds ont d'autant plus de chances d'être connectés qu'ils partagent un grand nombre de voisins communs.
 - **Jaccard Coefficient** : Mesure la similarité entre deux ensembles de voisins.
 - **Adamic/Adar Index** : Somme des inverses des logarithmes du degré des voisins communs.
- **Basées sur les Heuristiques** : Utilisation de heuristiques pour estimer la probabilité de la présence d'une arête. Par exemple :
 - **Preferential Attachment** : Les nœuds avec un degré élevé ont plus de chances d'attirer de nouvelles arêtes.
 - **Resource Allocation** : Distribution des ressources sur les chemins reliant les nœuds pour prédire les nouveaux liens.

Limitations des Méthodes Traditionnelles :

- **Échelle** : Les méthodes basées sur la similarité et les heuristiques peuvent être limitées en termes de scalabilité pour les grands graphes.
- **Complexité** : Elles peuvent ne pas capturer les relations complexes et les dépendances structurelles dans les graphes.
- **Précision** : Les performances peuvent être limitées lorsque les graphes sont très hétérogènes ou dynamiques.

Conclusion partielle

Les graphes sont des structures puissantes et flexibles pour modéliser les relations entre les entités dans de nombreux domaines. La prédiction de liens est une tâche essentielle pour découvrir de nouvelles connexions potentielles dans ces graphes, avec des applications variées allant des réseaux sociaux à la biologie. Cependant, les méthodes traditionnelles présentent des limitations qui peuvent être surmontées par l'utilisation de réseaux de neurones sur graphes (GNNs), ce qui nous amène à explorer ces modèles avancés dans les parties suivantes.

Partie 2 : Introduction aux Réseaux de Neurones sur Graphes (GNNs)

Les réseaux de neurones sur graphes (GNNs) sont une classe avancée de réseaux de neurones conçus pour travailler sur des données représentées sous forme de graphes. Contrairement aux données tabulaires ou séquentielles, les graphes contiennent des nœuds (entités) et des arêtes (relations) avec une structure complexe et interconnectée. Les GNNs sont capables de capturer cette complexité et de tirer parti des relations entre les nœuds pour des tâches d'apprentissage automatique telles que la classification de nœuds, la classification de graphes, et la prédiction de liens.

1. Notions de Base des Graphes

Avant de plonger dans les GNNs, il est essentiel de comprendre les concepts de base des graphes.

- **Nœud (Node)** : Un point ou une entité dans le graphe. Par exemple, dans un réseau social, un nœud pourrait représenter une personne.
- **Arête (Edge)** : Une connexion entre deux nœuds, représentant une relation ou une interaction. Par exemple, une arête pourrait représenter une amitié entre deux personnes dans un réseau social.
- **Graphe (Graph)** : Une structure composée de nœuds et d'arêtes. Un graphe peut être dirigé ou non dirigé, pondéré ou non pondéré.
- **Adjacency Matrix (Matrice d'Adjacence)** : Une matrice carrée utilisée pour représenter un graphe, où une entrée non nulle indique la présence d'une arête entre deux nœuds.

2. Qu'est-ce qu'un Réseau de Neurones sur Graphes (GNN) ?

Un GNN est une architecture de réseau de neurones conçue pour effectuer des opérations directement sur les graphes. Les GNNs exploitent les informations structurelles du graphe pour apprendre des représentations des nœuds et des arêtes. Voici les composants clés des GNNs :

- **Message Passing (Propagation des Messages)** : Un processus itératif où chaque nœud échange des informations avec ses voisins pour mettre à jour ses représentations.
- **Aggregation (Agrégation)** : Les informations reçues des voisins sont agrégées pour obtenir une représentation collective.
- **Update (Mise à Jour)** : La représentation du nœud est mise à jour en utilisant les informations agrégées.

