الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة غرداية
Université de Ghardaïa
كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie
قسم الرياضيات والاعلام الي
Département Math et Informatique
مخبر الرياضيات والعلوم التطبيقية
Laboratoire des Mathématiques et Sciences Appliquées

N° d'enregistrement
/.../.../.../.../...

Mémoire de fin d'étude, en vue de l'obtention du diplôme

Master

Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Systèmes Intelligents pour l'Extraction de Connaissances

Thème

Découverte de communautés dans les réseaux complexes basée sur les CNNs

---

Haddaoui Alla et Krimat Rababe Roumaissa

Soutenue publiquement le 25/06/2023

Devant le jury composé de :

Youcef Mahdjoub	MAA	Univ.Ghardaia	Président
Asma Bouchekouf	MAA	Univ.Ghardaia	Examinateur
Attia Nehar	MCB	Univ. Z.A. Djelfa	Examinateur
Slimane Bellaouar	MCA	Univ.Ghardaia	Encadrant
Abdelfateh Bekkair	Doctorant	Univ.Ghardaia	Co-encadrant

Année universitaire 2022/2023

Remerciement

“

Nous tenons à remercier avant tout ***DIEU*** *le tout puissant qui nous a donné la foi et le courage pour réaliser ce travail.*

Tout d'abord, nous tenons à remercier sincèrement notre encadreur, ***Mr Slimane Bellaouar****, pour ses conseils, son expertise et sa disponibilité tout au long de ce processus. Ses précieux conseils et ses encouragements ont été d'une grande aide dans la réalisation de ce travail. Nous souhaitons également exprimer notre reconnaissance envers notre co-directeur,* ***Mr Abdelfateh Bekkair****, pour ses perspectives complémentaires et ses idées novatrices ont influencé nos réflexions et ont contribué à la qualité des résultats obtenus. Nous vous remercions pour son expertise et sa disponibilité tout au long de ce projet. Sa collaboration et son soutien ont été d'une grande valeur ajoutée pour nous, et nous sommes reconnaissants d'avoir pu bénéficier de son encadrement.*

Enfin, nous tenons à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail et nos chers professeurs qui nous ont épaulées et guidées durant notre cursus universitaire.

”

Dédicace

“

À mon cher père qui n'a pas pu voir mon travail. Votre départ a créé un vide immense dans ma vie, mais votre impact et votre héritage continuent de vivre en moi. Votre présence bienveillante a toujours été un phare dans ma vie, illuminant mon chemin et me donnant la force de surmonter les obstacles. Je suis fier d'être votre fille et je m'efforcerai de vivre ma vie en suivant les valeurs que vous m'avez transmises.

Je dédie ce travail à ma source de force dans ma vie, ma chère mère, le meilleur frère du monde, mon frère ***Aimen****, ma seconde moitié, ma chère sœur* ***Manel****, son mari et son cher fils.*

Je dédie également ce travail à toute ma famille ***Haddaoui****, la famille* ***Taouti*** *et la famille* ***Benmoulai****, avec beaucoup de respect et d'amour.*

Je souhaite exprimer ma reconnaissance à tous mes enseignants et professeurs qui m'ont conseillé et guidé.

Un grand merci à mes chers amis et collègues que j'ai eu le plaisir de connaître jusqu'à présent. Leur amour et leurs encouragements ont été précieux.

Je tiens à remercier tout particulièrement ma chère binôme ***Rababe*** *pour son entente, sa sympathie et ses efforts.*

Haddaoui Alla.

”

Dédicace

“

*Je dédie ce travail à mes **parents**, mes plus grands soutiens et sources d'inspiration. Votre amour inconditionnel, votre encouragement constant et vos sacrifices ont été les piliers de ma réussite académique. Votre soutien indéfectible et votre confiance en moi ont nourri ma détermination et m'ont permis d'atteindre mes objectifs. Votre amour et vos valeurs ont façonné la personne que je suis aujourd'hui.*

*Je dédie ce travail à mon âme sœur, **Maroua**, mes deux petits frères chéris, **Muhammad** et **Ibrahim**.*

Un merci spécial à tous mes professeurs qui m'ont encadré et ont été une source d'inspiration et de connaissances tout au long de ma carrière universitaire.

*À tous mes chers amis et à mon amie proche **Anfal**, Malgré nos chemins qui se sont séparés pendant six longues années, ton soutien inconditionnel a transcendé les distances et les épreuves. J'espère sincèrement avoir l'occasion de te revoir bientôt.*

*Un merci particulier à ma binôme **Alla** pour ses efforts, sa persévérance et son soutien inébranlable qui ont été d'une valeur inestimable tout au long de notre parcours commun.*

Krimat Rababe Roumaissa.

”

ملخص

اكتشاف التجمعات في الشبكات المعقدة مثل الشبكات الاجتماعية والشبكات البيولوجية هو مسألة هامة في تحليل الشبكات. تمثل التجمعات هياكل فرعية ذات أهمية تكشف عن تجمعات العناصر المماثلة أو المترابطة. الشبكات العصبية التلافيفية (CNN) تقدم نهجًا واعدًا لاكتشاف التجمعات. من أجل معالجة هذا الموضوع، أجرينا دراسة مقارنة بين نهج تصنيف العقد ونهج تصنيف الروابط باستخدام هياكل CNN. كان هدفنا الرئيسي تقييم أداء هذين النهجين باستخدام مقاييس مختلفة على مجموعات بيانات متنوعة. تظهر النتائج أن النهجين القائمين على CNN يقدمان أداءًا متقاربًا وجيدًا في اكتشاف التجمعات في الشبكات المعقدة. تهدف التوصيات المطروحة إلى تحسين فهم وتحليل التجمعات في الشبكات المعقدة عن طريق استغلال قدرات CNN. تحمل هذه الدراسة آثارًا هامة لتحسين طرق اكتشاف التجمعات في الشبكات المعقدة من خلال استغلال مزايا CNN.

كلمات مفتاحية : اكتشاف التجمعات، الشبكات المعقدة، الشبكات العصبية التلافيفية، تصنيف العقد، تصنيف الروابط.

Résumé

La découverte de communautés dans les réseaux complexes, tels que les réseaux sociaux et biologiques, est une problématique importante en analyse des réseaux. Les communautés représentent des sous-structures significatives révélant des regroupements d'entités similaires ou interconnectées. Les réseaux de neurones à convolution (CNN) offrent une approche prometteuse pour cette découverte. Afin d'aborder le sujet, nous avons mené une étude comparative entre l'approche de la classification de nœuds et celle de la classification d'arêtes se basant sur des architectures CNN. Notre objectif principal était d'évaluer les performances de ces approches en utilisant différentes mesures sur divers ensembles de données. Les résultats montrent que les deux approches à base de CNN offrent des performances de détection de communautés convergentes et bonnes dans les réseaux complexes. Les recommandations formulées visent à améliorer la compréhension et l'analyse des communautés dans les réseaux complexes en exploitant les capacités des CNN. Cette étude a des implications importantes pour améliorer les méthodes de découverte de communautés dans les réseaux complexes en capitalisant sur les avantages des CNN.

Mots clés : détection de communautés, réseaux complexes, CNN, classification de nœuds, classification d'arrêts.

Abstract

Communities detection in complex networks, such as social and biological networks, is an important issue in network analysis. Communities represent significant substructures that reveal clusters of similar or interconnected entities. Convolutional Neural Networks (CNNs) offer a promising approach for this detection. To address this topic, we conducted a comparative study between the node classification approach and the edge classification approach based on CNN architectures. Our main objective was to evaluate the performance of these approaches using different measures on various datasets. The results show that both CNN-based approaches offer convergent and good community detection performance in complex networks. The recommendations aim to improve the understanding and analysis of communities in complex networks by harnessing the capabilities of CNNs. This study has important implications for enhancing community discovery methods in complex networks by capitalizing on the advantages of CNNs.

Keywords : communities detection, complex networks, CNN, node classification, edge classification.

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Contexte

Dans notre monde moderne, une grande partie des données qui nous entourent se présentent sous forme de graphes ou de réseaux complexes tels que les réseaux sociaux, les réseaux biologiques, les réseaux cérébraux, les réseaux de transport, les communications, et bien d'autres types de réseaux. Avec l'expansion rapide d'internet, de vastes réseaux sont apparus, comprenant des millions de nœuds et d'arêtes qui partagent des caractéristiques générales ou des fonctions communes. Par exemple, les réseaux sociaux offrent aux individus la possibilité de communiquer et de partager des intérêts communs, générant ainsi une énorme quantité de données sur les relations et les interactions des utilisateurs. Ces graphes sont souvent dynamiques, évoluant au fil du temps avec l'ajout et la disparition continus de nœuds et d'arêtes. Cela a suscité un intérêt considérable de la part des chercheurs qui cherchent à analyser leur structure, à comprendre leur évolution et à découvrir des communautés. Ceci constitue une étape cruciale vers la compréhension des réseaux complexes dans le monde réel.

Cependant, l'absence d'une définition universellement acceptée de la communauté pose un défi majeur dans le domaine de la découverte des communautés. Cela a conduit à la proposition de nombreuses méthodes et algorithmes pour découvrir des communautés dans les réseaux complexes. Ces méthodes se divisent en plusieurs catégories, parmi lesquelles les plus importantes sont les méthodes hiérarchiques, telles que l'algorithme GN (Girvan & Newman, 2002), qui repose sur une division fréquente du réseau, les méthodes dynamiques, telles que l'algorithme de propagation d'étiquettes (LPA) (Raghavan et al., 2007), qui adopte un mécanisme de propagation des étiquettes pour sélectionner les communautés, l'algorithme InfoMap (Rosvall & Bergstrom, 2008), qui se base sur des marches aléatoires pour maximiser la pression d'information, l'algoritme Walktrap (Pons & Latapy, 2005) ainsi que les méthodes d'optimisation visant à maximiser les modules, comme l'algorithme de Louvain (Blondel et al., 2008), pour améliorer la précision de la détection des communautés. Ces exemples ne représentent qu'une fraction des nombreuses méthodes traditionnelles existantes pour la détection de communautés dans les réseaux complexes.

Cependant, la plupart de ces méthodes reposent sur l'apprentissage automatique traditionnel, ce qui les rend moins efficaces pour la détection de communautés. Elles peuvent présenter une convergence lente, une capacité de recherche limitée et nécessitent un calcul intensif, surtout pour les réseaux dynamiques à grande échelle. En conséquence, leurs résultats peuvent manquer de précision. C'est pourquoi notre motivation principale est

d'adopter l'apprentissage en profondeur à travers les deux approches de classification des nœuds et de classification des arêtes que nous allons comparer dans ce mémoire, en tant que méthode et solution pour découvrir des communautés dans des réseaux complexes. L'apprentissage en profondeur a connu une utilisation généralisée ces dernières années dans de nombreux domaines.

Notre choix de travailler sur les CNN est motivé par le fait qu'ils font partie de thèse de doctorat de notre co-encadrant Abdelfateh Bekkair. Les CNN offrent de nombreux avantages, notamment leur capacité à modéliser des transformations non linéaires et à apprendre automatiquement des caractéristiques pertinentes. De plus, leur structure en couches facilite leur adaptation aux tâches complexes. C'est cette combinaison d'avantages qui nous a conduit à sélectionner les CNN pour la découverte de communautés dans les réseaux complexes.

Problématique

Le problème que nous abordons est le suivant : Quelles sont les différences entre les approches de classification de nœuds et de classification d'arêtes sur CNN en termes de performances et d'informations fournies sur les communautés ? Est-ce que l'utilisation des réseaux de neurones à convolution (CNN) offre une solution prometteuse ?

Objectif

L'objectif fondamental de cette recherche est d'effectuer une analyse approfondie des différentes approches proposées pour la détection des communautés en utilisant les CNN. Notre démarche consistera à mener une étude comparative détaillée de ces approches afin de déterminer la plus performante parmi elles.

Structure du mémoire

Le mémoire est structuré en trois chapitres, décrits comme suit.

Chapitre 01 : Ce chapitre est consacré à la définition des concepts de base de la recherche et est divisé en plusieurs parties. La première partie aborde les concepts liés à la théorie des graphes. La deuxième partie traite le concept des réseaux complexes et présente les types les plus importants de ces réseaux. La troisième partie se concentre sur les concepts de base liés aux CNN, en décrivant les différentes couches qui composent ces réseaux neuronaux et en mettant en évidence les principaux domaines d'application des CNN. Enfin, la dernière partie aborde les concepts liés à la découverte des communautés dans les réseaux, en expliquant les notions de communauté et de détection de communautés.

Chapitre 02 : Ce chapitre examine l'état de l'art et se divise en deux parties distinctes. La première partie présente une revue des méthodes classiques de détection de

communauté. La deuxième partie aborde les approches basées sur les CNN pour la détection de communauté, en mettant en évidence les recherches récentes qui explorent l'utilisation des réseaux de neurones convolutifs dans la détection de communauté dans des réseaux complexes.

Chapitre 03 : Dans ce dernier chapitre, nous nous sommes penchés sur les détails de la mise en œuvre de notre étude comparative des deux approches, à base de noeuds et à base d'arêtes. Nous discutons également des résultats obtenus afin d'évaluer leurs performances.

Chapitre 1

Notion de base

1.1 Introduction

La théorie des graphes est une branche qui étudie les relations entre les objets. Elle est utilisée pour modéliser une grande variété de réseaux complexes. Les réseaux complexes, tels que les réseaux sociaux, les réseaux biologiques sont présents dans de nombreux domaines de la vie moderne. La compréhension de leur structure et de leur dynamique est essentielle pour de nombreuses applications, notamment pour la détection de structures communautaires. Les réseaux de neurones à convolution peuvent également être utilisés pour améliorer la détection de communautés en identifiant des motifs locaux dans les réseaux, ce qui permet de mieux comprendre leur structure et leur fonctionnement.

Dans ce chapitre nous allons aborder les concepts de base sur les réseaux complexes, nous ne pouvons pas parler sur des réseaux complexes sans parler sur la théorie des graphes et comment elle fournit une base solide pour modéliser, analyser les réseaux complexes et présenter les différents types de réseaux complexes. Par la suite nous présentons des principes et l'architecture des réseaux de convolution (CNN), ainsi que leur utilisation dans la détection de communautés.

1.2 Théorie des graphes

L'histoire de la théorie des graphes débute en 1736 quand Euler démontra qu'il était impossible de traverser chacun des sept ponts de la ville russe de konigsberg une fois exactement et de revenir au point de départ. Dans la Figure 1.1 suivante, les nœuds représentent les rives.

La théorie des graphes est une branche des mathématiques qui étudie les propriétés des graphes, c'est-à-dire des structures représentant des relations entre des objets. Les objets qui composent un graphe sont appelés nœuds ou sommets et les liens entre eux sont appelés relations, liens ou arêtes.

Elle permet de modéliser, d'analyser et pour résoudre un grand nombre de phénomènes et de situations dans différents domaines (Needham & Hodler, 2020).

La théorie des graphes a de nombreuses applications dans divers domaines, notam-

ment : les maps, les réseaux de communication et informatiques, les logiciels, le contenu des sites web, l'étude des circuits électriques, la sociologie et l'économie (Sedgewick & Wayne, 2011).

Fig. 1.1—Ponts de Königsberg (Müler, 2012).

1.2.1 Définitions

Les concepts fondamentaux de la théorie des graphes comprennent(Müler, 2012) :

Graphe permet de décrire un ensemble de nœuds et d'arêtes. Un graphe fini $G = (V,E)$ est défini par l'ensemble de nœuds $V = \{v_1, v_2, \ldots, v_n\}$ et par l'ensemble d'arêtes fini $E = \{e_1, e_2, \ldots, e_m\}$.

noeud est l'unité de base d'un graphe, il en représente une ressource.

Arête est une connexion entre deux noeuds. On parle également d'arc ou de lien.

Ordre d'un graphe est le nombre de noeuds de ce graphe.

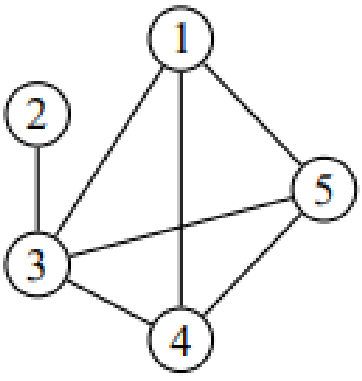


Fig. 1.2—Représentation du graphe G avec ordre 5 et 7 arêtes (Müler, 2012).

Chaîne est une liste ordonnée de noeuds reliés par des arêtes, tandis qu'un **chemin** est une chaîne dans laquelle chaque noeud n'apparaît qu'une seule fois.

Cycle est une chaîne fermée simple. tandis qu'un **Circuit** est un chemin dont les noeuds de départ et de fin sont les mêmes.

Graphe orienté est un graphe dont toutes les arêtes sont orientées (un sens aux arêtes). En revanche **Graphe non orienté** est un graphe dans lequel toutes les arêtes ne sont pas orientées.

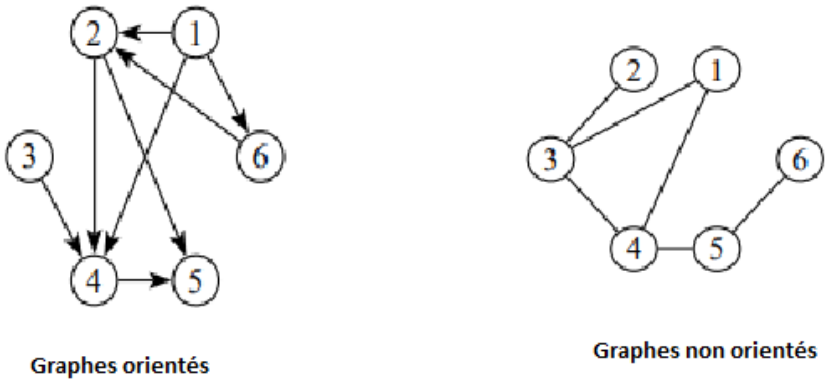


FIG. 1.3—Représentation d'un graphe orienté et non orienté (MÜLER, 2012).

Walk dans un graphe est une suite de sommets reliés par des arcs. Il peut s'agir d'un chemin simple, d'un chemin qui passe plusieurs fois par le même sommet, d'un cycle ou d'une boucle. La longueur d'un walk est définie comme le nombre de noeuds visités.

Arbre est un graphe connexe sans cycle, tandis qu'une arborescence est un arbre avec un sommet distingué, appelé racine.

forêt est un graphe sans cycle mais non connexe. Vous pouvez voir une représentation dans la Figure 1.4 ci-dessus.

FIG. 1.4—Représentation d'un Arbre et d'une Forêt (MÜLER, 2012).

1.2.2 Types des graphes

Voici quelques exemples de types de graphes :

Sous graphe

Les sous-graphes ne considèrent que certains des noeuds et ses arête.

Graphe simple

Un graphe est simple si au plus une arête relie deux noeuds et s'il n'y a pas de boucle sur un noeud.

Graphe connexe

Un graphe est connexe s'il est possible, à partir de n'importe quel noeud, de rejoindre tous les autres en suivant les arêtes.

Graphe complet

Un graphe complet est un graphe dans lequel chaque sommet du graphe est directement connecté à tous les autres noeuds.

Multigraphe

Un multigraphe est un graphe qui a une arête reliant un noeud à lui-même, ou qui a plusieurs arêtes reliant les deux mêmes noeuds.

Graphe biparti

Un graphe est bipartite si ses noeuds peuvent être divisés en deux groupes de sorte que toutes les arêtes du graphe se connectent à un noeud d'un autre ensemble dans un ensemble.

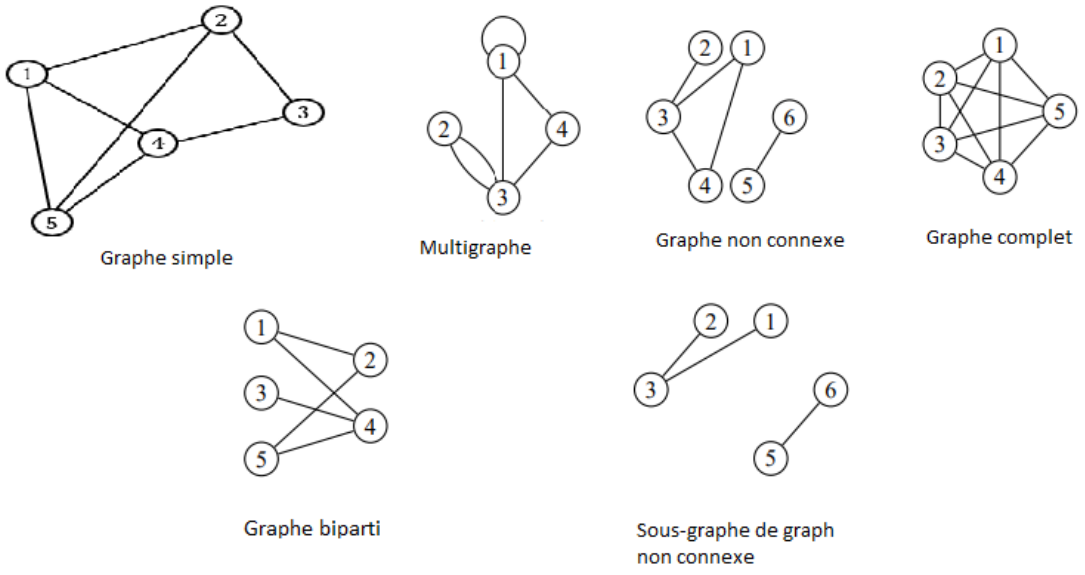


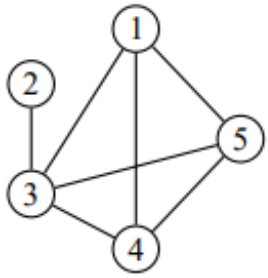
Fig. 1.5—Représentation des types de graphes (Müler, 2012).

1.2.3 Représentation des graphes

Il est possible de représenter un graphe de différentes façons, mais les deux représentations les plus courantes sont la liste d'adjacences et la matrice d'adjacences.

