

GRAPH NEURAL NETWORK(GNN)

Patrick Sorrel Mvoto Kongo

16 août 2023

PLAN DE TRAVAIL

INTRODUCTION
INTÉRÊT DE L'UTILISATION DES GNN

1 MACHINE LEARNING DES GRAPHS

1.1 OBJECTIF ET APPLICATIONS DES GNN

1.2 DOMAINES D'APPLICATIONS

2 PRÉSENTATIONS DES GNN

2.1 DÉFINITION ET CARACTÉRISTIQUES PRINCIPALES D'UNE

GNN

2.2 PRESENTATION DES DIFFERENTES ARCHITECTURES DE

GNN

3 PRINCIPE DE FONCTIONNEMENT D'UNE GNN

3.1 GRAPHE ET EXTRACTION DES DONNÉES

3.2 MESSAGE ET AGRÉGATION

3.3 CONNEXION ET DIFFÉRENTES COUCHEs DE GNN

3.4 EMBEDDING

3.5 ENTRAÎNEMENTS DES GNN

5 PERSPECTIVES DES GNN

CONCLUSION

INTRODUCTION

Graph Neural Networks (GNN) sont une famille de réseaux de neurones qui ont été développés pour traiter des données structurées sous forme de graphes. Contrairement aux réseaux de neurones traditionnels, qui sont conçus pour traiter des données tabulaires ou des séquences, les GNN sont capables de prendre en compte la structure complexe des graphes pour effectuer des tâches d'apprentissage automatique. Récemment, les recherches sur l'analyse des graphes, avec l'apprentissage automatique ont reçu de plus en plus d'attention en raison de la grande puissance expressive des graphes, c'est à dire que les graphes peuvent être utilisés comme dénotation d'un grand nombre de systèmes dans divers domaines, y compris les sciences, elle est utilisée pour des tâches tels que la classification des nœuds, la prédiction de liens et le regroupement. Dans la suite de cet exposé, nous allons explorer plus en détail les différents types de GNN, leurs caractéristiques, leurs applications et les techniques de propagation d'information.

1 INTÉRÊT DE L'UTILISATION DES GNN

1.1 MACHINE LEARNING DES GRAPHS

ML (apprentissage automatique) sous-ensemble d'applications courantes de l'IA dans lequel une machine a accès à des données dans le but d'extraire des connaissances à partir des données et d'effectuer une reconnaissance de motifs. Elle permet entre autre :

La détection de communautés et prédiction de liens .On peut profiter d'un graphe pour prédire les groupes de nœuds et les nouvelles arêtes potentielles, faire de l'analyse de réseaux sociaux Prédiction de graphes dynamiques (nœuds et arêtes de systèmes physiques) Génération de graphes (conception de médicaments)

1.2 OBJECTIF ET APPLICATION DES GNN

Challenges :

- Complexité topologique : pas de régularité locale (grille)
- Pas d'ordre évident pour les nœuds, pas de points de référence
- modélisation de la dynamique.

application :

- Recommandation d'images chez Pinterest
- Très large échelles
- Proposer des images similaires à une autre
- Features multiples
- Classification de nœuds
- Prédiction de liens
- Classification de graphes

- Clustering
- Génération de graphes
- Évolution de graphes

2 PRÉSENTATION DES GNN

2.1 DÉFINITION ET CARACTÉRISTIQUES PRINCIPALES D'UNE GNN

Les réseaux de neurones graphiques (GNN) sont une classe de modèles de réseaux de neurones conçus pour traiter des données structurées sous forme de graphes. pour effectuer des tâches telles que la classification de nœuds, la segmentation de graphes et la prédiction de liens. Les GNN ont connu une évolution rapide au cours des dernières années, avec de nouveaux modèles et de nouvelles architectures qui ont été proposés pour améliorer leur performance. Les réseaux de neurones convolutifs graphiques (GCN)[2016 : Kipf et Welling], les réseaux de propagation de message (MPNN)[2016 par Gilmer et al], les graphes de les graphes attentionnels (GAT)[Veličković et al. 2018], et les graphes de convolution récurrents (RCGN)[2018 par Kipf et Welling], les graphes de convolutions géométriques (GCG)[2019 par Bronstein et al].