3. Types de GNNs

Il existe plusieurs variantes de GNNs, chacune ayant ses propres mécanismes pour la propagation des messages, l'agrégation, et la mise à jour des représentations :

Graph Convolutional Networks (GCN)

Les GCNs appliquent des opérations de convolution sur les graphes, similaires aux convolutions sur les images. Elles permettent de capturer les relations locales entre les nœuds. L'équation de mise à jour pour un nœud v dans un GCN est souvent formulée comme suit :

$$h_v^{(k+1)} = \sigma \left(\sum_{u \in \mathcal{N}(v)} \frac{1}{\sqrt{d_u d_v}} W^{(k)} h_u^{(k)} \right)$$

où $\mathcal{N}(v)$ représente les voisins du nœud v , d_u et d_v sont les degrés des nœuds u et v , $W^{(k)}$ est la matrice de poids à l'étape k , et σ est une fonction d'activation.

Graph Attention Networks (GAT)

Les GATs introduisent des mécanismes d'attention qui permettent aux nœuds de pondérer différemment les contributions de leurs voisins. Cela permet de se concentrer sur les relations les plus importantes. L'équation de mise à jour pour un nœud v dans un GAT est :

$$h_v^{(k+1)} = \sigma \left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu}^{(k)} W^{(k)} h_u^{(k)} \right)$$

où $\alpha_{vu}^{(k)}$ est le coefficient d'attention attribué à la contribution du nœud u au nœud v à l'étape k .

GraphSAGE (Graph Sample and Aggregation)

GraphSAGE échantillonne les voisins des nœuds et utilise différentes fonctions d'agrégation (moyenne, LSTM, pool) pour combiner les informations des voisins. Cela permet de gérer des graphes de grande taille. L'équation de mise à jour pour un nœud v dans GraphSAGE est :

$$h_v^{(k+1)} = \sigma \left(W^{(k)} \cdot \text{AGGREGATE} \left(\{h_u^{(k)}, \forall u \in \mathcal{N}(v)\} \right) \right)$$

où AGGREGATE est une fonction d'agrégation, telle que la moyenne ou le max pooling.

Variational Graph Auto-Encoders (VGAE)

Les VGAE utilisent des techniques d'auto-encodage variationnel pour apprendre des représentations latentes des nœuds dans un espace continu. Cela permet de modéliser les incertitudes et de générer de nouvelles structures de graphe. L'objectif est de maximiser la probabilité de reconstruction du graphe à partir des représentations latentes.

4. Applications des GNNs

Les GNNs sont utilisés dans de nombreux domaines pour diverses tâches, notamment :

- **Classification de Nœuds** : Prédire les étiquettes des nœuds dans un graphe, comme la classification des utilisateurs dans un réseau social.
- **Prédiction de Liens** : Prédire l'existence de liens futurs entre des nœuds, utilisé dans les recommandations d'amis ou la découverte de médicaments.
- **Classification de Graphes** : Classifier des graphes entiers, comme la classification de molécules en fonction de leurs propriétés chimiques.

5. Avantages et Défis des GNNs

Avantages

- **Capacité à Capturer les Relations Complexes** : Les GNNs exploitent les structures des graphes pour capturer les dépendances entre les nœuds et les arêtes.
- **Flexibilité** : Ils peuvent être appliqués à une large gamme de problèmes dans divers domaines.

Défis

- **Scalabilité** : Traiter de très grands graphes peut être coûteux en termes de mémoire et de calcul.
- **Apprentissage Supervisé vs. Non Supervisé** : Les GNNs nécessitent souvent des données annotées pour l'apprentissage supervisé, ce qui peut être coûteux à obtenir.

Conclusion partielle

Les réseaux de neurones sur graphes (GNNs) représentent une avancée majeure dans le traitement des données graphiques, permettant d'appliquer des techniques d'apprentissage profond à des structures de données complexes et interconnectées. Leur capacité à capturer les relations et les dépendances entre les nœuds et les arêtes ouvre de nouvelles opportunités pour des applications avancées dans divers domaines, de la bioinformatique aux réseaux sociaux en passant par la finance et le commerce électronique.