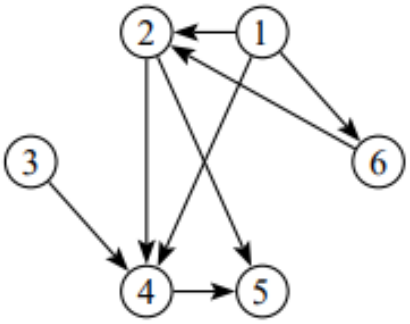
1.2.3.1 Listes d'adjacences

Les listes adjacentes sont représentées en donnant une liste des noeuds adjacents à chacun de ses noeuds dans un graphe les Figures 1.6 et 1.7 illustrent les listes adjacentes d'un graphe non orienté et orienté. Chaque noeud du graphe est associé à une liste de ses noeuds adjacents, c'est-à-dire les noeuds qui sont directement connectés à lui par une arête (Müler, 2012).



1: 3,4,5
2: 3
3: 1,2,4,5
4: 1,3,5
5: 1,3,4

Fig. 1.6—Liste adjacente d'un graphe non orienté (Müler, 2012).



1 : 2, 4, 6
2 : 4, 5
3 : 4
4 : 5
5 : –
6 : 2

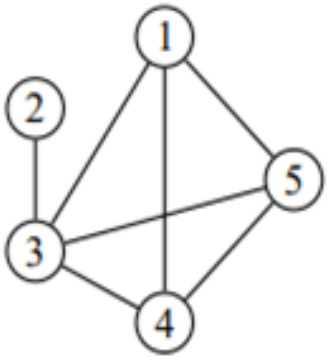
Fig. 1.7—Liste adjacente d'un graphe orienté (Müler, 2012).

1.2.3.2 Matrice d'adjacences

La Matrice d'adjacence d'un graphe $G = (V,E)$ est égale à la matrice $M = (m_{ij})$ de dimension $n * n$ où (i,j) désigne l'intersection de la ligne i et de la colonne j (Bretto et al., 2012). Chaque élément m_{ij} de la matrice représente la relation d'adjacence entre les noeuds i et j telle que :

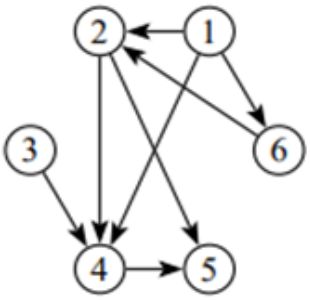
$$m_{ij} \begin{cases} 1 & \text{si } (i,j) \in E \\ 0 & \text{sinon} \end{cases}$$

les Figures ci-dessous 1.8 et 1.9 illustrent les matrices adjacentes d'un graphe non orienté et orienté.



$$M = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

FIG. 1.8—Matrice adjacence d'un graphe non orienté.



$$M = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

FIG. 1.9—Matrice adjacence d'un graphe orienté.

1.2.4 Parcours de graphes

Un parcours de graphe est un chemin qui suit les arcs d'un graphe pour relier des noeuds. Le but d'un parcours de graphe est de trouver tous les noeuds connectés à un noeud source donné et un chemin entre deux noeuds, la visite de tous les nœuds d'un graphe ou la détection de cycles ou de boucles dans un graphe. Il existe plusieurs algorithmes pour effectuer des parcours de graphes, par exemple :

1.2.4.1 Parcours en largeur (Breadth-First Search)

Le parcours en largeur est un algorithme de recherche utilisé pour explorer un graphe. Son principe est de visiter tous les voisins d'un nœud avant de passer au nœud suivant. Pour cela, on utilise une file d'attente pour se souvenir des nœuds qui doivent être visités.

Pour implémenter le parcours en largeur, on utilisera un tableau de booléens indexés par les sommets du graphe et initialisé à faux en plus une file d'attente de noeuds F et la file d'attente est initialisée à vide. Le parcours de G à partir d'un noeud s est réalisé par

l'algorithme 1 (Cohen, 2006).

Algorithme 1 : Parcours en largeur BFS

```
Data : (G : graphe, s : noeud)
Result : Marquage des noeuds atteignables depuis le noeud s
F ← File d'attente vide ;
marque[s] ← vrai ;
F.enfiler(s) ;
while F ≠ Vide do
    u ← F.defiler() ;
    for voisin v de u do
        if visite[v] = faux then
            visite[v] ← vrai ;
            F.enfiler(v) ;
        end
    end
end
```

Complexité

Soit G un graphe de n noeuds et m arêtes. Chaque noeud est enfilé au plus une fois. A chaque noeud visité, tous les noeuds adjacents sont testés pour le tableau marque ce qui se fait globalement en un temps proportionnel à m. L'algorithme demande donc un temps $\mathbf{O(n)}$ pour la boucle "tant que" et un temps $\mathbf{O(m)}$ pour les marques. Donc le temps de calcul est de l'ordre de $\mathbf{O}(\max(\mathbf{n},\mathbf{m}))$.

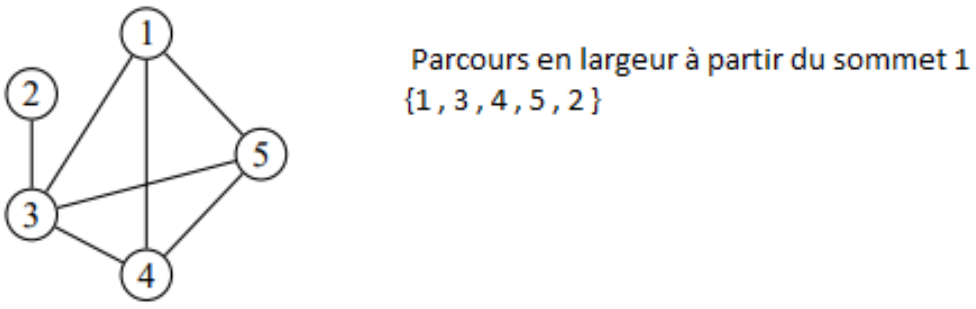


Fig. 1.10—Exemple de parcours en largeur BFS.

1.2.4.2 Parcours en profondeur (Depth-First Search)

Le principe du parcours en profondeur est de visiter le plus loin possible à partir d'un nœud donné, en marquant les nœuds visités au fur et à mesure.

Pour l'implémenter,Il s'agit essentiellement d'un processus récursif, nous allons utiliser un tableau de booléens marque indexé par les noeuds du graphe G. Ce tableau est initialisé à faux. Le parcours de G à partir d'un noeud v est réalisé par l'algorithme 2 (Cohen, 2006).

Algorithme 2 : Parcours en profondeur DFS

```
Procedure ParcoursProfondeur(G : graphe, v : noeud)
marque[v] ← vrai ;
for chaque noeud u adjacent à v do
    if marque[u] = faux then
        ParcoursProfondeur(G, u) ;
    end
end
```

Complexité

Soit G d'ordre n à m arêtes. Chaque noeud est paramètre de la procédure une et une seule fois donc il y a n appels récursifs. A chaque sommet visité, tous les sommets adjacents à v sont testés pour le tableau marque ce qui se fait globalement en un temps proportionnel à m. L'algorithme demande donc un temps $\mathbf{O}(\mathbf{n})$ pour les appels et un temps $\mathbf{O}(\mathbf{m})$ pour les marques. Donc le temps de calcul est de l'ordre de $\mathbf{O}(\max(\mathbf{n},\mathbf{m}))$.

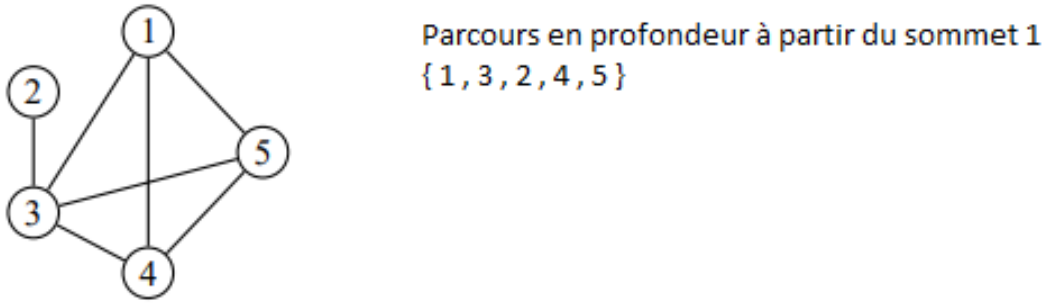


Fig. 1.11—Exemple de parcours en profondeur DFS.

1.2.5 Graphes et Réseaux

Pour comprendre un système complexe, il faut d'abord savoir comment ses composants interagissent les uns avec les autres. Autrement dit, on a besoin d'une carte de son schéma comme les relations entre des objets. En effet, on peut avoir trois réseaux assez différents ont exactement la même représentation graphe (Barabási, 2013).

La Figure 1.12 illustre différents réseaux modélisés dans le même graphe. Pour le premier graphe représente un ensemble des routeurs (ordinateurs spécialisés) connectés les

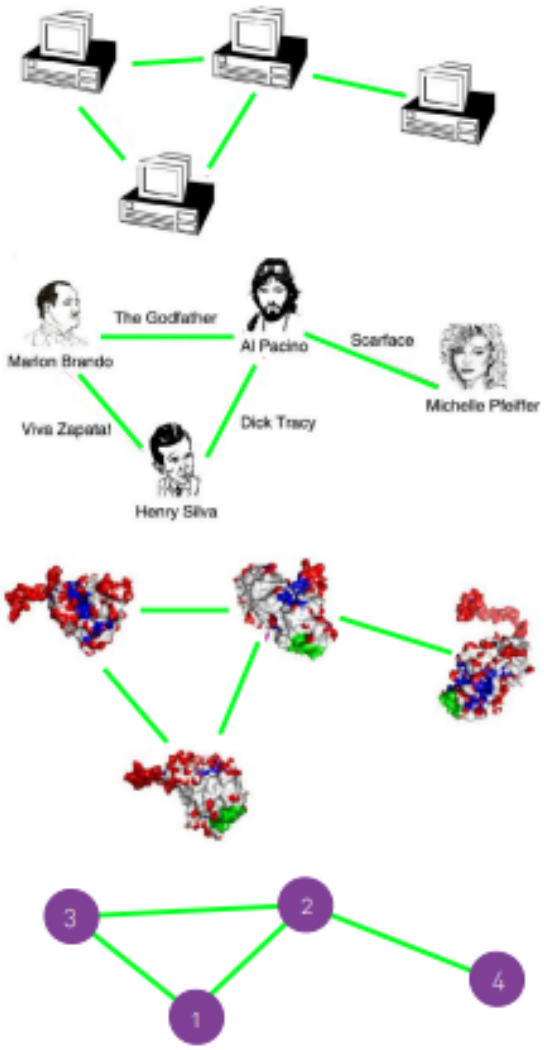


Fig. 1.12—Différents réseaux dans même graphe (Barabási, 2013)

uns aux autres. Les nœuds représentent les routeurs et les arcs représentent les connexions physiques entre eux. Le deuxième graphe est un réseau d'acteurs hollywoodiens représente les relations professionnelles entre les acteurs du cinéma hollywoodien. Les nœuds représentent les acteurs et les arcs représentent les relations professionnelles, s'ils ont joué dans le même film. Le troisième graphe est un graphe d'interaction protéine-protéine. Ce type de graphe représente les relations fonctionnelles entre les protéines en utilisant des arcs pour connecter les noeuds qui représentent les protéines individuelles. Les arcs reflètent des preuves expérimentales de la capacité des protéines à se lier l'une à l'autre dans la cellule, ce qui indique une interaction entre les protéines.

1.3 Types de réseaux complexes

Selon la théorie des graphes, un réseau complexe est un réseau composé d'ensemble de noeuds qui représentent des objets reliés entre eux par des liens. Le réseau complexe se

caractérise par une structure complexe et irrégulière, car il n'est pas possible de contrôler et de connaître ni le nombre de nœuds ni le nombre de liens de ce réseau (Salim, 2011).

On peut considérer qu'un réseau complexe est un système complexe constitué d'un ensemble d'objets qui forment des groupes interconnectés par interaction les unes avec les autres, ces interactions déterminent si ce système est complexe ou non et non le nombre d'objets, par exemple si le nombre d'objets est faible mais l'interaction entre eux est forte, dans ce cas elle forme inévitablement un système complexe (Dao, 2018).

L'étude des réseaux complexes a fait l'objet d'une grande attention de la part de la communauté scientifique, car ils représentent de manière purement abstraite les relations qui existent dans une gamme de systèmes biologiques et technologiques dans le monde réel, de plus, ils peuvent être utilisés pour un large éventail d'applications dans de nombreux domaines comme la découverte de communautés. Cette description s'applique à une variété de systèmes dans notre société d'aujourd'hui, tel que : les réseaux sociaux, les réseaux Web, les réseaux biologiques (Salim, 2011).

Dans cette section, nous nous intéresserons à deux types de réseaux complexes : les réseaux sociaux et les réseaux biologiques.

1.3.1 Réseaux sociaux

Ces dernières années, les réseaux sociaux ont occupé une part considérable de la vie de nombreuses personnes. Ils sont devenus célèbres dans le domaine de la recherche et ont rencontré un grand succès. De plus en plus utilisés par des personnes de différents domaines, les réseaux sociaux jouissent d'une grande popularité dans de nombreux pays. Ils permettent à des individus d'un système donné d'interagir en formant des communautés et de partager leurs souvenirs, expériences, sentiments et intérêts. (Figure1.13)(Peacock, 2010).

Un réseau social est un graphe composé de nœuds qui sont des personnes ou des organisations et de liens qui reflètent des relations sociales comme partager une famille, échanger des messages, avoir des intérêts similaires, etc. Ces dernières années ont vu beaucoup d'activité dans l'étude des réseaux sociaux, et les approches automatiques permettent maintenant d'examiner les attributs et les données de très vastes réseaux, tels que ceux constitués de pages internet ou tous les appels téléphoniques entrants à un fournisseur de télécommunications (Viennet, 2009).

1.3.1.1 Concepts fondamentaux en réseau sociaux

Nous allons prendre en connaissance de concepts de densité, centralité, proximité et coefficient de regroupement qui sont utilisés dans l'analyse des réseaux sociaux.

Densité du réseau est une mesure importante qui permet de connaître le niveau de connectivité dans un réseau. Il est défini comme le rapport entre le nombre d'arêtes dans le réseau et le nombre maximum possible d'arêtes. Ce rapport est compris entre 0 et 1, plus il est proche de 1, la densité du réseau est élevée, et plus il est proche de 0, la densité du réseau est faible et la connexion entre ses nœuds est mauvaise (Tabassum et al., 2018).

FIG. 1.13—Réseau social d'amitié (NORMAN, 2016).

Centralité est une mesure utilisée dans les réseaux pour évaluer la connectivité et l'influence des acteurs. Elle permet de distinguer les nœuds importants des autres dans le réseau. Les mesures de centralité, telles que la centralité de degré, l'intermédiarité et la proximité, sont utilisées dans l'analyse des réseaux sociaux pour étudier des phénomènes tels que les relations sociales, la propagation des maladies ou des informations (DENNY, 2014).

Proximité dans les réseaux sociaux est une mesure importante pour comprendre la rapidité des communications et des interactions entre les individus, en évaluant la distance la plus courte entre eux. Les plus courts chemins sont une méthode couramment utilisée pour calculer la proximité dans un réseau social, car ils permettent de déterminer le temps le plus court pour atteindre tous les nœuds à partir d'un nœud spécifique (TABASSUM et al., 2018).

Coefficient de regroupement est une mesure utilisée dans l'analyse des réseaux sociaux. Il permet de mesurer le degré de communication entre les individus du réseau en évaluant la tendance des nœuds à former des communautés bien connectées. Cette mesure détermine également l'efficacité du réseau dans la diffusion de l'information. Plus le coefficient de regroupement est élevé, meilleur est le réseau et plus la diffusion de l'information est rapide. En revanche, si le réseau a un coefficient de regroupement faible, cela indique que le réseau est fragmenté et que la diffusion de l'information est lente (DENNY, 2014).

1.3.1.2 Variétés de réseaux sociaux

Il existe de nombreux exemples de réseaux sociaux ici, énumérons quelques-uns des autres exemples de réseaux qui présentent également localité des relations.

1.3.1.3 Réseaux de communication

Dans ce cas, les nœuds remplacent les numéros de téléphone, qui sont en fait des personnes. Si un appel a été effectué entre deux nœuds dans un laps de temps prédéterminé, il y a un avantage entre eux. Les arrêts peuvent être pondérés en fonction du nombre d'appels effectués entre les nœuds en question (LESKOVEC et al., 2020).

1.3.1.4 Réseaux de messagerie

les nœuds représentent des adresses e-mail, qui correspondent à des individus. Les arêtes indiquent l'existence d'au moins un e-mail envoyé entre deux adresses dans une direction donnée. Si les e-mails sont échangés dans les deux sens, une seule arête est placée pour représenter cette relation bilatérale (Leskovec et al., 2020).

1.3.1.5 Réseaux de collaboration

Les personnes qui ont écrit des articles de recherche sont représentées par des nœuds. Deux personnes qui ont publié conjointement un ou plusieurs articles ont un avantage, nous pouvons également attribuer des étiquettes aux arêtes en fonction de la quantité de publications partagées. Les auteurs qui s'intéressent à un sujet particulier constituent les communautés de ce réseau (Leskovec et al., 2020).

1.3.2 Réseaux biologiques

Les graphes permettent de modéliser facilement les relations entre les objets, c'est pourquoi la biologie s'appuie sur les réseaux pour représenter les interactions entre les éléments biologiques. Les réseaux biologiques sont parmi les réseaux complexes les plus répandus avec une large portée, et ils se caractérisent par des mécanismes complètement différents des autres réseaux en raison de leur dépendance aux techniques expérimentales en biologie (Figure1.14) (Dao, 2018).

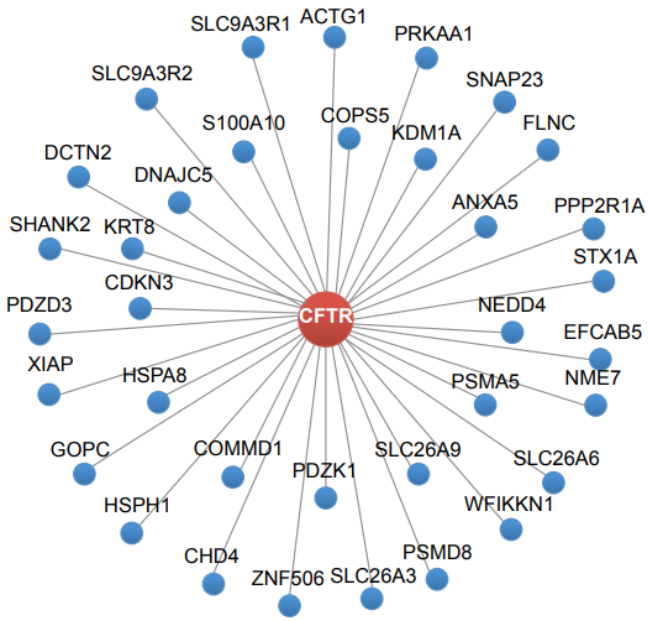


Fig. 1.14—Réseau d'interaction protéine-protéine (Brouard, 2013).

1.3.2.1 Types de réseaux biologiques

Parmi les différents réseaux d'interactions entre les molécules, nous nous intéressons plus particulièrement à :

1.3.2.2 Réseaux métaboliques

Les réseaux métaboliques sont des réseaux dirigés où chaque nœud représente un métabolite (une molécule) et chaque arête une réaction métabolique. Une réaction métabolique est une transformation chimique catalysée par voie enzymatique de composés chimiques ou de métabolites (également appelés réactifs) en d'autres substances (également appelées produits). Le résultat d'une réaction métabolique est souvent un réactif d'une autre réaction métabolique, démontrant l'interaction entre les réactions métaboliques (Zitnik, 2016). Le réseau peut être représenté de plusieurs manières équivalentes :

Par un graphe composé : Où les sommets représentent les métabolites et l'arête (orienté) entre le substrat et le produit est marqué avec l'enzyme qui catalyse la réaction (Sikora, 2011).

Par un graphe bipartite : Qui définit deux types de sommets différents, l'un pour les enzymes et l'autre pour les métabolites. Par conséquent, une réaction consiste en deux arcs qui voyagent du substrat au produit tout en passant par l'enzyme, Le graphique est bipartite tel que défini (Sikora, 2011).

Par un graphe des réactions : Lorsque deux réactions enzymes sont reliées par un arc , cela indique que un certain métabolite est produit par l'un et consommé par l'autre (Sikora, 2011).

1.3.2.3 Réseaux d'interactions entre les protéines

Les réseaux de protéines sont des réseaux où les nœuds représentent des protéines et les arêtes représentent les interactions entre ces protéines. Les protéines peuvent interagir les unes avec les autres pour former des complexes protéiques ou pour activer certaines fonctions. Ces interactions sont essentielles pour de nombreuses fonctions biologiques et peuvent résulter d'informations provenant de l'extérieur de la cellule. Par exemple, l'insuline, une protéine transportée dans le sang, se lie à une protéine membranaire de la cellule cible pour déclencher une réponse biologique (Zitnik, 2016).

1.3.2.4 Réseaux de régulation de gènes

Les réseaux de régulation des gènes représentent la façon dont les gènes sont activés dans une cellule. Les gènes peuvent contrôler le processus de transcription, produisant ainsi de l'ARN à partir de l'ADN, à la fois de manière inhibitrice et active. Certains gènes contiennent des instructions pour produire des protéines activatrices qui stimulent la transcription d'autres gènes. Ces réseaux de régulation des gènes coordonnent la réponse d'une cellule à des stimuli internes. Les nœuds du réseau représentent les gènes et les arêtes dirigées représentent les interactions régulatrices, telles que la liaison d'un facteur

de transcription à un gène cible (Zitnik, 2016).

1.4 Réseaux de neurones convolutifs (CNN)

Le concept de réseau neuronal a émergé dans les années 1940-1950, résultant d'une simplification des connaissances neurobiologiques de l'époque et de l'utilisation émergente de l'ordinateur en tant qu'outil d'exploration.

Les réseaux de neurones sont des modèles d'apprentissage automatique dont la conception est très schématiquements inspirés du fonctionnement de vrais neurones (humains ou non). Les réseaux de neurones sont constitués de couches de neurones interconnectés qui apprennent à partir de données pour effectuer des tâches spécifiques. Les réseaux de neurones peuvent être entraînés à l'aide de techniques telles que la rétropropagation du gradient et peuvent être de types variés tels que les réseaux de neurones convolutionnels (CNN), les réseaux de neurones récurrents (RNN) (Dumolard, 1994).