2.2 PRÉSENTATION DES DIFFÉRENTES ARCHITECTURES DE GNN

Il existe plusieurs types de Graph Neural Networks (GNNs) avec des architectures différentes, chacun étant adapté à des tâches spécifiques. Voici une liste des types de GNN les plus couramment utilisés avec leurs architectures associées

- Graph Convolutional Networks (GCNs) : Les GCNs sont les GNN les plus populaires et sont utilisés pour la classification de nœuds. Ils sont basés sur une convolution de graphes, qui permet de calculer les nouvelles représentations de chaque nœud en agrégeant les informations de ses voisins.
- Graph Attention Networks (GATs) : Les GATs sont une extension des GCNs qui utilisent des mécanismes d'atténuation pour pondérer les informations des voisins. Cela permet de donner plus d'importance aux voisins les plus pertinents pour la tâche à résoudre.
- Graph Recurrent Neural Networks (Graph RNN) : Les Graph RNN Ils permettent de modéliser les relations temporelles sur des graphes dynamiques en utilisant une couche de rétroaction qui prend en compte les états précédents des nœuds et des arêtes.
- GraphSAGE :

GraphSAGE est une architecture de GNN qui agrège les informations des voisins de chaque nœud en utilisant une fonction d'agrégation basée sur les réseaux de neurones. Cette méthode permet de traiter des graphes de grande taille et est utilisée pour la classification de nœuds

3 PRINCIPE DE FONCTIONNEMENT D'UNE GNN

3.1 GRAPHE ET EXTRACTION DES DONNÉES

Un graphe est une représentation visuelle d'un ensemble d'objets et des relations qui existent entre ces objets.

Différentes familles de graphes : le graphe orienté, les arêtes sont dirigées. Un graphe pondéré, les arêtes sont associées à des poids positifs qui caractérisent l'intensité du lien. Pour terminer, deux graphes attribués, les nœuds et/ou les arêtes portent une information. La liste n'est pas exhaustive .

- extraction des données dans un graphe : si on considère un graphe est dit complet s'il comporte une arête (v_i, v_j) pour toute paire de sommets $(v_i, v_j) \in E^2$.
- poids (fréquence de communication)
- rang (collaborateur préféré, 2e préféré)
- type (ami, collaborateur, famille)
- signé ou non : confiance, amitié/inimitié
- propriétés liées au graphe : nombre de voisins en commun
- Matrice d'adjacence : $m_{ij} = 1$ si l'arête (v_i, v_j) existe, 0 sinon.

3.2 MESSAGE ET AGRÉGATION

- La phase de message passing consiste à propager l'information (ou le "message") entre les nœuds du graphe en utilisant les représentations des nœuds et des arêtes. Cette étape permet aux nœuds du graphe de communiquer et de partager de l'information.
- La phase d'agrégation consiste à agréger les informations mises à jour pour chaque nœud afin de calculer une représentation globale du graphe ?

3.3 CONNEXION ET DIFFÉRENTES COUCHES DE GNN

Dans les réseaux de neurones graphiques (GNN), les connexions font référence aux relations entre les nœuds d'un graphe sur lequel le modèle est appliqué. Contrairement aux réseaux de neurones traditionnels qui opèrent sur des données tabulaires ou des vecteurs, les GNN traitent des données structurées sous forme de graphes. Les connexions dans les GNN sont généralement représentées par des arêtes reliant les nœuds du graphe. Ces arêtes peuvent avoir différentes

caractéristiques, telles que des poids ou des attributs, qui capturent les relations entre les nœuds. Il existe plusieurs couches dans les GNN :

- Les couches de propagation de messages : les couches de propagation de messages sont responsables de calculer les nouvelles représentations de chaque nœud en agréant les informations des nœuds voisins ?
- Les couches de pooling de graphes : les couches de pooling de graphes permettent de réduire la taille du graphe en agréant les informations de plusieurs nœuds en un seul
- Les couches de prédiction : les couches de prédiction permettent de générer des prédictions à partir des représentations de chaque nœud.