Partie 3 : Techniques et Modèles pour la Prédiction de Liens avec GNNs

La prédiction de liens est une tâche cruciale dans le domaine des réseaux complexes, où l'objectif est de prédire l'existence de liens futurs entre des nœuds dans un graphe. Les réseaux de neurones graphiques (GNNs) ont émergé comme une approche puissante pour cette tâche, offrant des méthodes sophistiquées pour capturer les dépendances et les structures des graphes.

1. Présentation Générale des GNNs

Les GNNs sont des réseaux de neurones spécialement conçus pour travailler sur les graphes. Ils utilisent des opérations de convolution sur les graphes pour apprendre des représentations des nœuds, des arêtes et des sous-graphes. Les principales étapes dans les GNNs incluent :

- **Message Passing** : Les nœuds échangent des informations avec leurs voisins directs.
- **Aggregation** : Les informations des voisins sont agrégées.
- **Update** : Les représentations des nœuds sont mises à jour en fonction des informations agrégées.

2. Modèles Couramment Utilisés pour la Prédiction de Liens

Graph Convolutional Networks (GCN)

Les GCNs utilisent des convolutions spectrales pour capturer les caractéristiques des nœuds et leurs voisins. Pour la prédiction de liens, les représentations des nœuds sont apprises et ensuite combinées pour estimer la probabilité de l'existence d'un lien entre deux nœuds.

Graph Attention Networks (GAT)

Les GATs introduisent des mécanismes d'attention dans le processus de message passing, permettant aux nœuds de pondérer l'importance de leurs voisins de manière adaptative. Cela permet de mieux capturer les dépendances complexes dans le graphe.

GraphSAGE

GraphSAGE (Graph Sample and Aggregation) échantillonne les voisins des nœuds et agrège leurs caractéristiques, ce qui permet de traiter des graphes de grande taille. Les agrégateurs peuvent être moyens, LSTM, ou basés sur des opérations de pool.

Variational Graph Auto-Encoders (VGAE)

Les VGAE utilisent des techniques d'auto-encodage variationnel pour apprendre des représentations latentes des nœuds, qui sont ensuite utilisées pour prédire les liens. Les auto-encodeurs graphiques visent à reconstruire le graphe original à partir des représentations latentes.

3. Méthodologies pour la Prédiction de Liens

Préparation des Données

1. **Échantillonnage de Liens Positifs et Négatifs** : Équilibrer les données en échantillonnant des liens existants (positifs) et des liens inexistantes (négatifs).
2. **Division des Données** : Diviser les données en ensembles d'entraînement, de validation et de test en fonction du temps ou de manière aléatoire.

Entraînement des Modèles

1. **Apprentissage des Représentations** : Utiliser les GNNs pour apprendre des représentations des nœuds.
2. **Construction des Paires** : Créer des paires de nœuds pour lesquelles la présence ou l'absence de lien doit être prédite.
3. **Classification** : Utiliser un classificateur pour prédire la probabilité de lien entre les paires de nœuds.

Évaluation des Modèles

1. **Métriques** : Utiliser des métriques telles que l'aire sous la courbe ROC (AUC), la précision, le rappel et le F1-score pour évaluer les performances des modèles.
2. **Courbes ROC et PR** : Analyser les courbes ROC et PR pour comprendre la performance du modèle sous différents seuils de classification.

4. Exemple de Pipeline de Prédiction de Liens

Voici un exemple de pipeline pour la prédiction de liens en utilisant les GNNs :

1. **Construction du Graphe** : Construire le graphe à partir des données disponibles.
2. **Initialisation des Représentations** : Initialiser les représentations des nœuds avec des caractéristiques initiales (ex. : embeddings aléatoires ou basés sur des attributs).
3. **Propagation des Messages** : Appliquer des couches de GNN pour propager les messages et mettre à jour les représentations des nœuds.
4. **Formation des Paires de Nœuds** : Former des paires de nœuds pour l'entraînement et le test.
5. **Prédiction des Liens** : Utiliser un classificateur pour prédire la probabilité de lien pour chaque paire de nœuds.
6. **Évaluation** : Évaluer les performances du modèle en utilisant les métriques appropriées.