1.4.1 Convolution

La convolution est une opération mathématique qui consiste à prendre deux fonctions et à en créer une troisième, qui représente la façon dont la forme de l'une des fonctions est modifiée par l'autre au fil du temps, les fonctions peuvent être remplacées par des signaux, des images ou d'autres types de données. Le produit de convolution est défini comme suit soient deux fonctions f(t) et g(t) définies sur un intervalle ou un ensemble quelconque, le produit de convolution est une intégrale qui exprime le degré de superposition de la fonction g lorsqu'on la fait glisser sur la fonction f (Gonzalez et al., 2009).

$f * g(t) \approx \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$.

1.4.2 Concepts de base CNN

Les réseaux de neurones convolutifs (Convolution Neural Network, CNN) sont inspirés biologiquement par le cortex visuel, qui comprend des zones de petites cellules sensibles à des zones spécifiques du champ visuel. Cette idée a été développée à partir d'une expérience menée par Hubel et Wiesel en 1962, qui ont montré que certains neurones individuels ne sont activés que par des contours dans une orientation particulière. Par exemple, certains neurones étaient activés lorsqu'ils étaient exposés à des bords verticaux, tandis que d'autres étaient activés lorsqu'ils étaient représentés par des bords horizontaux ou diagonaux. Hubel et Wiesel ont découvert que tous ces neurones sont organisés en une structure colonnaire et qu'ils sont responsables de la perception visuelle. Les machines utilisent également cette idée d'utiliser des composants spécialisés pour des tâches spécifiques, comme les neurones du cortex visuel à la recherche de caractéristiques spécifiques, ce qui constitue la base des CNN.

Les réseaux de neurones convolutifs (CNN) sont une catégorie de réseaux de neurones qui se sont avérés particulièrement performants dans la résolution de problèmes en vision

par ordinateur. Les CNN utilisent des couches de convolution pour extraire les caractéristiques des images en préservant leur relation spatiale. Ces caractéristiques sont ensuite traitées par des couches complètement connectées pour effectuer des classifications ou des prédictions. Les CNN ont été employés dans divers domaines tels que la reconnaissance d'images, la segmentation d'images, la détection d'objets, la classification de textes, et bien plus encore (Zeiler & Fergus, 2013). La Figure 1.15 présente la structure d'un CNN :

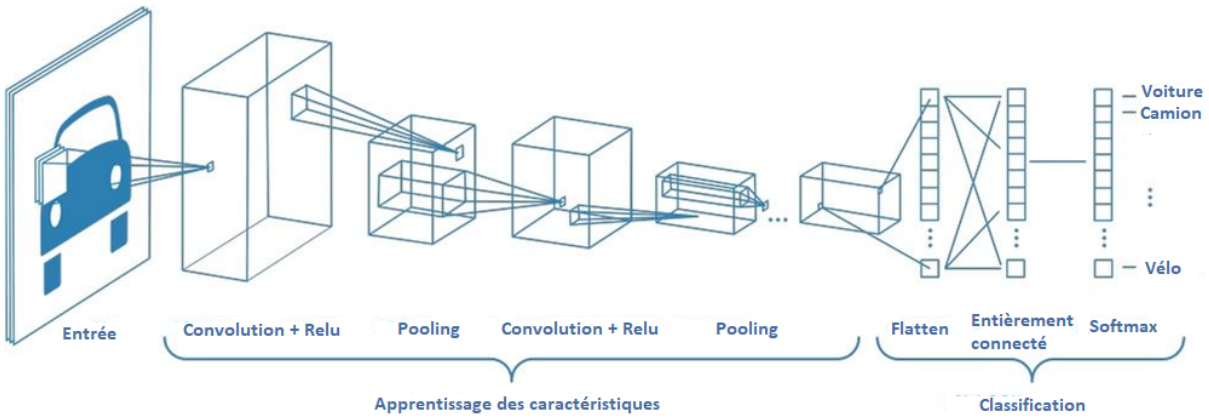


Fig. 1.15—Architecture CNN (Kulkarni & Shivananda, 2019).

1.4.3 Couches de CNN

Les réseaux de neurones convolutif se compose de trois couches divisées en deux étapes :

la première étape est constituée de (couche de convolution, couche pooling), qui représente l'étape d'extraction des caractéristiques et la deuxième étape est constituée de (couches entièrement connectées) qui représente l'étape de classification.

1.4.3.1 Couche de convolution (CONV)

La convolution est l'une des couches les plus importantes de toute l'architecture CNN, l'objectif principal de convolution est d'extraire des caractéristiques de l'image d'entrée à l'aide de filtres. Il se compose d'un ensemble de filtres appelés les filtres de convolution. Chaque filtre est responsable de l'extraction d'information spécifique, de sorte que chaque fois que nous répétons le processus de filtrage, le résultat sera de nouvelle caractéristique (Ghosh et al., 2020).

Le filtre est une matrice composée de valeurs, chaque valeur est appelée un poids de filtre, ces poids sont définis de sorte que ce filtre soit prêt à extraire des caractéristiques

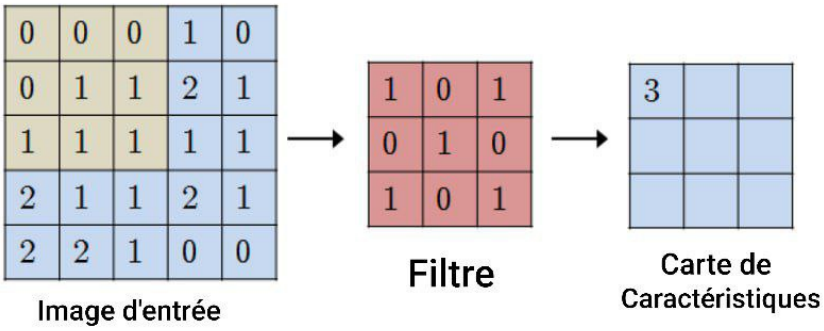


FIG. 1.16—Opération de convolution (AHAMED et al., 2020).

significatives. La taille de filtre $n*n$ avec $n>2$ et n doit être impaire (GHOSH et al., 2020).

Dans le CNN, l'image d'entrée peut être en niveau de gris ou multicanale (en couleur), et ici la différence est que la même information est représentée sur trois plans (GHOSH et al., 2020). Pour mieux comprendre l'opération de convolution on va faire l'exemple dans la Figure 1.16.

Nous avons une images de $5*5$ pixels représentée sous la forme d'une matrice et un filtre de $3*3$ pixels, et pour commencer l'opération de convolution nous prenons le filtre et le déplaçons sur toute l'image en partant du coin supérieur gauche de la matrice où il y a une correspondance avec le filtre, puis on calcule le produit scalaire. Nous répétons le processus horizontalement et verticalement sur la matrice, et l'image de sortie est le résultat de la convolution et s'appelle la carte de caractéristiques.

1.4.3.2 Fonctions d'activation

Les fonctions d'activation sont principalement utilisées dans les réseaux neuronaux, car elles rendent le réseau plus dynamique et capable d'apprendre des choses complexes et difficiles avec une grande précision, en plus de contrôler la sortie du réseau neuronal (SHARMA et al., 2020). Les fonctions d'activation convertissent les modèles linéaires en modèles non linéaires, car les modèles linéaires sont imprécis et fonctionnent mal (NWANKPA et al., 2018). Il existe de nombreuses fonctions d'activation (Figure 1.17) telles que :

Sigmoid : une fonction non linéaire qui est utilisée lorsque la sortie nécessite une prédiction, c'est-à-dire qu'elle est basée sur la probabilité, comme la classification binaire. Son résultat est toujours compris entre 0 et 1 (NWANKPA et al., 2018).

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}.$$

Tanh : Une fonction non linéaire similaire à la fonction sigmoïde, à la différence que le résultat est confiné entre -1 et 1 (SHARMA et al., 2020).

$$\text{Tanh}(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}.$$

SoftMax : Une fonction utilisée dans les problèmes de classification multiclasse, où elle calcule les probabilités de chaque classe sur toutes les classes possibles et utilise ces probabilités pour déterminer la classe cible qui prend la probabilité la plus élevée (Nwankpa et al., 2018).

$$\text{SoftMax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}, \quad i = 1, \ldots, K.$$

Parmi les fonctions d'activation les plus utilisées dans CNN est RELU.

RELU (Rectified Linear Unit) : Malgré sa simplicité, cela fonctionne bien dans la pratique, il accélère l'entraînement en activant les couches cachées et en convertissant les valeurs négatives en zéros (Ghosh et al., 2020).

ReLU est représenté mathématiquement comme suit : $f(x) = max(0,x)$.

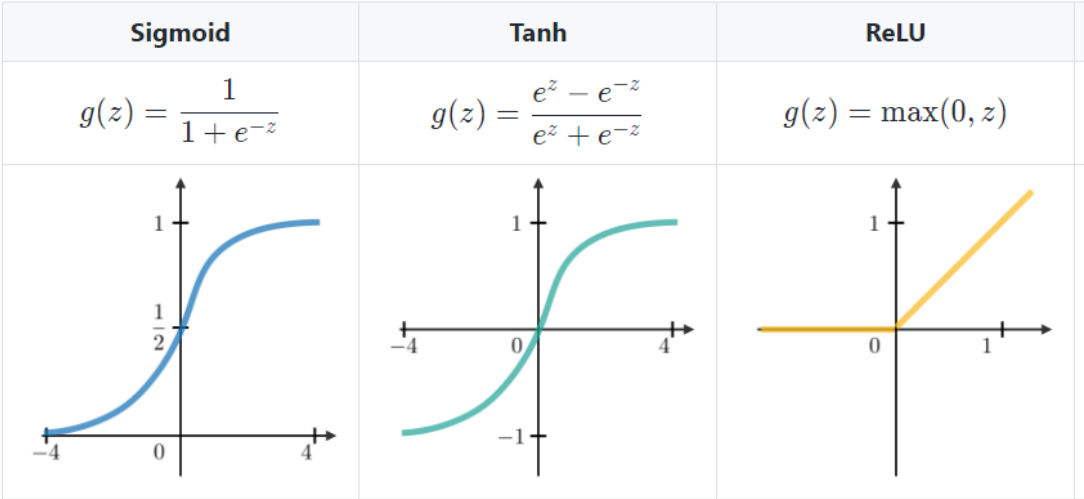


Fig. 1.17—Quelque fonctions d'activation (Amidi & Amidi, 2020).

1.4.3.3 Couche de Pooling

La couche de convolution est suivie du couche de pooling, ce qui est appliqué en détail aux cartes d'entités résultant du convolution. Son objectif principal est de réduire les dimensions de ces cartes afin de préserver et de garder les informations importantes sur chaque étape. Parmi les techniques de pooling les plus courantes et les plus utilisées est Max Pooling. Le filtre dans ce processus doit être paire (Ghosh et al., 2020).

L'exemple de la Figure 1.18 illustre le fonctionnement de la couche pooling.

1.4.3.4 Couches entièrement connectées

C'est la dernière couche des réseaux de neurones convolutifs, qui représente l'étape de classification, et apparaît comme un réseau entièrement connecté. Cette couche prend les résultats de l'étape d'extraction des caractéristiques comme entrées après l'avoir aplaties et converties en vecteur, et la sortie finale est le résultat final de CNN (Figure 1.19) (Ghosh et al., 2020).

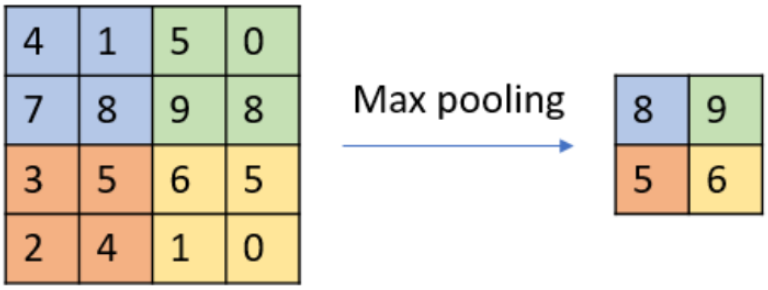


FIG. 1.18—Opération de Max Pooling (PODAREANU et al., 2019).

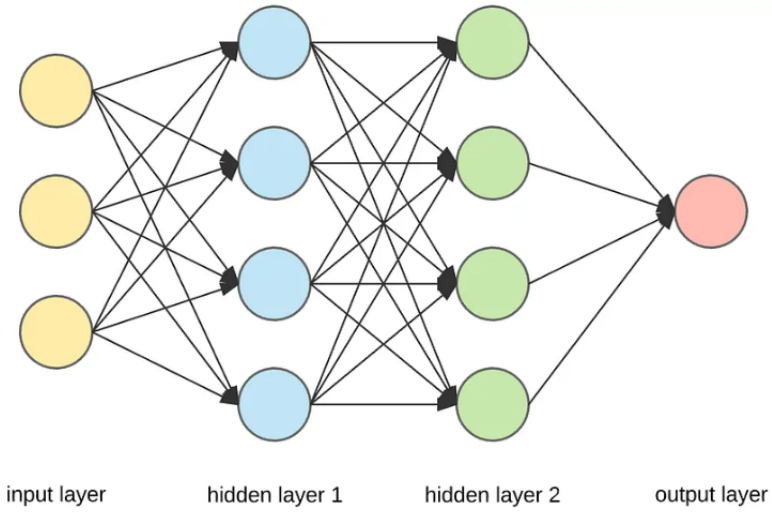


FIG. 1.19—Architecture de couches entièrement connectée (SOROKINA, 2022).

1.4.4 Domaines d'application des CNNs

Voici quelques domaines d'application des CNNs :

1.4.4.1 Classification des images

CNN est utilisé pour classer les images, en particulier dans le cas de grands ensembles de données en extrayant les caractéristiques de l'image de manière efficace et avec une grande précision (GHOSH et al., 2020).

Les CNNs se composent de trois couches principales interconnectées qui aident au traitement et à l'analyse des images, La première couche est la couche de convolution et elle est responsable de l'extraction des caractéristiques de l'image par des filtres apprenables, où l'image est traitée comme une matrice contenant des valeurs de pixels, qui sont présentées en entrée de la première couche, puis suivies par la couche de Maxpooling qui se charge de réduire les dimensions tout en veillant à préserver les informations de l'image, et la dernière couche à connecter dans laquelle se déroule le processus de classification. Les CNNs se caractérisent par la capacité de s'entraîner et d'apprendre à partir

des données qui lui sont fournies, car le réseau est entraîné en fournissant ces données, et ainsi il acquiert des informations et apprend les caractéristiques des images qui lui permettent de prendre une décision lors de la classification (Ramprasath et al., 2018). Et sont considérés le premier choix pour la classification des images en raison de sa plus grande précision par rapport aux autres méthodes (Ghosh et al., 2020). Sont utilisés de manière significative dans la classification des images médicales, telles que les images de diagnostic du cancer du sein et de nombreuses autres tumeurs, en plus de la classification des visages (Ramprasath et al., 2018).

1.4.4.2 Reconnaissance de texte

Les CNNs jouent un rôle efficace pour résoudre les problèmes de reconnaissance des textes dans les images, qui sont des nombres et des symboles, quelle que soit leur langue (Ghosh et al., 2020), où les données textuelles sont représentées sous forme de matrice. Les couches de CNNs sont précédées d'une couche chargée de convertir le texte en matrice, qui est la couche d'incorporation, où elle utilise ses techniques telles que GloVe, BERT, Word2Vec pour représenter chaque symbole ou mot avec une ligne de la matrice, de sorte que chaque symbole est représenté dans un vecteur. Une fois ce processus terminé, l'entrée est prête à être insérée dans la couche de convolution pour extraire les caractéristiques sous forme de vecteurs, suivie du processus Maxpooling qui gère le résultat de la convolution comme entrée, et le rôle de cette couche est de préserver les informations tout en la réduction de la représentation, suivie de la couche entièrement connectées dans laquelle les caractères sont classés, et le résultat est transmis à la fonction d'activation Softmax, et la sortie finale est la distribution de probabilité pour les différentes étiquettes (Bhandare et al., 2016).

La reconnaissance des données du groupe MNIST est la première contribution de CNN, et cela a été fait avec une grande précision (Ghosh et al., 2020).

1.4.4.3 Détection et segmentation d'objets

CNN a prouvé son efficacité sur les applications de vision par ordinateur, notamment la détection et la segmentation d'objets, dont la mission est d'identifier et de diviser différents objets dans les images. CNN a été amélioré plusieurs fois ces dernières années pour devenir l'une des méthodes de détection d'objets les plus utilisées (Bhatt et al., 2021).

La section suivante décrit les différentes méthodes utilisées par CNN pour détecter des objets et il existe deux types :

Les détecteurs à deux étapes, comme R-CNN (Region-based CNN), se composent de deux étapes principales. La première étape consiste à identifier les régions d'intérêt (ROIs) en utilisant la recherche sélective ou les boîtes englobantes. Dans la deuxième étape, les caractéristiques sont extraites de chaque région à l'aide du CNN, puis les objets sont classés à l'aide de SVM. Cependant, R-CNN nécessite beaucoup de temps pour l'apprentissage du réseau et la reconnaissance des objets, ainsi qu'un espace de stockage important pour extraire les caractéristiques de chaque région. En tant qu'amélioration, Fast R-CNN a été

proposé comme une alternative à R-CNN, offrant une meilleure vitesse de reconnaissance des objets et un coût de calcul réduit. Son idée principale est de faire passer l'image une seule fois à travers le CNN, la couche de pooling utilisant les ROIs pour créer une carte de caractéristiques et détecter les objets. Bien que Faster R-CNN soit plus rapide que R-CNN, il faut encore beaucoup de temps pour terminer le processus lorsqu'il s'agit de grands ensembles de données (Patel & Patel, 2020).

Détecteurs à un étage : Comme SSD, ainsi que YOLO dans lequel l'image est divisée en parties et chaque boîte classe l'objet et définit la boîte englobante. Les détecteurs à deux étages se caractérisent par leur grande précision, mais ils sont plus lents que les détecteurs à un étage (Patel & Patel, 2020).

1.4.4.4 Reconnaissance de la parole

Les CNNs sont connus pour être efficaces dans le traitement des problèmes des images, mais des études récentes ont révélé la capacité du CNN à reconnaître la parole avec une bonne efficacité (Bhatt et al., 2021).

Tout d'abord, le son est converti en un spectrogramme, qui montre les composantes de fréquence au fil du temps après leur analyse, ce qui représente l'onde sonore. Le spectrogramme est représenté par une image qui est introduite dans le réseau pour l'apprentissage. La première couche est la couche convolutive qui est responsable de l'extraction des caractéristiques de l'image du spectrogramme, suivie de la couche de regroupement pour réduire les dimensions de la carte des caractéristiques, et la dernière couche est la couche entièrement connectée dans laquelle le spectrogramme est classé en classes de parole basée sur Softmax (Alsobhani et al., 2021).

Les CNNs surpasse le DNN dans quatre domaines identifiés par les chercheurs de Microsoft en 2015 : Robustesse au bruit, Reconnaissance vocale à distance Modèles à faible encombrement, Conditions de test d'entraînement non concordantes pour les canaux (Bhandare et al., 2016).

1.4.4.5 Détection de la communauté

La popularité croissante des CNN et leur succès dans de nombreux domaines ont incité les chercheurs à proposer plusieurs approches pour détecter les communautés dans des réseaux complexes. Le premier modèle supervisé proposé par Xing et autres. est destiné aux réseaux topologiquement incomplets (TIN). La couche convolutive extrait les caractéristiques des relations adjacentes de chaque nœud en utilisant des filtres convolutifs. Ensuite, une opération de pooling extrême est appliquée pour réduire la taille des cartes de caractéristiques. Ce processus est répété avec une deuxième couche convolutive pour extraire des caractéristiques encore plus complexes. Une autre approche courante est le modèle de réseau ComNet-R, qui se base sur la classification des arêtes en utilisant un modèle E2I (Edge To Image) qui convertit les arêtes en images. ComNet-R supprime les arêtes entre les communautés pour former des communautés initiales, qui sont ensuite fusionnées en se basant sur des modules locaux afin d'améliorer la segmentation (Su et al., 2022).

1.5 Découverte de communautés

De nombreux systèmes complexes dans le monde réel ont pris la forme de réseaux constitués de nœuds qui interagissent entre eux, comme les réseaux sociaux, biologiques et technologiques, etc, et pour analyser et comprendre les structures de ces réseaux, il faut les démanteler et les diviser dans les communautés, ce qui conduit à un problème plus complexe et c'est le problème de la découverte des communautés qui est devenu un vaste champ de recherche qui a suscité beaucoup d'attention et donné lieu à de nombreux travaux. Le problème de la découverte des communautés est de connaître d'abord la signification d'une communauté et de chercher une définition générale pour celle-ci, car aucune définition n'a été atteinte ou convenue concernant la communauté (Slimani & Drif, 2016). Ci-dessous, nous mentionnons quelques définitions :

1.5.1 Définition d'une communauté

D'après (M. Newman, 2006) une communauté est un sous graphe que ne peut pas être divisé. Wasserman et Faust, 1994 ont considéré qu'un groupe de nœuds dans le graphe représente une communauté s'il est fortement connecté les uns aux autres, mais sa connexion aux nœuds extérieurs au groupe est faible. La communauté également été définie dans le sens de fort et de faible par Radicchi et al., 2004.

Une communauté est définie au sens fort si chaque nœud a plus de connexions avec le nombre d'autres au sein du groupe qu'avec les nœuds extérieurs au groupe.

$$K_i^{in}(V) > K_i^{out}(V), \forall i \in V,$$

Tel que : $K_i^{in}(V)$ est le nombre de connexions entre le nœud i et les autres nœuds appartenant au sous-graphe V.

$K_i^{out}(V)$ est le nombre de liens vers des nœuds dans le reste du réseau.

Une communauté est définie comme faible si le nombre de connexions au sein de la communauté est supérieur au nombre de connexions aux nœuds extérieurs à la communauté.

$$\sum_{i \in V} K_i^{in}(V) > \sum_{i \in V} K_i^{out}(V).$$

En général, les communautés de réseau sont des groupes de nœuds au sein desquels les nœuds sont beaucoup plus connectés les uns aux autres qu'au reste du réseau. Les communautés, également appelées clusters ou modules, désignent des regroupements de nœuds qui peuvent par exemple, partager des caractéristiques communes, souvent échanger des informations, ou effectuer des tâches comparables à l'intérieur du réseau (Radicchi et al., 2004).