3.4 EMBEDDING

- Embeddings = plongements Les embeddings sont utilisés pour pouvoir être traitées par la GNN, les données en entrée doivent être transformées en embeddings (représentations vectorielles) de chaque nœud et de chaque arête.
- Objectif : apprendre des représentations indépendantes des tâches
- On veut une représentation vectorielle des nœuds
- similarité d'embedding = similarité dans le réseau (proximité, symétrie)
- encoder l'information structurelle

3.5 ENTRAÎNEMENT DES GNN

L'entraînement d'un réseau de neurones graphiques (GNN) comprend généralement les étapes suivantes :

1. Préparation des données : Tout d'abord, vous devez préparer vos données sous forme de graphe. Cela implique de représenter vos entités (nœuds) et leurs relations (arêtes) à l'aide de structures de données appropriées.
2. Définition du modèle GNN : Ensuite, vous devez définir l'architecture de votre modèle GNN. Cela comprend la spécification des couches de propagation du graphe, des opérations d'agrégation, des fonctions d'activation, etc. Vous pouvez utiliser des bibliothèques comme PyTorch Geometric ou DGL pour créer votre modèle GNN.
3. Définition des fonctions de perte et d'optimisation : Vous devez définir une fonction de perte appropriée qui mesure l'écart entre les prédictions de votre modèle et les valeurs réelles. Ensuite, vous devez choisir un algorithme d'optimisation tel que la descente de gradient stochastique (SGD) ou l'optimisation d'Adam pour minimiser la fonction de perte.
4. Entraînement du modèle : À cette étape, vous devez alimenter vos données d'entraînement dans le modèle GNN. Vous itérez sur vos données en mini-lots (batches) et exécutez une passe d'entraînement à chaque itération. Pendant chaque passe, vous calculez les prédictions du modèle, calculez la perte, effectuez la rétropropagation du gradient et mettez à jour les poids du modèle à l'aide de l'algorithme d'optimisation choisi.

5. Évaluation du modèle : Une fois l'entraînement terminé, vous devez évaluer les performances de votre modèle sur des données de validation ou de test. Vous pouvez calculer des métriques telles que l'exactitude (accuracy), la précision, le rappel, le F1-score, etc., en fonction de la nature de votre tâche.
6. Réglage des hyperparamètres : Tout au long du processus d'entraînement, il est important de régler les hyperparamètres de votre modèle, tels que le taux d'apprentissage, le nombre de couches, le nombre de nœuds cachés, etc., pour obtenir de meilleures performances.
7. Itérations d'entraînement supplémentaires : Selon les résultats de l'évaluation, vous pouvez décider de répéter les étapes précédentes en modifiant certains aspects de votre modèle ou de votre pipeline d'entraînement. Cela peut inclure l'ajustement des hyperparamètres, l'augmentation des données, l'utilisation de techniques de régularisation, etc.

4 DOMAINES D'APPLICATION ET PERSPECTIVES DES GNN

4.1 DOMAINES APPLICATION DES GNN

les GNN ont plusieurs domaines d'applications parmi lesquelles on peut citer

1. Informatique et réseaux : Les GNN sont utilisés pour la classification des documents, la recommandation de produits et l'analyse de réseaux sociaux.
2. Biologie et médecine : Les GNN sont utilisés pour prédire la structure de protéines, identifier les gènes associés à des maladies et prédire les interactions entre les molécules.
3. Chimie : Les GNN sont utilisés pour prédire les propriétés des molécules, concevoir de nouveaux matériaux et accélérer la découverte de médicaments.
4. Physique : Les GNN sont utilisés pour prédire les propriétés des matériaux, modéliser la dynamique moléculaire et étudier les interactions entre les particules.
5. Finance : Les GNN sont utilisés pour la prévision de prix d'actifs, la détection de fraude et l'analyse de risque de crédit.

