

## 1 Présentation générale

Les réseaux neuronaux graphiques (Graph Neural Networks; GNN) sont utilisés pour traiter des ensembles de données à grande échelle dans des domaines tels que les réseaux sociaux, l'analyse des citations et la biologie [1]. Cependant, leur coût de calcul augmente minimalement proportionnellement au nombre de arêtes dans le graphe, ce qui crée des limites en termes de scalabilité.

La sparsification des graphes, c'est-à-dire la réduction du nombre de arêtes tout en essayant de préserver les propriétés essentielles du graphe, offre une voie prometteuse pour améliorer la scalabilité des GNN. Bien qu'il existe de nombreux schémas de sparsification (échantillonnage aléatoire, sparsification spectrale [2], sparsification métrique [3], etc), leur effet sur les performances des GNN reste une question ouverte.

## 2 Objectifs

L'objectif de ce projet est de mener une étude comparative des méthodes de sparsification des graphes pour les GNN, en mettant l'accent sur le compromis entre performances et efficacité.

Plus précisément :

- Mise en œuvre de différentes méthodes de sparsification des graphes.

- Évaluer leur impact sur le temps d'entraînement, l'utilisation de la mémoire et la précision prédictive des GNN.

- Identifier les méthodes de sparsification qui offrent le meilleur équilibre entre rapidité d'entraînement et performances.

Les questions de recherche sont les suivantes:

- Quelles stratégies de sparsification sont les plus efficaces pour préserver les performances des GNN ?

- Comment la sparsification influence-t-elle l'efficacité de l'entraînement (temps d'exécution, performance prédictive) ?

- Différents datasets réagissent-ils différemment à la sparsification ?

Pour répondre à ces questions, l'étudiant analysera l'impact de différentes techniques de sparsification sur les performances de modèles de GNN de référence (tels que GCN, GraphSAGE ou GAT, généralement implémentés avec PyTorch Geometric ou Deep Graph Library), en s'appuyant sur des jeux de données de référence (Cora, Citeseer, PubMed, OGB).

## 3 Prérequis

Une bonne maîtrise de Python et des bibliothèques scientifiques courantes est indispensable.

Une première expérience avec des frameworks de deep learning (tels que PyTorch) et/ou des bibliothèques de graphes (comme igraph) est appréciée, mais n'est pas obligatoire.

Les deux tutoriels suivants sont une excellente introduction aux GNN :

- <https://distill.pub/2021/understanding-gnns/>

- <https://distill.pub/2021/gnn-intro/>

## Bibliography

[1] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in International Conference on Learning Representations, 2017.

[2] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," in Proceedings of the fortieth annual ACM symposium on Theory of computing, 2008, pp. 563–568.

[3] M. Drevet, C. Chuc, M. Grossglauser, and P. Thiran, "Why the metric backbone