Les communautés dans les réseaux sociaux reflètent des groupes de personnes ayant des intérêts et des antécédents communs et suggèrent des modèles de groupes sociaux réels (Radicchi et al., 2004).

1.5.2 Définition de la découverte de communautés

La découverte de communautés est un domaine de la recherche en analyse de réseau qui se concentre sur la détection de groupes ou de sous-groupes de nœuds qui ont des caractéristiques similaires et des relations étroites entre eux dans un réseau donné tels que des groupes d'amis dans un réseau social, des groupes de gènes dans un réseau de régulation génétique. Il existe plusieurs approches pour détecter les communautés et les résultats de la découverte de communautés peuvent être utilisés pour comprendre les caractéristiques de chaque groupe, les relations entre les groupes, et les dynamiques de réseau à l'intérieur et entre les groupes (Combe, 2013).

1.5.3 Classification des algorithmes de la découverte de communautés

La détection de communautés dans un réseau complexe est une tâche complexe qui peut être approchée de différentes manières en fonction de la nature du réseau et des objectifs d'analyse. Ces approches peuvent être classées selon deux axes principaux : statique ou bien dynamique et avec ou sans chevauchements, ce qui donne quatre approches distinctes pour détecter les communautés dans les réseaux(Cazabet, 2013).

1.5.3.1 Détection de communautés statiques

La détection de communautés statiques est une approche qui suppose que les communautés dans un réseau sont stables et ne subissent pas de changements au fil du temps (Cazabet, 2013).

1.5.3.2 Détection de communautés dynamiques

Une communauté est dite dynamique si sa structure ou sa composition change au fil du temps (Sarr & Moctar, 2016).

1.5.3.3 Détection de communautés sans chevauchements

La détection de communautés sans chevauchement est une approche qui consiste à partitionner les nœuds d'un réseau en communautés mutuellement exclusives, c'est-à-dire qu'un nœud n'appartient qu'à une seule communauté(Fortunato, 2010).

1.5.3.4 Détection de communautés avec chevauchements

Une communauté est chevauchante si une partie de ses nœuds appartient simultanément à d'autres communautés (Sarr & Moctar, 2016).

1.6 Conclusion

En conclusion, ce chapitre introduit la théorie des graphes et les réseaux complexes, met en évidence l'importance de la détection de communautés dans ces réseaux, et présente également les réseaux de neurones à convolution et leur utilisation pour la détection de communautés.

Dans le prochain chapitre, nous examinons en détail l'état de l'art des méthodes de détection de communautés.

Chapitre 2

État de l'art

2.1 Introduction

La découverte des communautés est un domaine de recherche essentiel en sciences des réseaux, car il est reconnu que comprendre et analyser la structure d'un réseau offre de nombreux avantages précieux dans de nombreux domaines. Les chercheurs se sont intéressés à ce sujet et ont proposé de nombreuses méthodes et algorithmes pour détecter les communautés. Récemment, les techniques d'apprentissage en profondeur, en particulier les réseaux de neurones convolutifs (CNN), ont suscité un vif intérêt en raison de leur succès dans divers domaines. Cela a motivé les chercheurs à proposer des approches basées sur CNN pour la découverte de communautés dans des réseaux complexes.

Dans cette partie, nous examinerons les méthodes traditionnelles de détection de communautés, en passant en revue les algorithmes les plus importants utilisés dans chaque méthode. Ensuite, nous nous concentrerons sur les approches basées sur CNN, en examinant à la fois la classification des nœuds et la classification des arêtes.

2.2 Méthodes de découverte de communautés traditionnelles

Dans cette section, nous explorons les méthodes classiques de détection de communautés et présentons une sélection d'algorithmes qui sont utilisés en considérant les méthodes hiérarchiques, dynamiques et d'optimisation comme l'illustre la figure 2.1.

2.2.1 Méthodes hiérarchiques

En général, on sait très peu de choses sur la structure communauté d'un graphe. Il est rare de connaître le nombre de communautés dans lesquels le graphe est divisé, ou d'autres indications sur l'appartenance des nœuds. Parmis les algorithmes de détection de communautés, nous trouvons les méthodes hiérarchiques qui sont des approches de détection de communauté dans des réseaux complexes qui cherchent à regrouper les nœuds du réseau en communautés distinctes de sorte que les nœuds d'une même communauté

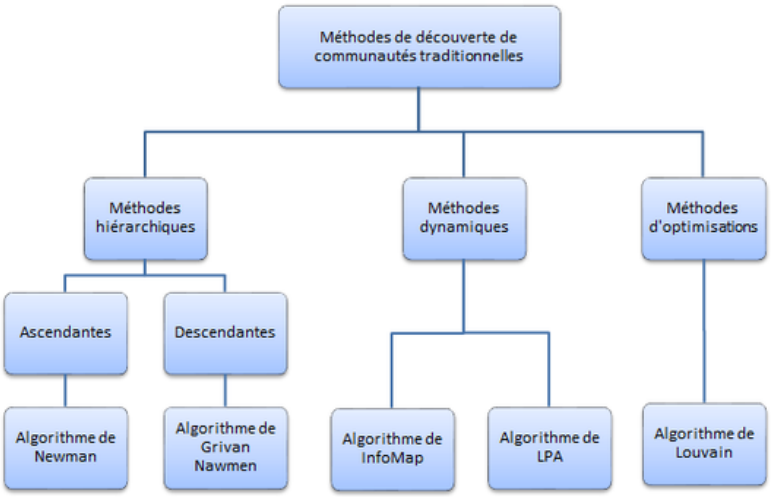


FIG. 2.1—Diagramme des méthodes traditionnelles de détection de communautés.

soient aussi similaires que possible et que les nœuds de différentes communautés soient aussi différents que possible. Les méthodes hiérarchiques ont l'avantage de fournir une vue globale de la structure des communautés. Les réseaux sociaux par exemple ont souvent une structure hiérarchique. Les méthodes hiérarchiques peuvent être coûteuses en termes de temps de calcul pour les grands réseaux, ce qui limite leur application aux petits réseaux. Il existe deux types principaux de méthodes hiérarchiques : les méthodes ascendantes et les méthodes descendantes (FORTUNATO, 2010).

2.2.1.1 Méthodes ascendantes

Les méthodes ascendantes, également appelées agglomératives présentent l'avantage de ne pas nécessiter de spécification préalable du nombre de communautés à détecter. L'un des algorithmes les plus couramment utilisés pour détecter les communautés de manière ascendante est l'algorithme de Newman (M. E. NEWMAN, 2004).

Algorithme de Newman

L'algorithme de Newman, proposé par Mark Newman en 2004, qui s'agit d'un algorithme ascendant qui commence avec chaque nœud dans sa propre communauté et fusionne progressivement les communautés en groupes plus larges selon une mesure de similarité entre les communautés appelée *modularité*, une métrique qui permet de trouver directement la partition communautés correspondant à la modularité maximale pour un graphe donné. L'algorithme de Newman a été largement utilisé dans la communauté scientifique pour l'analyse de divers réseaux, tels que les réseaux sociaux, les réseaux biologiques et les réseaux de transport. Depuis sa création, plusieurs variantes et améliorations

de l'algorithme ont été proposées pour améliorer sa précision et sa performance.

Algorithme 3 : Algorithme de Newman

Data : un réseau G
Result : la division de G en communautés
Initialiser chaque nœud comme sa propre communauté ;
Calculer la modularité de la partition initiale ;
while *il y a des communautés à fusionner* **do**
  Calculer la modularité de chaque paire de communautés fusionnables ;
  Fusionner les deux communautés qui augmentent le plus la modularité ;
  Mettre à jour la liste des communautés fusionnées ;
end

L'Algorithme 3 fonctionne de la manière suivante : initialement, chaque nœud est considéré comme une communauté distincte. Ensuite, la modularité est calculée pour toutes les paires de communautés voisines. La modularité est une mesure de la qualité du découpage d'un réseau en communautés. Elle est définie comme étant la différence entre le nombre d'arêtes au sein d'une communauté et le nombre attendu si les arêtes sont distribuées de manière aléatoire. Mathématiquement, l'algorithme de Newman calcule la modularité comme suit : $Q = \sum_i (e_{ii} - a_i^2)$,

où, e_{ij} est le nombre d'arêtes reliant les nœuds i et j, $a_i = \sum_j e_{ij}$ est la somme des degrés des nœuds dans la communauté i, et e_{ii} est le nombre d'arêtes à l'intérieur de la communauté i. La modularité est toujours comprise entre -1 et 1. Les communautés sont fusionnées de manière itérative en sélectionnant la fusion qui maximise la modularité, et les nœuds fusionnés sont combinés en des nœuds virtuels. La partition est ensuite mise à jour en fusionnant les communautés choisies, et ce processus est répété jusqu'à ce qu'il n'y ait plus de fusion possible. Enfin, la partition finale est obtenue en remplaçant les nœuds virtuels par les nœuds originaux correspondants, aboutissant à une partition qui maximise la structure interne des communautés tout en minimisant les connexions entre les communautés.

Les étapes de l'Algorithme 3 sont illustrées par la Figure 2.2.

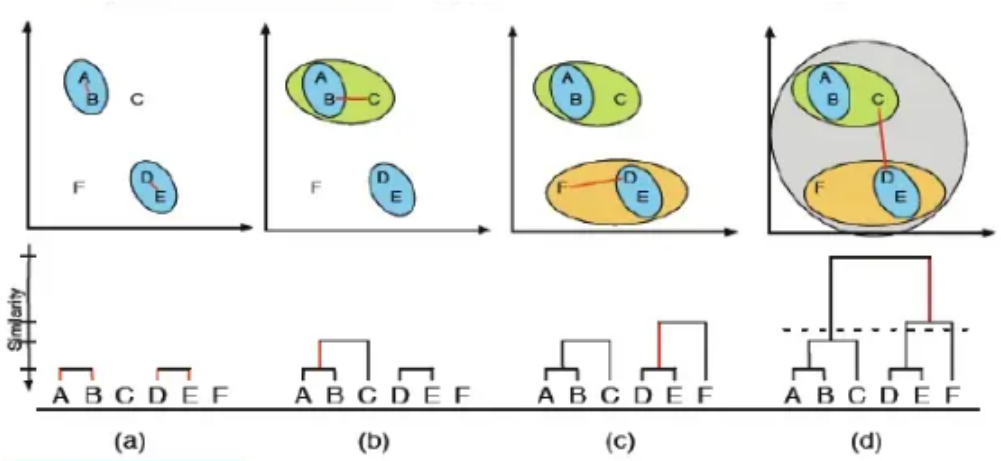


Fig. 2.2—Dendrogramme et différentes étapes d'un algorithme hiérarchiques Newman (Janssen, 2012).

La complexité de l'algorithme de Newman dépend de la méthode utilisée pour calculer la modularité. L'algorithme calcule la modularité en parcourant toutes les paires de nœuds du réseau. Pour chaque paire, il compte le nombre d'arêtes reliant les nœuds, ainsi que la somme des degrés des nœuds dans leurs communautés respectives. Cette approche a une complexité quadratique de l'ordre de $O(n^2)$, où n est le nombre de nœuds dans le réseau.

2.2.1.2 Méthodes descendantes

La méthode descendante, également appelée divisive, est une approche plus rapide que la méthode ascendante. Cependant, elle nécessite la spécification préalable du nombre de communautés souhaité, ce qui peut être difficile à déterminer à l'avance pour les grands réseaux. L'algorithme de Girvan-Newman est l'un des algorithmes de détection de communautés descendants les plus couramment utilisés (Girvan & Newman, 2002).

Algorithme de Grivan Nawman

L'algorithme de Girvan-Newman a été proposé en 2002 par les physiciens Mark Girvan et M. E. J. Newman. Depuis, l'algorithme a été utilisé dans de nombreuses applications, notamment dans l'étude des réseaux sociaux, des réseaux biologiques, de la physique et de l'informatique. C'est une méthode de détection de communautés séparative qui utilise une mesure de centralité appelée *centralité d'intermédiarité des arêtes* (Edge-Betweenness Centrality). L'idée de base est de supprimer itérativement les arêtes ayant la plus grande centralité d'intermédiarité jusqu'à ce que le réseau se divise en un certain nombre de communautés souhaité.

Algorithme 4 : Algorithme de Girvan-Newman

```
Data : Un graphe G = (V, E)
Result : Les communautés du graphe
Q_max ← 0 ; C_max ← ∅ ;
while |E| > 0 do
    Calculer la centralité d'intermédiarité de tous les arêtes de G ;
    Trouver le arête avec la plus grande centralité d'intermédiarité ;
    Supprimer le arête trouvé ;
    Trouver les communautés en utilisant les composantes connexes de G ;
    Calculer le modularity Q pour les communautés trouvées ;
    if Q > Q_max then
        Q_max ← Q ;
        C_max ← les communautés trouvées ;
    end
end
return C_max ;
```

Dans ce qui suit, nous décrivons sommairement l'Algorithme 4.

Tout d'abord, la centralité d'intermédiarité de toutes les arêtes du réseau est calculée à l'aide de la formule $C_B(e) = \sum_{i \neq e \neq j} \frac{\sigma(i,j|e)}{\sigma(i,j)}$, où e représente l'arête, $\sigma(i,j)$ représente le nombre total de chemins les plus courts du nœud i au nœud j, et $\sigma(i,j|e)$ représente le

nombre total de chemins les plus courts du nœud i au nœud j qui passent par l'arête e. Ensuite, l'arête ayant la plus grande centralité d'intermédiarité est supprimée. Les centralités d'intermédiarité des arêtes affectées sont recalculées, et ce processus est répété jusqu'à ce qu'il ne reste plus d'arêtes dans le réseau.

Les Figures 2.3 et 2.4 représentent les différentes étapes de algorithme Girvan-Newman.

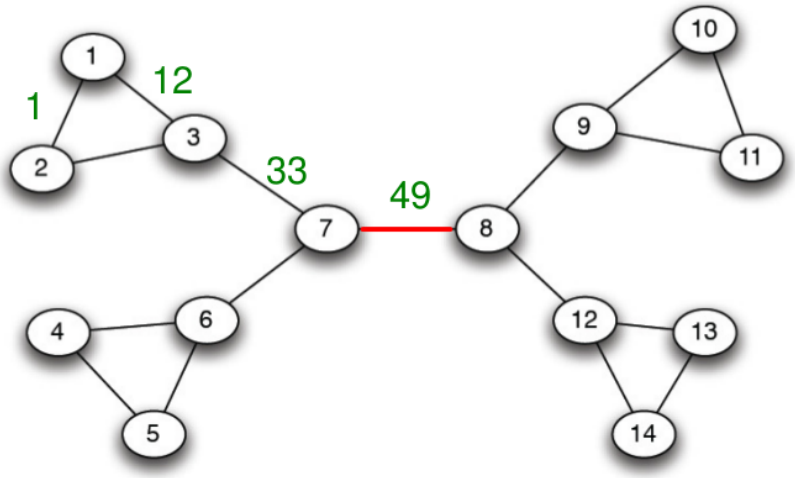


Fig. 2.3—Graphe pour la présentation de l'algorithme de Girvan-Newman (Michel, 2023).

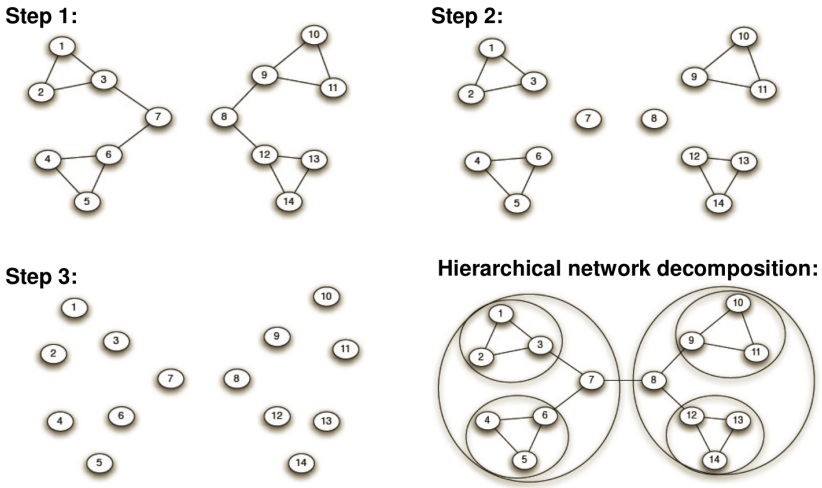


Fig. 2.4—Étapes de l'algorithme de Girvan-Newman (Michel, 2023).

L'algorithme de Girvan-Newman calcule la centralité d'intermédiarité de toutes les arêtes dans un graphe de n sommets et m arêtes en temps $O(mn)$. Cette opération doit être répétée une fois pour la suppression de chaque arête, entraînant une complexité temporelle dans le pire des cas de $O(m^2n)$. Cependant, après la suppression de chaque arête, seules les centralités d'intermédiarité des arêtes affectées par la suppression doivent être recalculées, ce qui est au maximum celles appartenant à la même composante que l'arête supprimée. Cela signifie que le temps d'exécution peut être meilleur que dans le

pire des cas pour les réseaux ayant une forte structure de communauté, c'est-à-dire ceux qui se séparent rapidement en composantes distinctes après les premières itérations de l'algorithme.

2.2.2 Méthodes dynamiques

La structure topologique dans le monde réel des réseaux est en constante évolution et change de manière inattendue au fil du temps, ce qui les rend instables. D'autre part, il est difficile d'appliquer des méthodes statiques, car elles ne leur sont pas adaptées. Par conséquent, les méthodes dynamiques des découvertes des communautés ont été proposées. Elles visent à suivre l'évolution des sociétés et à prédire les changements qui s'y produisent au fil du temps. Ces changements sont représentés dans l'émergence et la disparition des communautés à travers les fusions et les divisions qui se produisent dans le réseau (Sun et al., 2020). De nombreux algorithmes ont été proposés dans ce domaine et nous avons choisi les deux plus populaires, à savoir, l'algorithme InfoMap et celui de la propagation d'étiquettes.

2.2.2.1 Algorithme InfoMap

L'algorithme proposé par Rosvall et Bergstro en 2008 est une méthode de découverte de communautés basée sur le flux d'informations, où les nœuds entre lesquels les informations circulent rapidement et facilement représentent une unité ou une communauté. En effet, ceci dépend de la marche aléatoire, où la transition se fait d'un objet à un autre au hasard. Cet algorithme cherche à compresser le flux d'informations, c'est-à-dire à coder le chemin aléatoire dans le réseau, et cela se fait par codage de Huffman, où chaque nœud est représenté par un mot de code, et donc le problème de la détection des communautés se transforme en compression et cryptage des informations (Figure 2.5). L'algorithme cherche à réduire la longueur de la description du chemin aléatoire dans le réseau en utilisant l'équation de carte (Rosvall & Bergstrom, 2008).

L'équation de la carte est utilisée pour diviser les communautés et trouver la représentation compacte en réduisant la longueur de description du chemin de marche aléatoire dans le réseau. C'est la fonction objective de l'infomap qu'elle cherche à améliorer. L'équation de la carte est basée sur l'idée de représenter des chemins aléatoires par deux niveaux : le premier niveau dans lequel les communautés sont nommées et le deuxième niveau dans lequel les nœuds au sein des communautés sont nommés par le codage de Huffman. Cette idée permet aux chiffrements des nœuds internes d'être réutilisés dans différentes communautés, ce qui raccourcit la longueur du chiffrement (Rosvall & Bergstrom, 2008).

L'équation de la carte calcule la longueur de description minimale pour le chemin aléatoire et est donnée par l'Équation 2.1 :

$$L(M) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} p_i \curvearrowright H(P_i), \tag{2.1}$$

Où $q_{\curvearrowright} H(Q)$ représente la longueur moyenne du code des mouvements entre les communautés, et $\sum_{i=1}^{m} p_i \curvearrowright H(P_i)$ représente la longueur moyenne du code des mouvements

au sein des communautés (Rosvall & Bergstrom, 2008).

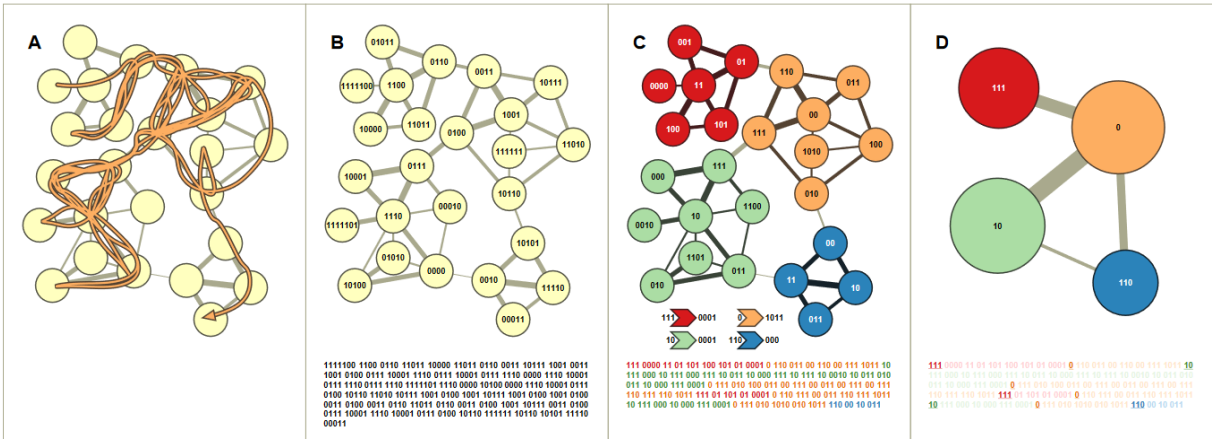


Fig. 2.5—Détecter les communautés en compressant la description des flux d'information sur les réseaux (Rosvall & Bergstrom, 2008)

Algorithme 5 : Pseudocode pour l'algorithme Infomap

Data : Réseau $G = (V, E)$, où V est l'ensemble de N sommets, E est l'ensemble des arêtes, et seuil d'amélioration minimale de qualité τ.
Result : Carte des nœuds vers les modules M.
Exécuter PageRank pour calculer le taux de visite des sommets pour chaque sommet ;
$M = m_i = v_i | v_i \in V$;
$L = L(M)$;
repeat
  $L_{prev} = L$;
  R = séquence aléatoire d'entiers de 1 à N ;
  for $i = 0$ *à* $N - 1$ **do**
    $m_{new} = \text{bestNewModule}(M, v_{R[i]})$;
    Déplacer $v_{R[i]}$ vers le module m_{new}, et mettre à jour M et L ;
  end
until $L_{prev} - L < \tau$
return M ;

Comme nous l'avons mentionné précédemment, l'algorithme Infomap (Algorithme 5) est divisé en deux phases : La première phase consiste à identifier les nœuds importants du réseau à l'aide de l'algorithme de PageRank en calculant la probabilité de visiter chaque nœud. Initialement, les modules initiaux sont définis de sorte que chaque nœud représente un module. Dans la deuxième phase, l'appel bestNewModule(M, v) prend en compte tous les mouvements possibles du sommet v et choisit celui qui réduit le plus la valeur $L(M)$. La pile de modules et la valeur $L(M)$ sont mises à jour à chaque fois jusqu'à ce que la valeur de la longueur minimale de la description soit inférieure au seuil minimal d'amélioration de la qualité τ. L'algorithme s'arrête et renvoie les modules finals qu'il considère comme la meilleure division du réseau (Barabasi & Posfai, ©2016).