Conclusion partielle

La prédiction de liens avec les GNNs est une tâche complexe mais puissante, qui permet de capturer les dépendances et les structures sous-jacentes des graphes. Les techniques et les modèles mentionnés ici offrent une variété d'approches pour aborder ce problème, en fonction des caractéristiques spécifiques du graphe et des besoins de l'application. En combinant une préparation adéquate des données, des modèles de pointe et des méthodologies robustes, il est possible d'obtenir des prédictions de liens précises et utiles.

Partie 4 : Applications et Cas d'Utilisation des GNNs pour la Prédiction de Liens

Les réseaux de neurones graphiques (GNNs) pour la prédiction de liens ont des applications vastes et variées dans plusieurs domaines. Voici un aperçu des applications les plus notables ainsi que des cas d'utilisation spécifiques :

1. Réseaux Sociaux

Les réseaux sociaux sont un terrain propice pour la prédiction de liens en raison de leur nature intrinsèquement graphique. Les GNNs peuvent être utilisés pour :

- **Recommandation d'Amis** : Prédire de nouvelles connexions potentielles entre utilisateurs en se basant sur leurs relations existantes et leurs interactions.
- **Détection de Communautés** : Identifier des groupes d'utilisateurs ayant des intérêts ou des connexions similaires.
- **Prévention de Fraude** : Détecter des comportements anormaux et des comptes frauduleux en analysant les motifs de connexion entre utilisateurs.

Cas d'Utilisation : Facebook et LinkedIn

Facebook et LinkedIn utilisent des algorithmes de prédiction de liens pour suggérer des amis ou des connexions professionnelles basées sur des caractéristiques telles que les amis communs, les interactions et les intérêts partagés.

2. Systèmes de Recommandation

Les GNNs sont utilisés pour améliorer les systèmes de recommandation en analysant les relations entre utilisateurs et items (produits, films, articles, etc.).

- **Recommandation de Produits** : Prédire quels produits un utilisateur est susceptible d'acheter en se basant sur ses achats précédents et ceux d'utilisateurs similaires.
- **Recommandation de Contenus** : Suggérer des films, des livres ou des articles basés sur les préférences et les interactions passées.

Cas d'Utilisation : Amazon et Netflix

Amazon et Netflix utilisent des modèles de recommandation sophistiqués, incluant des GNNs, pour analyser les interactions entre utilisateurs et items et fournir des recommandations personnalisées.

3. Bioinformatique

En bioinformatique, les GNNs sont appliqués à des problèmes tels que la prédiction des interactions protéine-protéine, l'identification des relations entre gènes, et la découverte de médicaments.

- **Prédiction des Interactions Protéine-Protéine** : Identifier les protéines susceptibles d'interagir en se basant sur leurs structures et leurs fonctions.
- **Découverte de Médicaments** : Identifier de nouvelles interactions entre les composés chimiques et les cibles biologiques.

Cas d'Utilisation : Recherche Pharmaceutique

Des entreprises pharmaceutiques utilisent des GNNs pour accélérer le processus de découverte de médicaments en identifiant rapidement des interactions potentielles entre les molécules et les cibles biologiques.

4. Réseaux de Télécommunications

Les opérateurs de télécommunications peuvent utiliser les GNNs pour analyser et optimiser leurs réseaux.

- **Optimisation du Routage** : Prédire les défaillances dans le réseau et optimiser les routes de données pour améliorer la performance.
- **Détection d'Anomalies** : Identifier des comportements anormaux dans le réseau qui pourraient indiquer des problèmes ou des attaques.

Cas d'Utilisation : Gestion des Réseaux

Les opérateurs utilisent des GNNs pour analyser les flux de données, détecter les pannes potentielles et optimiser le routage du trafic pour maintenir la qualité du service.

