DIALLO Ibrahima sambegou

28 janvier 2024

1 Partie 1 : Application des modèles GNN sur les données ogbn-proteins et ogbn-arxiv

Dans cette partie, nous avons exploré l'application de plusieurs modèles de GNN sur les jeux de données issus de OGB. L'objectif était de se familiariser avec les GNN et de comprendre les procédures d'évaluation.

1.1 ogbn-proteins : Protein-Protein Association Network

Pour les données ogbn-proteins, nous avons utilisé un RandomLoader pour entraîner le modèle. Après l'entraînement, nous avons appliqué son évaluateur pour évaluer le modèle, comme indiquer sur les site OGB. Les résultats ont été prometteurs et nous avons réussi à reproduire à peut prêt les résultats de la table 6 de l'article de référence.

Voici les meilleurs résultats obtenus lors nos experiences :

TABLE 1 – Résultats de ROC-AUC pour différents modèles

Méthode	ROC-AUC Entraînement	ROC-AUC Validation	ROC-AUC Test
GCN	0.6977	0.6963	0.5502
Node2Vec	0.5101	0.5008	0.5001
MLP	0.7894	0.7830	0.5430
GraphSAGE	0.8242	0.7763	0.7517

1.2 ogbn-arxiv : Paper Citation Network

Malheureusement, pour les données ogbn-arxiv, nous n'avons pas pu terminer nos expérimentations en raison de limitations matérielles. Notre ordinateur n'était pas assez puissant pour entraîner les modèles avec le loader. De plus, sans utiliser de loader, le modèle n'arrivait pas à converger. Par contre le modèle MLP de sklearn à étét toujour facile d'utilisation par rapport aux autres, nous avons pu obtenir des bons resultats avec.

2 Partie 2 : Application d'une méthode de GNN sur un jeu de données choisi

Nous avons utilisé les données du site IMDB pour la classification des critiques de films. Les résultats ont été encourageants. Tout d'abord pour faciliter le processing des données nous allons juste prendre les 10000 premiers commentaires et ensuite nous avons proceder comme suit :

- Calcul de la similarité : Nous avons calculé la similarité entre chaque paire de commentaires en utilisant la méthode TF-IDF pour convertir les commentaires en vecteurs, puis en calculant la similarité cosinus entre chaque paire de vecteurs.
- Construction du graphe : Nous avons construit un graphe où chaque nœud représente un commentaire et chaque arête représente une similarité supérieure à 0.6 entre deux commentaires.
- Préparation des données pour le GNN : Enfin, nous avons converti le graphe en une forme que torchgeometric peut utiliser. Nous avons converti la liste des arêtes en un tensor PyTorch et nous avons créé un objet Data pour notre graphe. Nous avons également créé un DataLoader pour gérer le chargement des données pendant l'entraînement.

Vous trouverz dans le notebook dédier l'esperimentation et les resultats

3 Conclusion

Ce projet nous a permis de nous familiariser avec les GNN et de comprendre les procédures d'évaluation et de construction de graphe (La manière dont on peut representer un problème sous forme de graphe). Bien que nous ayons rencontré des défis, notamment des limitations matérielles, nous avons néanmoins réussi à obtenir des résultats prometteurs dans nos expérimentations.