La complexité de la découverte des communautés par l'algorithme InfoMap est $O(NlogN)$ où N est le nombre de nœuds dans le réseau. Ceci dépend de la structure du réseau, car

l'optimisation de l'équation de la carte nécessite de traverser le réseau plusieurs fois.

2.2.2.2 Algorithme de propagation d'étiquettes

L'algorithme de propagation d'étiquettes (Label Propagation Algorithm, LBA) est parmi les algorithmes les plus courants pour découvrir les communautés. Il a été proposé par (Raghavan et al., 2007). LBA se caractérise par sa facilité et sa rapidité et ne nécessite pas de connaissance préalable des communautés. Il peut donc être appliqué à des réseaux contenant des millions d'individus (Xuegang et al., 2016).

L'idée de LPA est de propager les étiquettes des nœuds dans le réseau où il détermine l'étiquette d'un nœud particulier (L_i) en fonction des étiquettes de ses voisins. Au début, chaque nœud du réseau est nommé avec une étiquette unique, ensuite les étiquettes des nœuds sont mises à jour à chaque étape à plusieurs reprises, de sorte que les étiquette des voisins d'un nœud $N_L(i)$ contrôlent son nom. Il est donc mis à jour en fonction de la plupart de ses voisins (Xuegang et al., 2016).

$$L_i = \arg\max_l |N_l(i)|.$$

L'opération de mise à jour s'arrête lorsque l'algorithme converge, c'est-à-dire lorsque chaque nœud prend l'étiquette la plus commune parmi ses voisins tel que illustré dans la Figure 2.6. Cette opération est effectué de manière synchrone ou asynchrone.

Par synchrone nous entendons que chaque fois que l'opération de mise à jour est répétée, le nœud prend la même étiquette que ses voisins sur l'itération $(t-1)$. Alors que dans le cas des mises à jour asynchrone, les nœuds sont mis à jour séquentiellement, c'est-à-dire qu'à une certaine itération (t) le nœud porte l'étiquette commune entre la majorité de ses voisins qui ont été mis à jour au même instant (t), et le reste des nœuds à l'instant $(t-1)$.

Enfin, les communautés sont déterminées en fonction des nœuds connectés entre eux et portant la même étiquette (Raghavan et al., 2007).

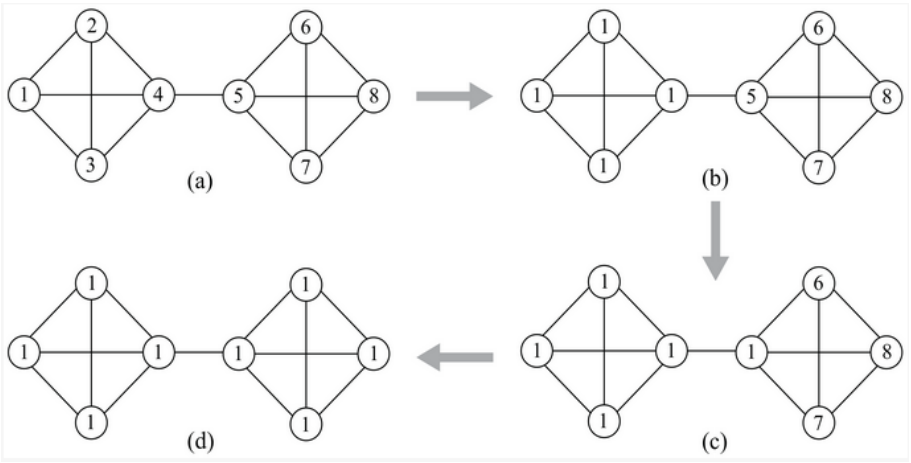


Fig. 2.6—Visualisation des étapes de l'algorithme de propagation des étiquettes (Xuegang et al., 2016).

Algorithme 6 : Algorithme de propagation des étiquettes

Data : Un graphe G
Result : Les communautés dans G
Initialiser chaque nœud dans G avec une étiquette unique ; **repeat**
  Permuter aléatoirement les nœuds dans G ; **pour** *chaque nœud dans G* **faire**
    Récupérer les étiquettes des voisins du nœud ; Mettre à jour l'étiquette du nœud pour qu'elle corresponde à l'étiquette majoritaire parmi ses voisins ; Si plusieurs étiquettes ont la même fréquence, en choisir une aléatoirement ;
  fin
until *jusqu'à ce que plus aucune étiquette ne change*
Regrouper les nœuds par leurs labels finales pour obtenir les communautés dans G ;

LBA (Algorithme 6) prend un temps quasi-linéaire. La première étape dans laquelle les nœuds sont initialisés avec des labels uniques nécessite O(n) temps et l'étape dans laquelle un nœud est nommé en regroupant ses voisins nécessite O(m) pour chaque itération de l'lagorithme, où n représente le nombre de nœuds et m représente le nombre d'arêtes dans le graphe(Raghavan et al., 2007).

Malgré les avantages de cet algorithme, il n'est pas précis, et son problème réside dans l'instabilité et la similitude des résultats obtenus lors de la découverte des communautés, en raison du processus de mise à jour aléatoire répété qui entraîne des solutions très différentes, en particulier dans les communautés à structure faible (Xuegang et al., 2016).

2.2.3 Méthodes d'optimisations

Les méthodes basées sur l'optimisation visent à maximiser la modularité, qui est une fonction objective connue sous le nom de fonction de qualité, utilisée pour évaluer la qualité des communautés en identifiant les communautés densément connectées et les communautés discrètes. Il est conseillé d'appliquer ces méthodes à de grands réseaux en raison de la souffrance des algorithmes d'optimisation de la modularité du problème de limite de résolution, ce qui provoque la disparition de petites communautés lorsqu'elles sont regroupées avec des communautés similaires, bien que la maximisation de la modularité soit un problème difficile, mais c'est la fonction la plus utilisée, et son amélioration reste un sujet d'intérêt et d'étude (Kanawati, 2013). Par la suite nous nous intéressons à l'algorithme Louvain qui se base sur l'optimisation de modularité.

2.2.3.1 Algorithme Louvain

L'algorithme de Louvain (Blondel et al., 2008) est un algorithme de regroupement hiérarchique basé sur l'optimisation gourmande utilisée pour découvrir des communautés dans de grands réseaux complexes en peu de temps. Il s'appuie sur la modularité et la considère comme une fonction objective qu'il vise à maximiser de manière répétée. La modularité permet d'identifier et de comparer la qualité des communautés, elle concerne

la partie interne de communauté et se définit par l'Équation 2.3.

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{2.2}$$

où m représente la somme pondérée des arêtes du réseau communautaire et c_i est la communauté dans laquelle réside le nœud i, A_{ij} est le poids des arêtes reliant les nœuds i et j, k_i est la somme des poids des arêtes connectées au nœud i, et la fonction $\delta(c_i, c_j) = 1$ si les nœuds i et j sont dans la même communauté sinon $\delta(c_i, c_j) = 0$ (Blondel et al., 2008).

L'algorithme Louvain (Algorithme 7) est scindé en deux phases : la première phase dans laquelle les communautés sont divisées de sorte que chaque nœud est une communauté, après quoi les voisins du nœud i sont déterminés et le changement de modularité est calculé lors du déplacement de ce nœud vers la communauté qui réalise le gain de modularité le plus élevé. Ce processus est répété sur tous les nœuds du réseau et s'arrête lorsque le déplacement des nœuds n'améliore pas ou n'augmente pas la modularité (Blondel et al., 2008).

La relation de calcul du changement de modularité qui résulte du transfert d'un nœud vers une autre communauté est donnée par l'Équation 2.3.

$$\Delta Q_i = [\sum in + 2k_{i,in}/(2m) - ((\sum tot + k_i)/(2m))^2] - [(\sum in/(2m) - (\sum tot/(2m))^2 - (k_i/(2m))^2], \tag{2.3}$$

où m représente la somme des poids des liens du réseau et $\sum in$ est le poids total des liens au sein de C et $\sum tot$ est Le total de tous les liens connectées à C, k_i représente le degré du nœud i et $k_{i,in}$ représente la somme des poids des liens sortant du nœud i vers les nœuds de C (Blondel et al., 2008).

La deuxième phase dans laquelle un nouveau réseau est créé dont les nœuds sont les communautés obtenues à la phase précédente, de sorte que les nœuds appartenant à la même communauté sont fusionnés en un nœud et la somme des poids des liens entre les nœuds dans l'étape initiale communautés devient le poids du lien entre les nouveaux nœuds qui leur correspondent. Lorsque la deuxième phase est terminée, les étapes des deux phases sont réappliquées au réseau résultant, et le processus d'itération s'arrête lorsque aucun changement ne se produit, et lorsque la modularité ne peut plus être améliorée. Ceci est un témoin idéal de l'obtention d'un division idéale tel que montré par la Figure 2.7 (Blondel et al., 2008).

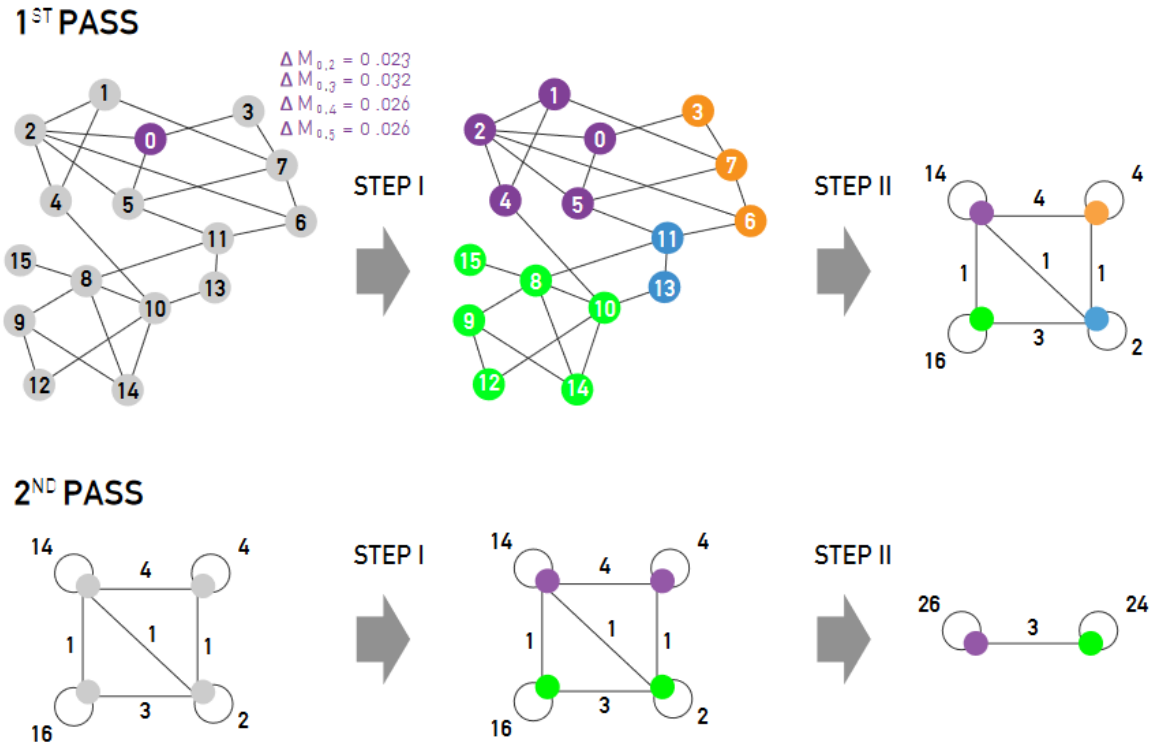


Fig. 2.7—Visualisation des étapes de l'algorithme de Louvain (Barabasi & Posfai, ©2016)

Algorithme 7 : Pseudo-code de la méthode Louvain

```
Data : G : le réseau initial
Result : Communautés de G
Placer chaque nœud de G dans sa propre communauté ; repeat
    for chaque nœud n de G do
        le placer dans sa communauté voisine, y compris la sienne, qui maximise
         le gain de modularité ;
    end
    if la nouvelle modularité est supérieure à l'initial then
        G = le réseau entre les communautés de G ;
    end
    else
        Terminer ;
    end
until aucune augmentation supplémentaire de la modularité n'est possible
```

L'algorithme de Louvain a une complexité temporelle linéaire, ce qui signifie que son temps d'exécution varie de manière directe avec le nombre d'arêtes du réseau. La complexité est généralement notée $O(m)$, où m représentant le nombre d'arêtes (Traag, 2015).

La Table 3.5 résume les méthodes traditionnelles de détection des communautés ainsi que les techniques utilisées dans chacune de ces méthodes.

Tab. 2.1 : Résumé des Méthodes traditionnelles de découverte de communautés

Méthodes	Algorithmes	Complexité	Références
Hiérarchique	**Newman**	$O(n^2)$	(M. E. Newman, 2004)
	Girvan-Newman	$O(m^2n)$	(Girvan & Newman, 2002)
Dynamique	**InfoMap**	$O(NlogN)$	(Rosvall & Bergstrom, 2008)
	LBA	$O(m)$	(Raghavan et al., 2007)
Optimisation	**Louvain**	$O(m)$	(Blondel et al., 2008)

2.3 Découverte de communautés basée sur les CNN

Les méthodes traditionnelles et les méthodes supervisées telles que les SVM ont leurs limitations en termes de capacité de représentation et de généralisation dans la détection de communautés. Ces méthodes ont souvent du mal à apprendre des fonctionnalités de haut niveau, en plus de ses performances instables. Pour surmonter ces limites, les méthodes d'apprentissage profond ont été développées. La détection de communautés basée sur les CNN utilise des architectures de réseaux neuronaux convolutifs pour identifier des communautés. Les deux approches principales dans ce domaine sont la classification des arêtes et la classification des nœuds. Ces méthodes exploitent les capacités de représentation des modèles d'apprentissage profond pour offrir de nouvelles perspectives dans la détection de communautés. Elles permettent de capturer des motifs complexes et d'obtenir des performances améliorées par rapport aux méthodes traditionnelles (Xin et al., 2017).

2.3.1 Approche de classification des arêtes

Dans (Cai et al., 2020), les auteurs ont proposé une méthode pour découvrir des communautés dans des réseaux complexes basée sur la classification des arêtes en fonction de leur emplacement dans le réseau. Cette approche tire parti de la propriété de regroupement dans des réseaux complexes, ce qui signifie que lorsque deux nœuds sont connectés par une arête au sein d'une communauté, la relation entre les groupes de voisins de ces deux nœuds est plus forte que la relation entre les groupes de voisins de deux nœuds connectés par une arête entre d'autres communautés.

Les arêtes sont classées par l'algorithme de segmentation ComNet-R qui consiste en trois étapes de base :

1. À l'aide du modèle Edge-to-image (E2I), les arêtes sont transformées en images dans

l'ensemble d'apprentissage, afin d'être introduites dans le modèle CNN.

2. ComNet classe les arêtes dans le réseau et supprime les arêtes entre les communautés dans le but de diviser le réseau et de former les communautés initiales.

3. Pour obtenir la partition finale, une fonction locale modulaire R est combinée avec les communautés initiales.

Le modèle E2I proposée par Cai et al., 2020, est une méthode de conversion qui permet de transformer les arêtes du réseau en images. Ainsi, les relations entre les voisins des nœuds peuvent être représentées dans une matrice tridimensionnelle de dimensions $(x*y*3)$, où x représente la taille des voisins du nœud i (set_i) et y représente la taille des voisins du nœud j (set_j). Dans la Figure 2.8, l'arête $(8,5)$ se situe dans la même communauté. Les nœuds adjacents au nœud 5 sont $(2,9,6)$ et les nœuds adjacents au nœud 8 sont $(6,7,9)$.

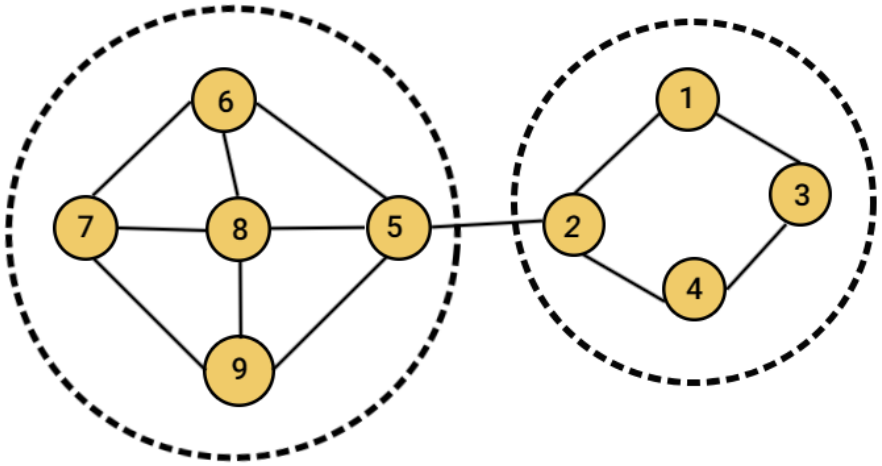


Fig. 2.8—Exemple de structure de communautés.

La représentation de l'arête $(8,5)$ dans une image par E2I est indiquée dans la Table 2.2, Ainsi, l'élément de la matrice est représenté par le vert (v) si les nœuds correspondant à la ligne et à la colonne sont tous deux dans l'intersection $N\_S(i,j)$ où $N(i)$ et $N(j)$ représentent les voisins de deux nœuds i, j d'une arête (i,j) respectivement et $N\_S(i,j)$ représente l'ensemble des nœuds qui sont voisins à la fois de i et de j dans, par le rouge (r) si le réseau contient l'arête formée par les nœuds correspondant à la ligne et à la colonne, et par le bleu (b) dans les autres cas.

Tab. 2.2 : Représentation de l'arête (8, 5) par le modèle E2I.

Nœud	6	7	9
2	b	b	b
9	v	r	v
6	v	r	v

Étant donné que les degrés des nœuds connectés à une arête ne sont pas uniformément répartis dans le réseau, les images générées par E2I ne peuvent pas être directement utilisées par le CNN, car il nécessite des images d'entrée de taille identique. Par conséquent, les

images ont été pré-traitées en utilisant la méthode de recadrage par interpolation de zone. Cette méthode divise l'image originale en zones plus petites, calcule la valeur moyenne des pixels dans chaque zone, puis utilise ces valeurs pour créer une nouvelle image aux dimensions requises afin de générer des images tridimensionnels ont la même taille.

La notation $\text{matr}(x,y,3) \rightarrow \text{IMG}(N,N,3)$ représente la conversion d'une matrice tridimensionnelle appelée $matr$ avec des dimensions x, y et 3. Cette conversion aboutit à une nouvelle matrice tridimensionnelle appelée IMG avec des dimensions unifiées N, N et 3. Le nombre 3 représente les canaux de couleur (rouge, vert et bleu). Avec cela, les données d'entrée pour le CNN sont prêtes.

Nous passons maintenant à la description de l'architecture CNN utilisée pour la classification des arêtes. Elle est généralement composée de trois couches de base : la couche de convolution, la couche de pooling et la couche entièrement connectée.

La couche de convolution est responsable de l'extraction des caractéristiques des images d'entrée à l'aide de plusieurs filtres de convolution. La sortie est un ensemble de cartes de caractéristiques où chaque carte est générée par un filtre différent.

Quant à la couche de pooling, elle est utilisée entre les couches de convolution et est implémentée indépendamment sur chaque canal. Elle est responsable de la réduction de la dimensionnalité des caractéristiques afin de réduire le coût de calcul ComNet et de diminuer l'effet du bruit dans l'image.

La dernière couche consiste en une ou plusieurs couches entièrement connectées qui prennent les cartes de caractéristiques générées par les couches de convolution et pooling et les convertissent en vecteurs, qui sont utilisés comme entrées pour la classification.

L'algorithme proposé vise à segmenter le réseau en supprimant les arrêts afin d'obtenir des communautés non connectées. Après l'entraînement, ComNet est capable de classifier arêtes, qu'elles soient intercommunautaires ou intracommunautaires. La classification s'effectue en se basant sur les étiquettes des nœuds. Ainsi, pour chaque arête, si les deux nœuds qui y sont connectés partagent la même étiquette, cette arête est classée à l'intérieur de la communauté, dans le cas contraire, elle est considérée comme située à l'extérieur de la communauté avec quelques erreurs de classification. Le résultat de la partition ne donne que des communautés primaires, ce qui indique la possibilité d'obtenir une partition améliorée (Figure 2.9).