5. Commerce Électronique

Les plateformes de commerce électronique utilisent les GNNs pour améliorer l'expérience utilisateur et augmenter les ventes.

- **Recommandation de Produits** : Suggérer des produits complémentaires ou similaires en analysant les comportements d'achat des utilisateurs.
- **Personnalisation** : Offrir une expérience d'achat personnalisée en fonction des interactions passées des utilisateurs.

Cas d'Utilisation : Alibaba et eBay

Ces plateformes utilisent des algorithmes de recommandation basés sur des GNNs pour analyser les comportements des utilisateurs et fournir des recommandations précises et personnalisées.

6. Finance et Assurances

Les institutions financières et les compagnies d'assurance appliquent les GNNs pour évaluer les risques et détecter les fraudes.

- **Évaluation des Risques** : Analyser les relations entre les entités financières pour évaluer les risques de crédit et d'investissement.
- **Détection de Fraude** : Identifier des transactions suspectes et des comportements frauduleux en analysant les motifs de connexion entre les entités.

Cas d'Utilisation : Détection de Fraude Bancaire

Les banques utilisent des GNNs pour détecter des transactions frauduleuses en analysant les relations complexes entre les comptes et les transactions.

Conclusion partielle

Les GNNs pour la prédiction de liens offrent des solutions puissantes et polyvalentes pour une multitude d'applications à travers divers domaines. Que ce soit pour recommander des amis, des produits, ou des contenus, optimiser des réseaux de télécommunications, ou découvrir de nouveaux médicaments, les GNNs permettent de capturer et d'exploiter les structures complexes des données relationnelles. Leur capacité à apprendre des représentations riches des nœuds et des liens dans un graphe ouvre de nouvelles opportunités pour améliorer les performances et l'efficacité des systèmes dans de nombreux secteurs.

Partie 5 : Implémentation

Nous allons utiliser la bibliothèque PyTorch Geometric pour présenter un exemple pratique d'utilisation des réseaux de neurones graphiques (GNN) avec 1/4 du dataset Cora. Ensuite, nous allons préparer le dataset, définir le modèle GNN, et entraîner le modèle.

Étapes pour réaliser le cas pratique

1. **Installation des bibliothèques nécessaires**
2. **Chargement et préparation des données**
3. **Définition du modèle GNN**
4. **Entraînement et évaluation du modèle**
5. **Évaluation et c du Modèle**

1. Installation des bibliothèques nécessaires

Installation de PyTorch et PyTorch Geometric (voir annexe).

2. Chargement et Préparation des Données

Nous allons charger le dataset Cora fourni par PyTorch Geometric et utiliser 1/4 de ce dataset pour simplifier l'exemple (voir annexe).

3. Définition du Modèle GNN

Nous allons utiliser un simple réseau de neurones graphiques à base de Graph Convolutional Network (GCN) (voir annexe).

4. Entraînement du Modèle

Enfin, nous allons définir les fonctions d'entraînement et d'évaluation, puis entraîner le modèle (voir annexe).

5. Évaluation et prédiction du Modèle

Après l'entraînement, nous faisons des prédictions sur les données de test. Nous comparons les étiquettes prédites aux vraies étiquettes et affichons les résultats. Nous visualisons le graphe en coloriant les nœuds correctement et incorrectement prédits (voir annexe).