4.2 PERSPECTIVES DES GNN

Les réseaux de neurones graphiques (GNN) ont suscité un grand intérêt dans la communauté de la recherche ces dernières années et continuent de présenter de nombreuses perspectives prometteuses. Voici quelques-unes des perspectives clés des GNN :

1. Modélisation de graphes complexes : Les GNN offrent un cadre puissant pour modéliser des graphes complexes et capturer les relations entre les entités dans divers domaines. Ils peuvent être appliqués à des problèmes tels que la recommandation personnalisée, l'analyse de réseaux sociaux, la bioinformatique, la chimie, la vision par ordinateur et bien d'autres. Les GNN permettent d'exploiter la structure du graphe pour améliorer les prédictions et les représentations des entités.
2. Interprétabilité améliorée : Les GNN offrent la possibilité d'améliorer l'interprétabilité des modèles en fournissant des explications basées sur la structure du graphe. Les GNN peuvent identifier les nœuds et les connexions clés qui contribuent aux prédictions du modèle, ce qui facilite la compréhension des facteurs importants influençant les résultats.
3. Généralisation à de nouveaux graphes : Les recherches actuelles se concentrent sur la capacité des GNN à généraliser à de nouveaux graphes qui n'ont pas été vus pendant l'entraînement. Des techniques telles que le transfert de connaissances et l'adaptation de domaine visent à permettre aux GNN de s'adapter à de nouveaux graphes avec des structures différentes, tout en conservant les connaissances apprises sur des graphes similaires.
4. Modélisation de graphes dynamiques : Les GNN sont étendus pour prendre en charge des graphes dynamiques, où la structure et les attributs des nœuds évoluent au fil du temps. La modélisation des graphes dynamiques permet de capturer les changements temporels, les dépendances à long terme et d'autres aspects dynamiques des données.
5. Combinaison avec d'autres architectures : Les GNN peuvent être combinés avec d'autres architectures de réseau neuronal, telles que les réseaux de neurones récurrents (RNN) ou les réseaux de neurones convolutifs (CNN), pour bénéficier des avantages de chaque architecture. Ces combinaisons permettent de modéliser des données multidimensionnelles complexes et d'exploiter à la fois les relations locales et globales.
6. Développement de nouvelles architectures : Les chercheurs explorent continuellement de nouvelles architectures de GNN pour améliorer leurs performances. Des variantes telles que les graphes de neurones à attention (GAT), les graphes de neurones à propagation graphique (GraphSAGE) et les graphes de neurones à diffusion (Graph Convolutional Networks) sont développées pour capturer des informations plus fines sur la structure du graphe et améliorer les performances de prédiction.

Conclusion

les réseaux de neurones graphiques (GNN) ont émergé comme une méthode puissante pour le traitement de données structurées sous forme de graphes. Les GNN ont un certain nombre d'avantages, tels que la capacité à prendre en compte les relations entre les nœuds de manière explicite, la flexibilité pour traiter des graphes de taille variable, la capacité à généraliser à des graphes non observés

et une performance supérieure à d'autres modèles d'apprentissage automatique pour le traitement de données structurées sous forme de graphes. Cependant, les GNN ont également certaines limites, telles que la sensibilité à la topologie du graphe, la difficulté à gérer les attributs hétérogènes, le manque d'explicabilité et le besoin de données d'entraînement massives. Il est important de comprendre ces limites afin de pouvoir utiliser les GNN de manière appropriée. Les GNN ont montré des résultats prometteurs dans de nombreux domaines, tels que la biologie, la chimie, la physique et les réseaux sociaux. Ils ont le potentiel de transformer la manière dont nous traitons les données structurées sous forme de graphes et de conduire à de nouvelles découvertes dans ces domaines. Cependant, il est important de continuer à explorer les limites des GNN et de développer de nouvelles méthodes pour surmonter ces limites afin de maximiser leur potentiel.