Afin d'améliorer les résultats de la classification, les communautés préliminaires sont

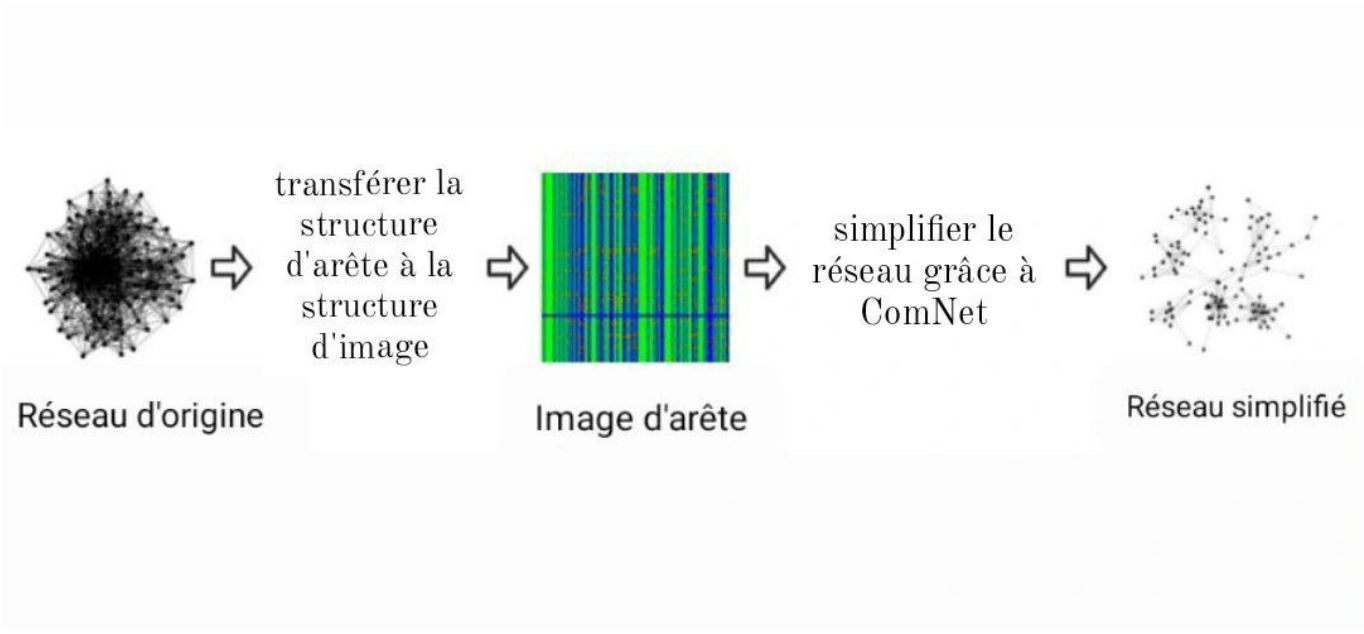


FIG. 2.9—Diviser le réseau à l'aide de ComNet (CAI et al., 2020).

combinées à l'aide de la mesure de la modularité locale R.

Algorithme 8 : Fusion des communautés préliminaires basée sur la modularité locale R

Entrée : Réseau G(V, E), communautés préliminaires FC$(c_1, c_2, \ldots, c_p)$, nombre de communautés m

Sortie : Communautés finales EC$(c_1, c_2, \ldots, c_m)$

Les premières m communautés avec la plus grande modularité locale R dans FC $(c_1, c_2, \ldots, c_p)$ sont définies comme les communautés réelles RC $\{r_{com}(1), r_{com}(2), \ldots, r_{com}(m)\}$. Les autres communautés sont des communautés virtuelles VC $\{v_{com}(1), v_{com}(2), \ldots, v_{com}(n)\}$.

Calculer la modularité locale de toutes les communautés réelles RC comme R_i, pour chaque $i \in$ RC.

while *VC n'est pas vide* **do**

  Prendre une communauté v_{com} au hasard dans VC et la fusionner avec chaque communauté r_{com} dans RC, puis calculer la modularité locale de toutes les communautés fusionnées comme r_i, pour chaque $i \in$ RC.

  Calculer $\Delta R = r_i - R_i$, sélectionner le r_{com} qui maximise ΔR et le fusionner avec v_{com}.

end

L'algorithme 8 commence par identifier les premières m communautés avec la plus grande modularité locale en tant que communautés réelles, tandis que les communautés restantes sont considérées comme virtuelles. L'algorithme calcule la modularité locale Ri pour chaque communauté réelle et fusionne les communautés virtuelles avec les réelles, puis calcule la modularité locale ri de chaque communauté fusionnée et calcule ΔR, puis choisit la communauté avec le gain modulaire le plus élevé pour la fusion. Ce processus est répété jusqu'à ce que toutes les communautés virtuelles soient fusionnées avec

communauté réelle.

Grâce à ces étapes, une division finale précise des communautés est obtenue.

La modularité locale R des communautés primaires est calculée par l'Équation 2.4.

$$R = \frac{B_{in}}{B_{in} + B_{out}} \tag{2.4}$$

où : B_{in} représente le nombre d'arêtes à l'intérieur de la communauté.

B_{out} représente le nombre d'arêtes dépendant de la frontière de la communauté aux nœuds externes à l'extérieur de la communauté.

L'algorithme ComNet-R a fait l'objet d'une évaluation approfondie de ses performances et de son efficacité dans la détection de réseaux générés par ordinateur ainsi que de réseaux réels de différentes tailles. Les résultats ont été comparés à ceux obtenus par des méthodes traditionnelles, et les performances de ComNet-R se sont révélées remarquables, surpassant celles des méthodes traditionnelles. L'algorithme a démontré une grande efficacité dans la découverte de la structure communautaire des réseaux.

Cependant, il est important de noter que l'implémentation actuelle de l'algorithme proposé est limitée aux graphes non orientés. Les auteurs ont indiqué qu'ils se concentreront sur le développement futur de l'algorithme et sur l'amélioration de la précision des résultats de découverte de communauté.

2.3.2 Approche de classification des noeuds

Xin et al (Xin et al., 2017) ont proposé le premier modèle de détection de communauté basé sur un CNN pour les réseaux topologiquement incomplets (TIN). Les TIN font référence à des réseaux dont la structure complète n'est pas observable ou connue en raison de données manquantes ou d'une incomplétude intrinsèque. Par exemple, les réseaux sociaux peuvent avoir des informations partielles sur les individus.

L'article (Xin et al., 2017) propose un modèle de détection de communauté basé sur un CNN pour les réseaux topologiquement incomplets. Le modèle utilise des couches convolutives pour extraire des caractéristiques locales à partir des relations de voisinage et une couche de connexion complète pour la classification des nœuds. Les résultats expérimentaux montrent que cette méthode est efficace, même avec peu de données étiquetées.

2.3.2.1 Données d'entrée

Ce modèle de détection de communauté utilise des matrices 1D pour représenter les relations d'adjacence entre les nœuds. Les entrées sont des nombres réels calculés en fonction de la distance entre les nœuds,Chaque élément de la matrice d'entrée correspond à une paire de nœuds n et n' et est calculé en fonction de la distance entre ces nœuds dans le réseau. Si la distance entre les nœuds est inférieure ou égale à s_0, l'entrée est calculée comme $e^{\sigma(1-s)}$, où σ est un facteur d'atténuation et s est la distance ou le nombre de sauts entre les nœuds n et n' et s_0 un seuil de nombre de sauts défini par l'utilisateur. L'inclusion des relations de contiguïté indirectes peut améliorer les performances, mais

un seuil approprié pour le nombre de sauts doit être choisi pour éviter une frontière floue entre les communautés. La taille de la matrice 1D est déterminée par le nombre de caractéristiques. Le choix d'un seuil adéquat est crucial pour le succès du modèle, et son impact a été évalué lors des expérimentations comme l'illustre la figure 2.10.

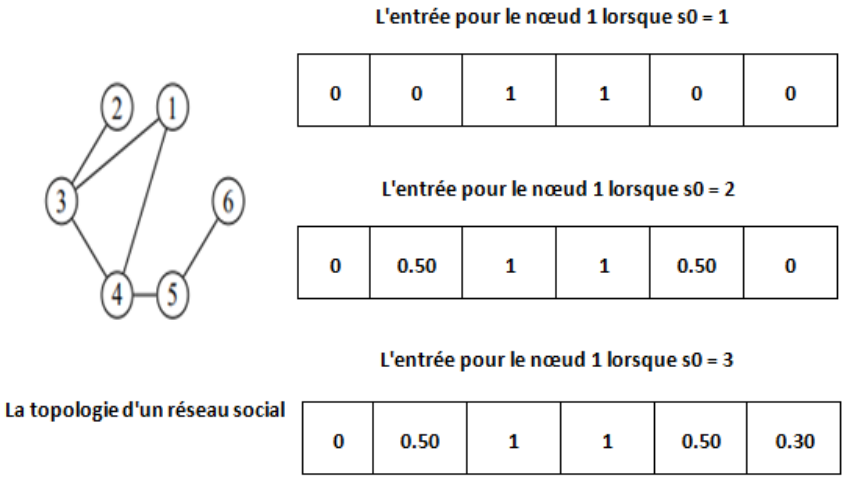


FIG. 2.10—Données d'entrée ($\sigma = 0.6$).

2.3.2.2 Couche convolutionnelle

Dans ce modèle, les noyaux convolutifs sont utilisés pour extraire des caractéristiques locales à partir des relations d'adjacence de chaque nœud. Les valeurs de la carte de caractéristiques V sont calculées en appliquant une fonction sigmoïde à la somme pondérée des entrées de la matrice de relations d'adjacence autour du nœud, en utilisant des poids et des biais spécifiques pour chaque noyau de convolution. Les noyaux convolutifs permettent de générer différentes cartes de caractéristiques pour chaque nœud, capturant ainsi des aspects différents des relations de voisinage dans le réseau. Cela permet au modèle de détecter des motifs complexes et de haute qualité, conduisant à une détection plus précise des communautés dans les réseaux.

$$v_{xy}^{n} = sigmoid\left(b_W + \sum_{i=0}^{w'-1}\sum_{j=0}^{h'-1} W_{ij} \cdot p_{(x+i)(y+j)}^{n}\right), \tag{2.5}$$

Le terme $p_n(x+i)(y+j)$ désigne la valeur d'entrée à la position (x+i, y+j) de la matrice de relations d'adjacence p_n, où x et y sont les coordonnées spatiales de la carte de caractéristiques, et i et j sont les coordonnées spatiales du noyau de convolution. La valeur de W_{ij} est un paramètre appris dans le processus d'entraînement du modèle, tandis que b_W est un biais partagé pour chaque entrée de la carte de caractéristiques.

2.3.2.3 Opération Max-pooling

Après la première couche convolutive et l'opération de pooling maximum, des cartes de caractéristiques c_1 sont obtenues pour chaque nœud d'entrée. Ensuite, une deuxième couche convolutive est appliquée à ces cartes de caractéristiques, suivie d'une opération de pooling maximum similaire à la première couche. Les noyaux de cette deuxième couche génèrent un nouvel ensemble de cartes de caractéristiques c_2 pour chaque entrée. Finalement, les auteurs obtiennent des cartes de caractéristiques c_1c_2 pour chaque nœud d'entrée après cette deuxième couche convolutive. Cette approche permet de capturer des caractéristiques locales et complexes des relations de voisinage, conduisant à une détection plus précise des communautés dans les réseaux.

2.3.2.4 Couche de connexion complète (Full Connection Layer)

La dernière couche de notre modèle proposé est une couche de connexion complète composée de K neurones de sortie, où la valeur de K est la même que le nombre de communautés prédéfinies. Chaque neurone de sortie, disons le k-ème neurone, représente si le nœud d'entrée actuel, disons n, appartient à la k-ème communauté. Si le nœud n appartient à la kième communauté, alors la valeur de sortie du kième neurone est réglée sur un nombre positif (par exemple 1) et les valeurs de sortie des autres neurones sont réglées sur zéro. Notez que chaque entrée de chaque carte de caractéristiques est connectée à tous les K neurones de la couche de connexion complète. Soit f la dernière couche de connexion complète. Ensuite, la valeur du k-ième neurone de sortie o_k^n dans la couche de connexion complète pour le nœud d'entrée n peut être définie comme suit :

$$o_k^n = sigmoide(b_f^k + W_f^k q_{c1 \cdot c2}^n) o q_{c1 \cdot c2}^n), \tag{2.6}$$

Où $q_{c1 \cdot c2}^n$ est la sortie de la deuxième couche de convolution, et W_f^k et b_f^k sont respectivement les poids et le biais du k-ième neurone dans la dernière couche de connexion complète.

2.3.2.5 Entraînement

Dans cette étape de l'apprentissage, l'objectif est d'optimiser les paramètres du modèle $P = (W, W^f, b, b^f)$ afin de détecter avec précision les communautés. Les noyaux convolutifs sont mis à jour lors de la phase d'apprentissage. Pour ce faire, les auteurs utilisent la rétropropagation pour optimiser les paramètres du modèle. Les données d'apprentissage sont composées d'un ensemble de T échantillons $(s_n, l_n)_{1 \leq n \leq T}$, où s_n est la relation d'adjacence du nœud n et l_n est le vecteur d'étiquette de longueur K pour le nœud n et $l_n^k \in 0, 1$, indiquant s'il appartient ou non à la k-ème communauté. Dans notre modèle, la fonction de coût peut être donnée comme suit :

$$J(P) = \frac{1}{2} \sum_{n=1}^{T} ||o_n - l_n||_2^2 = \frac{1}{2} \sum_{n=1}^{T} \sum_{k=1}^{K} (o_n^k - l_n^k)^2, \tag{2.7}$$

Où o_n^k représente la sortie du k-ème neurone et l_n^k est l'étiquette correspondante. Avec la dernière couche entièrement connectée, en appliquant l'algorithme de rétropropagation, les paramètres P sont optimisés et le coût du modèle $J(P)$ est diminué. L'algorithme s'arrête lorsque la fonction de coût converge.

2.3.2.6 Expériences et évaluation

Les résultats expérimentaux ont montré que la méthode CNN était plus performante que les méthodes non supervisées et supervisées, et qu'elle était plus robuste lorsque les arrêts des réseaux du monde réel manquaient gravement. Les auteurs ont également discuté de l'influence de la taille des données étiquetées ainsi que du nombre de sauts sur leur modèle profond. Ils ont présenté les performances du modèle CNN sur des ensembles de données. L'ensemble de données Football est un réseau à petite taille avec des communautés authentiques. LiveJournal et Youtube sont des réseaux à grande taille. Pour évaluer les performances des méthodes de détection de communauté, plusieurs métriques peuvent être utilisées. Dans cet article, les auteurs ont choisi d'utiliser Fsame, Jaccard, NMI et Precision. Les auteurs ont noté que cet article était le premier à explorer comment trouver des communautés dans TIN par des méthodes supervisées et non supervisées.

En conclusion, cet article propose des approches intéressantes pour résoudre le problème de la détection de communauté dans le TIN. Les résultats expérimentaux montrent que leur méthode CNN est plus performante que les méthodes non supervisées et supervisées, ce qui suggère que les méthodes d'apprentissage profond pourraient être une solution prometteuse pour ce problème. Les futures recherches devraient se concentrer sur l'exploration de méthodes de prétraitement et d'apprentissage en profondeur supplémentaires pour améliorer encore les performances du modèle.

2.4 Conclusion

Dans ce chapitre, nous avons examiné en détail les méthodes traditionnelles de détection de communautés, ainsi que les principaux algorithmes proposés dans chaque méthode. Ensuite, nous avons exploré les approches basées sur CNN pour la détection de communautés, en comprenant leur fonctionnement spécifique. Cette compréhension approfondie nous permettra de mettre en œuvre facilement ces approches dans le chapitre suivant.

Dans ce prochain chapitre, nous évaluerons la précision de ces approches sur des ensembles de données et mènerons une étude comparative entre elles.

Chapitre 3

Approches à base d'arêtes vs. Approches à base de noeuds

3.1 Introduction

Dans ce chapitre, nous procédons à une étude comparative expérimentale des deux approches de classification des arêtes et de classification des nœuds sur divers ensembles de données. Nous fournissons une explication détaillée des des étapes de déroulement des expérimentations. En fin, nous discutons les résultats obtenus par les deux méthodes sur chaque jeu de données.

3.2 Environnement de l'implémentation

Dans la présente, nous mettons en lumière les composants essentiels du logiciel et du matériel qui jouent un rôle crucial dans l'implémentation de notre programme.

3.2.1 Logiciel

Dans la suite, nous explorons l'environnement de développement, le langage de programmation et les bibliothèques fondamentales utilisées lors de l'implémentation.

3.2.1.1 Python

C'est un langage de programmation open source, simple, facile à apprendre et de haut niveau. Il dispose de nombreuses bibliothèques disponibles gratuitement. Il se distingue par sa flexibilité et sa capacité à représenter des concepts de programmation avec un code concis, ce qui contribue à réduire les coûts de maintenance. De plus, il est largement utilisé dans le domaine de l'intelligence artificielle en raison de sa capacité à exécuter facilement des tâches complexes d'apprentissage automatique [1].

[1] https://www.python.org

3.2.1.2 Tensorflow

C'est une bibliothèque Python versatile et open source, connue pour son écosystème flexible offrant de nombreuses bibliothèques et outils permettant de créer une multitude de modèles d'apprentissage automatique et d'applications d'apprentissage en profondeur. Grâce à ses caractéristiques, elle est devenue un framework très répandu [2].

3.2.1.3 Keras

C'est une bibliothèque Python open source pour la construction de réseaux de neurones. Elle se distingue par sa rapidité et sa facilité d'utilisation, ce qui en fait un choix privilégié pour le développement et l'entraînement de modèles d'apprentissage en profondeur [3].

3.2.1.4 Scikit-learn (Sklearn)

C'est une bibliothèque open source de machine learning en Python qui offre une large gamme de fonctionnalités, notamment le clustering, la régression et la classification. Cette bibliothèque est très polyvalente et peut être utilisée dans de nombreux contextes d'apprentissage automatique [4].

3.2.1.5 NetworkX

Il s'agit d'une bibliothèque Python qui permet la création, le traitement, l'analyse et l'étude de réseaux complexes ou de graphes complexes à l'aide d'une vaste gamme d'outils fournis [5].

3.2.2 Matériel

Google Colab gratuit dispose de 12,7 Go de RAM, 15 Go de RAM GPU et un disque dur de 78,2 Go.

3.3 Description des ensembles de données

Dans le cadre de notre expérience, nous évaluons les deux approches sur quatre ensembles de données distincts, chacun caractérisé par son nombre de nœuds, d'arêtes et de communautés. Cette section se consacre à la description détaillée de ces ensembles de données utilisés dans notre étude.

[2] https://www.tensorflow.org
[3] https://keras.io/
[4] https://scikit-learn.org/stable/
[5] https://networkx.org

3.3.1 Club de karaté de Zachary

L'ensemble de données du Club de karaté de Zachary[6] a été collecté par Zachary, 1977 dans les années 1970. Il représente un club de karaté qui a connu une scission en deux clubs distincts. Le réseau est composé de 34 membres, chaque membre étant représenté par un nœud dans le graphe. Les interactions amicales entre les membres sont représentées par des arêtes, totalisant 78 arêtes. Les membres du premier club sont représentés par des sommets carrés et grisés, symbolisant leur fidélité à l'administrateur du club. Les membres du deuxième club sont représentés par des sommets ronds et blancs, indiquant leur alignement avec l'instructeur comme l'illustre la Figure 3.1. Cette base de données est largement utilisée car la structure de communautés est préalablement connue.

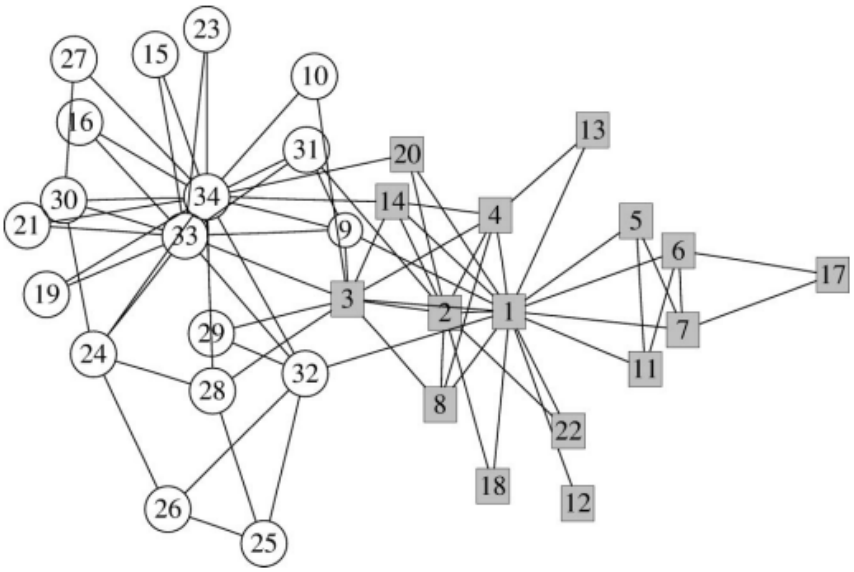


Fig. 3.1—Réseau de club de karaté de Zachary (M. E. Newman & Girvan, 2004).

3.3.2 Dauphins de Lusseau

Le réseau des dauphins de Lusseau[7] représente les interactions sociales entre les dauphins communs vivant dans le Parc Naturel Marin du Golfe du Lion, en France. Ce réseau est représenté sous forme de graphe non orienté, où les nœuds représentent les dauphins, les arêtes du graphe représentent les interactions sociales entre les dauphins. Ce réseau contient 62 nœuds et 159 liens. Dans ce réseau, les dauphins se sont séparés en deux groupes distincts (Figure 3.2). Cette séparation en groupes distincts offre des opportunités d'étude de la structure sociale et des comportements des dauphins dans leur environnement naturel (Lusseau et al., 2003).

[6] https://networkx.org
[7] https://networkrepository.com/soc-dolphins.php

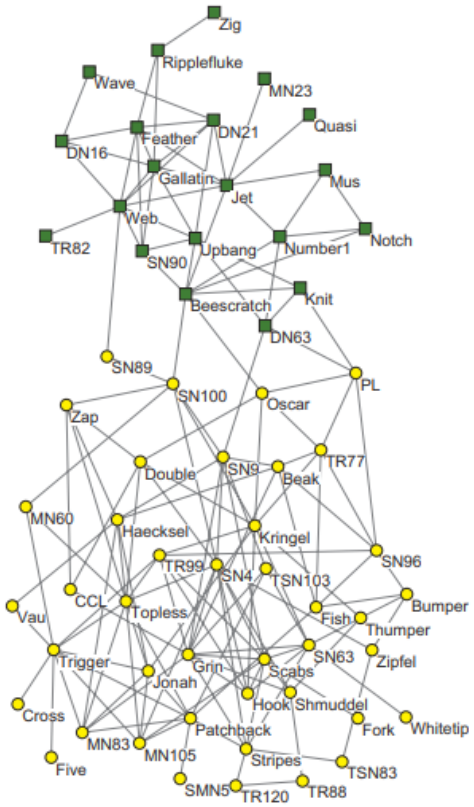


FIG. 3.2—Réseau social des dauphins (ARENAS et al., 2008).