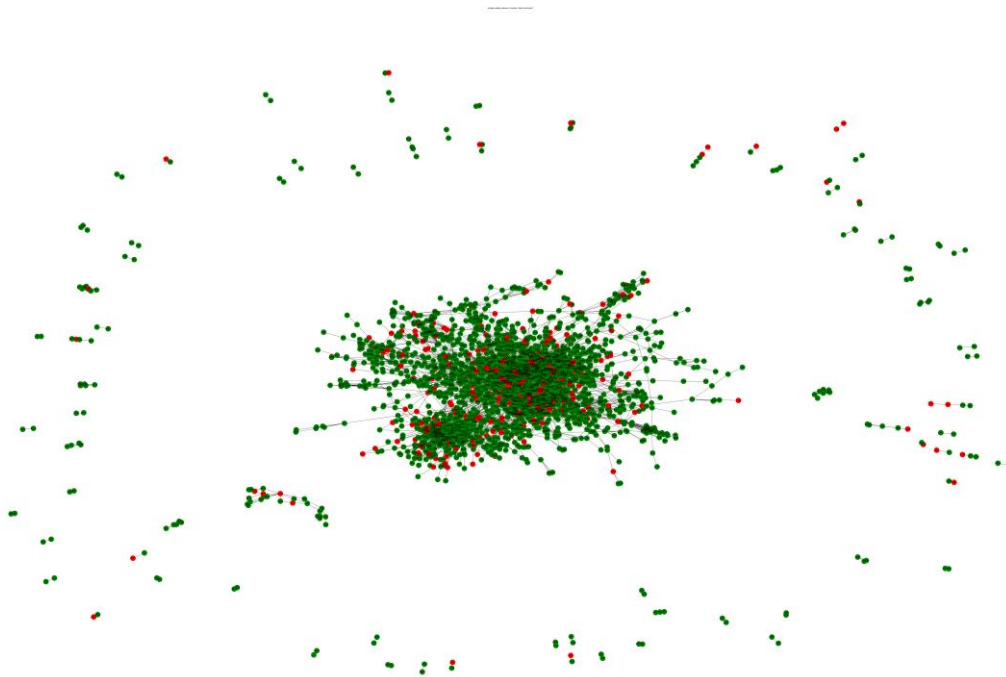


Figure 3 : visualisons le graphe en coloriant les nœuds correctement et incorrectement prédits.

Interprétation des résultats :

La figure représente un graphe de la base de données Cora, où les nœuds correctement classifiés par le modèle GNN sont colorés en vert et ceux incorrectement classifiés sont en rouge.

1. Distribution des prédictions correctes et incorrectes :

- La majorité des nœuds sont colorés en vert, ce qui indique que le modèle a correctement prédit les étiquettes pour ces nœuds.
- Les nœuds rouges, qui représentent les prédictions incorrectes, sont principalement dispersés dans le graphe, mais certains clusters de nœuds rouges sont visibles.

2. Densité du graphe :

- Le centre du graphe est très dense, avec de nombreuses connexions entre les nœuds. Cela pourrait indiquer une région du graphe avec une forte interconnexion et potentiellement des nœuds appartenant à des classes similaires.
- Les régions périphériques sont moins denses et montrent des nœuds plus isolés ou des petits clusters de nœuds.

3. Précision globale :

- Visuellement, la proportion de nœuds verts par rapport aux nœuds rouges suggère que le modèle a une bonne précision globale.

Référence :

«Évaluation des réseaux de neurones graphiques pour le lien Prédiction: Pièges actuels et nouvelles analyses d'analyse comparative» Juanhui Li¹ *, Harry Shomer¹ *, Haitao MAO¹, Shenglai Zeng¹, Yao MA², Neil Shah³, Jiliang Tang¹, Dawei Yin⁴, ¹ Michigan State University, ²Rensselaer Polytechnic

nsi-graphes, <https://info.blaisepascal.fr/nsi-graphes/>, consulté le 20 juin 2024.

graph-neural-networks-tout-savoir, <https://datascientest.com/graph-neural-networks-tout-savoir>, consulté le 21 juin 2024.

<https://graphsandnetworks.com/the-cora-dataset/>, consulté le 21 juin 2024.

CoraDataset, <https://github.com/mathematiger/CoraDataset/tree/main>, consulté le 22 juin 2024.

CoraDataset-main, <https://github.com/appistore/UQO/tree/main/CoraDataset-main>, consulté le 23 juin 2024.

Annexe :

Code source : <https://github.com/appistore/Graph-Neural-Networks>