3.3.3 Football américain

Le réseau Football américain présente une compilation complète d'information sur le football universitaire aux États-Unis[8] pour la saison 2000. Il se compose de 115 nœuds et 613 arêtes qui représentent les matchs entre les équipes de football. Les équipes ont été regroupées en 12 communautés distinctes, comme illustré dans la Figure 3.3, en fonction de leurs similitudes. Chaque communauté représente un groupe d'équipes partageant des caractéristiques communes dans le réseau (DU et al., 2007).

3.3.4 Email-Eu-core

Le réseau Email-Eu-core a été construit à partir de données de courrier électronique provenant d'une grande institution de recherche européenne[9]. La version sur laquelle nous avons travaillé est composée de 1005 nœuds et de 16706 arêtes, qui représentent les communications et les connexions entre les individus à l'intérieur et à l'extérieur de l'institution de recherche. Chaque individu de ce réseau est affilié à l'un des 42 départements comme indiqué dans la Figure 3.4, formant ainsi 42 communautés départementales distinctes.

[8] https://networkx.org
[9] https://snap.stanford.edu/data/email-Eu-core.html

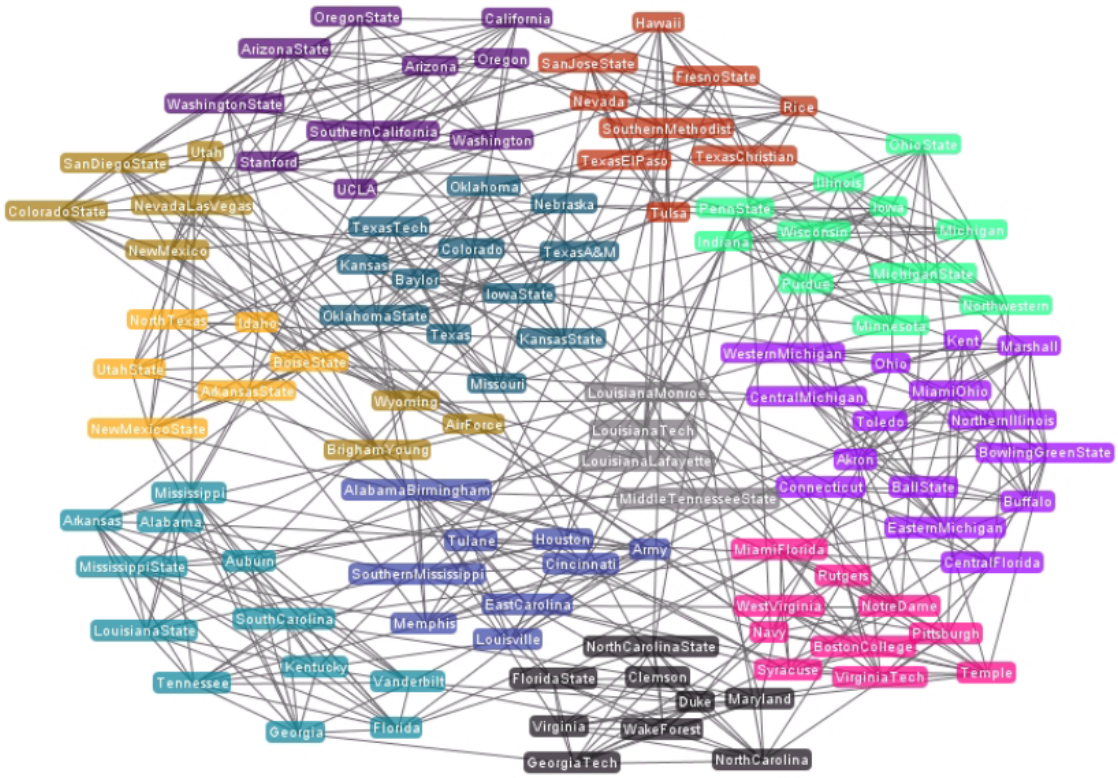


Fig. 3.3—Réseau football universitaire américain (Du et al., 2007).

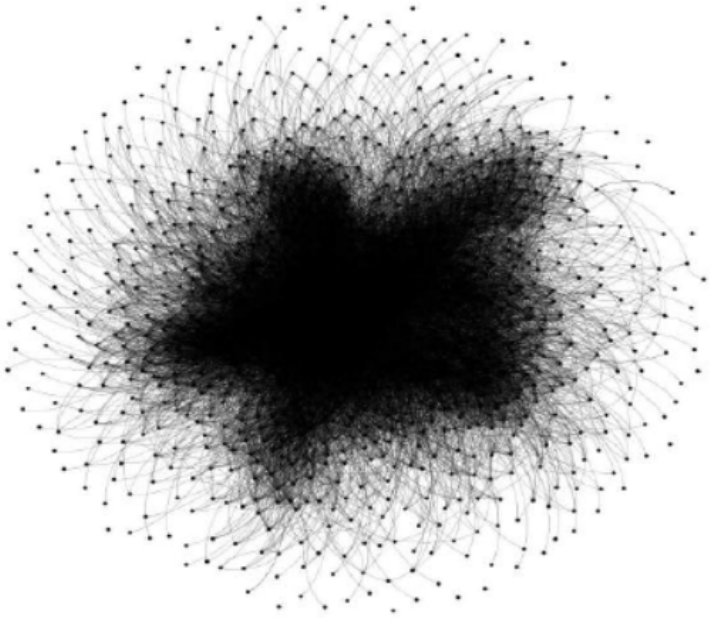


Fig. 3.4—Réseau Email-Eu-Core (Bharali, 2018).

3.4 Mesures d'évaluation des performances

Dans notre expérimentation, nous utilisons l'information mutuelle normalisée (NMI) (Équation 3.1) comme mesure pour évaluer la similarité entre les communautés détectées et les communautés réelles dans le réseau.

$$\mathrm{NMI}(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}\log\left(\frac{N_{ij}n}{N_i N_j}\right)}{\sum_{i=1}^{C_A} N_i \log\left(\frac{N_i}{n}\right) + \sum_{j=1}^{C_B} N_j \log\left(\frac{N_j}{n}\right)}, \tag{3.1}$$

où A représente l'ensemble des communautés réelles et B représente l'ensemble des communautés détectées. C_A et C_B correspondent respectivement au nombre de communautés réelles et au nombre de communautés détectées. Les valeurs N_i et N_j représentent respectivement le nombre de nœuds dans les communautés de l'ensemble A et de l'ensemble B. Quant à N_{ij}, il représente le nombre de nœuds présents à la fois dans la communauté i et la communauté j. Plus la valeur de N_{ij} est élevée, plus les communautés réelles i et détectées j sont similaires.

La plage de valeurs de sortie de la mesure NMI est de 0 à 1. Plus la valeur de NMI est proche de 1, plus les communautés détectées sont similaires aux communautés réelles.

En plus de cela, nous utilisons également le score F1 micro et F1 macro. Ces mesures permettent d'évaluer les performances de l'algorithme de détection de communauté au niveau de la communauté individuelle (macro F1) car il traite chaque catégorie individuellement et fait la moyenne des scores entre les catégories. et la performance globale dans toutes les communautés (micro F1). Ils fournissent des informations sur la précision, le rappel et les scores F1 globaux, ce qui permet d'évaluer l'efficacité de l'algorithme dans la détection et la classification des nœuds au sein des communautés (Lipton et al., 2014).

Pour calculer F1 micro et F1 macro :

1. Calculer les vrais positifs (TP), les faux positifs (FP) et les faux négatifs (FN) pour chaque communauté individuelle ainsi que les valeurs globales agrégées dans toutes les communautés.

2. Calculer la précision (P_i) et le rappel (R_i) pour chaque communauté i à l'aide de (Équation 3.2) et (Équation 3.3) respectivement :

$$Prcision(P_i) = \frac{TP_i}{TP_i + FP_i} \tag{3.2}$$

$$Rappel(R_i) = \frac{TP_i}{TP_i + FN_i} \tag{3.3}$$

3. Calculer le score F1 ($F1_i$) pour chaque communauté i en utilisant (Équation 3.4).

$$F1_i = \frac{2 \cdot (P_i \cdot R_i)}{P_i + R_i} \tag{3.4}$$

4. Calculer le score macro F1 en prenant la moyenne des scores F1 dans toutes les communautés à l'aide de (Équation 3.5).

$$MacroF1 = \frac{1}{C}\sum F1_i \tag{3.5}$$

Où C est le nombre total de communautés.

5. Pour calculer le score F1 micro (Équation 3.6), additionner les valeurs TP, FP et FN dans toutes les communautés, puis calculer la précision, le rappel et le score F1 à l'aide des valeurs agrégées :

$$Prcisionmicro = \frac{\sum TP}{\sum TP + \sum FP}$$

$$Rappelmicro = \frac{\sum TP}{\sum TP + \sum FN}$$

$$F1micro = \frac{2(\text{Précision micro} \cdot \text{Rappel micro})}{\text{Précision micro} + \text{Rappel micro}} \tag{3.6}$$

3.5 Implémentation

Cette section se focalise sur les détails d'implémentation des deux approches, ainsi que sur les résultats obtenus dans le cadre de notre étude.

3.5.1 Expérimentation de l'approche de classification des arrêts

Pour mettre en œuvre l'approche de classification des arêtes, nous adoptons l'algorithme proposé dans Cai et al., 2020 et vérifions son efficacité sur quatre réseaux de tailles différentes : le club de karaté de Zachary, le réseau social de dauphins, le réseau de la ligue de football et le réseau email-Eu-core. Ces réseaux sont représentés sous forme de graphes, et pour les utiliser dans notre modèle, nous accédons à ces graphes en utilisant la bibliothèque NetworkX.

Étant donné que les CNNs sont principalement utilisés pour le traitement d'images, nous convertissons les arêtes des réseaux en images en utilisant la méthode *edge_to_image (E2I)* proposée par Cai et al., 2020. L'idée derrière $E2I$ est de représenter les relations entre les voisins de chaque paire de nœuds sous la forme d'une matrice $3D$.

La matrice de sortie pour chaque arête reliant les nœuds i et j est composée de trois couleurs : le rouge indique que l'arête formée par les nœuds adjacents de i et j est présente dans le réseau, le vert indique que les nœuds adjacents de i et j sont tous deux présents dans l'intersection $N\_S(i, j)$, où $N\_S(i, j)$ représente l'ensemble des nœuds qui sont voisins à la fois de i et de j dans le réseau, et le bleu indique l'absence de relation entre eux.

Par la suite, nous appliquons une fonction appelée *image_cropping* pour recadrer l'image obtenue à partir de la conversion E2I en utilisant la méthode de recadrage *AREA*

(recadrage par interpolation de zone) fournie par Tensorflow. L'objectif est de transformer toutes les images résultantes en images $3D$ de taille fixe $(N * N * 3)$ avec une valeur N prédéfinie de 128 (Figure 3.5).

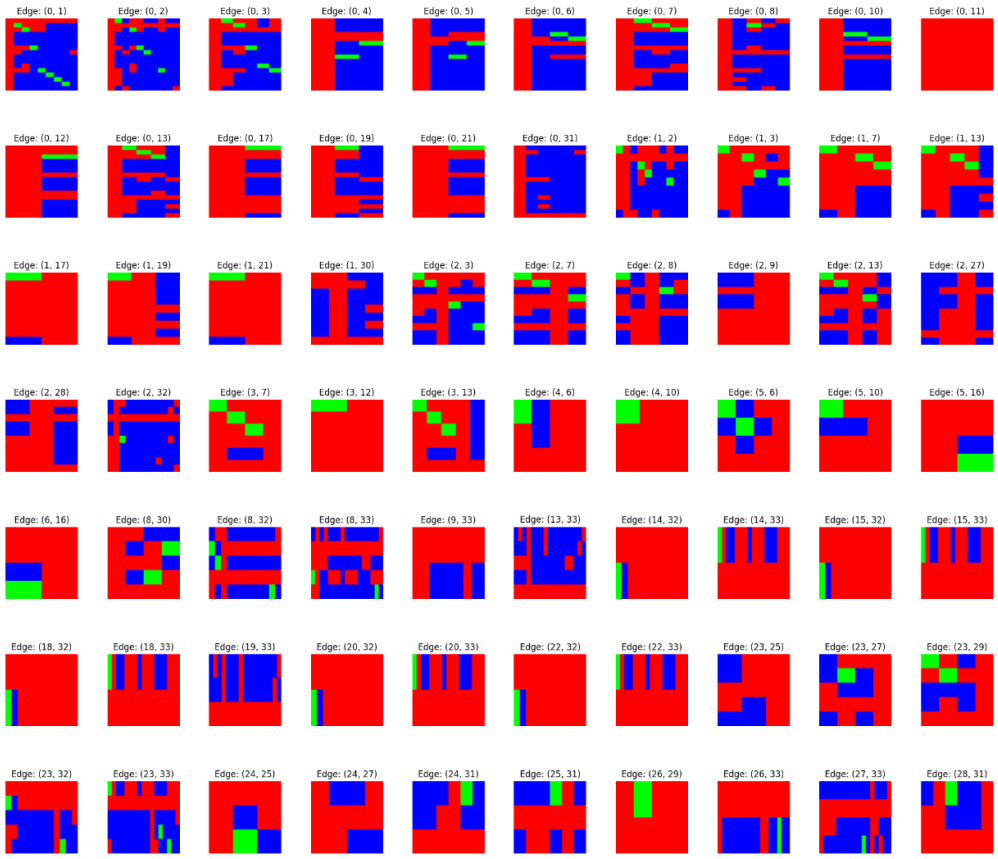


FIG. 3.5—Des images générées par toutes les arêtes dans le réseau Zachary.

Pour implémenter l'approche de classification des arêtes, nous avons utilisé le modèle de réseau de classification binaire *ComNet* dont l'architecture comprend 11 couches de convolution, 4 couches de MaxPooling, 2 couches de BatchNormalization, 1 couche d'aplatissement et 2 couches entièrement connectées. Les détails de ce modèle sont récapitulés dans la Table 3.1.

La première couche convolutive de modèle ComNet capture les caractéristiques de bas niveau des images d'entrée $(128 * 128)$ en utilisant 64 filtres $(7 * 7)$, une fonction d'activation ReLU pour introduire de la non-linéarité dans les opérations d'activation, et un padding pour préserver les informations dans les coins et les bords des images de manière plus efficace.

Ensuite, nous appliquons une couche de MaxPooling avec une fenêtre de pool de taille $(2*2)$, un padding et un décalage de 2 pour réduire les dimensions spatiales des activations de la couche précédente.

La troisième couche est la couche de batch normalization, qui est chargée de normaliser l'activation de la couche précédente.

Les cartes de caractéristiques résultantes sont utilisées comme entrée pour la deuxième couche convolutive. Cette couche contient 64 filtres de taille $(3 * 3)$ et une autre couche

qui utilise 192 filtres de taille $(5*5)$. Les deux couches utilisent la fonction d'activation ReLU et le padding.

La Couche de batch normalization, suivie de la couche de MaxPooling avec une fenêtre de pool de taille $(2*2)$ et un décalage de 2.

Une quatrième couche convolutive utilise 192 filtres de taille $(3*3)$, tandis qu'une cinquième couche utilise 384 filtres de taille $(5*5)$. Les deux couches utilisent la fonction d'activation tanh.

Une troisième couche de MaxPooling est suivie de six couches convolutionnelles. La première couche convolutive utilise 384 filtres de taille $(3*3)$, tandis que les deuxième, quatrième et sixième couches utilisent 256 filtres de taille $(5*5)$. Les troisième et cinquième couches convolutives utilisent également 256 filtres de taille $(3*3)$, toutes ces couches utilisent la fonction d'activation tanh.

Ensuite, une quatrième couche de MaxPooling est ajoutée pour réduire les dimensions spatiales avant de passer aux couches entièrement connectées.

Par la suite, une couche de Flatten est ajoutée pour convertir les cartes de caractéristiques en un vecteur unidimensionnel, ce qui permet de les transmettre à une couche dense.

La couche finale est une couche entièrement connectée composée de 512 neurones et utilisant la fonction d'activation tanh. Cette couche est suivie d'une couche de sortie qui utilise la fonction d'activation softmax qui donne la probabilité de classification de chaque classe.

Le taux d'apprentissage initial est défini à $1e-4(0,0001)$, et l'optimiseur AdaGrad est utilisé avec cette valeur. L'optimiseur AdaGrad adapte le taux d'apprentissage individuellement pour chaque poids du modèle, ce qui peut accélérer la convergence en donnant plus d'importance aux paramètres moins fréquents.

Après l'entraînement de ComNet, nous utilisons une méthode de partitionnement (split_network_using_ComNet) pour découper le réseau en éliminant les arêtes qui connectent les communautés. En utilisant les sorties E2I en tant qu'entrée, le modèle ComNet classe les arêtes comme appartenant ou non à la communauté, en se basant sur la similarité des étiquettes des nœuds.

Les résultats de la méthode (split_network_using_ComNet) ne sont que des communautés préliminaires, car il existe une possibilité d'erreurs dans la reconnaissance des arêtes. Afin d'améliorer la qualité de la segmentation, nous mettons en œuvre une stratégie de fusion des communautés initiales en utilisant la méthode (merge_preliminary_communities) qui se base sur le calcul de la modularité (calculate_modularity) R. Cette approche vise à regrouper de manière optimale les communautés afin d'obtenir une segmentation plus précise.

Dans la phase d'entraînement du modèle, nous configurons le nombre d'époques à 100 et la taille du lot (batch_size) à 40. Après avoir évalué les performances du modèle en utilisant les métriques F1 micro, F1 macro et NMI, nous affichons les résultats obtenus pour chaque métrique dans la Table 3.2 afin d'évaluer la qualité du modèle.

TAB. 3.1 : Architecture du modèle ComNet.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 64)	9472
max_pooling2d (MaxPooling2D)	(None, 64, 64, 64)	0
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256
conv2d_1 (Conv2D)	(None, 64, 64, 64)	36928
conv2d_2 (Conv2D)	(None, 64, 64, 192)	307392
batch_normalization_1 (BatchNormalization)	(None, 64, 64, 192)	768
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 192)	0
conv2d_3 (Conv2D)	(None, 32, 32, 192)	331968
conv2d_4 (Conv2D)	(None, 32, 32, 384)	1843584
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 384)	0
conv2d_5 (Conv2D)	(None, 16, 16, 384)	1327488
conv2d_6 (Conv2D)	(None, 16, 16, 256)	2457856
conv2d_7 (Conv2D)	(None, 16, 16, 256)	590080
conv2d_8 (Conv2D)	(None, 16, 16, 256)	1638656
conv2d_9 (Conv2D)	(None, 16, 16, 256)	590080
conv2d_10 (Conv2D)	(None, 16, 16, 256)	1638656
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 256)	0
flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 512)	8389120
dense_1 (Dense)	(None, 2)	1026

Total params : 19,163,330

Tab. 3.2 : Résultats expérimentaux (F , NMI) de l'algorithme de détection de communauté.

Dataset	F1 micro	F1 macro	NMI
Zachary	1	1	1
Dauphins	1	1	1
Football	0.078	0.0937	0.9339
Core	0.0039	0.0062	0.267

3.5.2 Expérimentation de l'approche de classification des nœuds

L'implémentation du modèle de classification des nœuds pour la détection de communautés dans les réseaux complexes s'est inspirée de la méthode décrite par Xin et al., 2017 dans leur article "Deep Community Detection in Réseaux topologiquement incomplets". Il est important de souligner que l'implémentation précise du modèle peut varier en fonction des détails spécifiques de l'article et des modifications apportées pour améliorer ses performances. Les étapes générales du modèle comprennent :

Dans le domaine de l'analyse des réseaux, les jeux de données représentés sous forme de graphes sont couramment utilisés. Pour notre projet, nous exploitons la bibliothèque NetworkX pour collecter et manipuler ces données de réseau. Les datasets que nous allons utiliser sont le graphe de la Karate Club, le réseau des dauphins de Lusseau, le réseau de football universitaire américain et le réseau Email-Eu-Core. Chacun de ces jeux de données présente des caractéristiques spécifiques et peut être utilisé dans diverses analyses et tâches de détection de communautés. Grâce à NetworkX, nous pouvons charger et prétraiter ces jeux de données en vue de les utiliser dans notre modèle de classification des nœuds.

L'étape suivante consiste à convertir les données du réseau en matrices d'adjacence en utilisant les fonctions de NetworkX. Parmi ces fonctions *nx.adjacency_matrix(G)*. Cette fonction retourne la matrice d'adjacence du graphe G et *nx.to_numpy_matrix(G)* convertit le graphe G en une matrice d'adjacence au format numpy. Ces fonctions permettent de convertir les données du réseau en une représentation matricielle, ce qui facilite le traitement et l'analyse. Ensuite, la fonction *calculate_locality_feature2* est appliquée à la matrice d'adjacence pour calculer les caractéristiques de localité. Cette fonction prend en entrée une matrice d'adjacence adj_matrix ainsi que deux paramètres : σ, qui est un facteur d'atténuation égal à 0.6, et $s0$, qui est un seuil de nombre de sauts égal à 3, le résultat est illustré dans la figure 3.6. Les caractéristiques de localité fournissent des informations sur la connectivité des nœuds dans un réseau. Elles aident à détecter les communautés en mettant en évidence les arêtes forts internes et les arêtes plus faibles externes. Le calcul des caractéristiques de localité implique la multiplication de la matrice d'adjacence par elle-même, ce qui permet de mettre à jour les valeurs pour souligner les arêtes forts. Ces caractéristiques de localité améliorent la détection des communautés et les performances

```
Matrice d'adjacence du graphe du Club de Karaté de Zachary :
[[0 1 1 ... 1 0 0]
 [1 0 1 ... 0 0 0]
 [1 1 0 ... 0 1 0]
 ...
 [1 0 0 ... 0 1 1]
 [0 0 1 ... 1 0 1]
 [0 0 0 ... 1 1 0]]
Caractéristique de localité du Club de Karaté de Zachary:
[[0.         1.         1.         ... 1.         0.54881164 0.54881164]
 [1.         0.         1.         ... 0.54881164 0.54881164 0.54881164]
 [1.         1.         0.         ... 0.54881164 1.         0.54881164]
 ...
 [1.         0.54881164 0.54881164 ... 0.         1.         1.        ]
 [0.54881164 0.54881164 1.         ... 1.         0.         1.        ]
 [0.54881164 0.54881164 0.54881164 ... 1.         1.         0.        ]]
```

FIG. 3.6—Résultats de la matrice d'adjacence et les caractéristiques de localité du graphe Zakary.

des modèles de classification des nœuds.

Le modèle CNN utilisé dans ce code est une architecture séquentielle qui comprend plusieurs couches pour l'extraction de caractéristiques et la classification. La première couche du modèle est couche de redimensionnement (Reshape) qui ajuste la forme des données d'entrée pour qu'elle corresponde à la taille du réseau. Ensuite, il y a deux couches de convolution (Conv1D) qui appliquent 10 filtres. La taille de noyau est 5 sur les données en entrée. Chaque couche utilise une fonction d'activation tangente hyperbolique (tanh) pour introduire une non-linéarité. Après chaque couche de convolution, une couche de pooling (MaxPooling1D) est utilisée avec un filtre de taille 2. Une couche de normalisation par lots (BatchNormalization) est ajoutée pour normaliser les activations des couches précédentes, améliorant ainsi la stabilité de l'apprentissage. Ensuite, les caractéristiques extraites sont aplaties (Flatten) pour être fournies en tant qu'entrée à la couche dense suivante. La dernière couche du modèle est une couche dense avec une fonction d'activation sigmoïde dans le cas de classification de 2 communautés (Karate, Dolphin). Cela permet de réaliser une classification binaire en produisant une probabilité de classe. Pour les tâches de classification multi-classes dans l'ensemble football (12 classes) et l'ensemble Eu-Core(42 classes), la fonction d'activation softmax est utilisée pour prédire la probabilité de chaque classe. Les détails de ce modèle sont récapitulés dans la Table 3.3.

La fonction de perte utilisée pour évaluer la différence entre les prédictions du modèle et les vérités terrain est l'entropie croisée binaire (binary_crossentropy) et dans le cas classification multi-classes (sparse_categorical_crossentropy). Le modèle est compilé avec ces paramètres d'optimisation et de perte, et les métriques d'évaluation de l'exactitude (accuracy) sont également spécifiées.

Pour entraîner le modèle CNN sur les données de réseau, nous avons suivi les étapes suivantes :

1. Extraction des étiquettes de communauté : Les étiquettes de communauté sont extraites à partir du graphe en attribuant à chaque nœud la valeur de la communauté à laquelle il appartient.

TAB. 3.3 : Résumé du modèle de classification binaire des nœuds sur le dataset Zakary.

```
Model : "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 34, 1)             0

 conv1d (Conv1D)             (None, 30, 10)            60

 max_pooling1d (MaxPooling1D  (None, 15, 10)           0
 )

 conv1d_1 (Conv1D)           (None, 11, 4)             204

 max_pooling1d_1 (MaxPooling  (None, 5, 4)             0
 1D)

 batch_normalization (BatchN  (None, 5, 4)             16
 ormalization)

 flatten (Flatten)           (None, 20)                0

 dense (Dense)               (None, 1)                 21

=================================================================
Total params : 301
Trainable params : 293
Non-trainable params : 8
_________________________________________________________________
```

2. Division des données en ensembles d'entraînement et de test : L'ensemble de données est divisé en ensembles distincts pour l'entraînement et les tests. Pour le premier ensemble, une division de 50% est utilisée pour l'entraînement et 50% pour les tests. Pour les deuxième et troisième ensembles, une division de 70% est utilisée pour l'entraînement et 30% pour les tests. Dans les trois cas, 90% des données sont utilisées pour l'entraînement et 10% pour les tests.

3. Entraînement du modèle : Le modèle est entraîné en spécifiant le nombre d'époques (epochs), dans ce cas, 100 époques, et la taille du lot (batch_size), fixée à 40.

4. Évaluation des performances du modèle : Après l'entraînement du modèle, ses performances sont évaluées en utilisant les métriques F1 micro, F1 macro et NMI, Les performances du modèle sont ensuite affichées dans le tableau des résultats 3.4.

Tab. 3.4 : Résultats de modèle entraîné avec différents pourcentages d'étiquetage d'ensemble d'entraînement.

Data	**Zakary**			**Dolphins**			**Football**			**Eu-core**		
/	**50%**	**70%**	**90%**	**50**	**70%**	**90%**	**50**	**70%**	**90%**	**50%**	**70%**	**90%**
F1-Micro	1.0	1.0	1.0	1.0	1.0	1.0	0.896	0.942	1.0	0.713	0.721	0.956
F1-Macro	1.0	1.0	1.0	1.0	1.0	1.0	0.837	0.865	1.0	0.515	0.547	0.896
NMI	1.0	1.0	1.0	1.0	1.0	1.0	0.921	0.956	1.0	0.747	0.775	0.978

3.6 Résultats et discussion

Dans cette section, nous discutons des résultats de l'implémentation des deux approches, à savoir la classification des nœuds et la classification des arêtes, et évaluons leur efficacité dans la détection des communautés sur différents ensembles de données.

Dans la classification des arêtes, les ensembles de données Zachary et Dolphin obtiennent des scores parfaits de 1 dans chacune des trois mesures, comme indiqué dans le Tableau 3.2. Cela s'explique par la petite taille et la densité de ces réseaux. Ces réseaux ont un nombre limité d'arêtes et un petit nombre de communautés, ce qui facilite la tâche de classification des arêtes.

En revanche, dans le cas de l'ensemble de données Football, le score F1 présente de mauvais résultats, tandis que le score NMI est élevé. Le problème avec F1 lors du processus de clustering réside dans le fait que les étiquettes ne correspondent pas adéquatement, tandis que NMI n'est pas sensible aux noms spécifiques des étiquettes.

Par ailleurs, l'ensemble de données Eu_Core donne des scores très faibles dans chacune des trois mesures en raison de sa taille importante, de sa densité élevée, du grand nombre d'arêtes et de la complexité structurelle des communautés. Ces caractéristiques rendent la tâche de classification des arêtes plus complexe et peuvent entraîner une performance réduite du modèle.

Dans la classification des nœuds, les résultats présentés dans la Table 3.4 indiquent les performances du modèle entraîné avec différents pourcentages d'étiquetage d'ensemble d'entraînement sur quatre jeux de données. Pour les jeux de données Zakary, Dolphins et Football, les performances du modèle sont presque parfaites avec tous les pourcentages d'ensemble d'entraînement. Pour le jeu de données Eu_core, les performances du modèle sont légèrement inférieures.

Les performances du modèle sont proches de la perfection, quel que soit le pourcentage d'étiquetage de l'ensemble d'entraînement utilisé. Cela signifie que le modèle a réussi à apprendre efficacement à partir de ces données et à effectuer une classification précise des nœuds.

la Table 3.5 présente les résultats expérimentaux des deux approches de détection de communauté, évalués à l'aide des mesures F1 et NMI. La classification des nœuds a été effectuée avec un étiquetage de 90

TAB. 3.5 : Résultats expérimentaux des deux approches.

DATASET	Mesures de performance	Classification des nœuds	Classification des arêtes
Zakary	**F1 micro**	1.0	1.0
	F1 macro	1.0	1.0
	NMI	1.0	1.0
Dolphins	**F1 micro**	1.0	1.0
	F1 macro	1.0	1.0
	NMI	1.0	1.0
Football	**F1 micro**	1.0	0.078
	F1 macro	1.0	0.0937
	NMI	1.0	0.933
Eu-core	**F1 micro**	0.956	0.003
	F1 macro	0.896	0.006
	NMI	0.978	0.267

En analysant les résultats expérimentaux de la Table 3.5, nous tenons à discuter de manière comparative les performances des deux méthodes de détection de communauté. Voici les points communs et les différences entre ces approches :

1. Points communs

   - Les deux approches sont capables de détecter des communautés.

2. Points de différence

   - Architecture du modèle : L'architecture du modèle de classification des arêtes est caractérisée par sa complexité en termes de couches et de paramètres. Cette complexité peut conduire à une spécialisation excessive dans certains groupes de données, ce qui signifie que le modèle peut être très performant

pour classer les arêtes d'un type spécifique, mais peut rencontrer des difficultés à généraliser et à obtenir de bons résultats pour d'autres types d'arêtes. En revanche, l'architecture du modèle de classification des nœuds est plus simple et moins compliquée, ce qui lui permet de s'adapter efficacement à différents types de données.

- Type d'apprentissage : Dans la classification des nœuds, il s'agit d'un apprentissage semi-supervisé, tandis que dans la classification des arêtes, il s'agit d'un apprentissage supervisé.
- Temps d'exécution : Le temps d'exécution peut varier en fonction de la taille du réseau. En général, la classification des noeuds est plus rapide car elle est basée sur des caractéristiques calculées directement à partir de matrices d'adjacence, contrairement à la classification des arêtes qui nécessite plusieurs étapes, ce qui prend plus de temps d'exécution.
- Date de publication : L'approche de classification des nœuds a été publiée en 2016, tandis que la classification des arêtes a été publiée en 2020.

Notre étude comparative a conclu que l'approche de classification des nœuds est la meilleure en termes d'efficacité et de précision.

3.7 Conclusion

Dans ce chapitre, nous analysons les résultats expérimentaux des deux approches et avons comparé leur précision en termes de performances. Ces résultats confirment que la classification des nœuds est une méthode robuste et fiable pour découvrir les communautés dans les réseaux. Par conséquent, nous recommandons d'adopter cette approche pour de telles tâches.

Conclusion générale

La détection de communautés dans les réseaux complexes est une problématique essentielle dans de nombreux domaines de recherche tels que les réseaux sociaux, biologiques. Les méthodes traditionnelles de détection de communautés peuvent être coûteuses et inefficaces, ce qui a conduit les chercheurs à s'orienter vers l'apprentissage en profondeur, en particulier les réseaux de neurones à convolution (CNNs), en raison de leur efficacité remarquable dans de nombreux domaines. La question générale du sujet est de savoir comment les CNNs peuvent être utilisés pour détecter efficacement les communautés dans les réseaux complexes.

Ce mémoire sur la découverte de communautés dans les réseaux complexes basée sur les CNNs a exploré deux approches principales, à savoir la classification des nœuds et des arêtes et contribue ainsi à l'amélioration des méthodes existantes de détection de communautés en exploitant les avantages des réseaux de neurones convolutifs. Les résultats expérimentaux obtenus sur différents ensembles de données ont démontré la capacité des CNNs à identifier efficacement les communautés dans les réseaux.

La démarche de recherche consiste à implémenter les deux approches, objets de la comparaison, et à évaluer leurs performances sur différents ensembles de données en utilisant des métriques telles que F1 micro, F1 macro et NMI.

Les obstacles de ce mémoire peuvent inclure la dépendance des performances des approches proposées aux caractéristiques spécifiques des réseaux étudiés, ainsi que la complexité de l'architecture de CNN dans l'approche de classification des arêtes. De plus, en raison de contraintes de ressources telles que la mémoire RAM, il n'a pas été possible de travailler sur des ensembles de données volumineux tels que le jeu de données YouTube avec plus d'un million de nœuds et 2 987 624 arêtes, ainsi que le jeu de données DBLP avec 317 080 nœuds et 1 049 866 arêtes.

Les résultats de cette recherche ont montré que les approches basées sur les CNNs ont obtenu des performances élevées dans la détection de communautés sur certains ensembles de données, comme le Club de karaté de Zachary et le réseau social des dauphins. Cependant, des performances plus faibles ont été observées sur des ensembles de données plus complexes, tels que le réseau de la ligue de football et le réseau email-Eu-core.

Ces résultats ouvrent de nouvelles perspectives de recherche et offrent des opportunités d'application concrètes pour la prise de décisions éclairées et l'amélioration des systèmes dans diverses industries. Les perspectives futures incluent l'exploration des CNNs et d'autres architectures de réseaux de neurones pour améliorer les performances de détection de communautés, ainsi que l'étude de l'impact des techniques de prétraitement des données et des paramètres de modélisation sur les résultats.

Bibliographie

Ahamed, P., Kundu, S., Khan, T., Bhateja, V., Sarkar, R. & Mollah, A. (2020). Handwritten Arabic numerals recognition using convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing*, *11*. https://doi.org/10.1007/s12652-020-01901-7

Alsobhani, A., ALabboodi, H. M. A. & Mahdi, H. (2021). Speech Recognition using Convolution Deep Neural Networks. *Journal of Physics : Conference Series*, *1973*(1), 012166. https://doi.org/10.1088/1742-6596/1973/1/012166

Amidi, A. & Amidi, S. (2020). *Quelque fonction d'activation* [Accessed on March 20, 2023]. https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-deep-learning

Arenas, A., Fernandez, A. & Gomez, S. (2008). Analysis of the structure of complex networks at different resolution levels. *New journal of physics*, *10*(5), 053039.

Barabási, A.-L. (2013). Network Science Chapter 2: Graph Theory. *Book*, *371*.

Barabasi, A.-L. & Posfai, M. (©2016). *Network science*. United Kingdom : Cambridge University Press, https://www.loc.gov/catdir/enhancements/fy1701/2016439537-b.html

Bhandare, A., Bhide, M., Gokhale, P. & Chandavarkar, R. (2016). Applications of Convolutional Neural Networks.

Bharali, A. (2018). An Analysis of Email-Eu-Core Network. *International Journal of Scientific Research in Mathematical and Statistical Sciences*, *5*, 100-104. https://doi.org/10.26438/ijsrmss/v5i4.100104

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K. & Ghayvat, H. (2021). CNN Variants for Computer Vision : History, Architecture, Application, Challenges and Future Scope. *Electronics*, *10*(20). https://doi.org/10.3390/electronics10202470

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, *2008*(10), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

Bretto, A., Faisant, A. & Hennecart, F. (2012). *Éléments de théorie des graphes* (T. 5). Springer.

Brouard, C. (2013). *Inférence de réseaux d'interaction protéine-protéine par apprentissage statistique* (thèse de doct.).

Cai, B., Wang, Y., Zeng, L., Hu, Y. & Li, H. (2020). Edge classification based on Convolutional Neural Networks for community detection in complex network. *Physica A : Statistical Mechanics and its Applications*, *556*, 124826. https://doi.org/https://doi.org/10.1016/j.physa.2020.124826

Cazabet, R. (2013). *Détection de communautés dynamiques dans des réseaux temporels* (thèse de doct.). Université Paul Sabatier-Toulouse III.

Cohen, J. (2006). Théorie des graphes et algorithmes. *Course notes. http ://www. univ-paris12. fr/lacl/cohen/poly-gr. ps*.

Combe, D. (2013). *Détection de communautés dans les réseaux d'information utilisant liens et attributs*. David Combe.

Dao, V.-L. (2018). *Characterizing community detection algorithms and detected modules in large-scale complex networks* (thèse de doct.).

Denny, M. (2014). Social network analysis. *Institute for Social Science Research, University of Massachusetts, Amherst*, *13*, 31.

Du, N., Wu, B., Pei, X., Wang, B. & Xu, L. (2007). Community detection in large-scale social networks. *Joint Ninth WebKDD and First SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, *16-25*. https://doi.org/10.1145/1348549.1348552

Dumolard, P. (1994). Les réseaux de neurones. *L'Espace géographique*, *23*(3), 287-288.

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, *486*(3-5), 75-174.

Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A. & De, D. (2020). Fundamental Concepts of Convolutional Neural Network. https://doi.org/10.1007/978-3-030-32644-9_36

Girvan, M. & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, *99*(12), 7821-7826.

Gonzalez, R. C., Woods, R. E. & Masters, B. R. (2009). Digital Image Processing, Third Edition. *Journal of Biomedical Optics*, *14*. https://doi.org/10.1117/1.3115362

Janssen, P. (2012). Cluster analysis to understand socio-ecological systems : a guideline.

Kanawati, R. (2013). Détection de communautés dans les grands graphes d'interactions (multiplexes) : état de l'art.

Kulkarni, A. & Shivananda, A. (2019). *Natural language processing recipes*. Springer.

Leskovec, J., Rajaraman, A. & Ullman, J. D. (2020). *Mining of Massive Datasets* (3e éd.). Cambridge University Press. https://doi.org/10.1017/9781108684163

Lipton, Z. C., Elkan, C. & Narayanaswamy, B. (2014). Thresholding Classifiers to Maximize F1 Score.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E. & Dawson, S. M. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations : can geographic isolation explain this unique trait ? *Behavioral Ecology and Sociobiology*, *54*, 396-405.

Michel. (2023). *GirvanNewman* [Accessed on March 17, 2023]. https://cedric.cnam.fr/vertigo/Cours/RCP216/coursGraphesCommunautes.html#mmds

Müler, D. (2012). Introduction à la théorie des graphes. *Cahiers de la CRM*, *6*.

Needham, M. & Hodler, A. E. (2020). *Graph algorithms : Practical examples in Apache Spark and Neo4j*.

Newman, M. (2006). Newman MEJ.. Modularity and community structure in networks. Proc Natl Acad Sci USA 103: 8577-8582. *Proceedings of the National Academy of Sciences of the United States of America*, *103*, 8577-82. https://doi.org/10.1073/pnas.0601602103

Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physical review E*, *69*(6), 066133.

Newman, M. E. & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, *69*(2), 026113.

NORMAN, F. (2016). *Réseau social d'amitié*. [Accessed on May 7, 2023]. https://ulfire.com.au/personal-networks/

Nwankpa, C., Ijomah, W., Gachagan, A. & Marshall, S. (2018). Activation Functions : Comparison of trends in Practice and Research for Deep Learning. *CoRR*, *abs/1811.03378*arXiv 1811.03378. http://arxiv.org/abs/1811.03378

Patel, S. & Patel, A. (2020). Object Detection with Convolutional Neural Networks. https://doi.org/10.1007/978-981-15-7106-0_52

Peacock, M. (2010). *PHP 5 Social Networking : Create a Powerful and Dynamic Social Networking Website in PHP by Building a Flexible Framework*. Packt Pub. https://books.google.dz/books?id=FU0-mQEACAAJ

Podareanu, D., Codreanu, V., Aigner, S., Leeuwen, C. & Weinberg, V. (2019). *Best Practice Guide - Deep Learning*. https://doi.org/10.13140/RG.2.2.31564.05769

Pons, P. & Latapy, M. (2005). Computing communities in large networks using random walks (long version).

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. & Parisi, D. (2004). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, *101*, 2658-63. https://doi.org/10.1073/pnas.0400054101

Raghavan, U. N., Albert, R. & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, *76*(3). https://doi.org/10.1103/physreve.76.036106

Ramprasath, M., Anand, M. V. & Hariharan, S. (2018). Image classification using convolutional neural networks. *International Journal of Pure and Applied Mathematics*, *119*(17), 1307-1319.

Rosvall, M. & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, *105*(4), 1118-1123. https://doi.org/10.1073/pnas.0706851105

Salim, B. (2011). *Une Nouvelle Approche Pour La Découverte De La Topologie Dans Les Réseaux Mobiles Ad Hoc Inspirée De La Communication Dans Les Essaims D'abeilles* (thèse de doct.). Université Mohamed Khider - Biskra.

Sarr, I. & Moctar, A. O. M. (2016). Détection de communautés statiques et dynamiques. *Revue d'intelligence artificielle*, *30*. https://doi.org/10.3166/ria.30.469-496

Sedgewick, R. & Wayne, K. (2011). Algorithms (4th edn). *Google Scholar Google Scholar Digital Library Digital Library*.

Sharma, S., Sharma, S. & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS.

Sikora, F. (2011). *Aspects algorithmiques de la comparaison d'éléments biologiques* (Theses 2011PEST1048). Université Paris-Est. https://pastel.archives-ouvertes.fr/pastel-00667797

Slimani, Y. & Drif, A. (2016). *Découverte de communautés dans les réseaux complexes* [working paper or preprint]. working paper or preprint. https://hal.science/hal-01389844

Sorokina, K. (2022). *Image classification with convolutional neural networks* [Accessed on May 28, 2023]. https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8

Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Jin, D. Et al. (2022). A comprehensive survey on community detection with deep learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Sun, Z., Sheng, J., Wang, B., Ullah, A. & Khawaja, F. (2020). Identifying Communities in Dynamic Networks Using Information Dynamics. *Entropy*, *22*(4). https://doi.org/10.3390/e22040425

Tabassum, S., Pereira, F., Fernandes, S. & Gama, J. (2018). Social network analysis : An overview. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, *8*, e1256. https://doi.org/10.1002/widm.1256

Traag, V. A. (2015). Faster unfolding of communities : Speeding up the Louvain algorithm. *Physical Review E*, *92*(3). https://doi.org/10.1103/physreve.92.032801

Viennet, E. (2009). Recherche de communautés dans les grands réseaux sociaux. *Revue des Nouvelles Technologies de l'Information, Apprentissage Artificiel et Fouille de Données, RNTI-A-3*, 145-160.

Wasserman, S. & Faust, K. (1994). *Social Network Analysis : Methods and Applications*. Cambridge University Press. https://doi.org/10.1017/CBO9780511815478

Xin, X., Wang, C., Ying, X. & Wang, B. (2017). Deep community detection in topologically incomplete networks. *Physica A : Statistical Mechanics and its Applications*, *469*, 342-352.

Xuegang, H., He, W., Li, H. & Pan, J. (2016). Role-based Label Propagation Algorithm for Community Detection.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, *33*(4), 452-473.

Zeiler, M. D. & Fergus, R. (2013). Visualizing and Understanding Convolutional Networks. *CoRR*, *abs/1311.2901*arXiv 1311.2901. http://arxiv.org/abs/1311.2901

Zitnik, M. (2016). Biological networks [Retrieved from]. https://snap.stanford.edu/class/cs224w-2016/slides/handout-bionets.pdf